



ORACLE

Oracle WebLogic Server for Oracle Cloud Infrastructureのディザスタ・ リカバリ



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

2023年7月 | バージョン19

Copyright © 2023, Oracle and/or its affiliates

公開

本書の目的

本書には、Oracle WebLogic Server for Oracle Cloud Infrastructureスタックのデザスタ・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 WebLogic Server、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8月	1	初版
2020年12月	2	ベスト・プラクティスを追加 付録CにDR環境でのスケール・アウト手順を追加 DRシナリオでのIDCS認証のサポートを追加
2021年3月	3	その他のセットアップ・オプション（DBFSではなくFSS/rsyncの方法をベースとしたもの）を追加。 複数のセクションを更新 付録Cに「スタンバイDBシステム再作成」の説明を追加
2021年4月	4	「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」のセクションを追加
2021年5月	5	表現を改良
2021年7月	6	付録Eに追加のセットアップ・オプション（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もの）を追加
2021年9月	7	5ページの脚注を追加 スイッチオーバー操作にOCI DNSスイッチオーバーの例を追加
2021年10月	8	「前提条件」→「データベース」の図および追加情報、推奨されるDB接続文字列の注記を更新
2022年1月	9	Data Guardの手動セットアップ・スクリプトを更新し、「RTOおよびRPOの概要」セクションを追加
2022年4月	10	「EXACSに関する考慮事項」の箇所を追加
2022年6月	11	「付録C - その他のライフサイクル運用」に「バッチ適用」に関する箇所を追加
2022年6月	12	Site Guardの非推奨化のため、リンクを更新し、Oracle Site Guardの参照を削除 (Doc ID 2875372.1を参照)
2022年7月	13	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	14	「付録F - 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
2022年11月	15	「セカンダリ・サイトでOracle WLS for OCIのプロビジョニング」にOCIポリシーの説明を追加 CRSサービス作成時のロール定義を更新（フィジカル・スタンバイ・ロールは必須ではない）
2023年2月	16	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	17	ブロック・ボリュームDR手法の考慮事項を更新。
2023年6月	18	“カスタム・ファイル”ポイントの改善
2023年7月	19	開始/停止スクリプトへのリンクを追加

目次

本書の目的	1
改訂履歴	1
はじめに	4
Oracle WLS for OCI向けのディザスタ・リカバリ・ソリューション	6
トポロジの説明	6
前提条件	9
要件	13
スクリプトのダウンロード	16
ディザスタ・リカバリのセットアップ	17
1. 仮想フロントエンド名を選択する	18
2. 仮想フロントエンドのプライマリ中間層を準備する	18
3. プライマリ中間層でのTNSエイリアスの使用の準備	19
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	20
5. セカンダリ・サイトのWLS for OCIをプロビジョニングする	25
6. 仮想フロントエンドのセカンダリ中間層を準備する	28
7. セカンダリ中間層でのTNSエイリアスの使用の準備	28
8. 必要な中間層ホスト・エイリアスの構成	29
9. WebLogic構成レプリケーションのためのステージング・マウントを構成する	31
10. プライマリWebLogicドメインでDRセットアップ・スクリプトを実行する	35
11. セカンダリWebLogicドメインでDRセットアップ・スクリプトを実行する	36
12. IDCS認証の使用時にDR構成を完了する	38
13. DRセットアップを検証する	43
ライフサイクル手順	44
構成変更のレプリケーション	44
スイッチオーバー	48
フェイルオーバー	50
検証のためのセカンダリ・サイトのオープン	50
構成レプリケーションのエンド・ツー・エンド検証	51
RTOおよびRPOの概要	53
予測されるRTO	53
予測されるRPO	54
ベスト・プラクティス	56
結論	57
付録A – Oracle RAC DBシステムに関する考慮事項	58
Oracle RACを使用する場合のOracle WLS for OCIエディションについて	58
Oracle RAC DBシステムを使用する場合のネットワークに関する考慮事項	58
CRSデータベース・サービスの作成と使用	58

その他のベスト・プラクティスの適用	59
付録B – 手動構成されたData GuardでのDBシステム・バックアップ	61
付録C – その他のライフサイクル運用	63
Oracle WLS for OCIのDR環境のバッチ適用	63
DBFSウォレットの再作成方法	64
スタンバイ・サイトでコンピュータ・インスタンスを停止させる場合	64
DR環境でのスケール・アウト手順とスケール・イン手順	65
スタンバイDBシステム再作成後のWLS for OCI DRの再構築	69
付録D – その他のベスト・プラクティス	71
TLOGおよびJMSストアでのJDBCストアの使用	71
付録E – 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・リカバリ	72
トポロジの説明	72
BVレプリケーションDRモデルの考慮事項	73
BVレプリケーションDRモデルの前提条件	75
BVレプリケーションDRモデルの要件	75
BVレプリケーションDRモデルのセットアップ・プロセス	77
BVレプリケーションDRモデルのライフサイクル操作	81
OCI CLIコマンドを使用したブロック・ボリューム・レプリケーション・タスクの自動化	88
付録F – プライマリでの追加スタンバイ・データベースの使用	89
構成前の手順	89
リージョン間のDR構成	90
構成後の手順	90
ローカル・データベースのスイッチオーバー	94

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、各種アプリケーション）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・プラクティスを実装することは、すべてのOracleデプロイメントにおいて重要な要件の1つとなっています。Oracle WebLogicとOracle Databaseには、プロセス停止の検出と再起動、クラスタ化、サーバーの移行、クラスタウェアの統合、GridLink、ロードバランシング、フェイルオーバー、バックアップとリカバリ、ローリング・アップグレード、ローリング構成の変更など、豊富な高可用性機能が含まれています。これらの機能により、アプリケーション環境が計画外停止時間から保護され、計画停止時間が最小限に抑えられます。

Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WLS for OCI) は、Oracle Marketplaceを介してプロビジョニングされるPlatform as a Service (PaaS) コンピューティング・プラットフォーム・ソリューションです。このソリューションを使用すると、Oracle Cloud InfrastructureにJava Enterprise Edition (Java EE) アプリケーション環境を素早く作成できます。標準的なOracle WLS for OCIデプロイメントでは、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）を1つのデータベースに格納します。Oracle WLS for OCIは、JRFが有効化されたドメインとして構成することで、Oracle Application Development Framework (Oracle ADF) をサポートし、Oracle Fusion Middleware Controlコンソールを使用して管理と監視を行うことができます。この場合、WebLogicはデータベースを使用して必要なスキーマ（Oracle Platform Security Servicesの情報など）を格納します。トランザクション一貫性を確保し、管理を簡素化するために、アプリケーション・データとWebLogic固有のスキーマは、通常は同じデータベースに格納されます。

すべての重要なシステムは、予期せぬ障害や自然災害から保護する必要があります。Oracle WLS for OCIなどのクラウドにデプロイされたシステムも、同じように保護する必要があります。フロントエンドのアクセス（ロードバランサーベースのフロントエンド・ホスト名など）、中間層（Oracle WebLogic）、およびデータ層（Oracle Database Cloud Service）の保護も必要です。そのためのソリューションとして、本番サイトとは地理的に異なる場所にあるOracle Cloud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数とリソース数は、本番サイトよりも少ない場合がありますが、オラクルでは、同じ容量でミラー構成を実行することを推奨しています。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、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 WLS for OCIのディザスタ・リカバリでは、Oracle Fusion MiddlewareとOracle Databaseに搭載されている高可用性機能と障害からの保護機能を活用して、**リカバリ時間目標 (RTO) とリカバリ・ポイント目標 (RPO) を最適化**します。クラウド・ディザスタ・リカバリ (DR) 構成に特有の考慮事項もありますが、Oracle Fusion Middleware (FMW) とOracle Databaseを使用する他のデプロイメントの場合と同じOracle MAAベスト・プラクティスがこの構成にも適用されます。このOracle MAA構想では、Oracle MAAベスト・プラクティスについて詳しく説明し、Oracle WLS for OCIに対応したDRをデプロイする手順の概要を示します。Oracle WLS for OCIのディザスタ・リカバリ・ソリューションは、Oracle WebLogic Serverドメインの構成ファイルと、使用するデータベースのランタイムとメタデータをレプリケートすることで実現します。アプリケーションによっては、他の構成ファイルをレプリケートすることが必要な場合もあります。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 WLS for OCIで使用するOracle Database Cloud Serviceの障害からの保護機能は、Oracle Data Guardを介して提供されます。

本書の対象読者は、**Oracle Cloud Infrastructure**、**Oracle WebLogic Server**、**Oracle Database**、**Oracle Data Guard**、**Oracle Databaseバックアップ/リカバリ**に関する知識を持つ技術者です。

Oracle WLS for OCIのディザスタ・リカバリ・ソリューション

トポロジの説明

Oracle WLS for OCI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。Oracle WLS for OCIドメイン・スタック、ロードバランサ、Oracle Cloud Infrastructure DBシステムで構成された**プライマリ・システム**が**特定のリージョン**にあり、Oracle WLS for OCIドメイン・スタック、ロードバランサ、Oracle Cloud Infrastructure DBシステムで構成された**セカンダリ・システム**が**別のリージョン**にあります。この同じトポロジを、複数の可用性ドメインが設定された1つのOracle Cloudデータセンターの範囲内に実装することもできますが、ディザスタ・リカバリによる保護を最大化するために、異なるリージョンを使用することが強く推奨されます。

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、アッシュバートとフェニックスは、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。

プライマリとセカンダリのOracle Cloud Infrastructure DBシステムには、[Data Guard](#)が構成されます。**Data Guard**の機能を頼りに、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。

セカンダリのOracle WebLogic Serverドメインは、プライマリ・ドメインのレプリカであり、使用する名前、スキーマ、パスワードなどは同じですが、セカンダリデータベースを参照します。WebLogic Serverのリッスナー・アドレスには、プライマリ中間層のホスト名が設定されているため、セカンダリ中間層のホストでは、セカンダリIPに解決できるように、該当するエイリアスがホスト・ファイル内に作成されます。

フロントエンドでは、システムで実行されるアプリケーションにアクセスするため、**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アドレスを指します。スイッチオーバーの場合、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。フロントエンド名は必ず、スイッチオーバーごとに、プライマリ・ロールを持つサイトのLBR IPを参照する必要があります。

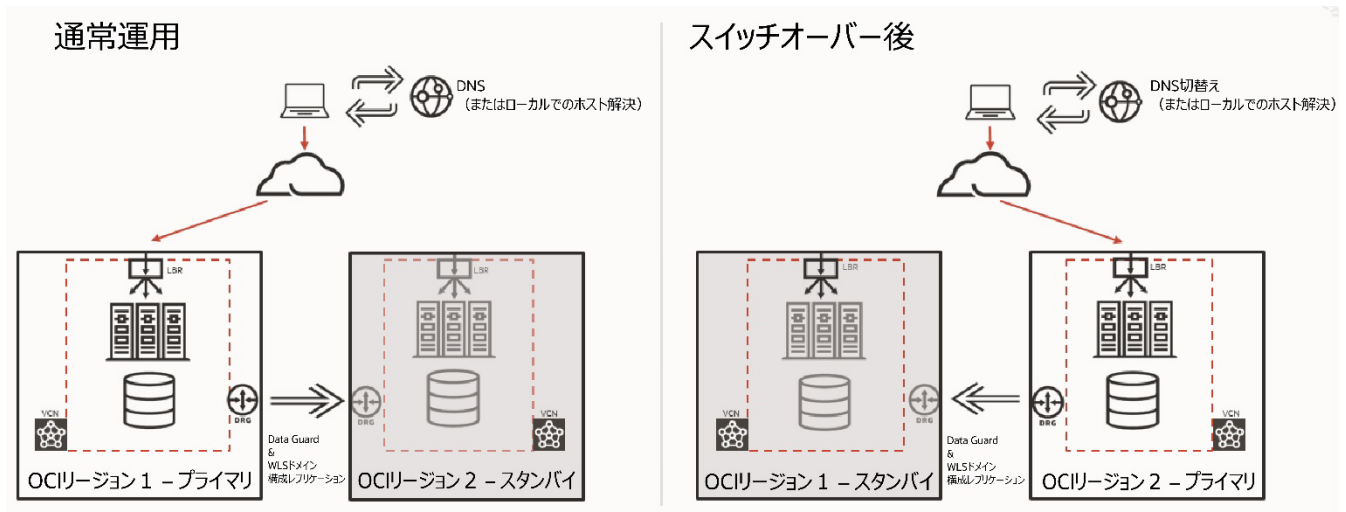


図1. Oracle WebLogic for OCIのディザスタ・リカバリ・トポロジ

通常業務では、スタンバイデータベースは**フィジカル・スタンバイ**になります。マウント状態になっているか、Active Data Guardを使用する場合は読み取り専用モードで開かれています。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

¹ Oracle WLS for OCIは、複数の可用性ドメインを持つリージョンで、可用性ドメインの障害に対する中間層の保護を標準提供しています。この保護を有効にするには、Oracle WLS for OCIをリージョン・サブネットにデプロイする必要があります。このタイプのネットワークにデプロイすると、Oracle WLSクラスタ内のノードは、利用できるすべての可用性ドメイン間で均等に分散されます。ただし注意点として、Oracle Databaseでは異なる可用性ドメインに同じRACクラスタのインスタンスを配置する構成はサポートされていません。そのため、リージョンを横断したDR保護を使用しない場合でも、可用性ドメイン・レベルの障害からデータベース層を保護するために、Oracle Data Guardを使用する必要があります。

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DRのセットアップとライフサイクルの手順の実行中、一部の動作の場合には、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**データベースは、その全体が更新可能なスタンバイデータベースです。スナップショット・スタンバイデータベースは、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は、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。

データベース内に存在するすべての情報が、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、MDS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。各サイトのローカル・ファイル・システムに格納されたWebLogicドメインの構成も、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。このレプリケーションは必須であり、初期DRセットアップ中に実行されます。また、システムのライフサイクル中も、プライマリ・ドメインで構成変更メンテナンスが実行されるたびに必要になります。オラクルは、このWebLogicドメイン構成をレプリケートする方法として、**Oracle Database File System (Oracle DBFS)**、**OCI File Storage Service (FSS)** と**rsync**、**リージョン間ブロック・ボリューム・レプリケーション**の3つの方法をサポートしています。これらはすべて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ベースの手法**では、ドメイン構成のコピーがDBFSファイル・システムのマウント・ポイントにステーキングされ、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あるファイル・システムで、NFSマウントのようにマウントすることができます。プライマリ・ドメインの構成がDBFSマウントにコピーされ、これが基盤のData Guard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、中間層ホストでスタンバイデータベースから同じDBFSマウント・ポイントをマウントできます。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。

この手法は、**Data Guardレプリケーションの堅牢性**を利用したものです。どのようなシナリオでも使用できますが、リージョン間の待機時間が中程度または長いDRシナリオには強く推奨されます。

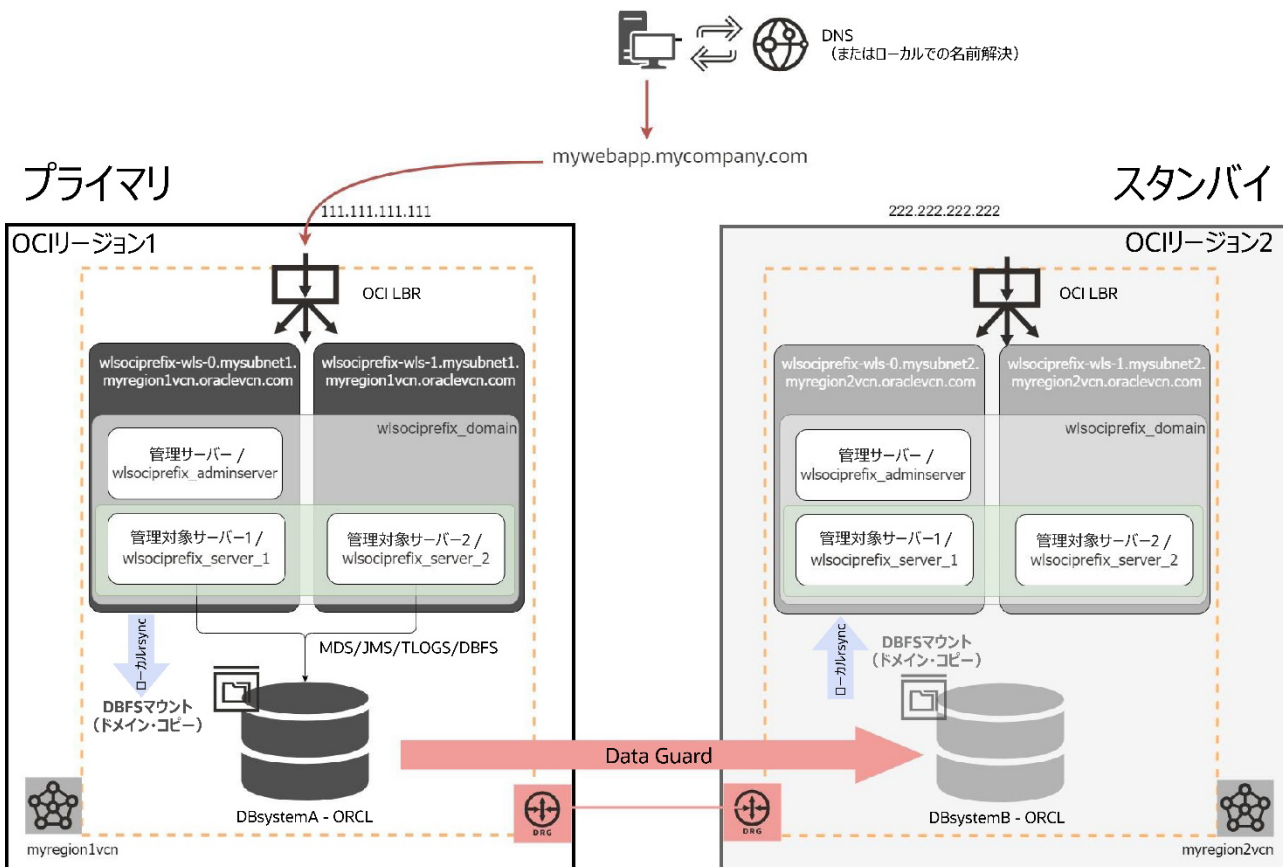


図2. Oracle WLS for OCIのDRトポロジ全体図 (WLSドメインの構成レプリケーションにはDBFSベースの手法を使用)

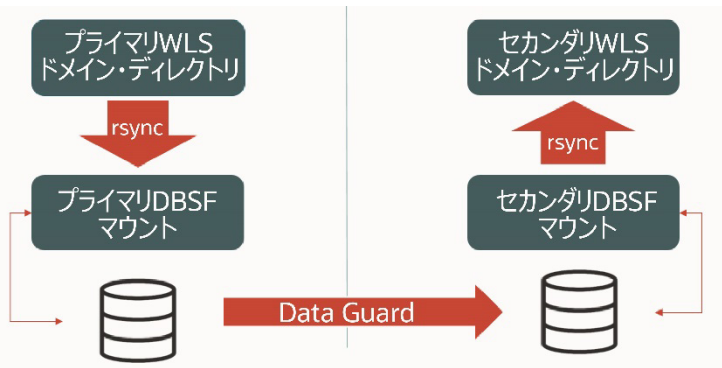


図3. DBFSベースの手法を使用したWLSドメイン構成レプリケーションの論理フロー・ダイアグラム

- OCI File Storage Service (FSS) /rsync手法**では、プライマリとスタンバイに1つずつある2つのファイル・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の場合と同様に、中間の“ステージング”ポイントとしてファイル・システムを使用します。そのために、OCI FSマウントを各リージョンで使用します。プライマリ・ドメインの構成をレプリケートするため、はじめにWLSドメインがローカル/ステージングOCI FSマウントにコピーされ、次にrsyncによってリモートOCI FS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 FS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。この手法は、データセンター間の待機時間が短く、接続の信頼性が高い場合に推奨されます。Oracle OCIリージョン間で動的ルーティング・ゲートウェイを使用して通信するケースがこれに当てはまります。

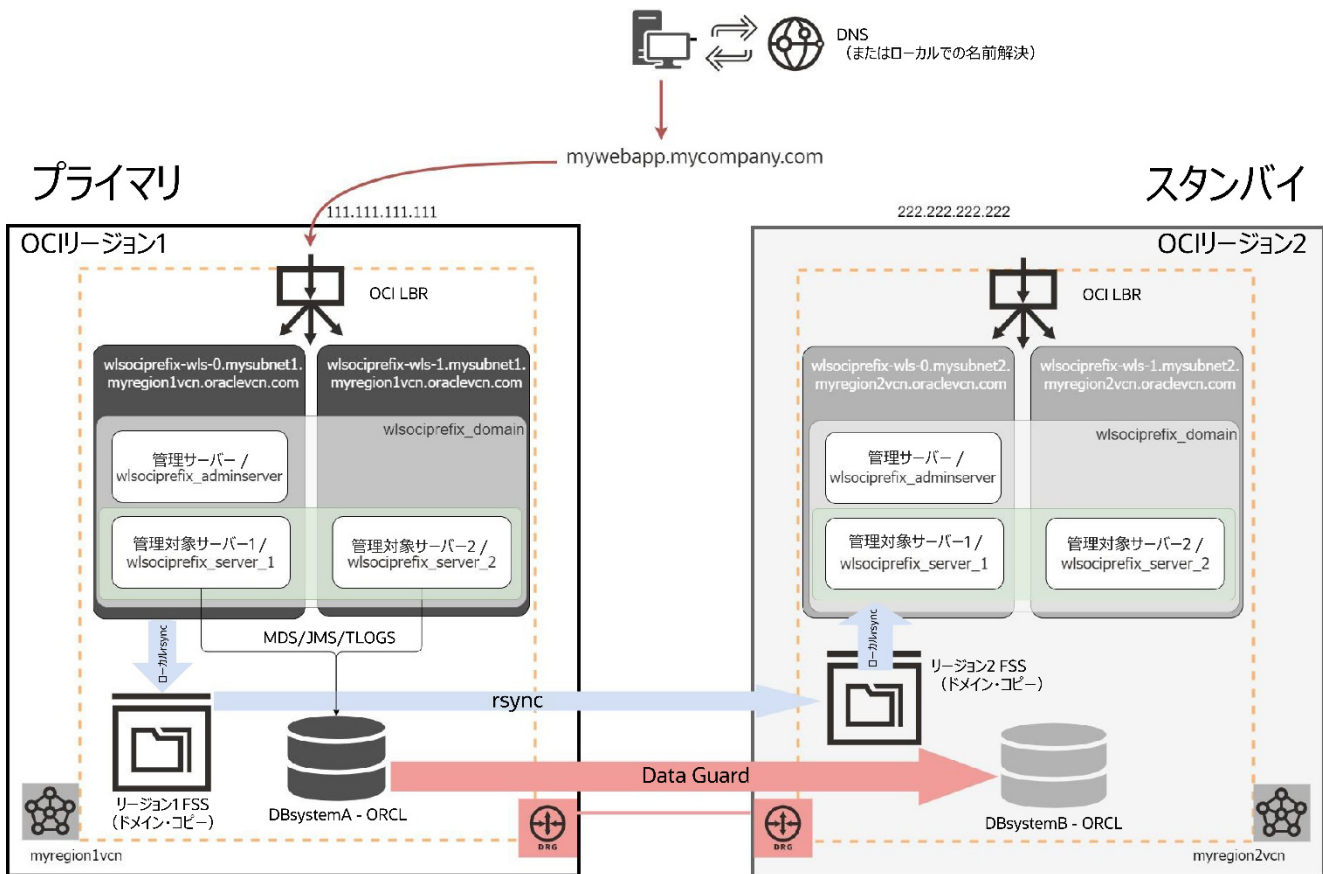


図4. Oracle WLS for OCIのDR/TO全体図（WLSドメインの構成レプリケーションにはOCI FS/rsync手法を使用）。

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・サイトのいずれかの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。

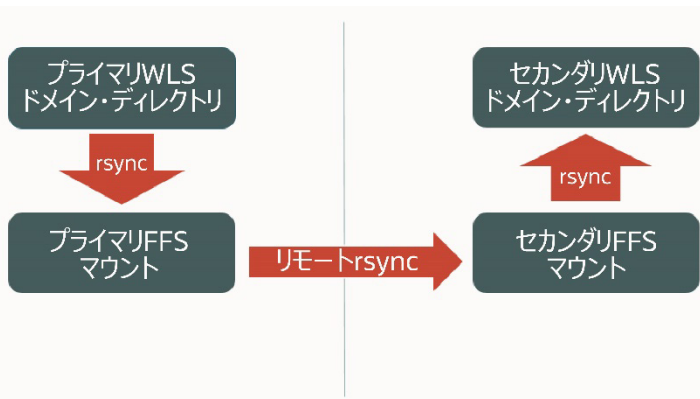


図5. OCI FS/rsyncの方法を使用したWLSドメイン構成レプリケーションの論理フロー・ダイアグラム

FSS/rsyncと比べると、**DBFS手法**は、Oracle Driverの再試行ロジックにより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。DBFSの方法はあらゆるシナリオで使用でき、データセンター間の待機時間が中程度であるか長くかかるDRシナリオの場合に強く推奨されます。ただし、構成レプリケーションにOracle DBFSを使用すると、セットアップ、データベース記憶域、ライフサイクルの観点で追加の影響があります。

FSS/rsyncの方法は、保守と構成が容易です。ただし、この方法は、動的ルーティング・ゲートウェイを使用するOracle OCIデータセンター間のDRシナリオでのみ推奨されます。データセンター間の通信で、たとえばパブリック・インターネットが使用される場合は、十分短い待機時間とならず、FSS/rsyncの動作が信頼できなくなる可能性があります。また、FSS/rsync手法では、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担当者にお問い合わせください）。

