

Oracle DatabaseのAmazon Web Servicesから Oracle Cloud Infrastructure Databaseへの移行

ORACLE WHITE PAPER | 2018年5月



目次

このホワイト・ペーパーの目的	4
対象範囲と前提事項	4
はじめに	4
Oracle Cloud Infrastructure Database	5
オンライン移行とオフライン移行	6
移行ユーティリティ	7
SQL Developer	7
RMAN	7
Data Pump	8
SQL*Loader	8
Oracle GoldenGate	8
AWS DMS	8
移行するオペレーティング・システム	8
AWSとOracle Cloud Infrastructureの間の相互接続性	9
接続の概念	10
接続オプション	11
接続オプションの選択	17
AWSからOracle Cloud Infrastructure Databaseへの移行ガイダンス	19
Amazon RDSからOracle Cloud Infrastructureへの移行オプション	22
RDS移行オプション1 : SQL Developer Data Pump : エクスポート/インポート	23
RDS移行オプション2 : SQL Developer : コピー	27
RDS移行オプション3 : SQL Developer : SQL*Loader	29
RDS移行オプション4 : SQL Developer : マテリアライズド・ビュー	32
RDS移行オプション5 : AWS DMSを使用した移行	34

Amazon EC2からOracle Cloud Infrastructureへの移行オプション	39
EC2移行オプション1：Data Pump：従来型の表領域のエクスポート/インポート	40
EC2移行オプション2：Data Pump：フル・トランスポータブル	41
EC2移行オプション3：Data Pump：トランスポータブル表領域	43
EC2移行オプション4：リモート・クローン：PDB	46
EC2移行オプション5：リモート・クローン：非CDB	46
EC2移行オプション6：RMAN：トランスポータブルPDB	47
EC2移行オプション7：RMAN：トランスポータブル・バックアップ・セット	49
EC2移行オプション8：RMAN：アクティブ・データベースからの複製	52
EC2移行オプション9：PDBのアンプラグとプラグ	55
EC2移行オプション10：非CDBのアンプラグとプラグ	56
EC2移行オプション11：Data Guardを使用した移行	58
EC2移行オプション12：Recovery Managerを使用した移行	62
レプリケーションのためのOracle GoldenGate	66
移行後の手順：透過的データベース暗号化	72
結論	72
参考資料	72

このホワイト・ペーパーの目的

Oracle Cloud Infrastructureは、エンタープライズ・データ・センターのパフォーマンス、制御、およびガバナンス要件を十分に満たしつつ、パブリック・クラウドのスケール、伸縮性、およびコスト節約性も提供する、優れた環境です。Oracle Cloud Infrastructureは、Oracleワークロードを実行するのに最適なクラウド・プラットフォームです。データベースをOracle Cloud Infrastructureに移行するにあたっては、検討すべきいくつかのオプションがあります。このホワイト・ペーパーでは、このような移行時に利用できる各種のオプションについて詳しく説明します。

対象範囲と前提事項


プライベートまたはパブリック・クラウド上にデプロイされたアプリケーションに使用できるデータベース・プラットフォームは、いくつか存在します。このドキュメントでは、読者が現在OracleデータベースをAmazon Web Services（AWS）環境にデプロイしていること、およびOracle Cloud Infrastructureへの移行を希望していることを前提とします。お客様は、デプロイしているOracle Relational Database Management System（RDBMS）のリリースとデプロイのモードに応じて、いくつかの異なる移行オプションが利用できます。このドキュメントでは、利用可能なすべてのオプションについて説明するのではなく、Oracle Cloud Infrastructureを最も迅速かつ簡単に移行できるオプションについて重点的に説明します。

このドキュメントで紹介する一部の例では、移行時にバージョンのアップグレードを実行しますが、データベース・バージョンのアップグレードに対応することは、このドキュメントの趣旨ではありません。

はじめに

Oracle Cloud Infrastructureは、最新のテクノロジーを利用してセキュリティと信頼性に優れた環境を提供し、大規模で複雑なインフラとアプリケーションのデプロイを効果的に管理できるようにする目的で設計されました。高い可用性を実現するため、Oracle Cloud Infrastructureでは、複数の地理的リージョンにクラウドがデプロイされます。各リージョンには、フォールト・インディペンデントな（つまり、他のエンティティとの間で障害が波及しない）可用性ドメインが少なくとも3つあり、それらの各ドメインには、電力、温度、およびネットワークの管理が分離されている、独立したデータ・センターが配置されます。Oracleのフラットで高速なネットワークでは、隣接するラック間のようなレイテンシとスループットがネットワーク全体で提供されるため、同期的なレプリケーションとコンスタントなアップタイムが確保されます。ネットワークやCPUのオーバーサブスクリプションがないので、帯域幅やパフォーマンスの予測可能性も確保されます。

Oracle Databaseは、38年前から広く使用されている、ACID準拠のリレーショナル・データベース管理システム（RDBMS）です。Oracleデータベースは、全世界の大規模エンタープライズ・ワークロードの多くに使用されており、通常は、それらのワークロードの中でも最も重要な部分に使用されています。Oracle Corporationは、データベース市場での38年間にわたるリーダーシップと、エンタープライズIT環境に関する深い知識を活かして、次世代のクラウド・インフラである、Oracle Cloud Infrastructureを設計、構築しました。Oracle Cloud Infrastructure Databaseサービスは、これまでと同じOracle RDBMSを、パブリック・クラウド環境でできるようにするサービスです。



Oracle Cloud Infrastructure Databaseでは、AWSにデプロイされたOracleデータベース（RDSまたはEC2インスタンス）と比べて、次のようなユニークな利点が提供されます。

- 最新バージョンのOracleデータベースをデプロイできます
- Oracleデータベースのすべての機能とオプションをデプロイし、利用できます（たとえば、Oracle Database In-Memory、Oracle Multitenantオプションなど）
- クリティカルなOracleデータベース・ワークロードのパフォーマンス需要や可用性需要に十分対応できるようエンジニアリングされた、最適なクラウド・プラットフォームにOracleデータベースをデプロイできます
- パブリック・クラウド上のOracleデータベースに最善のコスト・パフォーマンスが提供されます
- エディションの異なるOracle（SE1、EEなど）を、同じライセンス・モデル（および単一のデータベース・サポート・モデル）で使用できます
- ハイブリッドのOracleデータベース環境（オンプレミスとパブリック・クラウド）をシームレスに実装して管理でき、ベンダー・ロックインも発生しません
- Real Applicationクラスタを実装して、Oracleデータベースの高可用性とスケーラビリティを確保できます
- 特に高負荷なOracleデータベース・ワークロードをExadata Cloud Serviceに実装することで、Oracleデータベースを実行するための最善のプラットフォームとしてエンジニアリングされたプラットフォームならではの、最高のパフォーマンス、スケーラビリティ、およびセキュリティを享受できます