ドメイン構成のレプリケートにどの手法を使用するかにかかわらず、スタンバイ・データベースは、通常業務中にシャットダウンされると、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は請求額を減らすために停止させることが可能ですが、それらの中間層ホストを起動していない場合、プライマリ・サイトからレプリケートされるWLSドメイン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。スイッチオーバーが発生したときに、中間層ホストを起動してプライマリの変更を同期する必要があると、RTOが長くなります。セカンダリの中間層ホストを起動しておくこと（マネージャ・サーバーは停止）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。

- 3つ目のDRモデルは、**リージョン間ブロック・ボリューム・レプリケーション**をベースとした方法です。このアプローチでは、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[リージョン間ボリューム・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。この機能によって、ブロック・ボリュームとブート・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を継続的に自動で実行できます。構成レプリケーションでステージング・ロケーションを使用しないので、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²。このモデルの考慮事項、トポロジ、セットアップに関する具体的な詳細については、「[付録E – 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・リカバリ](#)」を参照してください。

前提条件

以下の考慮事項は、WebLogic Cloudのディザスタ・リカバリをセットアップする上での重要な側面です。

² 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。

WebLogicのエディション

OCIのWebLogicには次の3つのエディションがあります。

- [Oracle WebLogic Server Standard Edition](#)
- [Oracle WebLogic Server Enterprise Edition](#)
- [Oracle WebLogic Suite](#)

機能の詳細については、[ライセンス情報ユーザー・マニュアルの「Oracle WebLogic Server」](#)を参照してください。Active GridLink for RACと呼ばれるOracle WebLogic Server GridLinkデータソースは、ライセンス供与されるWebLogic Suiteの一部としてのみ使用できます。その他のエディションでは、Active Gridlinkが含まれていないため、マルチ・データソースを使用してOracle RACデータベースに接続します。Oracle RACを使用する場合は、GridLinkデータソースを使用するのがMAAベスト・プラクティスです。

このドキュメントは、Oracle RAC DBシステムを使用している場合を除き、Oracle WebLogic SuiteとOracle WebLogic Server Enterprise Editionに適用されます。**データベースがOracle RAC DBシステムの場合は、このドキュメントは、Oracle WebLogic Suiteを使用していることを前提とします。**このDRのセットアップでは、Oracle RACシナリオ向けのこのエディションのみがサポートされます。このエディションのみが、GridLinkデータソースを使用するためです。

認証

Oracle WLS for OCIは、"Oracle Identity Cloud Serviceを使用した認証の有効化"オプションを使用してプロビジョニングできます。その場合、デフォルトのOracle WebLogic Serverのアイデンティティ・ストアではなく、Oracle Identity Cloud Service (Oracle IDCS) を使用してドメインのユーザーを認証します。Oracle Cloudのアプリケーション・ゲートウェイ・エージェントとIdentity Cloud Servicesなどのコンポーネントがシステムに追加されます。

このドキュメントでは、**デフォルトのOracle WebLogicアイデンティティ・ストア**を使用するWLS for OCIシステムについて、およびDRイベントのコンテキストでWLS for OCIに**IDCS認証を使用することによる影響についても説明**します。ただし、IDCS固有の障害からの保護機能は本書の対象外です³。

これによって、WebLogicアイデンティティ・ストアをカスタマイズして別の外部LDAPを使用できなくなるわけではありません（WebLogicがOCIでサポートする限り）。WebLogicプロバイダの構成はドメイン構成内であれば、スタンバイ・ドメインにレプリケートされます。スタンバイのWebLogicシステムから、設定された外部LDAPのURLに到達可能な場合は、そのLDAPを使用できます。外部LDAPサービス向けのディザスタリカバリ・ソリューションは、WLS DRに関する本書の範囲外であり、個々のLDAP製品によって提供されます。

ロードバランサ

ここで示すディザスタリカバリ・ソリューションは、**Oracle WLS for OCIスタックがOCIロードバランサを使用して構成されてい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、クラスターに複数のサーバーが含まれる場合には必須です。

Oracle WLS for OCIの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、プライベートまたはパブリック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。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。

³ IDCSサポートに問い合わせせてご自身の地域でのIDCSディザスタリカバリ・ソリューションの提供状況を確認し、IDCS自体を障害から保護する方法の詳細を入手してください。

同様のトポロジがプライベートOCIロードバランサでも使用されます。プライベートOCIロードバランサは、システムをインターネットから隔離するために使用されますが、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。そのサブネットは、使用するサブネットの範囲に応じて、範囲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。サブネットが範囲内のリージョンに配置され、リージョン内に複数の可用性ドメインが存在する場合、プライマリとスタンバイの各ロードバランサは、別の可用性ドメインに配置されます。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）。

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ロードバランサをリージョン・サブネット内に配置します。

これは、デザイナー・リカバリ構成として十分な構成です。プライマリ・リージョンおよびセカンダリ・リージョンのロードバランサは、それぞれのローカルWebLogicクラスタにのみルーティングすればよいため、ロードバランサ間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でデフォルト構成が変更された場合（リスナー、バックエンド、または証明書が追加された場合など）に限り、セカンダリ・リージョンのロードバランサでも構成を手動で変更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 WLS for OCIは、プロビジョニング中に既存のデータベースと関連付けることができます。このようにOracle Cloud Infrastructureデータベース・システムをOracle WebLogic Cloudと一緒に使用するのは非常に一般的なシナリオです。

- 顧客が**Fusion Middleware Controlコンソール**などの機能を使用して、ドメイン、OPSSセキュリティ、ADF Frameworkなど（“JRFが有効化された”ドメイン）を監視、構成、管理したいと考えている場合、データベースは必須です。
- 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は、**MAAベスト・プラクティス**です。デザイナー・リカバリのトポロジでは特に、Data Guardレプリケーションによってセカンダリがプライマリになる場合に、セカンダリ・サイトでもこの情報が自動的に利用可能になります。
- アプリケーションの実行時データは、通常、データベースのカスタム・スキーマ内に保存されます。

以上のことから、このデザイナー・リカバリ・ソリューションでは、Oracle WLS for OCIに**Oracle Cloud Infrastructure DBシステム**が構成されていることを前提としており、本書はそのシナリオに適用されます。これにより、Oracle Database File System（Oracle DBFS）などの便利なデータベース機能を使用できるようになります。初期のデザイナー・リカバリ・セットアップ時とライフサイクル管理中、プライマリ・ドメインからセカンダリ・ドメインにWebLogic Serverドメイン構成をレプリケートする際、Data GuardとともにDBFSを使用できます（DBFSベースの手法を使用する場合）。

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⁴。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、後から追加する予定がある場合は、「[付録F – 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。

本書で説明するDRセットアップ手順は、**シングル・インスタンスおよびOracle RACデータベースのDBシステムとExadata Cloud Service (EXACS) データベース**で動作保証されています。Oracle WLS for OCIで使用されるDBシステムが**Oracle RACデータベース**である（つまり、複数のデータベース・ノードがある）場合、詳しい手順とベスト・プラクティスについて、「[付録A – Oracle RAC DBシステムに関する考慮事項](#)」を参照してください。

Oracle OCI Autonomous Databaseは、このドキュメントの対象外です。OCI Autonomousのシナリオについては、プレイブック[Configure Oracle Fusion Middleware DR on Oracle Cloud with an autonomous database](#)に従ってください。

⁴ <https://docs.oracle.com/ja-jp/iaas/Content/Database/Tasks/usingdataguard.htm>の「Oracle Data Guardの使用」を参照してください。

要件

WebLogic Cloudのディザスタリカバリをセットアップする際の重要な側面として、以下の要件があります。

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が、使用されるサイトに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・アドレスの名前を一意にする必要があります**、常にその時点でプライマリとなるシステムを参照する必要があります。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）を障害からの保護用の仮想フロントエンドとして再利用することが可能です。つまり、元のシステムで、“mywebapp.example.com”がプライマリのバニティURLとして使用されていた場合、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IPに再マッピングすることができます。

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、その他の商用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。WebLogic Cloudで仮想フロントエンド名を使用するための設定方法については、本書で説明します。

Weblogicリソース名プリフィックス

Oracle WLS for OCIスタックのプロビジョニング時に、“リソース名プリフィックス”を指定する必要があります。このプロパティは、WebLogic Serverのドメイン名、クラスタ名、WebLogicサーバー名、VMのホスト名など、すべてのリソース名の構成に使用されます。

このプロパティには、**プライマリとセカンダリのOracle WLS for OCIシステムで同じ値を設定する必要があります**。そうすることで、両方のシステムでWebLogicリソースの名前が一致します。同じ名前を使用することによって一貫性が保証され、これは、たとえばJMSメッセージやTlogのリカバリで必要になります。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。

1つのクラウド・テナンシー内の複数インスタンスで同じ“リソース名プリフィックス”を使用しても、インスタンスが作成されたリージョンまたはコンパートメントが異なる限り、何も問題はありません。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。

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。また、セカンダリ中間層のホストは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WebLogic管理ホストと通信する必要があります。

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、オラクル内部のネットワーク経由で通信できます。これは推奨されているアプローチです（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）。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、仮想クラウド・ネットワークごとにSecurity Listsで構成します（OCIコンソールで、「Core Infrastructure」→「Networking Section」→「Virtual Cloud Network」の順に選択）。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。

レプリケートされるデータの量は、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、同時実行性などと直接的な関連があります。構成をレプリケートする場合のDBFSの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セカンダリ・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、フォルト・トレラ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個のバケットのうち、損失は5バケット以下）と一貫した待機時間を実現します。詳しくは、[Oracle Cloudリージョン](#)を参照してください

WebLogic構成レプリケーションのためのステー징・ファイル・システム

本書では、サイト間でWLSドメイン構成をレプリケートする2つの手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징・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。リージョン間ブロック・ボリューム・レプリケーションに基づく手法は、ステー징・ファイル・システムを必要としない唯一の手法です。

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る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16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となります。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。**本書では、このアプローチを使用しています。**

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。

カスタム・ファイル

WebLogic Cloudで使用するWebLogic Serverドメイン構成の大半は、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WebLogic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) 。tnsnames.oraを含む**TNS管理フォルダはコピーから除外されます。**
- 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・デプロイメント (ear/warファイル、デプロイメント・プランなど) および**管理サーバーのWLSドメイン・ディレクトリ内にある** (一時データ以外の) あらゆるものを、本書で説明する手順を使用して、サイト間で最初に同期します。
- **weblogic_domain_name/config以下の構成**はすべて、WebLogicクラスタ機能ごとに同じサイトの他のノードに自動的に配布されます。管理対象サーバーが開始すると、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。**
- オラクルは、DRと関係なく**キーとIDに中央ストア**を使用することを推奨しています。たとえば、**非JRF対応ドメイン** (WLSバージョン14.1.1など) で、ノード・マネージャはKeystore Service (KSS) をIDキーストアとして使用しません。これは標準提供の\$DOMAIN_HOME/security/Demolidentity.jksを使用します。このファイルはホストごとに異なります。セカンダリでノード・マネージャに接続する場合の"ホスト名検証失敗"エラーを回避するには、DRセットアップの後に、Demolidentity.jksファイルをプライマリ・ノードwls-1からセカンダリ・ノードwls-1に、プライマリ・ノードwls-2からセカンダリ・ノードwls-2に、というようにコピーする必要があります。KSSを使用せず、JKS IDキーストア・ファイルをカスタマイズした場合も同じことが適用されます。そのため、これらのファイルのための中央のロケーションを使用することを推奨します。それを使用せず、自動的に配布も共有もされないフォルダに保存する場合には、手動での操作が必要になります (他のノードに自分でコピーする必要があります) 。これらのファイルが変更され、レプリケートされない場合、リスニング・アドレスの変更やDRなど、さまざまなシナリオでエラーが発生します。

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。**

スクリプトのダウンロード

このDR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**の“wls_mp_dr”および“app_dr_common”フォルダで入手できます。

- 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、**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。

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、特定のフォルダのダウンロードのみに使用できます。

ディザスタ・リカバリのセットアップ

本書では、開始時点で、プライマリ・サイトが稼働中であることを前提としています。プライマリ・サイトは、OCI DBシステム、Oracle WLS for OCIスタック、ロードバランサ（OCI LBR）で構成されています。この既存のプライマリ・システムに対するセカンダリDR構成を、地理的に離れたサイトに作成します。

プライマリ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）。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。

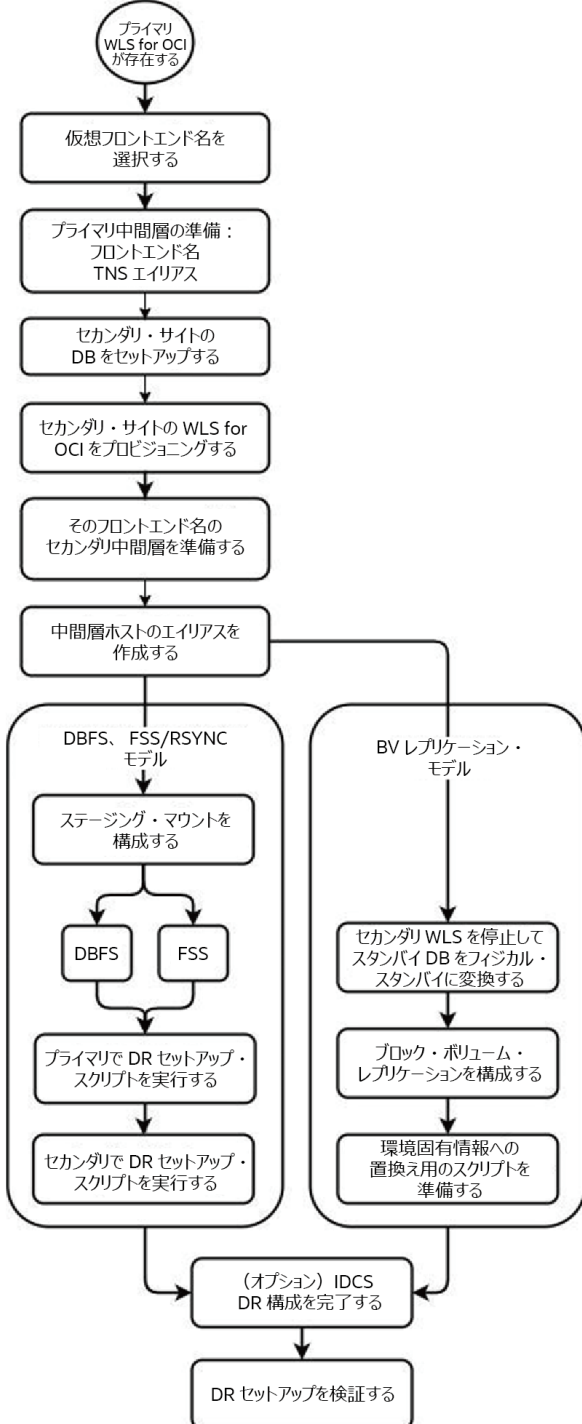


図6. 本書に記載したモデルのDRセットアップ手順を示すフロー・チャート

「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種類のDRモデルがあります。

次のポイントでは、DBFSモデルとFSS/rsyncモデルでのDRセットアップ方法について説明します。

リージョン間ブロック・ボリューム・レプリケーションに基づくDRモデルは、他の2つのモデルと大きく異なります（ステージング・ロケーションを使用しません）。このモデルについては、「付録E – 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・リカバリ」で別途説明しています。

1. 仮想フロントエンド名を選択する

Oracle WLS for OCIインスタンスが作成されると、プロビジョニングされたロードバランサは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

（“mywebapp.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に解決するには、次のポイントで説明するように、そのシステムのホスト・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。

e) 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、各サイ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WebLogicホストでこの処理を繰り返します。例：

```
[oracle@wlsociprefix-wls-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 mywebapp.mycompany.com
```

注：プライマリWLS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WebLogic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。

f) フロントエンド名をクラスタ・フロントエンドとして設定します。

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をフロントエンドのFQDN (“mywebapp.mycompany.com”など) に設定します。
- フロントエンドのHTTPおよびHTTPSポートに正しい値が設定されていることを確認します。
- 保存して有効化します。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。

Settings for wlsmkpl5_cluster

Configuration Monitoring Control Deployments Services Notes

General JTA Messaging Servers Replication Migration Singleton Services Scheduling Overload Health Monitoring **HTTP** Coherence

Save

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: mywebapp.mycompany.c

Frontend HTTP Port: 80

Frontend HTTPS Port: 443

Save

図7：クラスタのフロントエンド・ホストの構成

g) **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呼出しに使用されるURLは、各サイトで使用されるIPまたはホスト名にも依存しないようにする必要があります。host:port,host:portJNDIの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。⁵

同様に、システムで使用されるその他のすべてのカスタムJNDI URLも更新し、スイッチオーバーやフェイルオーバーが発生したときにセカンダリ・サイトでもURLが有効であるようにする必要があります。

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。

- 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、接続文字列がおのこのデータソースと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。
- 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、OCIコンソールを使用して手動で構成を行い、以下のようにjpsファイルを編集します。**
 - **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。

⁵ 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。ドメインの外部にあるt3/rmiクライアントではこのアプローチを使用することができず、セカンダリ・サイトへの切替えでは、host:portリスト形式の適切なDNSマッピングを使用する必要があります。JNDI InitialContextの取得にTCPロードバランサを使用することはできませんが、JMSクライアントからの後続リクエストはhost:portに直接接続するため、この場合もセカンダリ・サイトIPへのDNSマッピングが必要になります。

- **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.vcnash226.oraclevcn.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

The screenshot shows the 'Configuration' page for a 'Connection Pool'. The 'Properties' section is expanded, showing the following configuration:

```
user=SOAMPDR19_ESS
oracle.net.CONNECT_TIMEOUT=10000
oracle.net.tns_admin=/u01/data/domains/soampdr_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。
<property name="oracle.net.tns_admin" value="/u01/data/domains/wlsociprefix_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・サイトでのデータベースのセットアップ

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Data Guardの有効化機能を使用できない場合も（DBシステムの各フレーバー/エディションでData Guardのリージョン全域の機能を使用できるかどうかについては、Oracle Cloud Infrastructureドキュメントの「[DBシステムでのOracle Data Guardの使用](#)」を参照してください）、本書で提供しているスクリプトを使用して、手動でData Guardを構成できます。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。

EXACSDデータベースを使用する場合は、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する必要があります。「[3.3 EXACCSに関する考慮事項](#)」に進んでください。

3.1 オプション1) OCIコンソールを使用したData Guardの構成

OCIコンソールを使用してData Guardを有効化する場合、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。ただし、いくつかの要件があります。両方のDBシステムが同じコンパートメント内に存在すること、両方のDBシステムが同じシェイプ・タイプであること、DBシステムが異なるリージョンに存在する場合はそれらがリモートVCNピアリングを介して接続されていることなどです。これらの要件について、詳しくはOracle Cloud Infrastructureドキュメントの「[DBシステムでのOracle Data Guardの使用](#)」を参照してください。

プライマリデータベースに対してData Guardを有効にするには、OCIコンソールにログインし、プライマリDBシステムに移動して、プライマリデータベースをクリックします。Data Guardは、「Data Guard Associations」セクションで有効化できます。セカンダリDBシステムの構成プロパティのほとんど（バージョンやDB名など）は、プライマリから継承されるため事前定義されていますが、一部の構成プロパティについては指定する必要があります。次の表に、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	自動化DGの要件
Oracle Cloudテナンシー	XXXX/paasmaa	YYYY/paasmaa	必ず同じ値
コンパートメント	XXXX/wlsdr	XXXX/wlsdr	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/efXT:PHX-AD-1	必ず異なる値
DBシステム名	XXXX/drdbw1mp1a	YYYY/drdbw1mp1b	必ず異なる値
シェイプ	XXXX/VM.Standard2.1	XXXX/VM.Standard2.1	必ず同じ値
仮想クラウド・ネットワーク	XXXX/wlsdrvcn1ash	YYYY/wlsdrvcn1pho	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/wlsdrvcn1ashAD1	XXXX / wlsdrvcn1phoAD1	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）
ホスト名プリフィックス	XXXX/drdbw1mp1a	YYYY/drdbw1mp1b	必ず異なる値
管理者パスワード	XXXX/acme1234#	XXXX/acme1234#	必ず同じ値

3.2 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・テナンシー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。

3.2.1 セカンダリ・データベースのプロビジョニング

注：OCIコンソールを使用してData Guardを有効化している場合は、ここに示す手順をスキップし、セクション4「セカンダリ・サイトでのOracle WLS for OCIのプロビジョニング」に進んでください。

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。そのためには、スタンバイを作成する前にプライマリ・システムにパッチを適用することが必要になる場合があります（特に、プライマリ・システムの使用期間が長い場合）。オラクルでは、コンピュータ・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テナンシー	XXXX/paasmaa	YYYY/paasmaa	異なる値でも可
コンパートメント	XXXX/wlsdr	YYYY/wlsdr	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/drdbwmp1a	YYYY/drdbwmp1b	必ず異なる値
シェイプ	XXXX/VM.Standard2.1	XXXX/VM.Standard2.1	必ず同じ値
合計ノード数	N/1	N/1	必ず同じ値
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	必ず同一であること
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	異なる値でも可
SSH公開鍵	XXXX	YYYY	必ず同じ値
仮想クラウド・ネットワーク	XXXX/wlsdrvcn1ash	YYYY/wlsdrvcn1pho	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/wlsdrvcn1ashAD1	XXXX/wlsdrvcn1phoAD1	必ず異なる値
ホスト名接頭辞	XXXX/drdbwmp1a	YYYY/drdbwmp1b	必ず異なる値
データベース名	XXXX/ORCL	XXXX/ORCL	必ず同じ値
データベース・バージョン	XXXX/19c	XXXX/19c	必ず同じ値

PDB名	XXXX/PDB1	XXXX/PDB1	必ず同じ値
管理者パスワード	XXXX/acme1234#	XXXX/acme1234#	必ず同じ値
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

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。

3.2.2 プライマリ/セカンダリ間のData Guardの構成

注：Data GuardがOCIコンソールを使用して有効化されている場合は、これらの手順を省略し、セクション4「[セカンダリ・サイトのWLS for OCIをプロビジョニングする](#)」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。

- a) 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に到達できる必要もあります。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。

それぞれの通信を確認するには、ncコマンドを使用します（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@drdDBa ~]$ nc -vw 5 -z <secondary_db_ip> 1521
```

正しい出力の例：

```
[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⁶をダウンロードし、ソリューション・ブレイク「[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。

⁶ https://github.com/oracle-samples/maa/raw/main/dg_setup_scripts/dg_setup_scripts.zip

- 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)
```

3.3 EXACSに関する考慮事項

Exadata Cloud Serviceデータベースを使用している場合は、以下の推奨事項に従ってください。

- 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、Data Guardをセットアップします。
- EXACS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EXACS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。例：

```
[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
```

ただし、このサービスはスタンバイEXACS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.
```

EXACS Data GuardでOracle WLS for OCIのDRに適切なサービス構成を取得するには、以下の手順に従います。

- スタンバイEXACS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WLS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。例：

```
[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。**例：

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

- ExaCSデータベースはOracle RACデータベースです。RACシナリオでのセットアップについて、詳しくは「[付録A – Oracle RAC DB システムに関する考慮事項](#)」を参照してください。

5. セカンダリ・サイトのWLS for OCIをプロビジョニングする

セカンダリWebLogicはセカンダリDBシステムを参照するように作成されます。セカンダリDB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ります。

このため、セカンダリWebLogic for OCI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、読取り/書込みモードで開かれます。これにより、セカンダリWebLogic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@drdbw1mp1a ~]$dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> CONVERT DATABASE "secondary_db_unqname" to SNAPSHOT STANDBY;
Converting database "secondary_db_unqname" to a Snapshot Standby database, please wait...
Database "secondary_db_unqname" converted successfully
```

次に、「[Using Oracle WebLogic Server for Oracle Cloud Infrastructure documentation](#)」の説明に従って、セカンダリ・サイトのWebLogic Cloudシステムを作成し、セカンダリDBシステムを参照します。

スタック名は異なっても構いませんが、**リソース名プリフィックス**にはプライマリ・ロケーションで使用しているの**まったく同じもの**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ようお奨めします。

セカンダリ・ロケーションにプロビジョニングするWLS for OCI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。⁷

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。

⁷ WLS for OCIのプロビジョニングでは、2つの最新のPSUが提供されます。つまり、顧客は同じPSUを3〜6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WLS for OCIのPSUをプロビジョニングで使用できない場合、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WLS for OCIにパッチを適用する必要があります。詳しくは、<https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/patched-included-oracle-weblogic-cloud.html>を参照してください。

WebLogic Cloud 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/12.2.1.4.200414.03	XXXX/12.2.1.4.200414.03	必ず同一であること ⁷
スタック名	XXXX/wlsociprefixprim	YYYY / wlsociprefixstby	異なる値でも可
リソース名プリフィックス	XXXX/wlsociprefix	XXXX / wlsociprefix	必ず同一であること
SSH公開鍵	XXXX / my_public_key.pub	XXXX / my_public_key.pub	必ず同じ値
OCIポリシー	オンまたはオフ	オフ	プライマリ用のポリシーが存在し、スタンバイに適用可能であること。この表の下 「ポリシーに関する注記」を参照
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
IDCSを使用した認証の 有効化	オンまたはオフ	オンまたはオフ	必ず同じ値
ネットワーク・コンパートメント	XXXX/wlsdr	YYYY/wlsdr	異なる値でも可（ただし通常は同じ値）
VCN	XXXX/wlsvcnSite1	YYYY / wlsvcnSite2	必ず異なる値
サブネット	XXXX/wlsvcnSite1Subnet1	YYYY / wlsvcnSite2Subnet1	必ず異なる値
WebLogic Serverの シェイプ ⁸	XXXX/VM.Standard2.1	XXXX/VM.Standard2.1	必ず同じ値
管理者ユーザー名	XXXX/weblogic	XXXX / weblogic	必ず同じ値
管理者パスワード (シークレット)	XXXX / secretadmin_a	YYYY / secretadmin_b	パスワードが同じであること（ただし、セカン ダリ・ポールの場合はセカンダリ・リージョン内に 配置されるため、シークレットは異なる可能性 がある）
JRFを使用した プロビジョニング	XXXX / チェックを入れる	XXXX / チェックを入れる	必ず同じ値
データベース戦略	データベース・システム	データベース・システム	必ず同じ値
DBシステムの コンパートメント	XXXX/wlsdr	XXXX/wlsdr	異なる値でも可（ただし通常は同じ値）
DBシステム・ネットワーク	XXXX/wlsvcnSite1	YYYY / wlsvcnSite2	必ず異なる値 (リージョンは異なると想定される)
DBシステム	XXXX/drdbwImp1a	YYYY/drdbwImp1b	必ず異なる値
DBバージョン	XXXX/19	XXXX / 19	必ず同じ値
システム内のデータベース	XXXX/ORCL	XXXX/ORCL	必ず同じ値

PDB	XXXX/PDB1	XXXX/PDB1	必ず同じ値
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 (シークレット)	XXXX/ secretdb_a	XXXX / secretdb_b	パスワードが同じであること (ただし、ポールドはセカンダリ・リージョン内に配置されるため、シークレットは異なる可能性がある) Vaultサービスは、プロビジョニング中のパスワードの暗号化および復号化のみに使用される。WebLogicの実行時/ライフサイクル・タスクには使用されない
サンプル・アプリケーションのデプロイ	XXXX / チェックを入れる	XXXX / チェックを入れる	必ず同じ値
ポートの構成	XXXX	XXXX	必ず同じ値であること ポートをカスタマイズする場合は、必ずプライマリと同じポートを使用すること
コンピュート・シェイプ	XXXX	XXXX	必ず同じ値
WebLogic Serverノード数	N/2	N / 2	必ず同じ値
(IDCSがオンの場合) IDCSホスト・ドメイン名	XXXX/ identity.oraclecloud.com	XXXX/ identity.oraclecloud.com	必ず同じ値
(IDCSにチェックを入れた場合) IDCSポート	XXX/443	XXX / 443	必ず同じ値
(IDCSにチェックを入れた場合) IDCSテナント	XXXX/idcs-5e890de5903jr...	XXXX/ idcs-5e890de5903jr...	必ず同じ値
(IDCSにチェックを入れた場合) IDCSクライアントID	XXXX/f88aa64hh8hgd...	XXXX / f88aa64hh8hgd...	必ず同じ値
(IDCSがオンの場合) IDCSクライアント・シークレットのシークレット OCID	XXXX / secret1	YYYY / secret2	IDCSクライアント・シークレット (暗号化されていない値) は同じである必要があるが、シークレットは異なってもよい (各リージョンのポールド内に作成)
(IDCSにチェックを入れた場合) IDCSリダイレクト・ポート	XXXX/9999	XXXX / 9999	必ず同じ値

ポリシーに関する注記: “OCI Policies”フラグがオンになっている場合、プロビジョニング・ジョブで、WLS for OCIリソースの作成に必要なOCIポリシーと動的グループが自動的に作成されます。OCIポリシーと動的グループの名前は、インスタンスのプリフィックスに基づいて作成されます。プライマリのプロビジョニング時にこのフラグをオンにしており、セカンダリ・インスタンスをプロビジョニングするときに再度フラグをオンにすると、プロビジョニング・ジョブは、既存の名前でOCIポリシーと動的グループを作成しようとする。この場合、名前の競合が発生して、セカンダリのプロビジョニングが失敗します。この問題を回避するには、次のガイドラインに従います。

- プライマリのプロビジョニング時に“OCI Policies”フラグをオンしていない場合、Webサイト (https://docs.oracle.com/cd//E83857_01/paas/weblogic-cloud/user/you-begin-oracle-weblogic-cloud.html#GUID-C34C8258-5382-4386-8A38-4B5941CD3EB5) の説明に従って、OCIポリシーと動的グループは手動で作成しているはず。その場合、以下を確認します。
 - 動的グループにセカンダリ・リージョンのリソースが含まれていることを確認します。コンパートメント内のすべてのリソースに対して動的グループを作成しており、セカンダリ・コンパートメントが同じである場合、セカンダリWLS for OCIのリソースは動的グループに含まれています。
- プライマリのプロビジョニング時に“OCI Policies”フラグをオンにした場合、OCIポリシーと動的グループはすでに作成されています。以下の処理を実行します。
 - プライマリ・インスタンスで作成された動的グループを編集します (“<prefix>-wls- principal-group”のような名前、ルート・コンパートメント内に作成されています)。このタグのマッチング・ルールを削除します。残りのルールには、コンパートメント内のすべてのリソースが含まれます。

その上で、セカンダリのプロビジョニング時に「OCI Policies」フラグをオフにします。新しいリソースはすでに既存の動的グループに含まれているため、以前に作成されたポリシーで十分です。

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

注：データソースのDB接続文字列が、推奨されるフォーマットに準拠していることを確認してください。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接続文字列は次のとおりです。

```
jdbc:oracle:thin:@//<db-scan-address>:<port>/<pdb_service_name>
```

データベースがRACの場合、データソースはGridLinkデータソースである必要があり、推奨されるDB接続文字列は次のとおりです。

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=<db-scan-address>)(PORT=<port>)))(CONNECT_DATA=(SERVICE_NAME=<pdb_service_name>)))
```

セカンダリのWebLogic for OCI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。そのため、このケースでは、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。DRセットアップが完了したあとで、これらのサーバーを起動できます。

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WebLogicホストでこの処理を繰り返します。例：

```
[oracle@wlsociprefix-wls-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
222.222.222.222 mywebapp.mycompany.com
```

WebLogicセカンダリ・クラスタのフロントエンド・アドレスはプライマリのWebLogicドメイン構成からコピーされるため、これを更新する必要はありません。

注：セカンダリWebLogic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WebLogic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のと同じエイリアスが含まれており、それがセカンダリ・データベース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エイリアスが含まれるようにします。

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・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WLS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、プライベートDNSビューにエントリを追加する方法があります。

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/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ファイルのエントリが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。

この手法の短所は、すべてのWLS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。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「DR環境でのスケール・アウトおよびスケール・イン手順」を参照してください）。

このアプローチで、必要なホスト・エイリアスを作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。
- プライマリとスタンバイで、サーバーがリスニングする各WebLogicホストの**完全修飾ドメイン名（FQDN）を確認**します。FQDNを見つけるには、ドメイン構成ファイルでリスニング・アドレスを探します。たとえば、プライマリ・ドメインでは以下のとおりです。

```
[oracle@wlsociprefix-wls-0 config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-0 config]$ grep listen-address config.xml
<listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
```

セカンダリ・ドメインでの例を次に示します。

```
[oracle@wlsociprefix-wls-0 config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-0 config]$ grep listen-address config.xml
<listen-address> wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com </listen-address>
<listen-address>wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
<listen-address>wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
```

また、それぞれの中間層ホストでコマンド“hostname --fqdn”を使用してFQDN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ファイル**を編集し、スタンバイ・ホストのFQDN（ドメイン名付きの“長い名前”）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> <long_and_short_hostnames_secondary_node2>
```

注：“短い”名前はプライマリ・ホストとスタンバイ・ホストで同じになると想定されます。

プライマリ中間層ホストの/etc/hostsエントリは、次の例のようになります。

```
[oracle@wlsociprefix-wls-0 config]$ 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 mywebapp.mycompany.com

# WLS DRの場合
10.0.2.10 wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com wlsociprefix-wls-0 wlsociprefix-wls-
0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
10.0.2.11 wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com wlsociprefix-wls-1 wlsociprefix-wls-
1.wlsdrvcnfra1ad2.wlsdrvcnfra1.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ファイル**を編集し、プライマリ・ホストのFQDN（ドメイン名付きの“長い名前”）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>
```

FQDN（ドメイン名付きの“長い名前”）と非完全修飾ホスト名（ドメインなしの“短い名前”）セカンダリ中間層ホストの/etc/hostsエントリは、次の例のようになります。

```
[oracle@wlsociprefix-wls-1~]$ 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
222.222.222.222 mywebapp.mycompany.com

# WLS DRの場合
10.1.2.5 wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com wlsociprefix-wls-0 wlsociprefix-wls-
0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
10.1.2.4 wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com wlsociprefix-wls-1 wlsociprefix-wls-
1.wlsdrvcnlon1ad2.wlsdrvcnlon1.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。

(Oracle RACの場合の注記：使用しているデータベースがOracle RACデータベースである場合は、「付録A – Oracle RAC DBシステムに関する考慮事項」に記載された推奨事項をここで実装すると良いでしょう。ディザスタリカバリの構成を続行する前に、デフォルトPDBサービスではなく、PDBのクラスタウェア・データベース・サービスを使用するようにします。このベスト・プラクティスをこの時点で実装すると、次の手順でデータベース・クラスタウェア・サービス名を使用できます (代わりにデフォルトのPDBサービス名を使用し、DRセットアップ後の手順として、クラスタウェア・サービスを使用するための推奨事項を実装することもできます)。

9. WebLogic構成レプリケーションのためのステージング・マウントを構成する

前述した2つの手法のどちらかでWLSドメイン構成をレプリケーションする場合、このセクションの手順に従って、それぞれのディザスタリカバリ・トポロジに必要なオプションを実行します。

9.1 オプション1) DBFSベースの方法を使用する場合のDBFSマウントの構成

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。

DBFS手法を使用するWebLogic for OCIのDRでは、初期セットアップ時と、システムのライフサイクル中にWebLogicドメイン構成の変更をレプリケートする際にDBFSマウントを使用します。このDBFSファイル・システムは、プライマリからスタンバイに変更を同期するために使用される補助ファイル・システムです。必要な構成ファイルがプライマリ・ドメインからDBFSファイル・システムにコピーされ (ローカル・コピー) 、次に、基盤のData Guardレプリケーションによって自動的にスタンバイ・サイトにレプリケートされます。その後、スタンバイ・サイトで、レプリケートされた構成ファイルがDBFSファイル・システムからドメイン・フォルダにコピーされます。

本書では、プライマリおよびスタンバイでこのコピーを自動化するスクリプトを提供しています。このDBFSマウントは、他のWebLogic実行時操作では使用されないため、サービスにとって不可欠ではなく、システム・パフォーマンスにも大きな影響はありません。

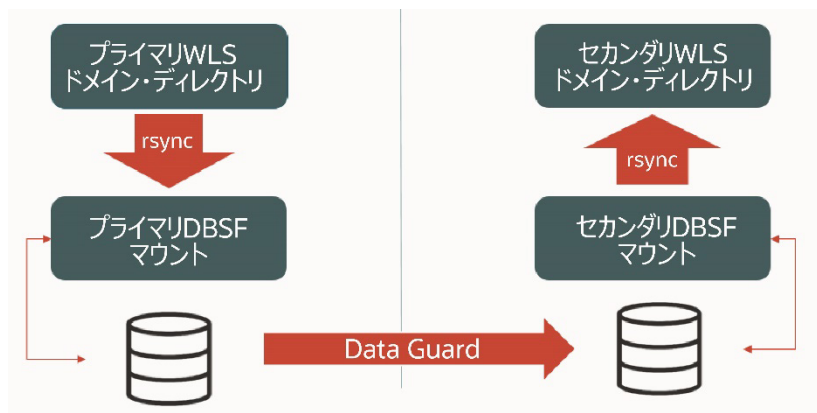


図8. 補助DBFSファイル・システムを使用したセカンダリ・サイトへのWLSドメイン構成のレプリケーション

デフォルトでは、WebLogicホストにDBFSマウントは構成されていないため、手動で構成する必要があります。手動で構成するには、**データベース・クライアントと複数のオペレーティング・システム・パッケージをインストールする必要があります**。各中間層ホストでこの操作を実行するには、以下の手順に従います。

- a) [Oracle E-delivery](#)からDBクライアントをダウンロードし、中間層ホストにアップロードします（まだインストールしないでください）。イメージベースのインストールではなく、必ずインストーラ・バージョンをダウンロードしてください。最新バージョンを使用することが推奨されます。Oracle E-deliveryから19.3 DBクライアントをダウンロードする例を以下に示します。“Oracle Database Client”を検索します。

All Categories

19.3を選択して続行します。

DLP: Oracle Database 19.3.0.0.0 (Oracle Database In-Memory, Oracle Database Vault)

データベース・クライアントのみを選択します。

<input type="checkbox"/> Download Queue	Terms and Restrictions	Platforms / Languages	Size
<input type="checkbox"/> Oracle Database 19.3.0.0.0	Oracle Standard Terms and Restrictions		
<input type="checkbox"/> Oracle Database 19.3.0.0.0			
<input checked="" type="checkbox"/> Oracle Database Client 19.3.0.0.0		Linux x86-64	2.0 GB

「Continue」をクリックし、インストーラ・バージョン（ゴールド・イメージではありません）を選択します。

<input type="checkbox"/> Oracle Database 19.3.0.0.0	
<input type="checkbox"/> Oracle Database Client 19.3.0.0.0 for Linux x86-64	
<input checked="" type="checkbox"/> V982064-01.zip	Oracle Database Client 19.3.0.0.0 for Linux x86-64, 1.1 GB
<input type="checkbox"/> V982065-01.zip	Oracle Database Client 19.3.0.0.0 for Linux x86-64 - Gold Image, 941.5 MB

ソフトウェアをダウンロードし、すべての中間層ホスト（/u01/install/V982064-01.zipなど）にアップロードします。

- b) ダウンロードしたスクリプトからdbfs_dr_setup_root.shスクリプトを見つけ、中間層ホストにアップロードします。このスクリプトは、ホストでDBFSマウントを準備するのに必要な次のタスクを実行します。DBクライアントと必要なオペレーティング・システム・パッケージをインストールし、DBFSユーザーとスキーマをデータベース内に構成します。次にDBFSファイル・システムをマウントし、cronを作成して、DBFSファイル・システムがホストのブート時にマウントされるようにします。
- c) 次に、**root**ユーザーとしてコマンドを実行します。構文は次のとおりです。

```
./dbfs_dr_setup_root.sh <local_db_scan_name> <db_port> <local_PDB_service> <pdb_sys_password>
<path_to_dbclient_installer>
```

入力パラメータとして、WLSが使用するローカル・データベースに接続するための接続データを指定する必要があります。プライマリ・サイト中間層ノードではプライマリPDBの接続データを、セカンダリ・サイト中間層ノードではセカンダリPDBの接続データを指定します。ローカル・スキャン名、ポート、PDBサービス名は、各ドメインのデータソースから取得できます。

注：この時点で、スタンバイ・データベースはスナップショット・スタンバイ・モードであることが必要です。

プライマリ中間層ホストでの実行例を示します（1行である必要があります）。プライマリPDBの値を指定する必要があります。

```
./dbfs_dr_setup_root.sh drdbwlp1a-scan.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com 1521
PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com acme1234# /u01/install/V982064-01.zip
```

セカンダリ中間層ホストでの実行例を示します（1行である必要があります）。セカンダリPDBの値を指定する必要があります。

```
./dbfs_dr_setup_root.sh drdbwlp1b-scan.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
1521 PDB1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com acme1234# /u01/install/V982064-01.zip
```

dbfs_dr_setup_root.shスクリプトをOracle RACシナリオで実行する際は、ローカルOracle RACのSCANアドレス名を使用します。つまり、プライマリ中間層ノードで実行するときはプライマリSCANアドレスを使用し、セカンダリ中間層ノードで実行するときはセカンダリOracle RACのSCANアドレスを使用します。また、適切なPDBサービス名を指定すると、DBFSはデフォルトPDBサービスではなく作成済みのCRSサービスに接続します。Oracle RACシナリオでの実行例を示します。

```
./dbfs_dr_setup_root.sh drdbbrac2a-scan.subnetlon1.myvcnlon.oraclevcn.com 1521
mypdbservice.mycompany.com acme1234 /u01/install/V982064-01.zip
```

d) 次の方法で、中間層ホストにあるDBFSマウントを**確認**します。

```
[root@ wlsociprefix-wls-1]# df -h | grep dbfs
dbfs-@PDB1:/ 32G 248K 32G 1% /u02/data/dbfs_root
[root@ wlsociprefix-wls-1]# ls /u02/data/dbfs_root
dbfsdir
```

これらの処理を**すべての中間層ホスト**（プライマリとセカンダリ）で繰り返します。

このスクリプトの結果は、次のようになります。

- **DBクライアント**・ソフトウェアが中間層ホストのフォルダ/u01/app/oracle/clientにインストールされます。DBクライアントに必要なパッケージが、yumを使用してインストールされます。
- DBFS用のPDBデータベースにユーザーが作成されます。**ユーザー名はdbfsuser**に、パスワードはスクリプトの入力パラメータで指定されたsysパスワードと同じものに設定されます。
- DBFSマウント用のPDBデータベースで**DBFS表領域**と**DBFSフォルダ**が作成されます（表領域の名前は**tbsdbfs**、**フォルダの名前はdbfsdir**です）。
- ドメインに**新規フォルダDOMAIN_HOME/dbfs**が作成されます。このフォルダには、dbfsuserのパスワードおよび必要なその他のアーティファクト（tnsnames.ora、sqlnet.ora）を格納するためのウォレットが保存されます。このウォレットは、中間層ホストでDBFSマウントをマウントするためにDBクライアントによって使用されます。
- DOMAIN_HOME/dbfsに**スクリプトdbfsMount.sh**も作成されます。このスクリプトは、中間層でDBFSをマウントするために使用されます。また、リポートのcronに追加されるので、マシンのリポート時にこのスクリプトが実行されます。
- DBFSマウントは**フォルダdbfsdir**として**/u02/data/dbfs_rootマウント・ポイント**にマウントされます。

このスクリプトは再実行が可能です。再実行する場合、作成済みの要素（DBユーザー、表領域など）に関する警告が表示されますが、このメッセージは無視して構いません。

9.2 オプション2) FSS/rsyncの方法を使用する場合のFSSマウントの構成

FSS/rsync手法で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FS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WLS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。すべてのノードでのマウントに必要な他のアーティファクトを保存していない場合は、他のWLSノードからはアンマウントしても構いません。

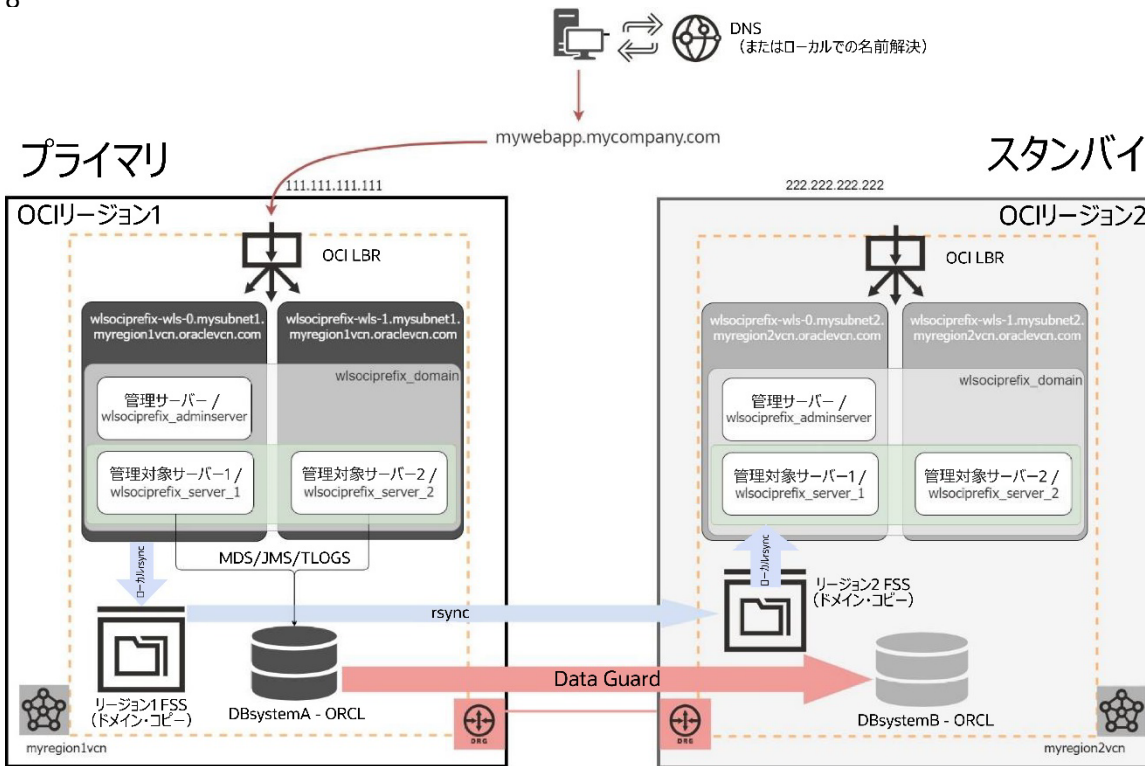


図9. 補助FSSファイル・システムとrsyncを使用したWLSドメイン構成のレプリケーション青色の矢印は構成のコピーの論理フローを示しています。rsyncコマンドは、プライマリ・サイトまたはスタンバイ・サイトのいずれかのWebLogic管理ホストで実行します。つまり、リモート・コピーの場合、プライマリ・サイトのWebLogic管理ホストは、rsyncによってスタンバイのWebLogic管理ホストに接続します。

Oracle WebLogic Server for Oracle Cloud Infrastructureスタックでは、**新しいファイル・ストレージを自動的に作成するかまたは既存のファイル・ストレージを再利用して、それをノードの/u01/sharedディレクトリにマウントできます。**WLS for OCIスタックでは、「Add File System」を必ずオンにします。このオプションは、プロビジョニング中とプロビジョニング後にオンにできます。

そうしない場合には、以下の手順で説明するように、ファイル・システムを手動で作成してマウントする必要があります。

	FSSマウントの準備	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)	-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WLSと同じ> 仮想クラウド・ネットワーク： <プライマリWLSと同じ> サブネット： <プライマリWLSと同じ>	新規マウント・ターゲット名： secondary_mt 可用性ドメイン： <セカンダリWLSと同じ> 仮想クラウド・ネットワーク <セカンダリWLSと同じ> サブネット： <セカンダリWLSと同じ>
2	各リージョンで ファイル・システム を作成する	-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 -それぞ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WLS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WLSと同じ> エクスポート： /secondaryfs 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、各サブネットの一部のネットワーク・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	中間層ホストにファイル・システムをマウントする	<p>すべてのWLS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ユーザーで、/etc/fstabを編集しマウントを追加して、適切なマウント・ターゲットのIPアドレス（プライマリ・サイトのプライマリ・マウント・ターゲットIP、セカンダリ・サイトのセカンダリ・マウント・ターゲットIP）を指定する： <mount_target_ip_address>:<export_name> <your_local_mount_point> nfs defaults,nofail,nosuid,rsvport 0 0 - 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に追加する行のサンプル（必ず1行で記述）：</p> <p>10.1.1.1:/primaryyfs /u01/shared nfs defaults,nofail,nosuid,rsvport 0 0</p> <p>セカンダリ・ホストの/etc/fstabに追加する行のサンプル（必ず1行で記述）：</p> <p>10.2.2.2:/secondaryyfs /u01/shared nfs defaults,nofail,nosuid,rsvport 0 0</p>	
5	マウントされたファイル・システムを確認する	<p>すべてのWLSホストで実行：</p> <pre># df -h grep /u01/shared # ls -la /u01/shared</pre>	<p>プライマリWLSホストでの実行結果のサンプル：</p> <pre>[root@wlsociprefix-wls-0 opc]# df -h grep primaryyfs 10.1.1.1:/primaryyfs 8.0E 0 8.0E 0% /u01/shared</pre> <p>セカンダリWLSホストでの実行結果のサンプル：</p> <pre>[root@wlsociprefix-wls-0 opc]# df -h grep secondaryyfs 10.2.2.2:/secondaryyfs 8.0E 0 8.0E 0% /u01/shared</pre>	
6	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>- 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</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>

10. プライマリWebLogicドメインでDRセットアップ・スクリプトを実行する

プライマリ中間層のOracle WebLogic管理サーバー・ホスト・ノードで、プライマリDRセットアップ・スクリプト`fmw_dr_setup_primary.sh`を実行します。このスクリプトによって、プライマリ・ドメイン構成がセカンダリ・サイトにコピーされます（DBFSまたはFSS/rsyncの方法が使用されます）。このスクリプトの実行によりWebLogicサーバーやバックエンドのデータベースで停止時間が発生することはありません。

- Githubからダウンロードしたスクリプトを**プライマリ管理ホストへアップロード**します。すべてのスクリプトを同じフォルダに入れ、それらのスクリプトに実行権限が付与されていることを確認します。
- プライマリWebLogic管理ホスト**で、**oracle**ユーザーとしてスクリプト`fmw_dr_setup_primary.sh`を実行します。このスクリプトは、対話形式でも非対話形式でも実行可能です。対話形式の使用法は次のとおりです。

```
fmw_dr_setup_primary.sh
```

非対話形式の使用法は次のとおりです。

```
fmw_dr_setup_primary.sh <dr_method> [db_sys_password] [remote_admin_node_ip] [remote_keyfile] [fss_mount]
```

コマンド説明：

<code>dr_method</code> ：	WLSドメイン構成のレプリケートに使用する手法。有効な値はDBFSまたはRSYNCです。 DBFS： DBFSベースの方法を使用する場合はこの値を指定します RSYNC： FSS/rsyncの方法を使用する場合はこの値を指定します
<code>db_sys_password</code>	<code>dr_method</code> がDBFSである場合のみ必須です。データベースのsysユーザーのパスワードです。
<code>remote_admin_node_ip</code> ：	<code>dr_method</code> がRSYNCの場合のみ必須。 セカンダリWLS管理サーバー・ホスト のIPアドレスです。プライマリ管理ホストから到達可能なIPアドレスである必要があります。 注：ホスト名ではなく IPを使用する必要がある があります。プライベートIPの使用を推奨します。そうすることで、サイト間通信が内部ネットワーク経由で行われます（プライマリ・ネットワークとセカンダリ・ネットワークは動的ルーティング・ゲートウェイを介して通信します）。
<code>remote_keyfile</code> ：	<code>dr_method</code> がRSYNCである場合のみ必須です。セカンダリ管理ホストへのSSH通信に必要な秘密鍵ファイルへの完全なパス。このファイルがプライマリのWLS管理サーバー・ホストにアップロードされており、oracleユーザーのみに読取り権が付与されていることを確認してください。
<code>fss_mount</code> ：	<code>dr_method</code> がRSYNCである場合のみ必須です。これは、OCI File Storageファイル・システムがマウントされているマウント・ポイントへのパスです。このOCI File Storageファイル・システムは、WLSドメイン構成のステージングに使用されます。例：/u01/shared