このホワイト・ペーパーの残りのセクションでは、Oracle Cloud Infrastructure Databaseサービスについて説明するとともに、AWS環境をOracle Cloud Infrastructureに接続するためのネットワーキング・オプション、移行に必要なツール、および各移行方法での移行手順について説明します。このホワイト・ペーパーでは、17種類の移行方法について説明します。各方法をどのような場合に使用するかについては、「AWSからOracle Cloud Infrastructure Databaseへの移行ガイドンス」セクションに記載されています。

Oracle Cloud Infrastructure Database

Oracle Cloud Infrastructureは、エンタープライズITのワークロードとクラウドネイティブのワークロードに対応した、比類のないパフォーマンスを提供できるよう設計されています。これは、OracleアプリケーションとOracleデータベースを実行するための最善のインフラであり、Oracle Real Applications Clusters（RAC）とOracle Exadataシステムが提供される唯一のデータベース・サービスです。Oracle Cloud Infrastructure上のDatabaseサービスを利用することで、クラウド上のOracleデータベースを簡単に構築し、スケーリングし、保護することができます。お客様は、パッチ適用、Data Guard、バックアップ/リカバリなどのシンプルなツールを使用して、データベースを管理できます。また、いずれのツールにも、Oracle Cloud Infrastructure REST APIまたはコンソールを使用してアクセスできます。また、データベース・ホストにアクセスし、既存のツールを使用して、クラウド上のデータベースをオンプレミスと同じように管理することもできます。

Oracle Cloud Infrastructure Databaseサービスでは、Oracle Databaseバージョン11.2.0.4、12.1.0.2、12.2.0.1、および18.0.0.1がサポートされています。

Databaseサービスでは、次のシェイプがお客様に提供されます。

- **仮想マシン・シェイプ**：Databaseサービスは、標準のVMコンピュート・シェイプをベースにした、様々な仮想マシン（VM）で利用できます。VMシェイプは、1～24基のコアと、256 GB～40 TBのスケラブルで高耐久なリモート・ストレージの選択肢から、コスト効率よく柔軟に選ぶことができます。VMは、データベース・アプリケーションに専用のサーバーを必要としないお客様に最適な選択肢です。VMでは、2ノードのOracle RAC環境を使用することもできます。
- **ベア・メタル・シェイプ**：Databaseサービスでは、クリティカルなエンタープライズ・アプリケーションの厳しいパフォーマンス・ニーズに対応できる、ベア・メタル・シェイプがサポートされています。このオプションでは、NVMEまたはSSDドライブをローカルにアタッチしたサーバー全体が、1社のお客様専用で使用されます。ベア・メタル・オプションでは、2～52基のコア、512 GB～768 GBのメモリ、および3.5 TB～16.5 TBの使用可能データベース・ストレージを選択できます。
- **Exadataシェイプ**：Oracle Exadata Database Machineは、Oracleデータベースを実行するための最善のパフォーマンスと、最高の可用性、および最高のセキュリティを提供します。お客様は、Exadata Database MachineをOracle Cloud Infrastructure上で使用し、22～335基のCPU、1440 GB～5760 GBのRAM、および84 TB～336 TBの使用可能ストレージを選択できます。これらの環境は、最大規模のデータベース・ワークロードに対応でき、最善のパフォーマンスと可用性を提供するものとなっています。

オンライン移行とオフライン移行


どの移行方法を選択するかは、ご使用のアプリケーションにおいて、どれくらいのダウンタイムなら許容されるかを基準に判断します。

- アプリケーションで許容されるダウンタイムが24～72時間の場合は、**オフライン移行**を検討してください。

オフライン移行の場合は、ソース・データベースが移行の間中シャットダウンされます。データベースは、1回のステップでAWSからOracle Cloud Infrastructureへと移行されます。移行の進行中にデータが変更されないようにして、データの一貫性を保つ必要があるため、シャットダウンは必須となります。

- データベースのアップタイム要件上、長時間のダウンタイムを許容できず、最小限のダウンタイムか、またはゼロ・ダウンタイムで移行を達成する必要がある場合は、**オンライン移行**を検討してください。

オンライン移行では通常、移行プロセスにおいて複数のステップが必要になります。最初のステップでは、Amazon Relational Database Service（Amazon RDS）内のソースから、



Oracle Cloud Infrastructure内のターゲット・データベースへと、データベース全体をコピーします。その後のステップでは、ソース・データベースで発生した変更の内容をターゲット・データベースに同期します。カットオーバーの際には、最終的な変更結果がレプリケートされ、アプリケーションはOracle Cloud Infrastructure内のターゲット・データベースを使用するように切り替えられます。オンライン移行に向けた準備として、Oracle GoldenGateやAWS Database Migration Service（DMS）などのツールを使用し、ソース・データベースとターゲット・データベースの同期を維持することもできます。

移行ユーティリティ

AWSからOracle Cloud Infrastructureへの移行には、いくつかのツールが利用できます。Oracle SQL Developer、Oracle SQL*Loader、Oracle Data Pump、AWS DMS、およびOracle GoldenGateは、データベースの論理移行（データをソースからアンロードし、ターゲットに再ロードする移行方法）を実行するために使用できます。Oracle Recovery Manager（RMAN）などのツールは、物理移行（データをそのままコピーする移行方法）を実行するために使用できます。

- 非常に大きなデータベースを移行する場合は、物理移行を実行したほうがメリットがある場合が多いです。なぜなら、移行後にインデックスの再構築や統計情報の収集などを行う必要がないからです。
- 論理移行は、部分移行を実行する場合や、データベース・レイアウトの変更、文字セットの変更、ブロック・サイズの変更、またはデータベースのデフラグを行う場合に適しています。論理移行は、ネットワークを介してソースからターゲットへと転送されるデータの量を大幅に減らしたい場合にも役立ちます。

SQL Developer

SQL Developerは、オラクルによって提供されている統合開発環境（IDE）です。Oracleデータベースの開発と管理を簡素化するのに役立ちます。JavaベースのプラットフォームであるこのIDEは、Linux、Mac OS X、およびWindowsプラットフォーム上で動作します。SQL Developerでは、Data Pumpエクスポート、データベース・コピー、SQL*LoaderなどのOracleツールを使用できるオプションが提供されるので、データベース移行を効率的に実行できます。

RMAN

RMANは、Oracleデータベースのリカバリ・マネージャ・ツールで、通常は、データベースをバックアップしてリカバリし、データを保護するために使用されます。RMANには、複製やトランスポートブル表領域など、データベース移行を容易にするための様々なオプションがあります。