DBFSベースの方法を使用する場合の実行例：

```
./fmw_dr_setup_primary.sh DBFS 'mypassword'
```

FSS/rsyncの方法を使用する場合の実行例：

```
./fmw_dr_setup_primary.sh RSYNC 10.1.0.2 /u01/install/sshkeys/my_priv_key.priv /u01/shared
```

残りのプライマリ中間層ホストでこのスクリプトを実行する必要はありません。

注：このスクリプトは、デフォルトではデータソース`opss-datasource.xml`を検索します。これは、TNS管理フォルダやDB接続文字列などのデータベース関連情報を収集する際の参考として使用されます。Oracle WLS for OCI環境がJRF対応ドメインではなく、`opss-datasource.xml`が存在しない場合、スクリプトはメッセージを表示して実行を停止します。その場合は、`fmw_dr_setup_primary.sh`スクリプトを編集して、定義済みの変数`datasource_name`に既存のデータソース構成ファイル名を指定できます。

11. セカンダリWebLogicドメインでのDRセットアップ・スクリプトの実行

すべてのセカンダリ中間層ホストで、**fmw_dr_setup_standby.sh**スクリプトを実行します。このスクリプトは、セカンダリ中間層クラスタがプライマリのスタンバイとして機能するための構成に必要な処理を実行します。

- a) DR手法がDBFSの場合、このスクリプトでは、セカンダリの中間層ホストからプライマリ・データベースに、リスナー・ポート（通常は1521）でアクセスできる必要があります。OCIコンソールを使用し、この通信に使用されるサブネットで適切な受信ルールを有効化します。

プライマリ・サイトとセカンダリ・サイトの間の通信がインターネット・ゲートウェイを経由する場合は、セカンダリ中間層ホストのパブリックIPからプライマリDBへの通信を開きます。

プライマリ・サイトとセカンダリ・サイトの間の通信が、動的ルーティング・ゲートウェイを使用して内部的にルーティングされる場合（この方法を推奨）、セカンダリ中間層ホストのプライベートIPからプライマリ・データベースへの受信接続を許可します。

これが必要になるのは、セットアップ時とライフサイクル運用時のみです。実行時には、中間層からリモート・データベースへのリージョン間通信は使用されません。

スクリプトを実行する前に、すべてのセカンダリ中間層でoracleユーザーで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 <system_password> <primary_db_ip>:1521/<primary_db_service>
```

- c) Githubからダウンロードしたスクリプトをすべてのセカンダリ中間層ホストへ**アップロード**します。すべてのスクリプトを同じフォルダに入れ、それらのスクリプトに実行権限が付与されていることを確認します。
- b) **すべてのセカンダリ中間層ホスト**で、セカンダリWebLogicサーバー（管理サーバーと管理対象サーバー）とノード・マネージャを**停止**します。
- c) プライマリDBに接続し、次のようにスタンバイ・データベースを**フィジカル・スタンバイに変換**します。

```
[oracle@drdbw1mp1a ~]$ 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...
Database "secondary_db_unqname" converted successfully
```

ステータスを確認して、“SUCCESS”と表示されるまで待ちます。例：

```
DGMGRL> show configuration
Configuration - ORCL_lhr2bb_ORCL_fra22g_12:09:
Protection Mode:MaxPerformance
Members:
  ORCL_lhr2bb - Primary database
  ORCL_fra22g - Physical standby database
Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 25 seconds ago)
```

- d) 最初のセカンダリ・ノードでoracleユーザーとしてスクリプト**fmw_dr_setup_standby.sh**を実行します。このスクリプトには、複数の入力パラメータが必要です。対話モードで実行することも、非対話モードで実行することもできます。実行の前にスクリプトを開いて指示を確認します。

対話形式を使用する場合：

```
fmw_dr_setup_standby.sh （注：プロンプトに従ってすべての値を入力します）
```

非対話形式を使用する場合：

```
fmw_dr_setup_standby.sh <DR_METHOD> [A_DB_IP] [A_PORT] [PDB_SERVICE_PRIMARY]
[SYS_DB_PASSWORD] [FSS_MOUNT]
```

コマンド説明：

- DR_METHODOD : WLSドメイン構成レプリケートに使用する手法。有効な値はDBFSまたはRSYNC。
 DBFS : DBFSベースの方法を使用する場合はこの値を指定します。
 RSYNC : FSS/rsyncの方法を使用する場合はこの値を指定します。
- A_DB_IP : *dr_method*がDBFSである場合のみ必須です。**プライマリ・データベースのIP**。内部IPを介したサイト間通信が可能な場合（つまり動的ルーティング・ゲートウェイを使用する場合）、**プライベートIP**を指定します（この方法を推奨）。そうでない場合は、プライマリ・データベースのパブリックIPを指定します。RACの場合は、プライマリのスキャンIPの1つを指定します。
- A_PORT : *dr_method*がDBFSである場合のみ必須です。プライマリ・データベースのリッスナー・ポート。
- PDB_SERVICE_PRIMARY : *dr_method*がDBFSである場合のみ必須です。プライマリPDBのサービス名。
- SYS_DB_PASSWORD : *dr_method*がDBFSである場合のみ必須です。プライマリ・データベースのsysパスワード。
- FSS_MOUNT : *dr_method*がRSYNCである場合のみ必須です。これは、OCI File Storageファイル・システムがマウントされているマウント・ポイントへのパスです。このOCI File Storageファイル・システムは、WLSドメイン構成のコピーをステージングするために使用されます。

DBFSベースの方法を使用する場合の実行例：

```
./fmw_dr_setup_standby.sh DBFS '10.0.2.2' '1521' 'PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com' mypassword'
```

データベースがRAC Databaseの場合、プライマリRACのスキャンIPアドレスの1つを指定します。デフォルトのPDBサービスの代わりに、クラスタウェア・データベースのサービス名を指定することもできます。そうすることで、DRセットアップではそのサービスを使用してPDBに接続されるようになります。Oracle RACシナリオでの実行例：

```
./fmw_dr_setup_standby.sh DBFS '10.0.2.34' '1521' 'mypdbservice.mycompany.com' mypassword'
```

FSS/rsyncの方法を使用する場合の実行例：

```
./fmw_dr_setup_standby.sh RSYNC '/u01/shared'
```

注：このスクリプトは、デフォルトではデータソース`opss-datasource.xml`を検索します。これは、TNS管理フォルダや接続文字列などのデータベース関連情報を収集する際の参考として使用されます。Oracle WLS for OCI環境がJRF対応ドメインではなく、`opss-datasource.xml`が存在しない場合、スクリプトはメッセージを表示して実行を停止します。その場合は、`fmw_dr_setup_standby.sh`スクリプトを編集して、定義済みの変数`datasource_name`に既存のデータソース構成ファイル名を指定できます。

- e) DRでDBFS手法を使用する場合、スクリプトによってデータベースがスナップショット・スタンバイ・モードに変換されるため、次のステップでノードのセカンダリ・サーバーを確認します。

RSYNC DRの方法では、このスクリプトによってDBロールの変換は行われません。スタンバイ・サイトでWLSプロセスを開始して確認する場合、次に進む前に**スタンバイ・データベースを手動でスナップショット・スタンバイに変換する必要があります**。

- f) このホストで**ノード・マネージャ**を起動します。ノード・マネージャの起動例は次のとおりです。

```
[oracle@wlsociprefix-wls-0 ~]$ cd $DOMAIN_HOME/bin
[oracle@wlsociprefix-wls-0 bin]$ nohup ./startNodeManager.sh >
$DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 &
```

- g) このホストで実行される**WebLogicサーバー**を起動します（これが管理サーバー・ホストである場合は管理サーバー、次にこのホスト内の管理対象サーバーの順で）。

WLSTを使用して管理サーバーを起動する例は次のとおりです。

```

[oracle@wlsociprefix-wls-0 ~]$ cd $MIDDLEWARE_HOME/oracle_common/common/bin
[oracle@wlsociprefix-wls-0 ~]$. /wlst.sh
wlst> nmConnect
('weblogic','passwd','<admin_host_name>','5556','<domain_name>','/u01/data/domains/<domain_name>','SSL')
wlst> nmStart('<admin_server_name>')

```

そのホストの管理対象サーバーを起動するには、WLS管理コンソールを使用します（「Environments」→「Servers」→「Control」）。

注：非JRF対応ドメイン（14.1.1など）で、ノード・マネージャはKSSをIDキーストアとして使用しません。これはデフォルトで \$DOMAIN_HOME/security/Demoldentity.jksを使用します。このファイルはホストごとに異なります。サーバーの起動時にノード・マネージャに接続する場合の「ホスト名検証失敗」エラーを回避するには、Demoldentity.jksファイルをプライマリ・ノードwls-1からセカンダリ・ノードwls-1に、プライマリ・ノードwls-2からセカンダリ・ノードwls-2に、という具合にコピーします。KSSを使用せず、JKS IDキーストア・ファイルをカスタマイズした場合も同じことが適用されます。

これは一度きりのアクションです。DRセットアップ後にこのコピーを実行するだけで済みます。

- h) サーバーがRUNNINGのステータスで起動されていることを**確認し**、サンプル・アプリのURLと管理コンソールを確認します。
- i) **セカンダリ・クラスタ**に含まれる**残りすべての中間層ノード**で、“c)”（フィジカル・スタンバイへのデータベース変換）から“h)”（確認）までのステップを繰り返してスクリプトを実行します。

完了したら、セカンダリ・ドメインの**管理サーバーと管理対象サーバーを停止**して、再度**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**します。以上で、セカンダリ・サイトはDR用の準備が整いました。

12.IDCS認証の使用時にDR構成を完了する

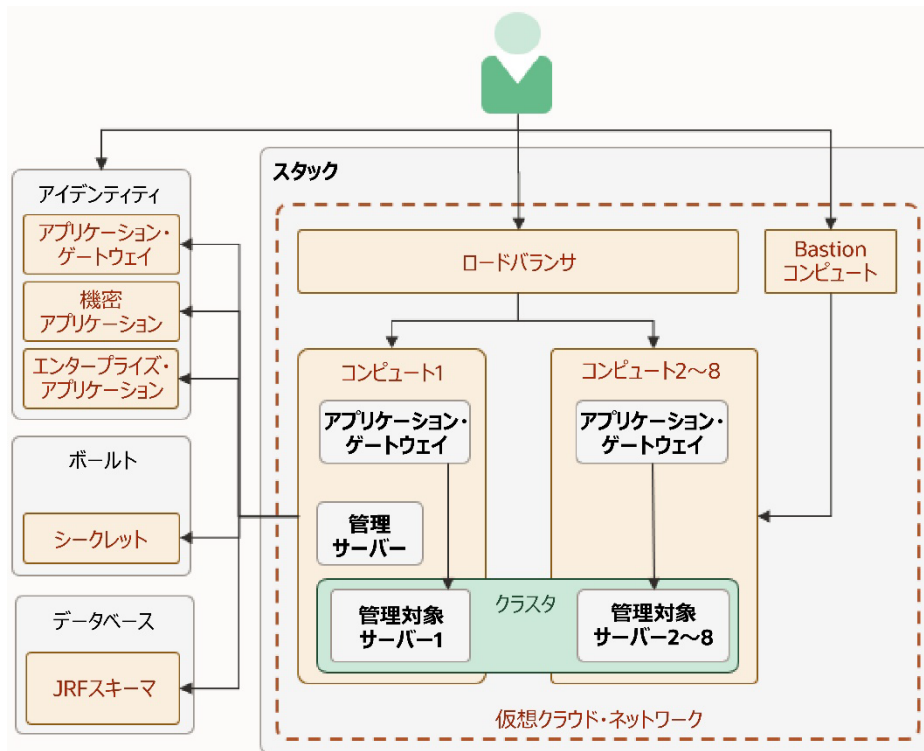
WLS for OCIシステムのIDCS認証を有効にした場合、DR構成を完了するには追加の手順が必要です。認証データの一貫性を保つために、プライマリとスタンバイの両方が同じIDCSサービスを使用する必要があります。

注：このドキュメントでは、DRイベントのコンテキストでWLS for OCIにIDCS認証を使用することによる影響について説明します。ただし、IDCS固有の障害からの保護機能は本書の対象外です⁸。

概要は次のとおりです。“Oracle IDCSを使用した認証の有効化”をチェックしてWLS for OCIシステムをプロビジョニングすると、以下が適用されます。

- **前提条件**として、事前に**機密アプリケーション**を手動で作成する必要があります。詳しくは、Oracle WLS for OCIドキュメントの「Oracle WebLogic Server for OCIを使用する前に」の「[機密アプリケーションの作成](#)」セクションに記載された情報を参照してください。この機密アプリケーションのクライアントID/シークレットIDは、プロビジョニング画面で顧客によって提供されます。これはプロビジョニング時に必要となりますが、WLS for OCIシステムの作成が完了したら使用されることはありません。その他
- **機密アプリケーション**は、WLS for OCI Serviceのプロビジョニング時にIDCSに自動的に作成されます。このアプリケーションは、WebLogicドメインの作成時にWebLogicへの認証プロバイダとして追加されます。アプリケーション名は、`service_name + _confidential_idcs_app_ + timestamp`を使用して構築されます。
- **エンタープライズ・アプリケーション**は、WLS for OCI Serviceのプロビジョニング時にIDCSに自動的に作成されます。このアプリケーションには、リソースを保護するためのWeb層ポリシーが（リソースおよび認証ポリシーの形式で）含まれます。アプリケーション名は、`service_name + _enterprise_idcs_app_ + timestamp`を使用して構築されます。
- **アプリケーション・ゲートウェイ**は、WLS for OCI Serviceのプロビジョニング時にIDCSに自動的に作成されます。これは、アプリケーション・ゲートウェイとエンタープライズ・アプリケーションを接続するために使用されます。このアーティファクトのクライアントIDおよびクライアント・シークレットは、インスタンスの各VMで実行されるアプリケーション・ゲートウェイのDockerイメージで構成されます。アプリケーション・ゲートウェイ名は、`service_name + _gateway_ + timestamp`を使用して構築されます。

⁸ IDCSサポートに問い合わせてご自身の地域でのIDCSデザスタ・リカバリ・ソリューションの提供状況を確認し、IDCS自体を障害から保護する方法の詳細を入手してください。



以上から、プライマリとセカンダリのOracle WLS for OCIシステムがプロビジョニングされた後で、IDCS内に存在するのは以下のアーティファクトになります。

- 前提条件の機密アプリケーション。プライマリおよびスタンバイのプロビジョニングに同じものを使用できます。プロビジョニング時に使用されますが、それ以降は使用されません。
- **プライマリ**WLS for OCIシステムの機密アプリケーション、エンタープライズ・アプリケーション、アプリケーション・ゲートウェイ。
例：
wlsmkpl7_confidential_idcs_app_2020-09-17T10:53:28.031194
wlsmkpl7_enterprise_idcs_app_2020-09-17T10:53:28.031194
wlsmkpl7_app_gateway_2020-09-17T10:53:28.031194
- **セカンダリ**WLS for OCIシステムの機密アプリケーション、エンタープライズ・アプリケーション、アプリケーション・ゲートウェイ。
例：
wlsmkpl7_confidential_idcs_app_2020-09-18T11:55:44.241334
wlsmkpl7_enterprise_idcs_app_2020-09-18T11:55:44.241334
wlsmkpl7_app_gateway_2020-09-18T11:55:44.241334

注：名前はリソース名接頭辞から作られるため、プライマリとスタンバイで名前が似通っていますが、タイムスタンプは異なります。セカンダリはプライマリの後に作成されるため、セカンダリ・アーティファクトのタイムスタンプの方が後になります。

このドキュメントでは、プライマリで使用されるのと同じIDCSサービスを使用するようにセカンダリWLS for OCIシステムを再配線して、プライマリとセカンダリのID情報を一致させる方法を説明しています。このため、セカンダリ・ドメインでは、プライマリと同じ機密アプリケーション、エンタープライズ・アプリケーション、およびアプリケーション・ゲートウェイを使用する必要があります。次の手順を実行してDRをセットアップします

a) セカンダリWebLogicドメインで、プライマリの機密アプリケーションを構成する

機密アプリケーションは、WebLogicドメインの作成時にWebLogicへの認証プロバイダとして追加されます。プライマリ・ドメイン構成はセカンダリにレプリケートされているため、セカンダリWebLogicドメインでのDRセットアップの実行後、**セカンダリ・ドメインはプライマリ機密アプリケーションですすでに構成されています。**

プライマリ機密アプリケーションがセカンダリで使用されることを**検証**するには、以下を実行します。

- IDCSコンソールにログインします。
IDCSコンソールのリンクにアクセスするには、「OCI Console」→「Identity」→「Federation」の順に移動します。
次に、アイデンティティ・プロバイダをクリックすると、**Oracle Identity Cloud Service Console**がIdentity Provider情報に表示されます。例：
<https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole>
- 「Applications」に移動し、**プライマリ機密アプリケーション**の名前を**特定**します。
機密アプリケーション名は、*ervice_name + _confidential_idcs_app_ + timestamp*となります。
例：wlsmkpl7_confidential_idcs_app_2020-09-17T10:53:28.031194

注：セカンダリではなく、プライマリ用に作成した機密アプリケーションを特定してください。名前は似ていますが、プライマリ用の機密アプリケーションのタイムスタンプの方が先になります。

- そのアプリケーションをクリックして、「Configuration」→「General Information」の順に移動します。
- **クライアントID**を特定します。例：87b58e51a4744f95897a2fd72ae5c987
- SSHを使用してセカンダリ・ドメインの管理ノードにログインし、以下のファイルのコンテンツを表示します
<DOMAIN_HOME>/config/config.xml
- IDCSIntegratorを見つけ、クライアントIDがプライマリの機密アプリケーションと同じであることを確認します。例：

```
<sec:authentication-provider xmlns:idcs="http://xmlns.oracle.com/weblogic/security/idcs"
xsi:type="idcs:oracle-identity-cloud-integratorType">
<sec:name>IDCSIntegrator</sec:name>
..
<idcs:tenant>idcs-5e890de598888888888d70064c5e00718</idcs:tenant>
<idcs:client-tenant>idcs-5e890de598888888888d70064c5e00718</idcs:client-tenant>
<idcs:client-id>87b58e51a4744f95897a2fd72ae5c987</idcs:client-id>
```

セカンダリの機密アプリケーションにアクセスできないようにするため、これを非アクティブ化します。

- IDCSコンソールにログインします。例：<https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole>
- 「Applications」に進み、**セカンダリ機密アプリケーション**を特定します。
機密アプリケーションの名前は、*service_name + _confidential_idcs_app_ + timestamp*です
例：wlsmkpl7_confidential_idcs_app_2020-09-18T11:55:44.241334
- 「Deactivate」をクリックしてこれを非アクティブ化します。

注：プライマリではなく、セカンダリの機密アプリケーションを非アクティブ化してください。名前は似ていますが、セカンダリの機密アプリケーションのタイムスタンプの方が後になります。誤ってプライマリ機密アプリケーションを非アクティブ化してしまった場合は、再度アクティブ化することができます。

b) セカンダリWebLogicドメインでプライマリ・エンタープライズ・アプリケーションとアプリケーション・ゲートウェイを構成する

アプリケーション・ゲートウェイは、それぞれのWebLogicノードで実行されるクラウド・アプリケーション・ゲートウェイで構成されます。この時点では、セカンダリ・ドメインで実行されるセカンダリ・クラウド・アプリケーション・ゲートウェイは、セカンダリ・エンタープライズ・アプリケーションとセカンダリ・アプリケーション・ゲートウェイを参照します。これらがプライマリと同じアーティファクトを参照するようにするため、以下の手順を実行します。

- **プライマリ・アプリケーション・ゲートウェイ**のクライアントIDとクライアント・シークレットを特定します。
これには、IDCSコンソールにログインして「Security」→「App Gateway」に移動し、プライマリ・アプリケーション・ゲートウェイをクリックします。アプリケーション・ゲートウェイ・アプリケーションの名前は、*service_name + _app_gateway_ + timestamp*です。例：wlsmkpl7_app_gateway_2020-09-17T10:53:28.031194

注：セカンダリではなく、プライマリ・アプリケーション・ゲートウェイを特定してください。名前は似ていますが、プライマリ用の機密アプリケーションのタイムスタンプの方が先になります。

- クライアントIDとクライアント・シークレットをメモします。例：

```
Client ID:      28dabe1c9de4410dab27b6e4febb2bfa
Client Secret: f1f9589c-ee71-4a73-ae2b-f66ef7bbd6a5
```

- SSHでセカンダリWLS for OCIドメインの管理ノードにログインします。
- アプリケーション・ゲートウェイのDockerコンテナを停止し、opcとしてログインして以下を実行します。

```
sudo docker stop appgateway
```

- 既存のウォレット・ファイル/u01/data/cloudgate_config/cwallet.ssoのバックアップを作成します。

```
cp /u01/data/cloudgate_config/cwallet.sso /u01/data/cloudgate_config/cwallet.sso.bak
```

- /u01/data/cloudgate_config/appgateway-envファイルに次の行を追加します。

```
CG_APP_NAME=<client_id>
CG_APP_SECRET=<client_secret>
```

ここで、<client_id>と<client_secret>には、プライマリ・アプリケーション・ゲートウェイの値を指定します。

- プライマリとセカンダリで、プロビジョニングに使用したIDCSテナントが異なる場合（大陸間DRの場合など）、次の箇所にはプライマリと同じIDCSを設定する必要があります。

ファイル/u01/data/cloudgate_config/appgateway-envでは、以下のプロパティで使います。

```
CG_APP_TENANT=<primary_idcs_tenant>
# IDCSのインスタンスへアクセスするにはURLが必要です。
IDCS_INSTANCE_URL=https://<primary_idcs_tenant>.identity.oraclecloud.com:443
```

ファイルcloudgate.configでは、以下のプロパティで使います。

```
"bootstrap": {
  "externalIdUrl": "https:// <primary_idcs_tenant>.identity.oraclecloud.com:443",
  ....
  "tenantName": "<primary_idcs_tenant>",
```

<primary_idcs_tenant>のサンプル値は、“idcs-82507b399af654467dccb94029c895ab0”です。

- ウォレットを再生成し、権限とモードを変更します。

```
sudo /opt/scripts/idcs/create_idcs_cloudgate_cwallet.sh
sudo chown opc:opc /u01/data/cloudgate_config/cwallet.sso
sudo chmod 775 /u01/data/cloudgate_config/cwallet.sso
```

- アプリケーション・ゲートウェイに含まれるDockerを再起動します。

```
sudo docker start appgateway
```

（プライマリ・エンタープライズ・アプリケーションを使用する）プライマリ・アプリケーション・ゲートウェイを参照するようになりました。

- 最後に、バックアップ・ファイルがある場合はそれを削除できます。

```
sudo rm /u01/data/cloudgate_config/cwallet.sso.bak
```

- 残りのセカンダリ・ノードで同じ操作を繰り返します。

セカンダリ・エンタープライズ・アプリケーションまたはセカンダリ・アプリケーション・ゲートウェイにアクセスできないようにするには、以下のよう
それらを非アクティブ化します。

- IDCSコンソールにログインします。たとえば次のとおりです。
<https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole>
- 「Applications」に進み、**セカンダリ・エンタープライズ・アプリケーション**を特定します。
例：wlsmkpl7_enterprise_idcs_app_2020-09-11T11:55:44.241334
- 「Deactivate」をクリックしてこれを非アクティブ化します。
- 次に、IDCSコンソールで「Security」→「App Gateway」の順に移動し、**セカンダリ・アプリケーション・ゲートウェイ**を特定します。
例：wlsmkpl7_app_gateway_2020-09-187T11:55:44.241334
- これを非アクティブ化します。

注：プライマリではなく、セカンダリ・エンタープライズ・アプリケーションおよびセカンダリ・アプリケーション・ゲートウェイを非アクティブ化してください。名前がよく似ていますが、セカンダリのアプリケーションのタイムスタンプの方が後になります。

c) IPではなくフロントエンド名でエンタープライズ・アプリケーションを構成する

デフォルトで、エンタープライズ・アプリケーションの「Application URL」にはプライマリIPが設定されています。

ORACLE Identity Cloud Service

wlsmkpl7_enterprise_idcs_app_2020-09-17T10:53:28.031194


Details OAuth Configuration SSO Configuration Import Users Groups

Details

Application Type Enterprise Application

* Name wlsmkpl7_enterprise_idcs_app_2020-09

Description Application to integrate IDCS with WebLogic Cloud service service wlsmkpl7 - Used by Application

Application Icon 

Upload

* Application URL http://152.67.136.170

Custom Login URL

まだ構成していない場合は、エンタープライズ・アプリケーションにフロントエンド名を構成します。

- IDCSコンソールにログインします。たとえば次のとおりです。
<https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole>
- 「Applications」に進み、**プライマリ・エンタープライズ・アプリケーション**を特定します。
例：wlsmkpl7_enterprise_idcs_app_2020-09-11T11:55:44.241334
- アプリケーションをクリックし、“Application URL”を編集してフロントエンド名を設定します。

ORACLE Identity Cloud Service

wismkpl7_enterprise_idcs_app_2020-09-17T10:53:28.031


Details OAuth Configuration SSO Configuration Import Users Groups

Details

Application Type Enterprise Application

* Name wismkpl7_enterprise_idcs_app_2020-09-17T10:53:28.031

Description Application to integrate IDCS with WebLogic Cloud service service wismkpl7 - Used by Application

Application Icon 

Upload

* Application URL https://mywebapp.mycompany.com

Custom Login URL

d) IDCSで保護されたURLをセカンダリ・ロケーションで検証する

IDCSによって保護されたセカンダリURLが、セカンダリ・ロケーションで正しく機能していることを確認します。

この確認を実行するには、セカンダリ・ロケーションへスイッチオーバーするか（「[スイッチオーバー](#)」で説明）、またはスタンバイ・データベースをスナップショットへ変換してセカンダリ・サーバーを起動します（「[検証のためのセカンダリ・サイトのオープン](#)」で説明）。

それから、IDCSに保護されたセカンダリのサンプル・アプリケーションにアクセスして、想定どおりに機能していることを確認します。

https://<secondary_frontend_ip>/_protected/idcs-sample-app/

例：

https://222.222.222.222/_protected/idcs-sample-app/

これにより、IDCSログイン・ページにリダイレクトされるはずですが、適切な資格証明を入力すると、IDCSサンプル・アプリケーションのページが表示されます。

13. DRセットアップを検証する

これで、DR環境のセットアップは終了です。オラクルは、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（本書の「[スイッチオーバー](#)」を参照）を実行するか、またはセカンダリ・サイトを検証用に関きます（本書の「[検証のためのセカンダリ・サイトのオープン](#)」を参照）。

ライフサイクル手順

構成変更のレプリケーション

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。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。**プライマリWebLogicドメインで構成を変更したら（新規デプロイメント、新規データソース、デプロイメント・プランの変更など）、何らかの方法でその変更をセカンダリ・ドメインにレプリケートしなければなりません。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。

- 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ドメイン構成が**定期的に変更される**場合、ドメイン構成をプライマリからセカンダリにレプリケートするスクリプトを使用できます。このために、`config_replica.sh`スクリプトが提供されています。DRセットアップで使用する方法（DBFSまたはFSS/rsync）に応じて、このスクリプトは選択した方法を使用したWLSドメイン構成のレプリケーションを実行します。DRセットアップでDBFS手法が使用された場合、Data Guardを使用した構成の同期で、Oracle Database File System（Oracle DBFS）が使用されます。FSS/rsync手法が使用された場合、構成のレプリケートにrsyncが使用されます。

この後、それぞれの方法について詳しく説明します。

a) ドメイン構成の変更を両方のサイトに適用する

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ユーザーとして次のコマンドを実行します。 <code>[oracle@drdbw1mp1a]\$dgmgrl sys/your_sys_password@primary_db_unqname</code> DGMGRL> CONVERT DATABASE secondary_db_unqname to SNAPSHOT STANDBY; データベース"secondary_db_unqname"をスナップショット・スタンバイ・データベースに変換しています。しばらくお待ちください... データベース"secondary_db_unqname"の変換が終了しました
3	セカンダリ・サイトでWebLogic管理サーバーを起動する (起動していない場合)	Oracle Cloudのドキュメントの手順に沿って管理サーバーを起動します。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WLS管理コンソールを使用して構成変更を適用します。変更を有効化し、想定どおり変更が機能していることを確認します。
5	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。 <code>[oracle@drdbw1mp1a ~]\$dgmgrl sys/your_sys_password@primary_db_unqname</code> 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 "ORCL" ...

```
Continuing to convert database " secondary_db_unqname" ...
データベース" secondary_db_unqname"の変換が終了しました
```

b) スクリプトを使用して構成変更を伝播する

システムのライフサイクル中にWebLogicドメイン構成が頻繁に更新される場合は、config_replica.shスクリプトを使用してレプリカプロセスを自動化できます。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。

オプション1) DBFSの方法

このアプローチでは、「DBFSマウントの構成」ステップで作成したDBFSファイル・システムに、プライマリ・サイトのドメイン構成のコピーが含まれているため、これを**補助ファイル・システム**として使用します。このファイル・システムにある情報は、Data Guardを介してスタンバイ・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・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）、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ている場合は、スタンバイ・サイトにDBFSファイル・システムをマウントすることもできます。

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プロセス、マウント・ポイントなどにも依存することになります）。

この方法の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にコピーされるファイルは、データベースに格納されるため、**Data Guardを介してスタンバイ・データベースに自動的に転送**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。

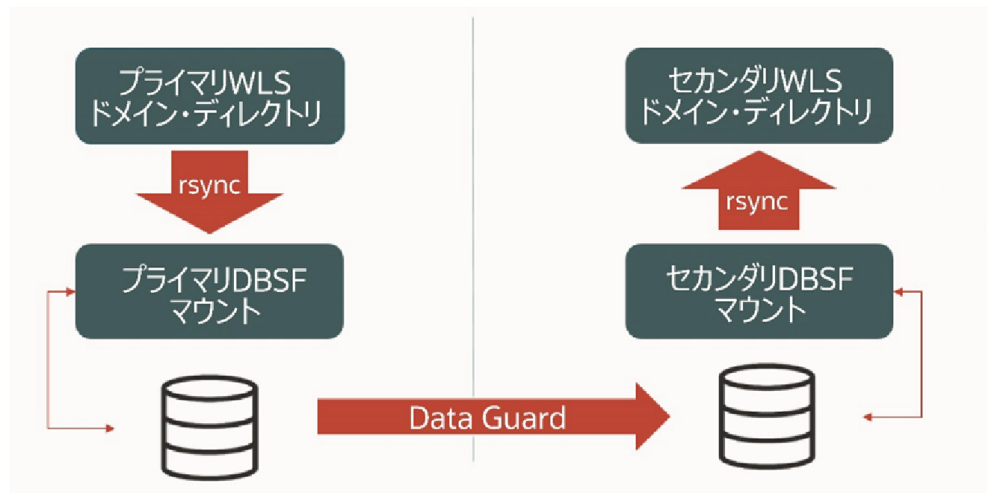


図10：DBFSベースの方法を使用した、WebLogicドメイン構成のスタンバイWebLogicクラウドへのレプリケート

この手順のメリットは、スタンバイ・サイトで構成の更新を適用するために、**Data Guardレプリケーションの堅牢性を頼り**にできることです。レプリカ方向はサイトのロールと全体的に一貫しており、スイッチオーバーやフェイルオーバーが発生すると自動的に変更されます。

注：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、中間層はDBFSマウントをマウントします。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。顧客はデフォルトで構成された値（合計10分。最小限のDBホストの再起動をサポートするための時間）を必要に応じて調整したり、少な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点に注意してください。

オプション2) FSS/rsyncの方法

rsyncコマンドを使用して、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できます。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・システムに同期されます。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、ドメイン構成がセカンダリ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。

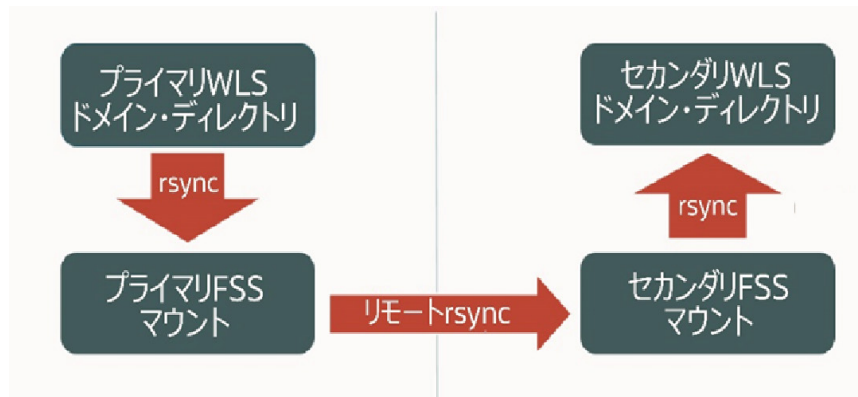


図11：FSS/rsyncの方法を使用した、WebLogicドメイン構成のレプリケート

DBFS手法とFSS/rsync手法の両方で、提供された`config_replica.sh`スクリプトを使用してこれらの手順を**自動化できます**。同じスクリプトがどちらの手法にも有効で、プライマリとスタンバイの両方で使用できます。このスクリプトにより、サイトの現在のロールがチェックされ、そのサイトがプライマリなのかセカンダリなのか、および使用する手法がDBFSなのかFSS/rsyncなのかに応じて、必要なアクションが実行されます。

`config_replica.sh`スクリプトを使用してWebLogic構成をレプリケートする手順は以下のとおりです。

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。

3. **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。パラメータはスクリプト内で説明されます。
4. **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。
5. **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。
6. 実行が終了したら、**セカンダリ中間層のWebLogic管理ホストで、（oracleユーザーで）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時に停止しています。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ります。セカンダリWebLogic管理サーバーは、いったん起動すると、スタンバイ・データベースがフィジカル・スタンバイに再び変換されても稼働状態のままになります（スタンバイ・データベースは、Data Guardを介してプライマリ・データベースからデータベースのREDOログを受信して適用するため、通常動作中はフィジカル・スタンバイ・ステータスになっている必要があります）。

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

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、config_replica.sh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。

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 WLS for OCI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witch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	レプリケートを保留している変更がある場合は、「構成変更のレプリケーション」を参照して、それらをセカンダリ・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。

b) スwitch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。 注：WLS for OCI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。
2	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。 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。 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。TTLに小さい値を設定するとこの時間が短くなりますが、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を通常の値に戻します。
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。ユーザー-oracleで次を実行します。 <pre>[oracle@drdbw1mp1a ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> switchover to "secondary_db_unqname"</pre>

<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>	<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：WLS for OCI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>
--	--

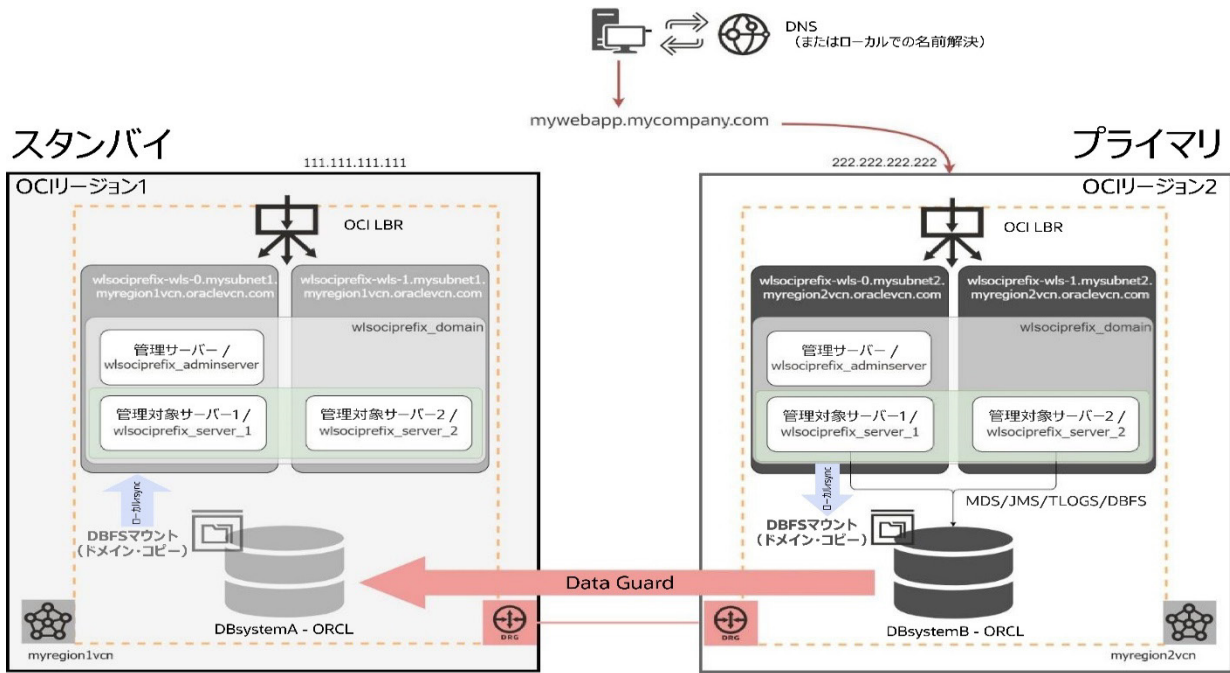


図12：スイッチオーバー後のWebLogic for OCIのディザスタ・リカバリ・システム（DBFSベースの方法）

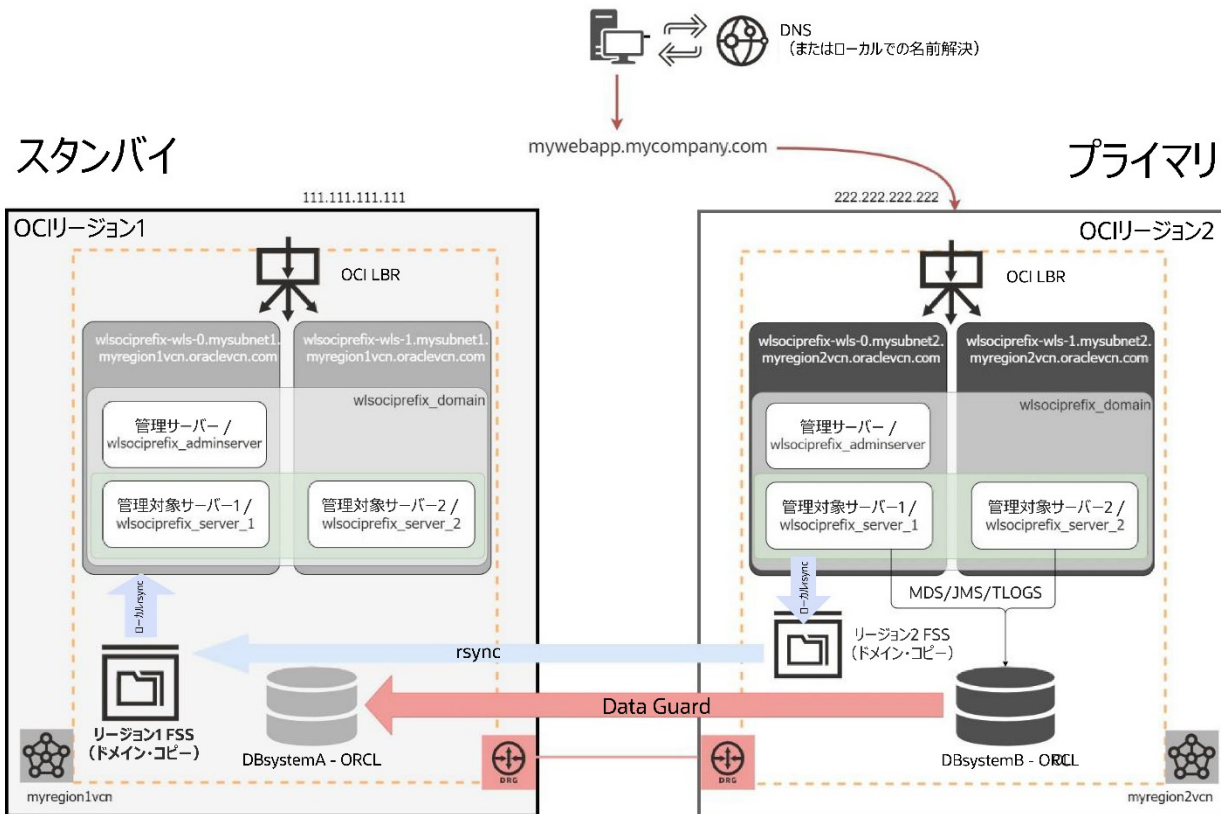


図13 : スイッチオーバー後のWebLogic for OCIのディザスタ・リカバリ・システム (FSS/rsyncの方法)

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 WLS for OCI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	<p>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。</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>
2	データベースをフェイルオーバーする	<p>セカンダリDBホストでDBブローカを使用してフェイルオーバーを実行します。ユーザー-oracleで次を実行します。</p> <pre>[oracle@drdbw1mp1b ~]\$ dgmgml sys/your_sys_password@secondary_db_unqname DGMGRL> failover to "secondary_db_unqname"</pre>

3	セカンダリ・サイト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コンソールカスクリプトを使用）。</p> <p>注：WLS for OCI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>
---	---------------------	--

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WLS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。

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	<p>プライマリDBホストでDGブローカを使用して、セカンダリDBをスナップショット・スタンバイに変換します。ユーザーoracleで次を実行します。</p> <pre>[oracle@drdbA ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to snapshot standby</pre> <p>“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。</p>
2	セカンダリ・サイト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します。例：</p> <pre>cd /u01/app/oracle/middleware/oracle_common/common/bin ./wlst.sh wlst> nmConnect ('weblogic','acme1234#','wlsociprefix-wls-0','5556','wlsociprefix_domain','/u01/data/domains/wlsociprefix_domain','SSL') wlst> nmStart('wlsociprefix_adminserver')</pre> <p>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コンソールカスクリプトを使用）。</p> <p>注：WLS for OCI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>

3	検証する	<p>これはスイッチオーバーではなく、プライマリは引き続きプライマリ・ロールのままであるため、フロントエンド名はプライマリを参照し、すべてのアクセスはプライマリにリダイレクトされます。</p> <p>セカンダリのWLSシステム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。</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WLS for OCI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わからず、フロントエンド名がプライマリIP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。</p>
---	------	--

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	<p>プライマリDBホストでDGブローカを使用して、セカンダリを再びフィジカル・スタンバイに変換します。ユーザーoracleで次を実行します。</p> <pre>[oracle@drdba ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to physical standby</pre> <p>“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。</p>
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。

構成レプリケーションのエンド・ツー・エンド検証

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징・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징・フォルダ（DBFSまたはFSS）から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。

- 1) 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。
 - a) 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。
 - b) 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。
 - c) 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。
 - d) 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。

- e) 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。
 - f) セカンダリのWebLogic管理サーバーを停止します。
 - g) 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。
 - h) プライマリで行った構成変更はテスト用であるため、ここで元に戻します。
- 2) 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。
- a) 本書の「スイッチオーバー」セクションの説明に従って、システム全体をセカンダリにスイッチオーバーします。
 - b) 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。
 - c) `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。新しいプライマリで`config_replica.sh`を実行してから、新しいスタンバイでこのスクリプトを実行します。
 - d) 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。
 - e) システムをスイッチバックして元の状態に戻します。

その他のライフサイクル運用については、「[付録C – その他のライフサイクル運用](#)」を参照してください。

RTOとRPOの概要

注：以下の値は、参照の目的のみに提供される一般的な値であり、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、お客様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RTO

リカバリ時間目標（RTO）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。スイッチオーバーとフェイルオーバーは停止時間を伴うイベントであるため、システムのRTOに影響を及ぼします。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Oracle WLS for OCI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。サンプルとして使用されたこの特定のシステムは、WLSホストのシェイプがVM.Standard2.1で、WLSサーバーのヒープ・メモリ・サイズが1Gです。JRF対応ドメインを使用しており、WebLogicサーバーの接続プールは標準構成で、デモ・アプリケーション（sample-app）がデプロイされています。

	スイッチオーバー手順	Oracle WLS for OCIのDRでのサンプル時間
1	保留中のすべての構成変更を伝播する	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	～30秒（強制） / ～2分（正常）
	2.2 管理サーバーを停止する	～8秒（強制） / ～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が、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	～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2分
	5.2 管理対象サーバーを（並列で）起動する	約3分
... 停止時間の終了		

1つの手順と別の手順の間の自然な遅延や、他の追加の検証は上記の時間に含まれません。これは、スイッチオーバーの手順の実行方法に依存するためです（例：手動、カスタム・スクリプトによる自動化、オーケストレーション・カスタム・ツールの使用、Oracle Site Guardの使用など）。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、顧客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。

- 停止時間を必要としないスイッチオーバーに関連するあらゆるアクティビティを、プライマリ・サーバーを停止する前に実行します。たとえば、`config_replica.sh`スクリプトに基づくWLS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平行で停止します。
- アプリケーション/ビジネスで可能な場合は、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完了したら、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⁹。OCIコンソールを使用したデータベース・スイッチオーバーはOCIコンソールのロールを自動的にリフレッシュしますが、DBスイッチオーバーはOCIコンソールを使用して実行した場合に時間がかかります。
- OCI LBRも、サーバーの起動とリクエストの送信開始を実現するのに時間がかかります。OCI LBRヘルス・チェックの頻度によりませんが、通常は数秒です。チェックに使用される間隔が短いほど、サーバーの起動が速くなります。ただし、短すぎる間隔の使用は慎重にしてください。ヘルス・チェックが大量だと、バックエンドが過負荷になる可能性があります。

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。Oracle WLS for OCIでRPOが関係するのは、トランザクション・ログ、JMSメッセージ、その他のアプリケーションの情報です。これらはすべてデータベースに格納されています。

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（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。場合によって、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ます（ファイル・アダプタ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は、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とされます。オブザーバの構成については、[Oracle DB Systemのドキュメント](#)を参照してください。
- Data Guardで 사용되는保護モードは、Maximum Availability、Maximum Protection、またはMaximum Performance（デフォルト）のいずれかです。
 - **Maximum Availability**モードは、スタンバイ・データベースの障害後のプライマリ・データベースの障害などの特定の二重障害の場合を除いて、データ損失ゼロを保証します。
 - **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なりますが、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。

⁹ DBシステムのロールが`dgmgrl`コマンドによるスイッチオーバー後に自動的にリフレッシュされない場合は、My Oracle SupportでService RequestをオープンしてOCIコンソールでロールを更新してください。

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。

- データベース外部のファイル・システムにランタイム・アーティファクトが保存されている場合（カスタム・ファイル・ストレージ・サービスに保管されたファイルが、顧客アプリケーションによって使用または生成される場合など）、これらのRPOは、セカンダリ・ロケーションへの同期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、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ます。RPOは、このコンテンツのレプリケーション間の時間になります。もう一つの方法として、これらのランタイム・ファイルをDBFSファイル・システムに保管することができます。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。

- 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。より頻繁に構成が変更されるほど、障害発生時に失われる構成変更の量が多くなります。
- WebLogic構成が**スタンバイに同期される頻度**。このドキュメントで説明したように、WebLogic構成は手動で、またはconfig_replica.shスクリプトを使用して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。もう一つの方法は、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。
- WebLogic構成のレプリケーションに使用される**プロシージャの信頼性**。DBFSの方法とFSS/rsyncの方法のどちらも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징・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・リカバリ・トポロジのライフサイクルにおいて、Oracleでは以下のベスト・プラクティスを推奨しています。

- JDBC永続ストアを使用する。カスタム永続ストアを作成する場合は、必ずJDBC永続ストアとして作成してください。この方法ではJMSメッセージがデータベース表に保存されるため、この情報はData Guard経由でセカンダリ・サイトにレプリケートされます。
- 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・サイトで同じパッチ・レベルを維持する。ソフトウェアはいずれの層においてもセカンダリ・システムに自動的にレプリケートされません。パッチをプライマリにインストールする場合は、同じパッチをスタンバイ・ロケーションにインストールする必要があります。データベースにパッチを適用する際、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、`config_replica.sh`スクリプトを使用してレプリケートされない）プライマリ・システムに適用される変更はすべて、セカンダリ・システムでも実行する必要があるため、プライマリ・システムとセカンダリ・システムには同じ構成が含まれます。例としては、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。
- 定期的にはスイッチオーバーを実行してセカンダリ・システムの健全性を検証する。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。詳しくは、「[検証のためのセカンダリ・サイトのオープン](#)」を参照してください。
- アプリケーション・デプロイメント操作の場合は、WebLogic管理コンソールでWebLogicのデプロイメント・オプションとして“Upload your files”を使用し、管理サーバーのアップロード・ディレクトリの下（ドメイン・ディレクトリ/`servers/admin_server_name/upload`の下）に、デプロイしたファイルが配置されるようにすることを推奨します。こうすることで、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。
- プライマリとスタンバイのWLSホストによって使用されるブロック・ボリュームでは、ブロック・ボリュームのバックアップを定期的に行うか、自動バックアップを設定する。詳細については、Oracle WLS for OCIドキュメントの「[ボリューム・バックアップについて](#)」を参照してください。

その他のベスト・プラクティスについては、「[付録D - その他のベスト・プラクティス](#)」を参照してください。

結論

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

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

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

WebLogic Serverで使用されるDBシステムがOracle Real Application Cluster (Oracle RAC) DBシステムの場合、このドキュメントで説明するようにデザスタ・リカバリを構成できます。その際、以下を考慮する必要があります。

Oracle RACを使用する場合のOracle WLS for OCIエディションについて

このドキュメントは、Oracle RAC DBシステムを使用している場合を除き、Oracle WebLogic SuiteとOracle WebLogic Server Enterprise Editionに適用されます。Oracle RAC DBシステムを使用している場合、**このドキュメントはOracle WebLogic Suiteのみに適用されます。**このエディションのみが、GridLinkデータソースを使用するためです。Oracle RACを使用する場合は、GridLinkデータソースを使用するのがMAAベスト・プラクティスです。

Oracle RAC DBシステムを使用する場合のネットワークに関する考慮事項

Single Client Access Name (SCAN) は、Oracle Real Application Clusters環境で使用される機能です。SCANを使用すると、クラスタ内で実行中のOracle Databaseにアクセスする際に、クライアントは単一の名前を使用できます。SCANアドレスは、SCANに割り当てられたすべてのアドレス（通常は3つのIPアドレス）に解決するように構成された、完全修飾ドメイン名です。SCANリスナーはクラスタ内のあらゆるノードで実行できます。SCANを使用すると、データベースはロケーションに依存しなくなるため、クライアント構成は、どのノードが特定のデータベースを実行しているかに依存する必要がなくなります。接続がSCANリスナーに対して確立されると、その接続は、各データベース・ノードで実行され**仮想IP (VIP)** でリスニングするデータベース・リスナーの1つにリダイレクトされます。データベース・クライアントからOracle RACデータベース・インスタンスへの接続は、通常は次のように試行されます。

- データベース・クライアントがSCANに接続し、有効なサービス名を提供します。
- SCANリスナーが、このサービスをホストしているデータベース・インスタンスを特定し、ローカルまたはそれぞれのノードのノード・リスナーにクライアントをルーティングする。
- ノードVIPと所定のポートでリスニングするノード・リスナーは、接続リクエストを取得し、クライアントをローカル・ノードのインスタンスに接続します。

OCIデータソースを使用するOracle WebLogic Server Suiteは、Oracle RAC DBシステムを使用してプロビジョニングされると、スキャン・リスナーを参照するように自動的に設定されます。これはベスト・プラクティスに準拠しています。

OCIでWebLogic用のDRをセットアップするとき、各中間層から**リモートOracle RAC DBシステムに接続できる**必要があります。この通信は、config_replica.shスクリプトを使用した構成のレプリケーションでも使用されますが（DBFSの方法のみ）、ランタイムには決して使用されません。**各中間層ホストから、ポート1521で、リモートOracle RACのSCAN IP、VIP、ホストIPに通信できるように、必要なネットワーク・ルールが定義されていることを確認してください。**

注：動的ルーティング・ゲートウェイでは、リモート・ネットワークにある名前に対してDNS解決は提供されません。リモートOracle RAC DBシステムを指すように各中間層で作成されたtnsnames.oraエイリアスは、SCANアドレス名ではなく、IPアドレスを使用します。

CRSデータベース・サービスの作成と使用

OCIにおけるOracle WebLogic Server Suiteは、お使いのOracle WebLogic ServerドメインでGridLinkデータソースをプロビジョニングし、選択したOracle Databaseクラスタに接続します。GridLinkは、Oracle Databaseクラスタ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、ノードが追加または削除された場合にクラスタ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。

注：これらのステップは、ディザスタリカバリ用にWLSを構成する前に、プライマリ・ドメインとセカンダリ・ドメインで実行することを推奨します。

e) **プライマリOracle RACで、Oracle Databaseサービスを作成します。**

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て、サービスを追加、構成、起動します。