Data Pump

Data Pumpユーティリティを使用すると、ソース・ファイルのデータをOracle固有の形式（OS非依存）の出力ファイルへと抽出し、その出力ファイルをターゲット・データベースへとロードすることができます。Data Pumpエクスポートは、要件に応じて、データベース全体を対象に実行することもできますし、特定のスキーマを対象に実行することもできます。このユーティリティでは、エクスポートとインポートをパーソナライズしたり、バックアップ・ファイルを暗号化したりできるオプションが提供されます。

SQL*Loader

SQL*Loaderは、オラクルによって提供されている、一括式のフラットファイル・ローディング・ユーティリティです。SQL*Loaderでは、ほぼすべての入力フラット・ファイルをOracleデータベースへのロード用にフォーマットできる、強力な宣言型言語が利用できるため、フラット・ファイルを適切にフォーマットするための手間が最小限に抑えられます。

Oracle GoldenGate

Oracle GoldenGateを使用すると、異なるOracleデータベース間でリアルタイムなデータ統合を実行できます。Oracle GoldenGateでは、ソース・データで発生する変更をキャプチャして変換し、ターゲット・データベースにプロパゲートして適用するための必要なプログラムがすべて提供されます。お客様は、Oracle GoldenGateを使用して、ソース・データベースの初期コピーをAmazon RDSからOracle Cloud Infrastructure上のOracle Databaseインスタンスへと移し、実際のカットオーバーが実行されるまで、それらの同期を維持することができます。これにより、新しい移行先データベースへのカット・オーバーを最小限のダウンタイムで達成することができます。

AWS DMS

AWS Database Migration Service（DMS）は、AWSにあるOracleデータベースを、Oracle Cloud Infrastructure Databaseサービス内のOracleデータベースへと移行するために使用できます。AWS DMSでは、ソース・データベースのフル・スキーマをターゲット・データベースにコピーできます。また、ソース・データベースでのすべての変更をターゲットにレプリケートして、Oracle Cloud Infrastructure内のデータベースへとカット・オーバーするときまで、表やデータをソースと同期しておくことができます。

移行するオペレーティング・システム

AWS（Amazon RDSまたはAmazon EC2）でのOracleデータベース実装には、多くの場合、Linux 64ビット・オペレーティング・システムが使用されます。Oracle Cloud Infrastructure Databaseサービスのインスタンスでも、64ビットLinuxオペレーティング・システムが使用されます。このドキュメントで説明している移行オプションはいずれも、ソース環境で64ビットLinuxオペレーティング・システムが使用されていることを前提としたものです。ソース・データベース・インスタンスでMicrosoft Windows 64ビット・オペレーティング・システムが使用されている場合は、Oracle Data Guardを使用してデータベースを移行することをお勧めします。

AWSとOracle Cloud Infrastructureの間の相互接続性

大きなデータベースを移行するには、ネットワークを介して、ソース・データベース・インスタンスの大量のデータを、できる限り迅速かつ安全に、ターゲット・インスタンスへと転送する必要があります。専用の帯域幅と一貫した低レイテンシを提供する堅牢なネットワークを確保すれば、このデータ転送を短い時間枠で完了することができます。

インターネット・ゲートウェイ（IGW）と動的ルーティング・ゲートウェイ（DRG）は、Oracle Cloud Infrastructureの仮想クラウド・ネットワーク（VCN）を、インターネット、オンプレミス・データ・センター、またはその他のクラウド・プロバイダへ接続するために使用できる、サービス・ゲートウェイ・オプションです。

このセクションでは、Oracle Cloudへの一般的なネットワーク接続を計画する際に利用できる接続サービス・オプションについて説明するとともに、クラウド・プロバイダ間の接続オプションについても説明します。すべての主要なクラウド・サービス・プロバイダ（CSP）では、3つのネットワーク接続サービス・オプションが提供されています。パブリック・インターネット、IPSec VPN、および専用接続です。オラクルの専用接続サービスは、Oracle Cloud Infrastructure FastConnectと呼ばれます。ワークロードや、転送する必要があるデータの量に応じて、1つ、2つ、または3つすべてのネットワーク接続サービス・オプションが必要となります。

	最大（Mb/秒）	レイテンシ	ジッター	コスト	セキュア
パブリック・インターネット	10,000以下	可変	可変	可変	いいえ
IPSec VPN	250以下	可変	可変	可変	はい
FastConnect	100,000以下	予測可能	予測可能	予測可能	はい

パブリック・インターネットでは、インターネットに接続された任意のデバイスからのアクセス性が提供されます。IPSec VPNは、ネットワークをクラウドに拡張することでアクセスが提供される、セキュアな暗号化ネットワークです。FastConnectでは、専用の接続と、インターネットへの代替接続が提供されます。このサービスでは、その排他的な性質を活かし、高い信頼性と低レイテンシ、専用の帯域幅、およびセキュアなアクセスが提供されます。

FastConnectサービスでは、次の[接続モデル](#)が提供されます。

- Oracleネットワーク・プロバイダまたは交換パートナーを経由した接続
- データ・センター内での直接ピアリングを経由した接続
- サードパーティ・ネットワークからの専用回線を経由した接続

接続の概念

移行に使用できる特定の接続モデルについて見ていく前に、Oracle Cloud Infrastructureのネットワーク接続に使用される、いくつかの重要なコンポーネントについて確認しておきましょう。

プライベート・ピアリング：既存のインフラを、Oracle Cloud Infrastructure内のVCNへと拡張できます。接続間の通信には、IPv4プライベート・アドレス（通常はRFC 1918）が使用されます。

パブリック・ピアリング：インターネットを使用せずに、Oracle Cloud Infrastructure内のパブリック・サービスにアクセスできます（たとえば、Oracle Cloud Infrastructure Object Storage、Oracle Cloud InfrastructureコンソールおよびAPI、またはVCN内のパブリック・ロード・バランサ）。接続間の通信には、IPv4パブリックIPアドレスが使用されます。

仮想クラウド・ネットワーク（VCN）：Oracle Cloud Infrastructure内の仮想ネットワークです。お客様は、VCNを使用してインフラをクラウドに拡張できます。詳細については、[「VCNとサブネット」](#)を参照してください。

仮想回線：1つ以上の物理ネットワーク接続上を通る、隔離されたネットワーク・パスです。既存のネットワークの終端とOracle Cloud Infrastructureとの間に単一の論理接続を提供します。プライベート仮想回線ではプライベート・ピアリングがサポートされ、パブリック仮想回線ではパブリック・ピアリングがサポートされます（[「FastConnectのプライベートおよびパブリック・ピアリング」](#)を参照してください）。各仮想回線は、お客様とオラクル、およびネットワーク・プロバイダまたは交換プロバイダ（Oracle FastConnectパートナーを通じて接続している場合）の間で共有される情報によって構成されます。お客様は、複数のプライベート仮想回線を使用することもできます（たとえば、組織内の別の部門からのトラフィックを分離する場合など）。

インターネット・ゲートウェイ（IGW）：クラウド・ネットワークの終端をインターネットに接続するルーターです。お客様のVCNから発信され、VCNの外部のパブリックIPアドレスへと向かうトラフィックは、インターネット・ゲートウェイを通過することになります。

動的ルーティング・ゲートウェイ（DRG）：お客様のVCNにアタッチされる仮想エッジ・ルーターです。プライベート・ピアリングに必要です。DRGは、FastConnect経由か[IPSec VPN](#)経由かにかかわらず、お客様のVCNへと向かうプライベート・トラフィックの単一のエントリ・ポイントとなります。DRGを作成したら、それをVCNにアタッチし、VCNのルート表にDRG用のルートを追加して、トラフィック・フローを有効にする必要があります。

これらの概念については、[FastConnectのドキュメント](#)で説明されています。

接続オプション

このセクションでは、データベース移行のための最適な接続オプションについて説明します。速度、コスト、および時間に基づいてオプションを比較するには、「接続オプションの選択」を参照してください。

オプション1：クラウド交換による接続

交換プロバイダは、オンプレミスと交換プロバイダとの間に確立された同一の専用物理接続を介して、クラウド・プロバイダの大規模なエコシステムへの接続性を提供するプロバイダです。利用可能なプロバイダとしては、Megaport、Equinix、Digital Realtyなどがあります。

クラウド間のルーティングについては、次のオプションがあります。

- 交換プロバイダからの仮想ルーター・サービスを使用する（たとえば、Megaport Cloud Router（MCR））。
- 物理カスタマ・エッジ（CE）デバイスを、交換プロバイダとのコロケーションで配置する。

次の表は、仮想ルーター・サービスを使用する場合と、物理ルーターを交換プロバイダとのコロケーションで配置する場合の、メリットとデメリットを示したものです。

メリット	デメリット
仮想ルーター・サービスを使用する	
<ul style="list-style-type: none">• デプロイが簡単• 帯域幅をオンデマンドで利用できる• デプロイとメンテナンスのコスト効率が良い	<ul style="list-style-type: none">• ルーティングを変更する柔軟性が、クラウド交換プロバイダからのサポート範囲に限定される• パブリックIP通信を利用できない
専用の物理ルーターを使用する	
<ul style="list-style-type: none">• ルーティング機能を柔軟に管理できる• デプロイするハードウェアを選べる	<ul style="list-style-type: none">• デプロイ時間が長い• スケーリングに制限がある• ハードウェア・メンテナンスの手間がかかり、それに関連する金銭的成本も生じる

このホワイト・ペーパーでは、パートナー非依存のアプローチによる最適な接続オプションの提供を目的としていますが、今回は、Megaport Cloud Router（MCR）オプションについて詳しく説明することにします（デプロイが簡単なため）。

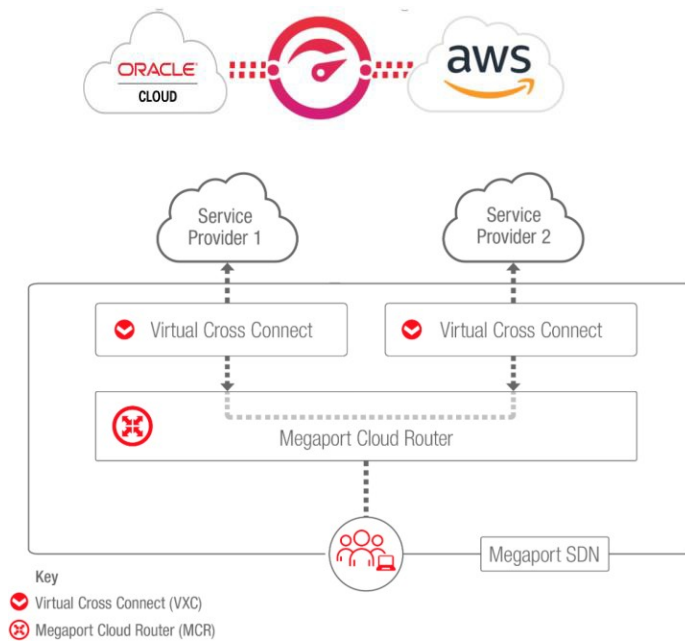
接続を設定するには、次のステップが必要となります。

1. Oracle Cloud Infrastructureコンソールを通じて、FastConnectをMegaportに接続します。
2. AWSコンソールを通じて、AWS Direct ConnectをMegaportに接続します。

3. MCRを作成します。

A. MCRからFastConnectへのVirtual Cross Connect（VXC）接続を作成します。

B. MCRからAWS Direct ConnectへのVXC接続を作成します。



Oracle Cloud Infrastructureコンソールを通じて、FastConnectをMegaportに接続する

この手順では、次の前提要件が満たされていることを前提とします。

- 設定しようとしているのは、プライベート・ピアリング接続である。
- [テナンシ](#)と[IAMポリシー](#)が、すでにOracle Cloud Infrastructureで設定されている。


Oracle Cloud Infrastructureコンソールで、次の手順を実行します。リンクをクリックすると、詳細な説明にアクセスできます。

1. [VCNを作成](#)します。
2. [DRGを作成](#)します。
3. [DRGをVCNにアタッチ](#)します。
4. [ルート・ルールを作成](#)して、ルート表を設定します。

5. 仮想回線を設定します。

- A. 「ネットワーキング」タブをクリックし、「FastConnect」を選択します。正しいコンパートメント内にあることを確認します。
- B. 「接続の作成」をクリックします。
- C. 「プロバイダを通じて接続」を選択し、「Megaport」を選択して、「続行」をクリックします。
- D. 「プライベート仮想回線」を選択し、DRGを選択します。
- E. BGP（Border Gateway Protocol）IPアドレスとASN情報を入力します。