```
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。

例：

```
[oracle@priracnode1 ~]$ srvctl add service -db ORCL_lhr3jg -service mydbservice.mycompany.com -preferred
ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY "
[oracle@priracnode1 ~]$ srvctl modify service -db ORCL_lhr3jg -service mydbservice.mycompany.com -rlbgoal
SERVICE_TIME - clbgoal SHORT -pdb pdb1
[oracle@priracnode1 ~]$ srvctl start service -db ORCL_lhr3jg -service
mydbservice.mycompany.com [oracle@priracnode1 ~]$ srvctl config service -db ORCL_lhr3jg -
service mydbservice.mycompany.com Service name: mydbservice.mycompany.com
Server pool:
Cardinality:2
Service role:PRIMARY,SNAPSHOT_STANDBY
Management policy:AUTOMATIC
...
Available instances:
CSS critical: no
```

f) **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。

```
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>
```

```
srvctl start service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME>
```

例：

```
[oracle@seccracnode ~]$ srvctl add service -db ORCL_fra3vb -service mydbservice.mycompany.com -preferred
ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY"
[oracle@seccracnode1~]$ srvctl start service -db ORCL_fra3vb -service
mydbservice.mycompany.com [oracle@seccracnode1~]$ srvctl config service -db ORCL_fra3vb -
service mydbservice.mycompany.com Service name: mydbservice.mycompany.com
Server pool:
Cardinality:2
Service role:PRIMARY,SNAPSHOT_STANDBY
Management policy:AUTOMATIC
DTP transaction: false
...
```

g) データソースとブート・ファイルの接続文字列を更新します。

データソースとjpsブート・ファイルで、デフォルトのPDBサービス名ではなく、この新しいサービス名が参照される必要があります。
tnsnames.oraのTNSエイリアスでは、デフォルトのサービス名ではなく適切なサービス名が使用されるようにしてください。

その他のベスト・プラクティスの適用

ディザスタ・リカバリのセットアップを問わず、WebLogic Serverデータソース構成には、Oracle RACデータベースに接続する場合に推奨されるベスト・プラクティスが他にも存在します。以下のプラクティスは任意の時点で適用できます。

- **auto-ons**の使用

Oracle 12c以降のデータベースを使用している場合、データベースから自動的にONSリストがドライバに提供されます。
データソース構成で、ONSノード・リストを空白のままにすることができます。

- **“Test Connections On Reserve”**

データソースで**“Test Connections On Reserve”**（予約テスト接続）が選択されていることを確認します。

Oracle RACインスタンスが使用できなくなると、GridLinkデータソースはFANイベントを受信しますが、アプリケーションに戻された接続が確実に良好であるようにするために、データソースでTest Connections On Reserveを有効にするのがベスト・プラクティスです。

- **“Seconds to Trust an Idle Pool Connection”**

テストの有効性を最大化するために、**“Seconds to Trust an Idle Pool Connection”**（アイドル・プール接続を信頼する秒数）を0に設定して、接続が常に検証されるようにすることもできます。0に設定した場合、アプリケーションに返される接続はすべてテストされます。このパラメータを10に設定した場合、前のテスト結果が10秒間有効になるため、この時間が経過する前に接続が再利用されると、結果は引き続き有効であると見なされます。

- **“Test Frequency”**

データソースの**“Test Frequency”**（テスト頻度）パラメータが0でないことを確認します。

これは、WebLogic Serverインスタンスが未使用の接続のテストを試行する間隔（秒数）です。通常は、デフォルト値の120で十分です。

付録B – 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。これに対して、Oracle Cloudはさまざまなアプローチを提供しています。バックアップをローカルに保存することも、クラウドに保存することもでき、実行方法は自動か、Oracle Recovery Manager (RMAN) またはdbcliを使用したカスタムを選択できます。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。
- リンク先の推奨事項に従ってRMAN設定を構成してください。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合わせてRMANバックアップ・スクリプトを作成し、crontabに含めます。以下はあくまでも、全体バックアップを実行する例です。

```
# RMANを実行します
export ORACLE_HOME=/u01/app/oracle/product/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
# 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、プライマリ・データベースで自動バックアップを有効にすることができます。こ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構成は、ほとんどのデータベース障害シナリオに対して保護を提供します。言い換えると、ほとんどの場合、プライマリ・データベースで障害が発生すると、スタンバイにスイッチオーバーすることによって操作を再開できます。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ある可能性があります。そのようなまれなシナリオでは、スタンバイ・データベースも再作成する必要があります。

手動Data Guard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](#)」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U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・システムでDRを再構築するために、WLS for OCI中間層で一部のプロパティを更新する必要があります。詳細については、「[スタンバイDBシステム再作成後のOracle WLS for OCI DRの再構築](#)」の手順に従ってください。

付録C - その他のライフサイクル運用

Oracle WLS for OCIのDR環境のパッチ適用

Oracle WLS for OCIの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**

Oracle WLS for OCIの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、パッチごとに異なります。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）**

これらのパッチの一部では、readmeに「FMW_ROLLING_ORACLE_HOME」という記載があります。その場合、DRをセットアップしているかどうかに関係なく停止時間は必要ありません。

しかし、たいいていのパッチはFMW_ROLLING_ORACLE_HOMEに対応しておらず、中間層のシャットダウンが必要です。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が問題なく動作したら、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. 古いプライマリ中間層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手順にかかる時間のみになります。DRをセットアップしていない場合、停止時間にパッチ適用時間が追加されます。

- **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、上記と同じ手順になります。ただし、DBスキーマ変更のために、中間層とDBへの同時パッチ適用が必要な場合、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。DRがセットアップされている場合、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が問題なく動作したら、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、パッチ適用とその手順をスタンバイで検証できる（問題の特定やパッチ適用手順自体の確認）というメリットがあります。

DBFSウォレットの再作成方法

注：これはDBFSベースの手法のみ適用されます。

DRセットアップ中、dbfs_dr_setup_root.shスクリプトの実行時にDBFSマウントが作成されました。「[DBFSマウントの構成](#)」で説明したように、このときPDB内にユーザー（dbfsuser）が作成され、DOMAIN_HOME/dbfs/walletにウォレットが作成されて、ユーザー名とパスワードが格納されます。このウォレットは、DBFSクライアントによって、データベースへの接続とDBFSのマウントに使用されます。

データベース内で“dbfsuser”のパスワードが更新された場合、新しいパスワードでウォレットを再作成する必要があります。ウォレットを再作成するには、以下の手順に従います。

- ホストにログインし、“oracle”ユーザーに切り替えます。以下のコマンドはすべてoracleユーザーで実行する必要があります。
- DOMAIN_HOME/dbfsフォルダに移動します。
- DBFSマウントでPDBへの接続に使用されているTNSエイリアスを特定します。通常はPDB名です。dbfsMount.shスクリプトで、dbfs_clientの行を探します。使用されているtnsエイリアスには、“@”が付いています。例：

```
..
$ORACLE_HOME/bin/dbfs_client -o /u01/data/domains/<domain_name>/dbfs/wallet/ @PDB1 -o direct_io -o allow_other
$MOUNT_PATH &>>dbfs.log &
...
```

- 次のように、古いウォレットをバックアップします。

```
mv wallet wallet_bckup
```

- 次のコマンドを実行して、Oracle Walletを生成します。

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -create
```

ウォレットのパスワードを入力するように求められます。

- 次のコマンドを実行して、新しい資格証明をウォレットに追加します。この例では、dbfsuserがDBFSのユーザー名です。

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -createCredential
<tnsalias_for_pdb> dbfsuser <dbfsuser_password>
```

例：

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -
createCredential PDB1 dbfsuser dbfsuser_password
```

ウォレットのパスワードを入力するよう促されます。

- ウォレットが新しいパスワードで更新されたことを確認するには、次のコマンドを実行します。

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -listCredential
```

DBFSを再マウントします。

- まずアンマウントします。

```
fusermount -u /u02/data/dbfs_root
```

- 再マウントします。次のコマンドを使用できます。

```
$DOMAIN_HOME/dbfs/dbfsMount.sh
```

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、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）。

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、プライマリとセカンダリのデータベース間の接続が長時間切断されないようにすることが推奨されます。これには、セカンダリが停止するシナリオや、通常の業務中に2つのサイト間での通信が阻止される可能性があるネットワーク・レベルの問題などが含まれます。

スタンバイ中間層のコンピュート・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・リカバリにおいて次のような影響があります。

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

これらの影響を最小限に抑えるため、いくつかのセカンダリWLSのコンピュート・インスタンスを停止させる場合には、セカンダリ管理サーバー・ホストを起動したままの状態を維持し、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できます。

注：

顧客の請求状態はこのドキュメントの対象外です。いくつかのサーバーを停止させることの請求への影響を確認するには、請求状態に関する確認を得るため、オラクルのライセンス・チームにお問い合わせください。

すべての場合において、インスタンスのOSを使用してインスタンスを停止しても、そのインスタンスの請求は停止されません。この方法でインスタンスを停止する場合は、必ずコンソールまたはAPIからも停止してください。

DR環境でのスケール・アウト手順とスケール・イン手順

Oracle WLS for OCIドキュメント「[Scale a Domain](#)」に記載された手順に従って、Oracle WLS for OCIシステムでスケール・アウトとスケール・インを実行できます。

Oracle WLS for OCI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Oracle WLS for OCIシステム（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。Oracle WLS for OCIの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

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、Oracle WLS for OCIインスタンスにパッチ34988073が含まれる必要があります。

Oracle WLS for OCIのDR環境をスケール・アウトするための推奨手順は以下のとおりです。

a) プライマリWLS for OCIシステムのスケール・アウト：

1. 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・サーバーに認識されていないWebLogicサーバー・ノードがセカンダリ・システム構成に含まれる場合（セカンダリWLS for OCIにはスケール・アウトが実行されるまで同じノードは含まれません）、一部のアプリケーションの起動がセカンダリで失敗する場合があります。詳細については、「[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](#)」を参照してください。

2. 「[Add or Remove WebLogic Server Nodes](#)」の手順に従い、プライマリ・スタックでノード数を1つ増やします（この例では、1ノード増加すると仮定しています）。
3. 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。
 - a. /etc/hostsを編集して、フロントエンドFQDNをプライマリのフロントエンドLBR IPとともに追加します。次に例を示します。

```
# 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LBRのIPを指しています
111.111.111.111 mywebapp.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。例：

```
10.0.0.82 <prim_midtier1_fqdn> <prim_midtier1_shortname> <sec_midtier1_fqdn> <sec_midtier1_shortname>
10.0.0.81 <prim_midtier2_fqdn> <prim_midtier2_shortname> <sec_midtier2_fqdn> <sec_midtier2_shortname>
```

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。

4. 新しい管理対象サーバーを再起動します。

b) セカンダリWLS for OCIシステムのスケール・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）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、プライマリ名ではなくセカンダリ名を使用します。この変更は後で元に戻されます。変更が必要なのは、新しく追加されるノードでは、config.xmlに設定されるプライマリ・サーバー名のエイリアスが/etc/hostsに含まれないためです。これには、次を実行します。

- **プライマリ**中間層ホストのFQDN名を見つめます（スケール・アウトの実行前に存在するノード）。例：

```
wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
```

プライマリ中間層1のFQDNは、*wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com*で、そのホスト名は、*wlsociprefix-wls-0*です。

プライマリ中間層2のFQDNは*wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com*で、そのホスト名は*wlsociprefix-wls-1*です。

- **セカンダリ**中間層ホストのFQDN名を見つめます（この時点で存在するノード）。例：

```
wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
```

注：ホスト名はプライマリとセカンダリのWLSホストで同じになるはずですが、FQDNの値のみが異なります。

- セカンダリ管理サーバー・ノードで、プライマリ・ホストのFQDNをセカンダリ・ホストのFQDNに置き換えます（<DOMAIN_HOME>/config/config.xmlファイル内）。

```
cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_pre_scale-out
sed -i 's/primary_midtier1_fqdn_name/secondary_midtier1_fqdn_name/g' config.xml
sed -i 's/primary_midtier2_fqdn_name/secondary_midtier2_fqdn_name/g' config.xml
```

例：

```
sed -i 's/ wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /g' config.xml
sed -i 's/ wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.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（ノード・マネージャをまだ起動していない場合は起動します）。
4. 「[Add or Remove WebLogic Server Nodes](#)」の説明に従い、セカンダリ・スタックで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に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、仮想フロントエンド名およびセカンダリ・フロントエンドIP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 mywebapp.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。例：

```
10.2.0.12 <secondary_midtier1_long_name> <secondary_midtier1_shortname> <primary_midtier1_long_name>
<primary_midtier1_shortname>
10.2.0.11 <secondary_midtier2_long_name> <secondary_midtier2_shortname> <primary_midtier2_long_name>
<primary_midtier2_shortname>
```

- /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. （IDCS認証使用時のみ）新しく追加されたノードで、本書のセクション**10b**「セカンダリWebLogicドメインで、プライマリのエンタープライズ・アプリケーションとアプリケーション・ゲートウェイを構成する」に示した手順を実行して、Cloudgateエージェントがプライマリと同じアプリケーション・ゲートウェイを参照するようにします。残りのセカンダリ・ノードでは、DRセットアップ中にこのステップが実行されています。
 7. 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。

8. 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。
9. **(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)** この段階で、必要に応じて、ステップ2の変更を元に戻し、もう一度プライマリFQDN名をリスニング・アドレスに設定することができます。その場合は、ステップ2と同じ置換を逆に行います。

```
cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_post_scale-out
sed -i 's/secondary_midtier1_fqdn_name/primary_midtier1_fqdn_name/g' config.xml
sed -i 's/secondary_midtier2_fqdn_name/primary_midtier2_fqdn_name/g' config.xml
```

ここで元に戻さない場合も、後でconfig_replica.shを使用して構成をレプリケートするとき（ステップd）に元に戻されます。

- 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。

1. プライマリで、すべての既存のプライマリ中間層ノード（と新規ノード）にエイリアスを追加します。例：

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

2. セカンダリで、すべての既存の中間層ノード（と新規ノード）にエイリアスを追加します。例：

```
<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ビューに新規ノードの名前を追加します。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（プライマリIPを参照）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（セカンダリIPを参照）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。

スケール・イン

WLS for OCI DR環境のスケール・イン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WLS for OCIシステムのスケール・イン：**
 1. 「[Add or Remove WebLogic Server Nodes](#)」の説明に従い、プライマリ・スタックのサーバー数を減らします。
- b) **セカンダリWLS for OCIのスケール・イン：**
 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・スタックで「[Add or Remove WebLogic Server Nodes](#)」に示された手順を実行します。
 5. 終了したら、セカンダリのプロセスを停止し、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。
- c) 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。
- d) （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。

自動スケーリング

自動スケーリング機能を使用すると、事前定義されたしきい値イベントの発生によって、スケール・アウトとスケール・インが自動で処理されます。この処理中に手動で行う手順はありません。Oracle WLS for OCIのDRシナリオでもこの機能を使用できますが、以下の考慮事項があります。

- 自動スケール・アウト/スケール・イン処理は、プライマリ・ロールを持つシステムで実行されるようにします。スタンバイ・ロールを持つサイトで実行しないでください。
- セカンダリ・サイトで（プライマリ・ロールを持つときに）自動スケーリング機能を使用するには、「[必要なホスト・エイリアスの構成](#)」の説明に従って、**DNSプライベート・ビュー**を使用して設定する必要があります。
- 自動スケール・アウト/スケール・イン処理により、プライマリ・サイトとセカンダリ・サイトでノード数が異なる結果になる場合があります。これが一時的な短時間である場合、問題にはなりません。しかし、長時間に及ぶ場合は、セカンダリ・サイトを手動でスケール・アウト/スケール・インして、プライマリのノード数に合わせることを推奨します。

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

オラクルでは、プライマリWLS for OCIシステムとスタンバイWLS for OCIシステムに含まれるリソース（ノード数、メモリ、その他）をまったく同じにし、プライマリ・ロケーションにスケール・アウトまたはスケール・インを実行する場合は前述したのと同じアクションをセカンダリで進めること強く推奨しています。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2ノードから3ノードへスケール・アウトされ、2つのノードしかないスタンバイにその構成がレプリケートされた場合、セカンダリ・ロケーションにとって未知の新しいノードがあるため（セカンダリ・サイトに同等のノードが存在しないため）、一部のアプリケーションは解決できない場合があります。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 "myCustomApp".
```