6. 仮想回線の作成を終了します。OCIDとカスタムBGP IPアドレスをメモしてください。



PENDING PROVIDER

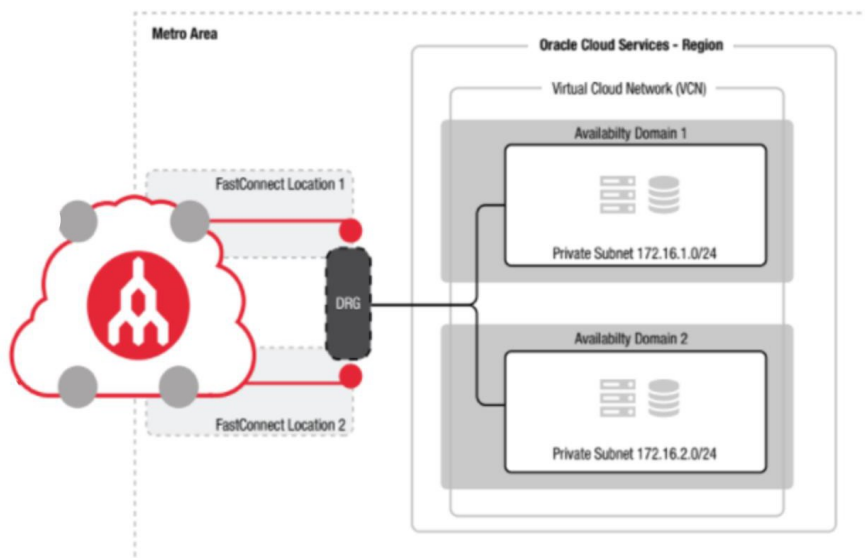
MyFastConnect

[Edit](#) [Terminate](#)

Provider Name: Megaport	Dynamic Routing Gateway: DRG1
Provider Service Name: VXC	Provisioned Bandwidth: 20 Gbps
Provider State: INACTIVE	Lifecycle State: PENDING PROVIDER
OCID: ...hmfxoq Show Copy	Created: Fri, 24 Mar 2017 00:30:48 GMT
Virtual Circuit Type: Private	

BGP Information:

Oracle BGP ASN: 31898	Oracle BGP IP Address: 10.0.5.19/31
Customer BGP ASN: 12345	Customer BGP IP Address: 10.0.5.18/31



AWSコンソールを通じて、AWS Direct ConnectをMegaportに接続する

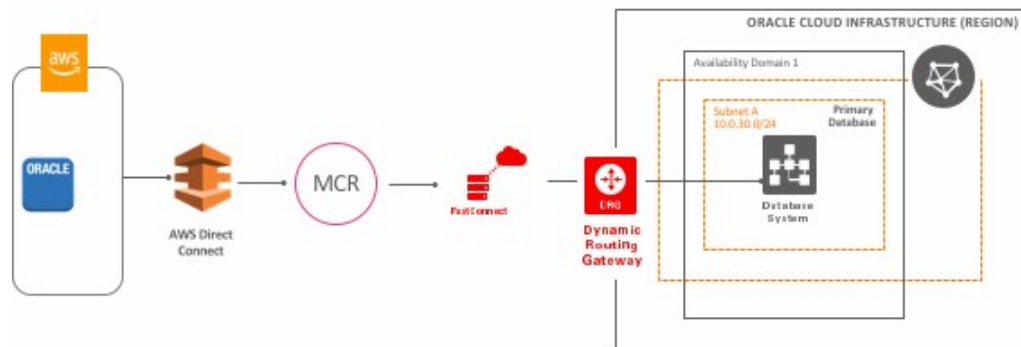
この手順では、次の前提要件が満たされていることを前提とします。

- Amazon Virtual Private Cloud (VPC) が、適切なCIDRを使用して（Oracle上のVCNと重複することなく）作成されている。
- ルート表とルート・タグが作成されていて、利用可能である。

次の手順は、仮想プライベート・ゲートウェイ、プライベート仮想インタフェース、およびDirect Connectゲートウェイを確立する方法を示したものです。

1. 仮想プライベート・ゲートウェイを作成します。
 - A. カスタマASNを選択し、プライベートASNを指定します。
 - B. 仮想プライベート・ゲートウェイをVPCにアタッチします。
2. 「AWS Direct Connect Connections」 ペインで、プライベート仮想インタフェースを作成します。
 - A. 仮想インタフェース・タイプとして、「**Private**」を選択します。
 - B. 適切な仮想インタフェース所有者を選択します。
 - C. 「**Connection To**」では、「**Virtual Private Gateway**」を選択します。
3. リモートのAWSリージョン内のVPCにアクセスするためのDirect Connectゲートウェイを作成します（他のリージョンのVPCへのアクセスが必要な場合に必要です）。
 - A. Direct Connectゲートウェイを作成します。
 - B. 仮想プライベート・ゲートウェイをDirect Connectゲートウェイにアタッチします。

仮想プライベート・ゲートウェイとDirect Connectゲートウェイの作成時に指定されたASNをメモしてください。リモート・リージョンのVPCに接続する必要がある場合は、Megaportポータルで対応するASNを指定する必要があります。



MCRとVXC接続を作成する

MegaPortポータルで、MCRと2つのVXC接続を作成します（1つはOracleへの接続、もう1つはMCRからAWSへの接続）。

MCRを作成する

1. MegaPortポータルにログインし、「**Create MCR**」を選択します。



2. 場所を選択し、レート制限を選択します。利用できるオプションは、100 Mbps、500 Mbps、および1 Gbps～5 Gbps（1 Gbpsきざみで指定）です。

FastConnectへのVXC接続を作成する

VXCを作成する前に、Oracle仮想回線ID（OCID）とOracle ASN（31898）を確認する必要があります。

1. MCRから、「**+Connection**」を選択し、「**Cloud**」を選択します。
2. クラウド・プロバイダのリストから、「**Oracle**」を選択します。
3. Oracle仮想回線ID（OCID）を貼り付けます。
4. 接続名を指定します。レート制限は、OCIDからの最大問合せ値に基づいて自動入力されます。

5. VCN IP CIDRとBGP詳細を指定します。次のスクリーンショットはサンプルです。

The screenshot shows the Oracle Cloud console configuration page for a VCN. At the top, there are two network endpoints: 'DL MCR Ashburn' (1 Gbps, Ashburn, USA) and 'OCI (us-phoenix) Primary (BMC)' (10 Gbps, Phoenix, USA). Below these, the 'IP Addresses' section shows a list with '192.168.100.129/30' selected. A red callout points to this field with the text: 'Enter the same IP address entered in the Oracle Console in the Customer IP address field'. The 'BGP Connections' section shows a table with one connection, 'BGP Connection 1', with details: Local ASN 65500, Peer ASN 31898, Local IP 192.168.100.129/30, Peer IP 192.168.100.130/30, and BGP Auth. A red callout points to the Peer IP field with the text: 'Provided in the Oracle Console by viewing the connection details of the FastConnect'. The interface also includes sections for 'Static Routes' and 'NAT IP Addresses', each with an 'Add New' button. At the bottom, there are buttons for 'Back To Dashboard', 'Back', and 'Next'.

6. プロセスを終了し、サービス・ステータスとBGPステータスを確認します。

MCRからFastConnectへのVXC接続を作成するための手順については、
<https://knowledgebase.megaport.com/megaport-cloud-router/mcr-oracle-bmc-fastconnect/>で説明されています。

MCRからAWS Direct ConnectへのVXC接続を作成する

1. MCRから、「+Connection」を選択し、「Cloud」を選択します。
2. クラウド・プロバイダのリストから、「AWS」を選択します。
3. 接続名を指定し、レート制限を選択します。
4. 「Connection Details for AWS Service」で、「Private」を選択します。
5. BGPピアリング情報を指定します。プライベートASNか、MegaportのデフォルトASNである133937を選択できます。
6. プロセスを終了し、VXCがMCRに正常にアタッチされたことを確認します。
7. AWS Direct Connectダッシュボードで、インバウンド仮想インタフェース要求をチェックします。
8. 「Accept Virtual Interface」をクリックします。
9. 先ほど作成した、仮想プライベート・ゲートウェイ（単一リージョン内での1対1のVPCマッピング）か、Direct Connectゲートウェイを選択します。

10. BGPステータスが「Pending Acceptance」から「Pending」になり、その後BGPが確立されると「Available」になるのを確認します。

11. VPCルートの伝播を有効にし、受信したルート表をチェックします。

A. 「**Routes Tables**」タブのVPCに移動します。

B. 「**Route Propagation**」タブで、「**Edit**」をクリックし、関連付けられた仮想プライベート・ゲートウェイの「**Propagate**」チェック・ボックスをオンにして、「**Save**」をクリックします。

MCRがルートの受信を開始します。

MCRからDirect Connectを経由してAWSへのVXC接続を作成するための手順については、<https://knowledgebase.megaport.com/megaport-cloud-router/mcr-aws-directconnect/>で説明されています。

オプション2：IPSec VPN経由の接続

IPSec VPNでは、データ・トラフィックを暗号化することで追加的なセキュリティが提供されます。VPNで達成可能な帯域幅は250 Mbpsに制限されるため、転送するデータの合計量や必要な転送レートによっては、複数のVPNトンネルが必要になる場合があります。

Oracle Cloud Infrastructureと他のクラウド・プロバイダの間にセキュアな接続を作成するための手順については、次のブログ記事で説明されています。

<https://blogs.oracle.com/cloud-infrastructure/creating-a-secure-connection-between-oracle-cloud-infrastructure-and-other-cloud-providers>

接続オプションの選択

以下に示す大まかな指針は、どの接続を選ぶかを定めるうえで参考になりそうなポイントをまとめたものです。これらのデータは全体としての判断材料を簡単にまとめたものです。最終的に最適と判断される接続オプションは、ケースごとに異なります。

速度

- FastConnectでは、1Gと10Gのポート速度が提供されます。
- Direct Connectでは、50M、100M、200M、300M、400M、500M、1G、および10Gのポート速度が提供されます。
- IPSec VPNの速度は、ほとんどの場合、500Mb/秒以下に制限されます。

コスト

- Oracle FastConnectでは、均一のポート時間料金が課金され、データ転送については一切課金はありません。詳細については、「[Oracle FastConnectの価格](#)」を参照してください。
- Oracle IPSec VPNサービスでは、インバウンド・データ転送の料金は課金されません。アウトバウンド・データ転送については、最大10 TBの転送まで無料です。10 TBの制限を超えた後は、少額の料金が課金されます。詳細については、「[Oracle IPSec VPNの価格](#)」を参照してください。
- Amazonの価格には、ポート料金とデータ転送料金があります。インバウンド・データは計測されませんが、アウトバウンド・データは計測され、課金されます。詳細については、「[Amazon Direct Connectの価格](#)」を参照してください。
- Megaportの価格は、MCRの作成時に選択したレート制限に基づいて決まります。利用可能なオプションは、100 Mbps、500 Mbps、および1、2、3、4、5 Gbpsです。課金レート（1ヶ月あたりの値）は、MCRのデプロイ先と接続するリージョンに基づいて、デプロイ時に表示されます。

時間

データ転送時間は、各ホップで選択した速度によって決まります。専用接続とIPSec VPNの比較で言うと、専用接続では、決定論的な時間枠が提供されます。これは、接続にプライベート・メディアが使用されることで、信頼性と一貫性が高まるためです。

次の表は、AWSからOracle Cloud Infrastructureへのデータ転送時間に対する帯域幅に基づいた、仮設的なコスト・シナリオを示したものです。

		データ (TB)			
		10	100	1,000	10,000
レート (Gb/秒)	1	22h13m12s	9d6h13m12s	92d14h13m12s	925d22h13m12s
	10	2h13m12s	22h13m12s	9d6h13m12s	92d14h13m12s
	100	13m12s	2h13m12s	22h13m12s	9d6h13m12s

100 Gb/秒の速度は、複数の10Gリンクをリンク集約することで達成できます。

AWSからOracle Cloud Infrastructure Databaseへの移行ガイド

次の表は、ニーズに合った移行オプションを選ぶうえで役立つガイドラインを示したものです。各移行オプションの詳細については、次のセクションで説明します。

ソース	方法	タイプ	オンライン/ オフライン	ゼロ・ダウ ンタイム?	データベース・ サイズ	注釈
RDS、 EC2	SQL Developer Data Pump : エク スポート/インポー ト	論理	オンライン	はい	5 TB以下	5 TBまでのデータベースに使用 できます。 ゼロ・ダウンタイムは、 Oracle GoldenGateと組み合 わせることで達成できます。
RDS、 EC2	SQL Developer : コピー	論理	オンライン	いいえ	200 MB以下	小規模のデータベースに使用 できます（オブジェクト数を 問わず）。
RDS、 EC2	SQL Developer : SQL*Loader	論理	オンライン	いいえ	10 GB以下	小/中規模のデータベースに 使用できます（オブジェクト 数に制限があります）。 この方法では、パッケージ、 プロシージャ、ファンクショ ンなどのオブジェクトを移行 することはできません。
RDS、 EC2	SQL Developer : マテリアライズ ド・ビュー	論理	オンライン	はい	500 GB以下	小規模のデータベースに使用 できます（オブジェクト数に 制限があります）。
RDS、 EC2	AWS DMS	論理	オンライン	はい	5 TB以下	データベースのサイズが5 TB 以下で、ゼロ・ダウンタイム の移行が必要な場合に使用で きます。
EC2	Data Pump : 従来 型の表領域のエク スポート/インポー ト	論理	オンライン	はい	20 TB以下	データベースのサイズが20 TB以下で、次の条件に該当 する場合に使用できます。 <ul style="list-style-type: none"> データベース全体、また はスキーマのサブセット を移動したい。 ソースからターゲットへ ネットワーク経由で転送 されるデータの量を最小 限に抑えたい。 ターゲットのデータベー スの物理レイアウトを変 更したい。 文字セットの変換を行 いたい。