```
weblogic.application.ModuleException: java.net.UnknownHostException: wlsmpdr1-wls-2.sub10171440110.vncash.oraclevcn.com
```

人為的エラーやリカバリ状況が原因でこのシナリオに直面した場合は、次善策として、プライマリには存在してもセカンダリには存在しないノードのためにセカンダリWLSホストで“偽の”エイリアスを追加し、既存のサーバーを起動することができます。偽のエイリアスは存在しないIP（またはセカンダリDBのIPが使用可能）を参照し、既存のセカンダリ・サーバーでアプリケーションを起動できます。セカンダリに“新しい”ノードは存在せず、反応することはありませんが、“unknownHostException”エラーは発生せず、アプリケーションは既存のノードで起動します。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ししないでください（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ため、オラクルはこれを推奨しません。**

スタンバイDBシステム再作成後のWLS for OCI 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。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。

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

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

- a) スタンバイDBシステムを終了する前に、これから終了する**元のスタンバイDBシステムのDBの一意の名前**（スタンバイDBホストの\$ORACLE_UNQNAME変数）、**プライベートIPとパブリックIP、VCN、サブネット、ホスト名接頭辞**をメモしておきます。
- b) スタンバイDBシステムが終了したら、プライマリDBシステム・ホストの/etc/hostsファイルを確認します。終了したスタンバイDBホストに関するエントリがある場合は、削除するかコメントアウトします。スタンバイDBホスト用の新しいエントリがその作成時に自動的に追加されます。
- c) OCIコンソールを使用してプライマリDBシステムでのData Guardを再び有効化するときに、**以前のスタンバイDBシステム**で使用していたものと**同じVCN、同じサブネット、同じホスト名接頭辞**を指定するようにしてください。そうすれば、新しいスタンバイDBシステムと以前のスタンバイDBシステムで異なる値は、DBの一意の名前、プライベートIP、パブリックIPだけになります。
- d) 新しいDBシステムが正常に作成され、OCIコンソールでData Guard構成が完了したら、新しいスタンバイDBシステムの**一意のDB名およびプライベートIP**の値をメモします。
- e) セカンダリ・ローケーションの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。
- f) **（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WLS for OCI DR環境でのみ必要とされます。**スタンバイWLS**ホストで実行：

- 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。

- g) **（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WLS for OCI DR環境でのみ必要とされます。

DBFS手法を使用している場合は、**スタンバイWLS**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。
- ファイル**\$DOMAIN_HOME/dbfs/tnsnames.ora**を編集します。このファイルには、エイリアスがいくつか含まれています。そのうちの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について、この元のスタンバイDBの一意の名前を、新しいスタンバイDBの一意の名前に置き換えます。

DBFS手法を使用している場合は、**プライマリWLS**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WLSホスト間で異なる場合があります。プライマリでは、セカンダリCDBの参照にスタンバイIPが使用されますが、スタンバイWLS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、これは想定どりの動作です。
- プライマリWLSホストでlocaldb.logファイルを更新する必要はありません。プライマリの一意の名前が含まれていますが、これは変更されていません。

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)	ORCL_phx1kg	ORCL_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WLS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	新しい内容
<code>/u01/data/domains/ local_CDB_jdbcurl.nodelete</code>	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ベースの方法を使用する場合のみ) <code>\$DOMAIN_HOME/ dbfs/localdb.log</code>	ORCL_phx1kg	ORCL_phx1c3
(DBFSベースの方法を使用する場合のみ) <code>\$DOMAIN_HOME/ dbfs/tnsnames.ora</code>	(その他のエントリ) ... ORCL_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 = ORCL_phx1kg. mysubnet.region2vcn.oraclevcn.com))) ...)	(その他のエントリ) ... ORCL_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 = ORCL_phx1c3 . mysubnet.region2vcn.oraclevcn.com))) ...)

プライマリWLSホストでは以下のようになります。

更新対象のファイル	元のコンテンツ	新しいコンテンツ
(DBFSベースの方法を使用する場合のみ) \$DOMAIN_HOME/ dbfs/tnsnames.ora	(その他のエントリ) ... ORCL_phx1kg = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=10.2.0.2)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL_phx1kg. mysubnet.region2vcn.oraclevcn.com)(UR=A)))	(その他のエントリ) ORCL_phx1c3 = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST= 10.2.0.5)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME= ORCL_phx1c3 .mysubnet.region2vcn.oraclevcn.com)(UR=A)))

- h) 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、Oracle WLS for OCIのDR環境で、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。

付録D - その他のベスト・プラクティス

JDBCストアを使用したTLOGおよびJMSストアの格納

DRトポロジでは、サーバーのTLOGとJMSストアを格納するために、JDBCストアを使用することが強く推奨されます。JDBCストアは、Oracle データベースの整合性とデータ保護を活用しており、クラスタのすべてのサーバーがリソースを利用できるようにします。JDBCストレージを使用することは、とりわけDRトポロジにとってメリットがあります。DBに格納されている情報は、Data Guardによってセカンダリにレプリケートされるため、この情報が、スイッチオーバー後にセカンダリ・サイトで利用できるようになり、データベースに格納されている残りの情報と連携されます。

TLOGに関しては、OCIのWebLogicではデフォルトで、サーバーのTLOGがデフォルトのストア（ファイル・システム）を使用するように構成されています。管理対象サーバーのTLOGストアをJDBCストアに変更するには、以下を実行します。

- WebLogicコンソールでプライマリ・ドメインにログインし、「Lock & Edit」をクリックします。
- WLSSchemaDataSourceに移動します。「Services」→「Data Sources」→「WLSSchemaDataSource」
- 「Targets」をクリックし、管理サーバーのみが対象となっている場合は、クラスタも対象にします。このデータソースが、管理対象サーバーのTLOGに使用されます。
- 管理対象サーバーに移動します。「Environment」→「Servers」→「<manager_server_name>」→「Configuration」→「Services」
- Advancedで「Transaction Log Store」に移動します。
 - o Typeに「JDBC」を選択します。
 - o Data Sourceに「WLSSchemaDataSource」を選択します。
 - o Prefix Nameではデフォルトの値を使用できます。例：

The screenshot shows the configuration page for the Transaction Log Store in the Oracle WebLogic console. The 'Type' dropdown is set to 'JDBC'. The 'Data Source' dropdown is set to 'WLSSchemaDataSource'. The 'Prefix Name' text field contains the value 'TLOG_wlsmkp15_server_'.

- クラスタの残りの管理対象サーバーで同じ操作を繰り返します（この変更は管理サーバーでは不要で、デフォルトのTLOGを使用できます）。
- 変更を適用します。
- 変更を適用するには、管理対象サーバーを再起動する必要があります。

これで、config_replica.shスクリプトを使用して構成変更をセカンダリ・ドメインにレプリケートできます。

JMSに関しては、OCIのOracle WebLogicでデフォルトで定義されているJMSサーバーや永続ストアはありません。アプリケーションでJMSストアを使用する場合は、必ずJDBC永続ストアを使用して構成してください。

付録E – 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・リカバリ

トポロジの説明

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DRソリューションは、ほとんどの面で、本書のFSSとDBFSの説明と同じです。トポロジは同じで、構成のレプリケート方法のみが異なります。トポロジの重要なポイントについて次に示します。

このトポロジは**アクティブ・パッシブ・モデル**です。Oracle WLS for OCIドメイン、ロードバランサ、Oracle Cloud Infrastructure DBシステムから構成される**プライマリ・システム**を1つのリージョンに配置し、Oracle WLS for OCIドメイン、ロードバランサ、Oracle Cloud Infrastructure DBシステムから構成される**スタンバイ・システム**を別のリージョンに配置します。リージョン間ブロック・ボリューム・レプリケーションを利用するには、プライマリとスタンバイに対して異なるリージョンを使用する必要があります。

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**の機能を頼りに、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。

セカンダリのOracle WebLogic Server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・ドメインと同じ名前、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。WebLogic Serverのリッスナー・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、セカンダリ中間層ホストでは、セカンダリIPに解決されるようホスト・ファイル内に関連する別名が作成されます。

フロントエンドでは、システムで実行されるアプリケーションにアクセスするため、**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OCILoadバランサのIPを参照します。スイッチオーバーの場合、このフロントエンド名は、セカンダリ・サイトのOCILoadバランサのIPを参照するように更新されます。フロントエンド名は必ず、スイッチオーバーごとに、プライマリ・ロールを持つサイトのLBR IPを参照する必要があります。

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。Active Data Guardを使用する場合、マウント状態か、または読取り専用モードで開かれています。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、読取り/書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DRのセットアップとライフサイクルの手順の実行中、一部の動作の場合には、スタンバイ・データベースがフィジカル・スタンバイからスナップショット・スタンバイに変換されます。**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は、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。

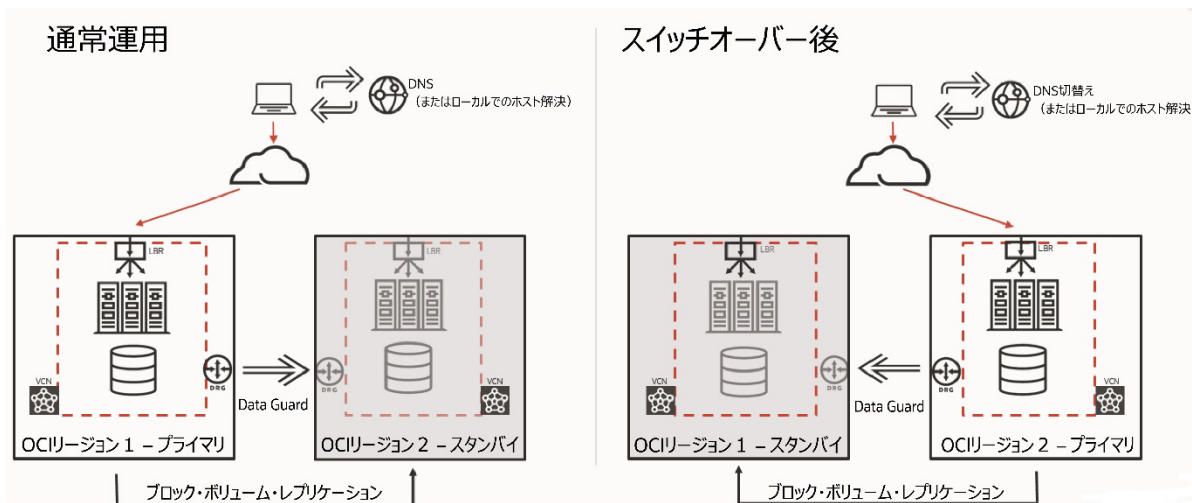


図14. リージョン間ブロック・ボリューム・レプリケーションに基づくOracle WLS for OCIのディザスタ・リカバリ・トポロジ

他の方法と同様に、データベース内に存在するすべての情報（カスタム・スキーマ、TLOG、JDBC永続ストア、OPSS情報など）が、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。各サイトのローカル・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・サイトからセカンダリにレプリケートする必要があります。本書で説明したその他の方法との**相違点**は、WebLogicドメイン構成をプライマリからスタンバイにレプリケートする方法です。この方法では、**リージョン間ブロック・ボリューム・レプリケーション**機能を使用して送信されます。この機能によって、ブロック・ボリュームとブート・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を継続的に自動で実行できます。詳しくは、Oracleドキュメントの「[ボリュームのレプリケート](#)」を参照してください。ここで説明されているトポロジと同様に、この機能をディザスタ・リカバリ・シナリオで使用できます。

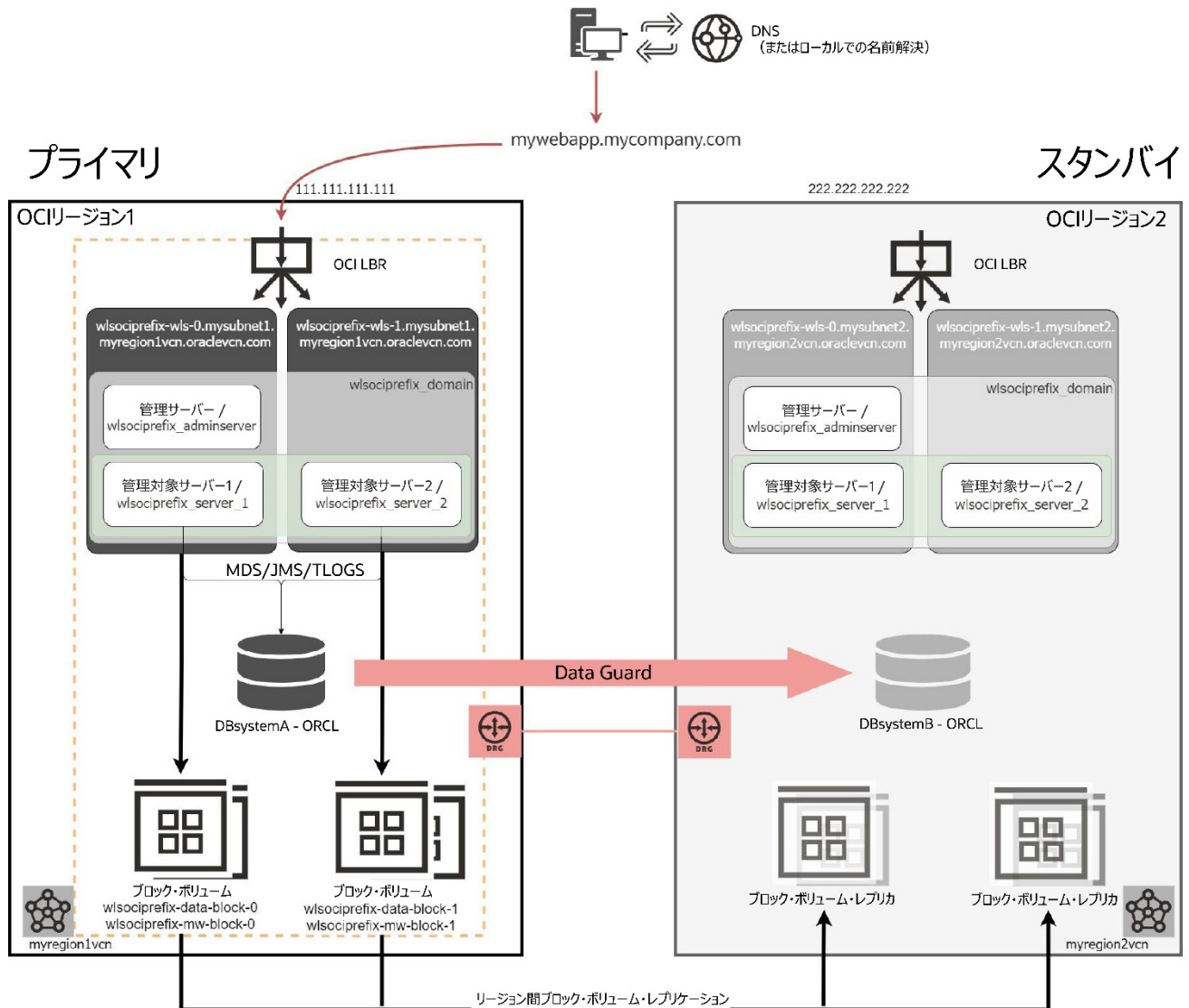


図15：リージョン間ブロック・ボリューム・レプリケーションをベースとしたWLS for OCIのディザスタ・リカバリ・トポロジ詳細

BVレプリケーションDRモデルの考慮事項

本書で説明した他の方法と比較した場合の**ブロック・ボリューム・レプリケーション手法のおもな考慮事項**を以下にまとめます。

- **汎用ソリューションである**

ブロック・ボリューム・レプリケーションをベースとしたDRソリューションは、個々のシステムの特異性に依存しにくくなっています。WLS for OCI以外にも他のシステムに汎用的に適用でき、システム内の他のコンピュート・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。

- **継続的な自動のレプリケーション・プロセスを使用している**

ブロック・ボリュームのレプリケーションはOCIインフラストラクチャによって自動で実行される継続的なプロセスです。手動での実行やスケジューリングが必要となるスクリプト・ベースではありません。

- **レプリケーションの対象がドメイン構成に限定されない**

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<wlsociprefix>-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。レプリケートされたブロック・ボリュームの情報は、プライマリ・ブロック・ボリュームの正確なコピーです。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. BVL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。

- **ミドルウェア・ブロック・ボリュームのレプリケーション**

WLS for OCIコンピュート・インスタンスでは、Oracleソフトウェアは別のブロック・ボリューム（/u01/appにマウントされた

<wlsociprefix>-mw- block-Nブロック・ボリューム）に格納されます。このブロック・ボリュームには、各コンピュート・インスタンスの**Oracleソフトウェア・ホームとそのインベントリ**が格納されます。これらのブロック・ボリュームのレプリケートも可能ですが、**継続的にレプリケートする必要はありません**。このボリュームのコンテンツは、WebLogicソフトウェアまたはJDKのパッチ適用/修正がない限り、変更されることはありません。

このボリュームを構成ボリュームと同じレプリケーション戦略に含めれば、コストと、ソリューションの複雑さが不必要に増すことになります。それでも、このボリュームを不定期にレプリケートして、プライマリからセカンダリにパッチを伝播することは可能です。これらのボリュームの管理には、次の異なる方法を使用できます。

1. これらのmwブロック・ボリュームを、（構成ボリュームと同様に）継続的なリージョン間レプリケーション戦略に追加できます。この場合、プライマリのOracleホームに適用されたパッチがセカンダリにレプリケートされるようになります。ただし、オプション3と比べると、管理オーバーヘッドとコストが増加します。
2. 最適な方法は、これらのボリュームを不定期にレプリケートすることです。たとえば、プライマリ・ピアにパッチが適用された場合にレプリケートを実行できます。これにより、コストを削減し、スイッチオーバー実行時の手順を削減することができます。このモデルでは、パッチ適用は1つのサイトのみで実行し、それがセカンダリにコピーされます。ただし、オプション3と比べると、管理オーバーヘッドとコストが増加します。
3. 代替策として、この“ソフトウェア”ボリュームをレプリケーション戦略から完全に省略できます。ただし、プライマリにパッチが適用された場合、同じパッチ適用の手順をセカンダリ中間層インスタンスに適用する必要があります。そうすることで、パッチのレプリケーションを自動化しない代わりに、スイッチオーバー操作の手順を減らし、より安価なソリューションを実現できます（セカンダリ・システムに手動でパッチを適用する必要がありますが、先にセカンダリで修正プログラムのテストを実行できるという点でも便利です）。

- **共有ストレージの要件**

ブロック・ボリュームを複数のコンピュート・インスタンスの共有ストレージとして使用することはできません（ブロック・ボリュームの上で実行する追加のクラスタ・ファイル・システムが必要です）。これにより、ブロック・ボリュームは、複数のWebLogicサーバーから同時にアクセスされる必要があるランタイム・アーティファクトのための有効な手法から除外されます。

- **管理の複雑さ：**

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。

- **コストの影響**

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ようになります。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・コストも請求に含まれるようになります。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。

Oracleドキュメント「[リージョン間ボリューム・レプリケーション](#)」の「リージョン間レプリケーションのコストに関する考慮事項」を参照してください。

- **リカバリ時間目標（RTO）**

スイッチオーバーRTOは、“DBFS”や“FSS/rsync”の手法で提供されるものと似ています。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必要とされる追加手順（レプリカのアクティブ化、ブロック・ボリュームのタッチなど）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・クラスタの通常のフェイルオーバーの場合、この時間は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目標に影響なく適用可能です。

- **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は他の方法と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時間以内となります。一方、ソース・ボリュームでのデータの変更率によっては、このRPOが変化します。たとえば、大量の書き込み/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。本書で説明したその他の方法の場合、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、レプリケートされる情報の量は少ないため、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。

BVレプリケーションDRモデルの前提条件

WebLogicのエディション

DBFSおよびFSS/rsyncモデルの「[前提条件](#)」→「[WebLogicのエディション](#)」と同じ。

認証

DBFSおよびFSS/rsyncモデルの「[前提条件](#)」→「[認証](#)」と同じ。

ロードバランサ

DBFSおよびFSS/rsyncモデルの「[前提条件](#)」→「[ロードバランサ](#)」と同じ。

データベース

DBFSおよびFSS/rsyncモデルの「[前提条件](#)」→「[データベース](#)」と同じ。

レプリケート対象のブロック・ボリューム

BVレプリケーションDRソリューションでは、他のサイトに**ブロック・ボリュームのみがレプリケートされ、ブート・ボリュームはレプリケートされない**ことを前提としています。

WLS for OCIドメインの各中間層ホストには、**2つのブロック・ボリューム**がアタッチされています。本書では、**データ・ブロック・ボリューム** (<wlsociprefix>-data-block-N) **のみが継続的にレプリケートされることを前提として**、mwブロック・ボリューム (<wlsociprefix>-mw-block-N) の手順はオプションとして記載します。

これらのフォルダ (/u01/dataと/u01/app) の外部にある他のコンテンツはブート・ボリュームに配置され、**レプリケートはされません**。オペレーティング・システムはブート・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のOSに対してパッチ適用や修正が行われた場合、同じパッチ適用手順をセカンダリWLS for OCIコンピュート・インスタンスでも実行する必要があります。

BVレプリケーションDRモデルの要件

フロントエンド・アドレス

DBFSおよびFSS/rsyncモデルの「[要件](#)」→「[フロントエンド・アドレス](#)」と同じ。

WebLogicリソース名接頭辞

DBFSおよびFSS/rsyncモデルの「[要件](#)」→「[Weblogicリソース名プリフィックス](#)」と同じ。

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

DBFSおよびFSS/RSYNCモデルの「[要件](#)」→「[WebLogicのデータソースおよびJPSファイルでTNSエイリアスを使用](#)」と同じ。

サイト間のネットワーク通信

「[要件](#)」→「[サイト間のネットワーク通信](#)」で説明したように、**プライマリ・データベースとスタンバイ・データベースは、REDO転送のために、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**。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、Oracle Cloudの内部ネットワーク経由で通信できます。これは推奨されているアプローチです（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）。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）、この方法はセキュリティと信頼性の観点から推奨されません。

リージョン間ブロック・ボリューム・レプリケーションDRモデルでは、**追加のネットワーク通信は必要ありません**。リージョン間のrsyncは使用されず、中間層からリモート・データベースへの接続は不要です。

リージョン間ブロック・ボリューム・レプリケーションでのリージョン

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ます。また、このDRトポロジで使用されるプライマリ・リージョンとセカンダリ・リージョンの間で、リージョン間ブロック・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BVのリージョン間レプリケーション用に相互接続されている訳ではありません。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「[リージョン間ボリューム・レプリケーション](#)」の「[レプリケーション・ターゲット・リージョン](#)」の表を参照してください。

BVレプリケーションによるDRモデルのセットアップ・プロセス

BVレプリケーションを使用したDRモデルのセットアップ手順は、本書で説明したDBFSおよびFSS/rsyncの方法のセットアップ・プロセスを少し変えたものです。次にそのフロー・チャートを示します。

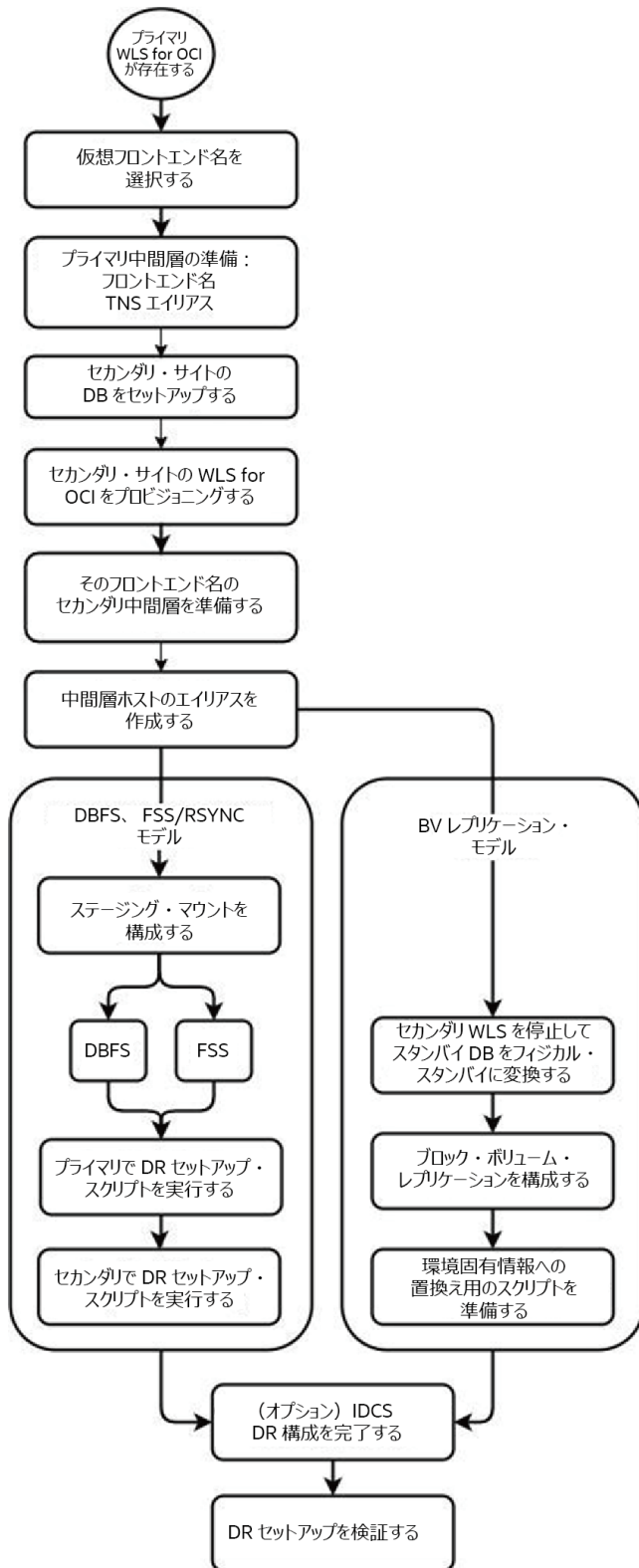


図16. 本書に記載したモデルのDRセットアップ手順を示すフロー・チャート

その他のモデルと同様に、開始時点で、OCI DBシステム、Oracle WLS for OCIドメイン、ロードバランサ（OCI LBR）で構成されたプライマリ・サイトが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・システムに対して作成されます。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・アドレスの変更のみです）。

次に、BVLレプリケーションに基づくセットアップ・プロセスの手順をまとめます。ステップ1～8までは、「[ディザスタ・リカバリのセットアップ](#)」で説明した手順と同じです。残りの手順は、BVLレプリケーションDRモデルに固有のものであり、これ以降の各ポイントで説明します。

1. 仮想フロントエンド名を選択する

「[ディザスタ・リカバリのセットアップ](#)」→「[1.仮想フロントエンド名を選択する](#)」

2. 仮想フロントエンドのプライマリ中間層を準備する

「[ディザスタ・リカバリのセットアップ](#)」→「[2.仮想フロントエンドのプライマリ中間層を準備する](#)」

3. プライマリ中間層でのTNSエイリアスの使用の準備

「[ディザスタ・リカバリのセットアップ](#)」→「[3.プライマリ中間層でのTNSエイリアスの使用の準備](#)」と同じ

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

「[ディザスタ・リカバリのセットアップ](#)」→「[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](#)」

5. セカンダリ・サイトのWLS for OCIをプロビジョニングする

「[ディザスタ・リカバリのセットアップ](#)」→「[5.セカンダリ・サイトのWLS for OCIをプロビジョニングする](#)」

6. 仮想フロントエンドのセカンダリ中間層を準備する

「[ディザスタ・リカバリのセットアップ](#)」→「[6.仮想フロントエンドのセカンダリ中間層を準備する](#)」

7. セカンダリ中間層でのTNSエイリアスの使用の準備

セカンダリ構成はプライマリ構成によって上書きされるため、この段階はスキップします。

8. 中間層ホストのエイリアスの構成

「[ディザスタ・リカバリのセットアップ](#)」→「[8.必要な中間層ホスト・エイリアスの構成](#)」と同じ

9. 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。

b) スタンバイ・データベースを再度フィジカル・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;
データベース" secondary_db_unqname"をフィジカル・スタンバイ・データベースに変換しています。しばらくお待ちください...
Oracle Clusterware is restarting database "orclb" ...
Continuing to convert database " secondary_db_unqname" ...
データベース" secondary_db_unqname"の変換が終了しました
```


10. 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WLS for OCIの**コンパートメント**を選択して、該当するブロック・ボリュームを探します。
各コンピュート・インスタンスには次の2つのブロック・ボリュームがあります。
 - **<wlsociprefix>-data-block-N** (/u01/dataにマウント)。このボリュームには、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。このブロック・ボリュームを**スタンバイ・サイトにレプリケートする必要があります**。
 - **<wlsociprefix>-mw-block-N** (/u01/appにマウント)。前述の「**レプリケートされるブロック・ボリューム**」の説明のとおり、mwブロック・ボリュームを継続的にレプリケートする必要はありません。このため、mwブロック・ボリュームのレプリケート手順はオプションとして記載しています。
- ブロック・ボリュームの**名前**、**OCID**、**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。

wlsociprefix-data-block-0、OCID ocid1.volume.oc1.eu-london-1.abtheljs...uhmlg2s3avjradeoirz3q、AD-1
wlsociprefix-data-block-1、OCID ocid1.volume.oc1.eu-london-1.abthelj...cdcayp6gw2irsri4nrgx7q、AD-2
オプション：

wlsociprefix-mw-block-0, with OCID ocid1.volume.oc1.eu-london-1.abtheljs...jjhbuhmlg2s3deoirkz4s, in AD-1
wlsociprefix-mw-block-1, with OCID ocid1.volume.oc1.eu-london-1.abthelj...wiphp6gw2irsri4nrh7jdy, in AD-2

これらのブロック・ボリュームは次の例のようにマウントされています。

```
[opc@wlsociprefix-wls-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 161M  7.1G  3% /run
tmpfs           7.2G  0  7.2G  0% /sys/fs/cgroup
/dev/sda3       39G  3.7G  35G  10% /
/dev/sda1       200M  8.6M  192M  5% /boot/efi
/dev/sdb        49G  1.4G  46G  3% /u01/app
/dev/sdc        49G  176M  47G  1% /u01/data
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
```

→ (オプション) このブロック・ボリュームもレプリケートできる
→ このブロック・ボリュームがレプリケート対象となる

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、OCID、可用性ドメインの情報を取得します。

c) プライマリ・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有効化する

プライマリWLS for OCIの各ブロック・ボリュームで、リージョン間レプリケーションを有効化します。

- ブロック・ボリュームを編集して、リージョン間レプリケーションを**ON**に設定します。
- **Region** : レプリケート先のリージョン。セカンダリ・リージョンを選択します。
- **Availability Domain** : セカンダリ・リージョン内で該当するブロック・ボリュームのあるADを選択します。例 : セカンダリ・リージョンのwlsociprefix-data-block-0がAD-3に配置されている場合は、プライマリのwlsociprefix-data-block-0のレプリカの場所としてAD-3を選択します。

- **Name** : 今後OCI CLIを使用して自動化を進めるために、レプリカの存在するリージョンにかかわらず、ブロック・ボリューム・レプリカ・オブジェクトに対して同じ名前を使用することを推奨します。使用例 : "**wlsociprefix-data-block-N_replicated_from_remote_site**"
- 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**」→「**Block Volume Replicas**」に移動します。

注 : 同じ可用性ドメイン内に複数のWLSコンピュート・インスタンスが含まれる場合、各ブロック・ボリュームを個別にアクティブ化する代わりに、ボリューム・グループを作成して、このグループに対してレプリケーションをアクティブ化することができます。

d) 元のブロック・ボリュームをセカンダリ中間層ホストからデタッチする

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、次の手順を実行します。

- データ・ブロック・ボリュームをアンマウントします。

```
[opc@wlsociprefix-wls-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。Oracleプロセスはこの時点で停止されているという前提ですが、このフォルダ上で実行中のプロセスがある場合は、umountは失敗します。

- アンマウントの完了後、OCIコンソールを使用してブロック・ボリュームをデタッチします。
各**ブロック・ボリューム**の「**Attached Instances**」→「**Detach from Instance**」に移動します。
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。これは、次の再起動で元のブロック・ボリュームがマウントされないようにするためです。例 :

```
..
UUID=7e5f896e-6e05-4be2-99e5-ddf2b3cbb021 /u01/app ext4 auto,defaults,_netdev,nofail 0 2
#このエントリを削除 :
#UUID=9e87cf72-a75c-4dff-9825-432f1668d8f9 /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。

mwブロック・ボリュームもレプリケートしている場合は、mwブロック・ボリュームと/u01/appマウント・ポイントに対しても同じステップを実行します。

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、名前を変更します。

f) セカンダリ・コンピュート・インスタンスでsystemd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”）へキャッシュされた参照をリフレッシュします。

11. 環境固有情報への置換用のスクリプトを準備する

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。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ます。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。

この置換作業を自動化するために、**replacement_script_BVmodel.sh**スクリプトが提供されています。

- a) 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
- b) それをプライマリおよびスタンバイで管理サーバーにアップロードします。レプリケート対象のブロック・ボリュームではない**フォルダに保存します**。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。
- c) 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）。
- d) 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。

12. IDCS認証の使用時にDR構成を完了する

「[ディザスタ・リカバリのセットアップ](#)」→「[12.IDCS認証使用時のDR構成の完了](#)」と同じです。

ただし、セカンダリ中間層ホストでCloudgate構成を変更する必要はありません（ステップb）。Cloudgate構成はレプリケートされるブロック・ボリューム（/u01/data）に含まれるため、すでにプライマリからコピーされています。