ソース	方法	タイプ	オンライン/ オフライン	ゼロ・ダウ ンタイム?	データベース・ サイズ	注釈
						Oracle GoldenGateと組み合 わせて、ゼロ・ダウンタイム の移行を実行できます。
EC2	Data Pump : フ ル・トランスポー タブル	物理	オフライン	いいえ	すべてのサイズ	データベース内のすべての表 領域を転送したい場合に使用 できます。
EC2	Data Pump : トラ ンスポータブル表 領域	物理	オフライン	いいえ	すべてのサイズ	ソース・データベースから ターゲットに表領域のサブ セットだけを転送したい場合 に使用できます。
EC2	リモート・クロ ン : PDB	物理	オンライン	いいえ	5 TB以下	この方法は、次の条件に該当 する場合に使用できます。 <ul style="list-style-type: none"> ソースで12cのマルチテ ナント・データベースを 使用している。 データベースが開発用ま たはテスト用である（こ の方法では、ターゲット でポイントインタイム・ リカバリが実行されるた め、ソース・データベー スとのトランザクション の一貫性は完全には保証 されません）。
EC2	リモート・クロ ン : 非CDB	物理	オンライン	いいえ	5 TB以下	この方法は、次の条件に該 当する場合に使用できま す。 <ul style="list-style-type: none"> ソースに12cの非CDB データベース（コンテ ナ・データベース）があ り、ターゲットでマルチ テナントを使用するよう に変換したい。 データベースが開発用ま たはテスト用である（こ の方法では、ターゲット でポイントインタイム・ リカバリが実行されるた め、ソース・データベー スとのトランザクション の一貫性は完全には保証 されません）。

ソース	方法	タイプ	オンライン/ オフライン	ゼロ・ダウ ンタイム?	データベース・ サイズ	注釈
EC2	RMAN : トランスポート ブルPDB	物理	オンライン	いいえ	すべてのサイズ	この方法は、ソースで12cの プラグブル・データベース (PDB) を使用して、 PDBを移行する必要がある、 かつダウンタイムを最小限に 抑えたい場合に使用できます (ダウンタイムは、前回の増 分バックアップ用にかかった 時間にまで抑えられます)。
EC2	RMAN : トランスポート ブル・バックアッ プ・セット	物理	オフライン	いいえ	すべてのサイズ	この方法は、ソースで12cの PDBを使用して、特定の 表領域セットだけを移行する 必要がある、かつダウンタイ ムを最小限に抑えたい場合に 使用できます (ダウンタイム は、前回の増分バックアップ 用にかかった時間にまで抑え られます)。
EC2	RMAN : アクティ ブ・データベース からの複製	物理	オンライン	いいえ	5 TB以下	この方法は、次の条件に該当 する場合に使用できます。 <ul style="list-style-type: none"> • 中間の一時ストレージを 必要としない、1ステッ プの方法を希望してい る。 • データベースが開発用ま たはテスト用である (こ の方法では、ターゲット でポイントインタイム・ リカバリが実行されるた め、ソース・データベー スとのトランザクション の一貫性は完全には保証 されません)。
EC2	PDBのアンプラグ とプラグ	物理	オフライン	いいえ	すべてのサイズ	この方法は、次の条件に該当 する場合に使用できます。 <ul style="list-style-type: none"> • ソースで12cとマルチテ ナント・データベースを 使用している。 • ビジネス要件上、フル移 行のためにデータベース のダウンタイムが発生す ることを許容できる。

ソース	方法	タイプ	オンライン/ オフライン	ゼロ・ダウ ンタイム?	データベース・ サイズ	注釈
EC2	非CDBのアンプラグとプラグ	物理	オフライン	いいえ	すべてのサイズ	この方法は、次の条件に該当する場合に使用できます。 <ul style="list-style-type: none"> ソースで12cと非CDBデータベースを使用している。 ビジネス要件上、フル移行のためにデータベースのダウンタイムが発生することを許容できる。
EC2	Data Guardを使用した移行	物理	オンライン	はい	すべてのサイズ	この方法は、Oracle GoldenGateなどの追加的ツールを使用せずに、ゼロ・ダウンタイムの移行を達成したい場合に使用できます。
EC2	RMANを使用した移行	物理	オンライン	いいえ	すべてのサイズ	この方法は、きわめて大規模なデータベースを最小限のダウンタイムで移行するために使用できます。ダウンタイムは、最後の増分バックアップを作成し、それをターゲットに転送し、そのバックアップを使用してリストアとリカバリを実行するのにかかる時間となります。

Amazon RDSからOracle Cloud Infrastructureへの移行オプション

Amazon Relational Database Service (RDS) では、データベース・サーバーへのオペレーティング・システム・ログインが提供されないため、データベース移行オプションは、SQL*Net接続経由で達成できるものに制限されます。できるだけシンプルなエンドユーザー・エクスペリエンスを提供するため、次のオプションの最初の4つでは、SQL Developerを使用して移行が実行されます。これらのオプションは、SQL Developerがすでにインストールされていることを前提としたものです。

- SQL Developer Data Pump : エクスポート/インポート
- SQL Developer : コピー
- SQL Developer : SQL*Loader
- SQL Developer : マテリアライズド・ビュー
- AWS DMS

注：このセクションにある手順はすべて、AWS EC2インスタンスからのデータベース移行にも使用できます。

RDS移行オプション1：SQL Developer Data Pump：エクスポート/インポート

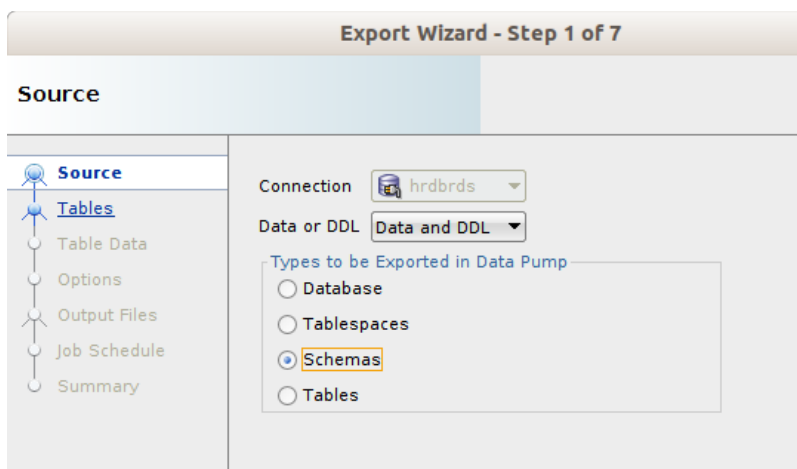
この移行方法では、Oracle Data Pumpユーティリティを使用してソース・データベースからデータが抽出されます。このデータベース移行方法は、非常にスケーラブルで高パフォーマンスです。データは、プラットフォーム非依存の形式のエクスポート・ファイルに抽出されます。これらのファイルは、DBMS_FILE_TRANSFERというデータベース・ユーティリティを使用してターゲットに転送され、その後、インポート・プロセスが実行されて、データがターゲット・データベースにロードされます。