13. DRセットアップを検証する

これで、DR環境のセットアップは終了です。オラクルは、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（本書の「[スイッチオーバー](#)」を参照）を実行するか、またはセカンダリ・サイトを検証用に開きます（本書の「[検証のためのセカンダリ・サイトのオープン](#)」を参照）。

BVレプリケーションDRモデルのライフサイクル操作

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。リージョン間ブロック・ボリューム・レプリケーションをベースとして、WLS for OCIのDR構成でSite1からSite2への**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witch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。

	スイッチオーバー前の手順	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	<p>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）。プライマリからレプリケートされたボリュームのみをデタッチします。たとえば、レプリケートしていない場合は、mwブロックをデタッチしないでください。</p> <p>次に、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します。これらのボリュームは今後使用されません。</p>
2	Site2 でレプリカをアクティブ化する	<p>この時点まで、BVがSite1からSite2へ継続的にレプリケートされています。レプリケートされたBVをSite2でマウントするには、Site2のBVレプリカをアクティブ化する必要があります。</p> <p>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をコンピュータ・インスタンスにアタッチできます（「BLOCK VOLUMES」→「BLOCK VOLUME REPLICAS」に表示される"ブロック・ボリューム・レプリカ"のエンティティはコンピュータ・インスタンスにアタッチできません）。</p> <p>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。</p> <ul style="list-style-type: none"> - Site2の「Block Storage」→「Block Volume Replicas」に移動します。 - BVレプリカをクリックして、「Activate」を選択します。 - レプリカ名としては、作成された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。できれば元のブロック・ボリュームと同じ名前を使用してください。そうすることで、OCI CLIを使用した自動化が容易になります。 例："wlsprefix-data-block-N"（mwもレプリケートされる場合は"wlsprefix-mw-block-N"） - Site2のすべてのBV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BVを Site2の中間層ホストに アタッチする	<p>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。</p> <p>- OCIコンソールの「Block Volume」で、「Block Volume」→「Attached Instances」→「Attach to Instance」をクリックします。</p> <p>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/en-us/iaas/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コマンドを実行します。</p> <p>- 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。</p> <pre>[root@wlsociprefix-wls-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-b372-43720aaae717" /dev/sdb:UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4" □</pre> <p>- /etc/fstabに該当するUUIDのエントリを追加して、マウントを実行し、再起動後もそのマウントが継続するようにします。 UUID=35e72262-979a-4d84-85ce-a6f91e3b1250 /u01/data ext4 auto,defaults,_netdev,nofail</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。

- 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、正常にマウントされたことを確認します。

```
[root@wlsociprefix-wls-0 opc]# mount -a
[root@wlsociprefix-wls-0 opc]# df -h | grep /u01/data
/dev/sdb 49G 1.4G 46G 3% /u01/data
```

オプション：mwブロック・ボリュームをレプリケートしている場合は、中間層ホストで、mwブロック・ボリューム（および/u01/appマウント・ポイント）に対してこれらのステップを繰り返します。必ず正しいパスで正しいUUIDにマウントするようにします。

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。

4 Site2で置換用スクリプトを実行する

Site2の管理者ホストで、**replacement_script_BVmodel.sh**スクリプトを実行します。

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。

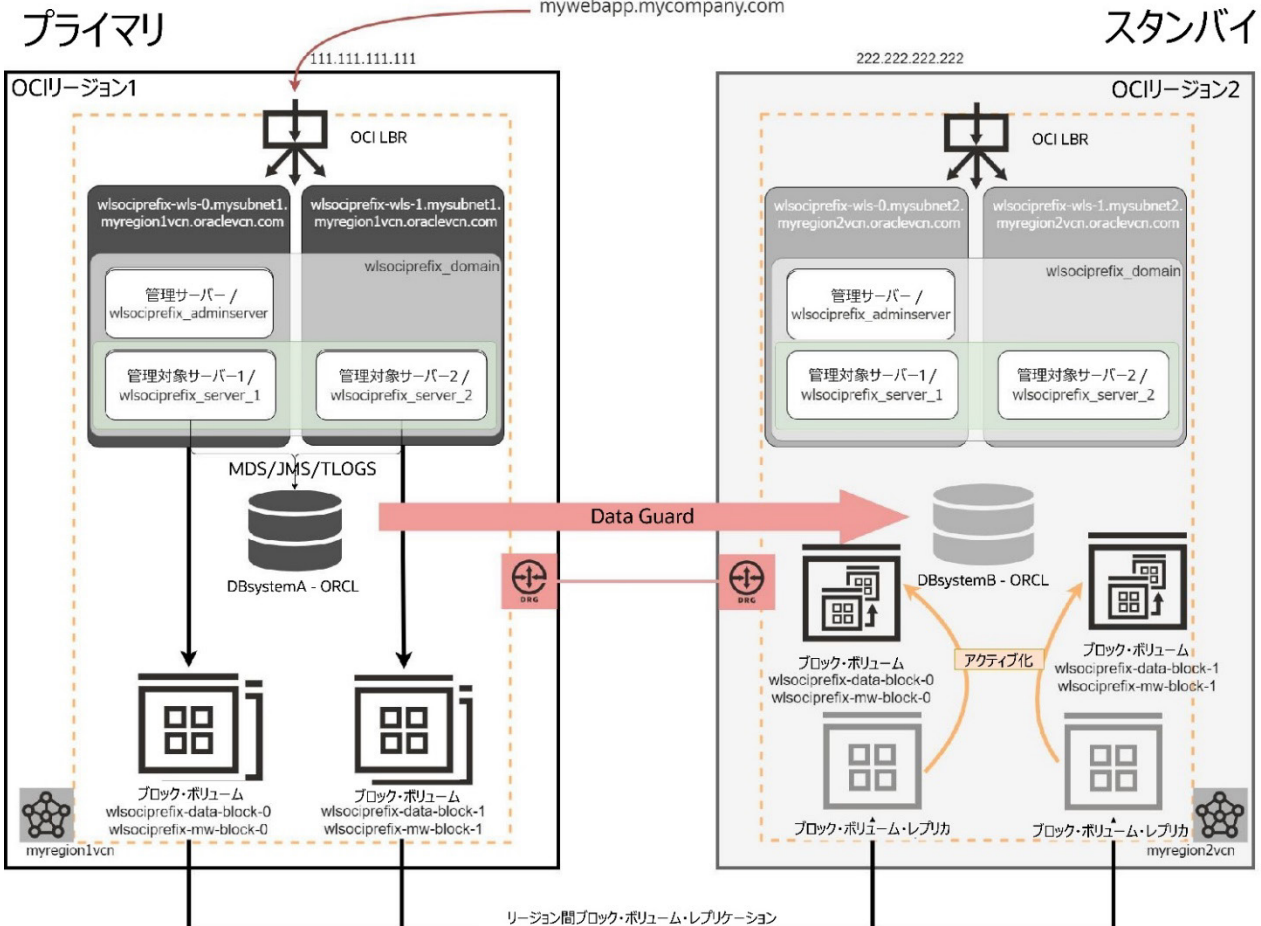


図17：スイッチオーバー前操作の実行中に、ブロック・ボリューム・レプリカがスタンバイ・サイトでアクティブ化されます。

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。

	スイッチオーバー手順	詳細
1	プライマリ・サイトのプロセスを停止する	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。 注：WLS for OCI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MAAI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。
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仮想アドレスがサイト2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@drdbwlp1a ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> 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	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： \$ cd \$DOMAIN_HOME/bin/ \$ nohup ./startNodeManager.sh > \$DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 & セカンダリ管理サーバーを起動します。例： \$ cd /u01/app/oracle/middleware/oracle_common/common/bin \$./wlst.sh wlst> nmConnect('weblogic','acme1234#','wlsociprefix-wls-0','5556','wlsociprefix_domain','/u01/data/domains/wlsociprefix_domain','SSL') wlst> nmStart('wlsociprefix_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）。 注：WLS for OCI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MAAI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、サービスはSite2でアクティブ状態です。そのため、追加の停止時間は必要ありません。ただし、スイッチオーバーの手順を完了し、システムのロールが適切に入れ替わった状態にするための、追加のタスクがあります。次のタスクをすぐに行うことを推奨します。

- Site2（新しいプライマリ）の中間層ホストのブロック・ボリュームで、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）のブロック・ボリューム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SCSI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、維持することができますが、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。
- 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。これらのボリュームは今後使用されません。

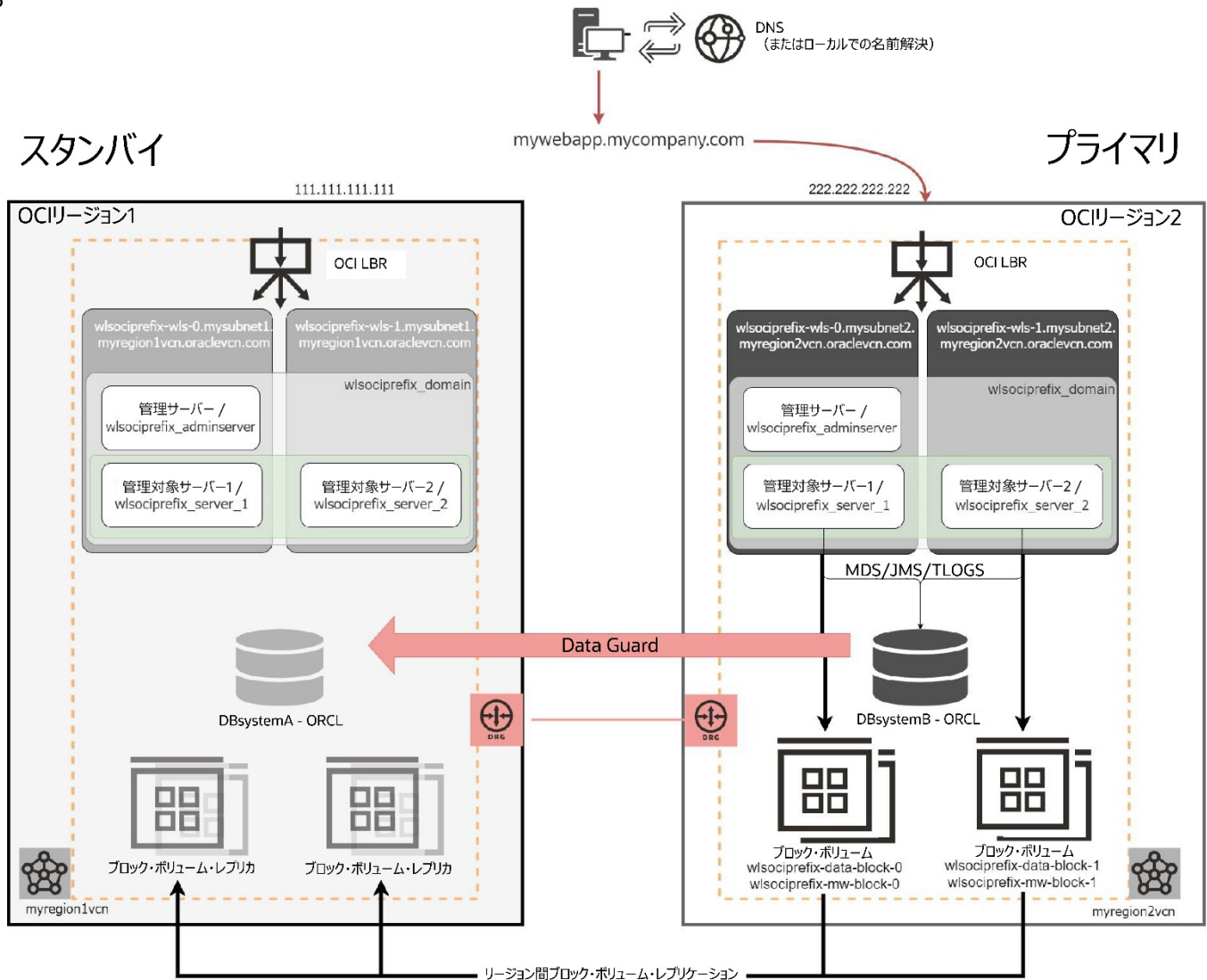


図18：スイッチオーバー後、逆方向のブロック・ボリューム・レプリケーションを有効化するための実行後手順

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。

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@drdbw1mp1b ~]$ dgmgml sys/your_sys_password@secondary_db_unqname
DGMGRL> failover to "secondary_db_unqname"
```

その他の手順は、事前処理および事後処理を含めて、スイッチオーバーと同じです。

検証のためのセカンダリ・サイトのオープン

このモデルでも、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せずにスタンバイ・サイトを検証できます。これにより、セカンダリWLS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。

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	Site2でレプリカをアクティブ化する	<p>この時点まで、BVがSite1からSite2へ継続的にレプリケートされています。レプリケートされたBVをSite2でマウントするには、Site2のBVレプリカをアクティブ化する必要があります。</p> <p>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（「BLOCK VOLUMES」→「BLOCK VOLUME REPLICAS」に表示される"ブロック・ボリューム・レプリカ"のエンティティはコンピュート・インスタンスにアタッチできません）。</p> <p>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。</p> <ul style="list-style-type: none"> - Site2の「BLOCK VOLUMES」→「BLOCK VOLUME REPLICAS」に移動します。 - 各BVレプリカをクリックして「Activate」を選択します。 - レプリカ名としては、作成された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。できれば元のブロック・ボリュームと同じ名前を使用してください。そうすることで、OCI CLIを使用した自動化が容易になります。例： "prefix-data-block-N"（mwもレプリケートされる場合は"prefix-mw-block-N"） - Site2のすべてのBVレプリカに対して、同じ操作を繰り返します。

2	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	<p>アクティブ化によって作成されたアタッチ可能なブロック・ボリュームが、Site2のOCIコンソールの「Storage」→「Block Volume」に表示されます。</p> <p>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。</p> <p>-この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。</p> <p>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タ・インスタンス上で実行できるようにするには、https://docs.oracle.com/en-us/iaas/Content/Block/Tasks/enablingblockvolumemanagementplugin.htm#blockplugin-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 コマンドを実行します。</p> <p>-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。</p> <pre>[root@wlsociprefix-wls-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-b372-43720aaae717" /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、正常にマウントされたことを確認します。</p> <pre>[root@wlsociprefix-wls-0 opc]# mount -a [root@wlsociprefix-wls-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> <p>オプション：mwブロック・ボリュームをレプリケートしている場合は、中間層ホストでmwブロック・ボリューム（マウント・ポイント/u01/app）に対して同じ手順を繰り返します。必ず正しいパスで正しいUUIDにマウントするようにします。</p>
3	Site2の中間層で置換を行う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>
4	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ブローカを使用して、セカンダリをスナップショット・スタンバイに変換します。ユーザーoracleで次を実行します。</p> <pre>[oracle@drdbA ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to snapshot standby</pre> <p>“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。</p>
5	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。例：</p> <pre>:\$ cd /u01/app/oracle/middleware/oracle_common/common/bin \$./wlst.sh wlst> nmConnect ('weblogic','acme1234#','wlsociprefix-wls-0','5556','wlsociprefix_domain','/u01/data/domains/wlsociprefix_domain','SSL') wlst> nmStart('wlsociprefix_adminserver')</pre> <p>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）。</p> <p>注：WLS for OCI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。</p>
6	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プロキシ経由の場合、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、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る可能性があります。</p> <p>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ために、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>

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

	スタンバイをスタンバイ・ロールに戻す手順	詳細
1	セカンダリの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ます。
2	スタンバイDBを再びフィジカル・スタンバイに変換する	プライマリDBホストでDGブローカを使用して、セカンダリを再びフィジカル・スタンバイに変換します。As user oracle: [oracle@drdbA ~]\$ dgmgml 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。
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。 - レプリケートされたブロック・ボリュームをアンマウントします。 [opc@wlsociprefix-wls-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。Oracleプロセスはこの時点で停止されているという前提ですが、このフォルダ上で実行中のプロセスがある場合は、umountは失敗します。 - アンマウントの完了後、OCIコンソールを使用してブロック・ボリュームをデタッチします。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。 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iscsciコマンドを実行するよう求められます 。 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。
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。これらのボリュームは今後使用されません。

スケール・アウトとスケール・イン

リージョン間ブロック・ボリュームのDRモデルにおけるスケール・アウトとスケール・インの手順は、DBFSモデルおよびRSYNCモデルの手順とは少し異なります。

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

- はじめに、1つ前の「[検証のためのセカンダリ・サイトのオープン](#)」セクショ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。
- 「[スケール・アウト](#)」セクションに記載された手順に従って、プライマリとセカンダリでスケール・アウトを実施します。config_replica.sh スクリプトに関する説明についてはすべて無視してください。このモデルには適用されません。
- 「[検証のためのセカンダリ・サイトのオープン](#)」セクションの説明に従い、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。
- 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。

注：WLS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BV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、マウントされてはいません。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、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。

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WLS for OCI DR環境のスケール・インの手順は次のとおりです。

- はじめに、1つ前の「[検証のためのセカンダリ・サイトのオープン](#)」セクショ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、またスタンバイ・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。
- 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。
- 「[スケール・イン](#)」セクションに記載された手順に従って、プライマリとセカンダリでスケール・インを実施します。config_replica.shスクリプトに関する説明についてはすべて無視してください。このモデルには適用されません。
- 「[検証のためのセカンダリ・サイトのオープン](#)」セクションの説明に従い、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：WLS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BV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・ボリュームを指し示したマウントを作成しないように、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。

OCI CLIコマンドを使用したブロック・ボリューム・レプリケーション・タスクの自動化

ブロック・ボリュームに関連する操作の大半（レプリケーションの有効化/無効化、アクティブ化、ブロック・ボリュームのアタッチ/デタッチ）はOCIコンソールを使用して実行できるだけでなく、OCIコマンドライン・インタフェース（CLI）のコマンドでも実行できます。

このDRモデルのセットアップやスイッチオーバー/フェイルオーバーでよく使用する操作について、以下にコマンドの一覧を示します。CLIの使用については、詳しくは[Oracle Cloud Infrastructure CLI Command Reference](#)と[ボリュームのレプリケート](#)を参照してください。

- ブロック・ボリュームのリージョン間レプリケーションを有効化する

```
oci bv volume update --volume-id <volumeId> --block-volume-replicas [{"displayName":"<replicaDisplayName>","availabilityDomain":"<availabilityDomain>"}]
```

- リージョン間ブロック・ボリューム・レプリケーションの無効化

```
oci bv volume update --volume-id <volumeId> --block-volume-replicas '[]'
```

- ブロック・ボリューム・レプリケーションのアクティブ化

```
oci bv volume create --source-volume-replica-id <volumeReplicaId> --compartment-id <compartmentID> --availability-domain <AvailabilityDomain> --display-name <DisplayName>
```

- コンピューター・インスタンスへのブロック・ボリューム（iscsiタイプ）のアタッチ

```
oci compute volume-attachment attach-iscsi-volume --instance-id <instanceId> --volume-id <volumeId>
```

- コンピューター・インスタンスからのブロック・ボリュームのデタッチ

```
oci compute volume-attachment detach --volume-attachment-id <volumeAttachmentId>
```

上記のとおり、OCI CLIコマンドを実行するには、ブロック・ボリュームのOCIDとボリューム・アタッチIDが必要です。OCIDはスイッチオーバーを実行するたびに変更されます。ただし、これらの値はOCI CLIスクリプトを使用して取得することもできます。ブロック・ボリューム・オブジェクトに使用する"表示名"が一貫している限り、表示名を使用して動的にOCIDを取得できます。

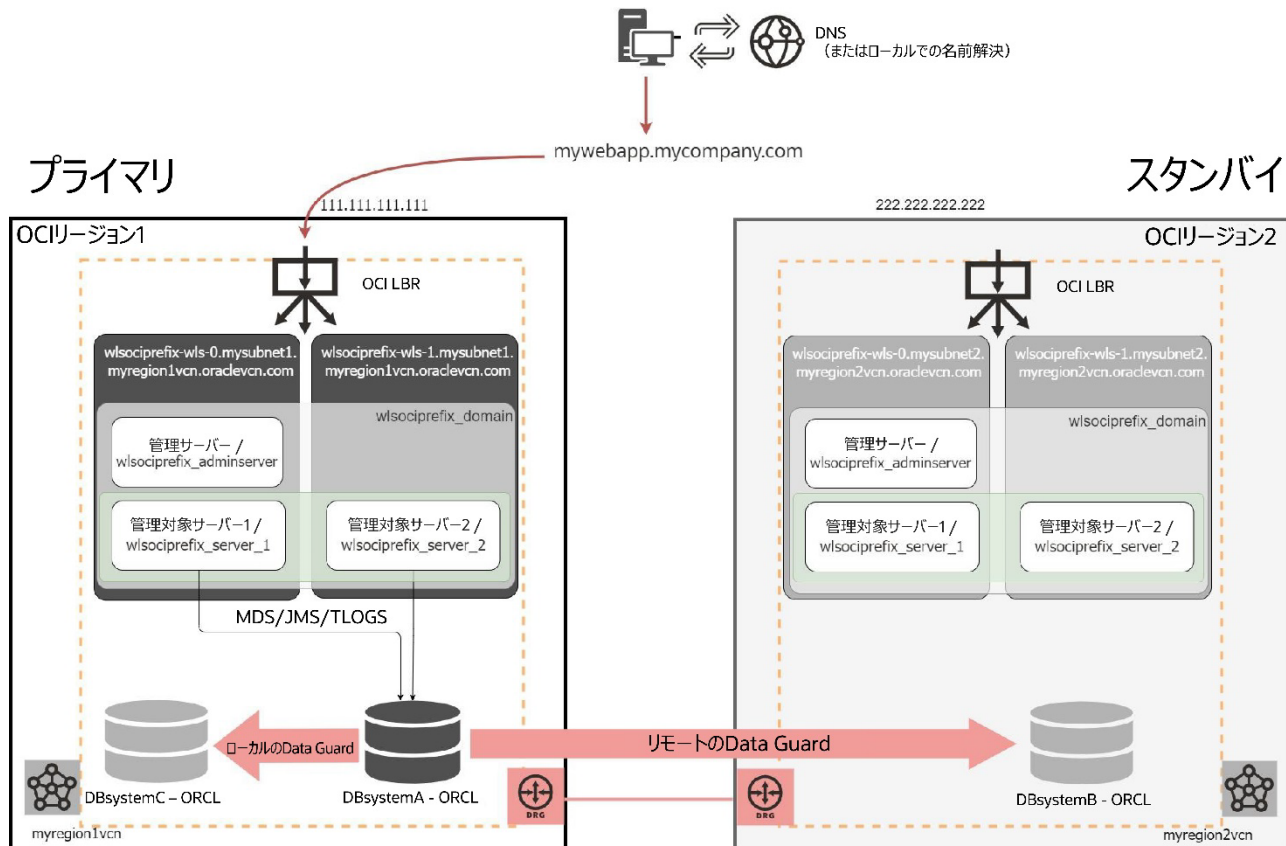
8

ただし、OCI CLIコマンドですべてのステップを自動化できる訳ではありません。ブロック・ボリュームのアタッチやデタッチなどの一部の操作では、OCI コンソール（またはOCI CLI）での操作とオペレーティング・システムのコマンド（fstabの編集、mount、iSCSIコマンドの実行など）を組み合わせる必要があります。ブロック・ボリューム・レプリケーションを使用したOracle WLS for OCIのDRモデルで、スイッチオーバーを半自動化するスクリプトの例については、https://github.com/oracle-samples/maa/tree/main/wls_mp_dr/Block_Volume_Replica_Methodを参照してください。

付録F - 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、リージョン間DRのセットアップを実行することはできますが、**DRセットアップ・スクリプトの実行前に、Data Guardブローカ構成からローカル・スタンバイを削除する必要があります**。DRセットアップ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。



構成前の手順

リージョン間DR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。DRセットアップ・スクリプトの実行前に、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。例：

```
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
  ORCL_frankfurt - Physical standby database
Fast-Start Failover:Disabled
Configuration Status:

SUCCESS (status updated 1 seconds ago)

```

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WLS for OCI 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の場合は、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>)))

リージョン間のDR構成

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。

構成後の手順

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。例：

```

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 specified threshold
Fast-Start Failover:Disabled
Configuration Status:
WARNING (status updated 57 seconds ago)

```

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。例：

```

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_lhr3xb:YES ORCL_lhr132:
  対応

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。

2. 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 mypdbservice.example.com -preferred ORCL1,ORCL2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db $ORACLE_UNQNAME -service mypdbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT
srvctl config service -db $ORACLE_UNQNAME -service mypdbservice.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。例：

```
srvctl add service -db $ORACLE_UNQNAME -service mypdbservice.example.com -preferred ORCL1,ORCL2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY "
srvctl modify service -db $ORACLE_UNQNAME -service mypdbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT
srvctl config service -db $ORACLE_UNQNAME -service mypdbservice.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ブローカによって自動的に停止/開始されます）。

```
srvctl start service -db $ORACLE_UNQNAME -service mypdbservice.example.com
```

3. 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=mypdbservice.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）**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。

6. (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WLS for OCI 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。例：

```
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。例：

```
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。

7. (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WLS for OCI DR環境でのみ必要とされます。

レプリケーションにDBFS手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。

- それぞれの中間層で、\$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。例：

```
..
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。例：

```
...
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)))
..
```

この例では、

“ORCL_london2”は、ローカル・スタンバイの一意的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・データベースのスイッチオーバー

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する必要があります。このためには、データソースとjps-config.xml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、いず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。WebLogic構成コンソールに変更が反映されていることを確認します。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。
6. 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までご連絡いただくか、oracle.comをご覧ください。

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



blogs.oracle.com



facebook.com/oracle



twitter.com/oracle

Copyright © 2023, 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黙示的保証を含め、商品性ないし特定目的適合性に関する黙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 WebLogic Server for Oracle Cloud Infrastructureのディザスタ・リカバリ、
2023年6月