利点：この方法では、データベースの論理移行に対応でき、並列処理が可能で、再起動の機能も提供されます。

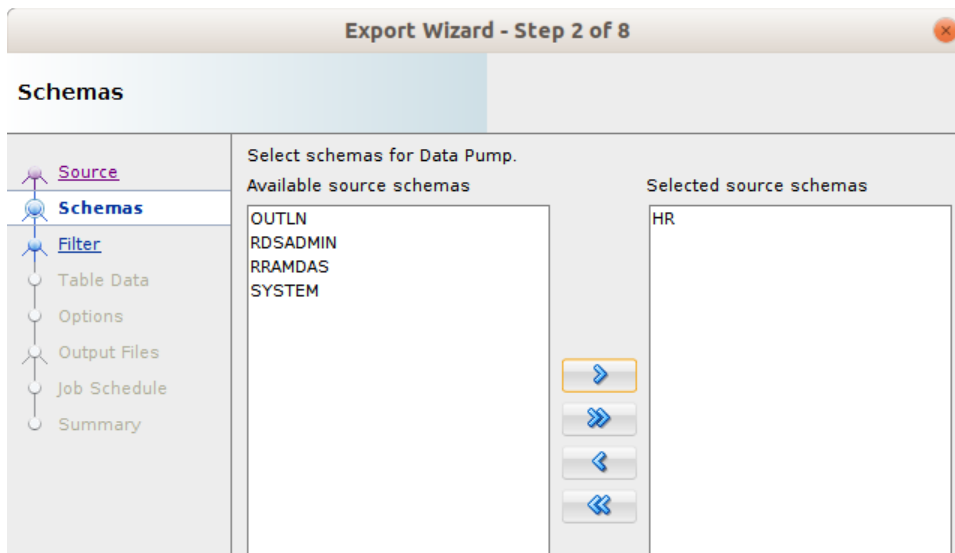
オンライン/オフライン：この方法は、オンラインで実行できます。トランザクションは、Oracle GoldenGateを使用して、キャプチャと適用について同期できます。

データベース・サイズ：5 TBまでのデータベースに使用できます。この方法は、RDSインスタンス上のファイル・システム（Data Pumpのダンプ・ファイルが作成される場所）に十分な空き容量があれば、比較的大規模なデータベースに使用できます。

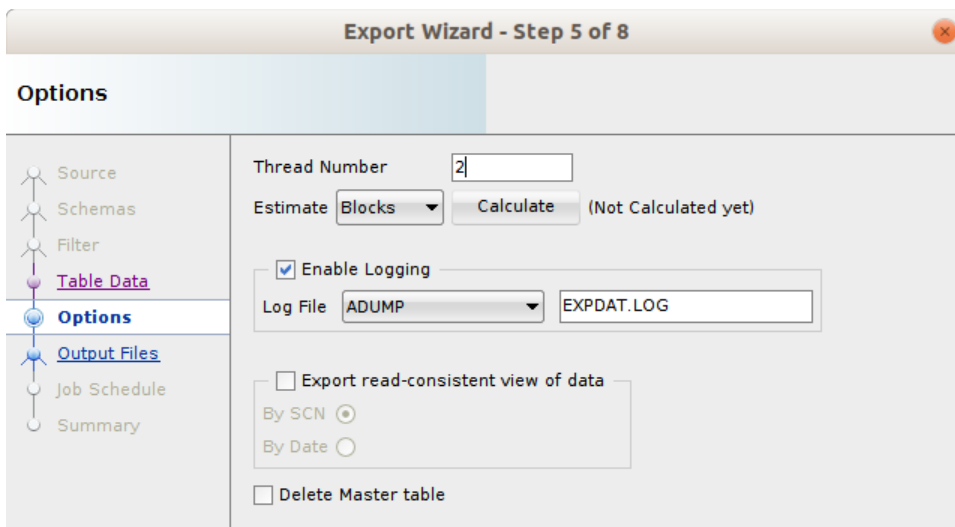
1. SQL DeveloperのUIで、「表示」>「DBA」に移動します。
2. データベース・インスタンスへの接続を展開し、「Data Pump」を右クリックして、「データ・ポンプ・エクスポートウィザード」を選択します。
3. ウィザードの「ソース」ページで、エクスポート・タイプを選択します。



4. 「スキーマ」ページで、エクスポートするスキーマを選択します。

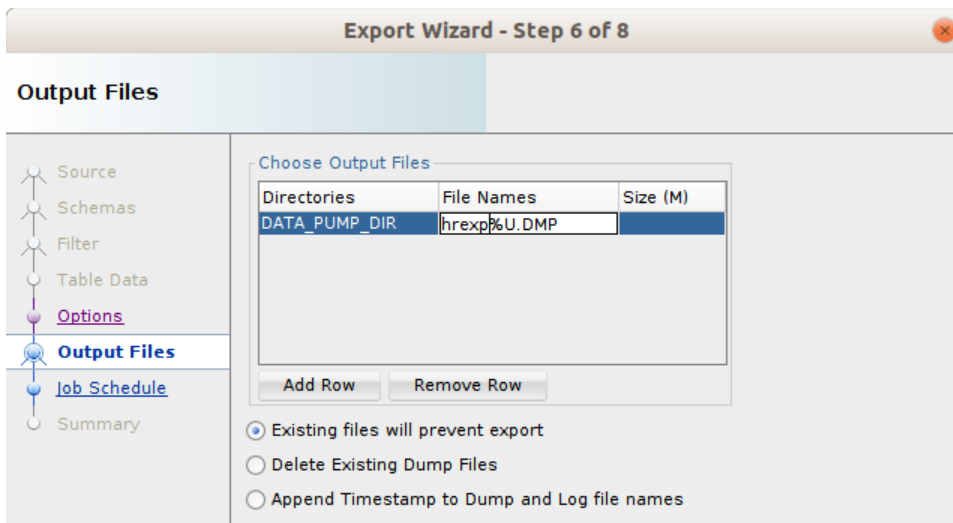


5. 次の2つのページ（「フィルタ」と「表データ」）は、何も変更せずに進みます。
6. 「オプション」ページで、「スレッド数」の値を指定してエクスポートの並列処理を選択します。



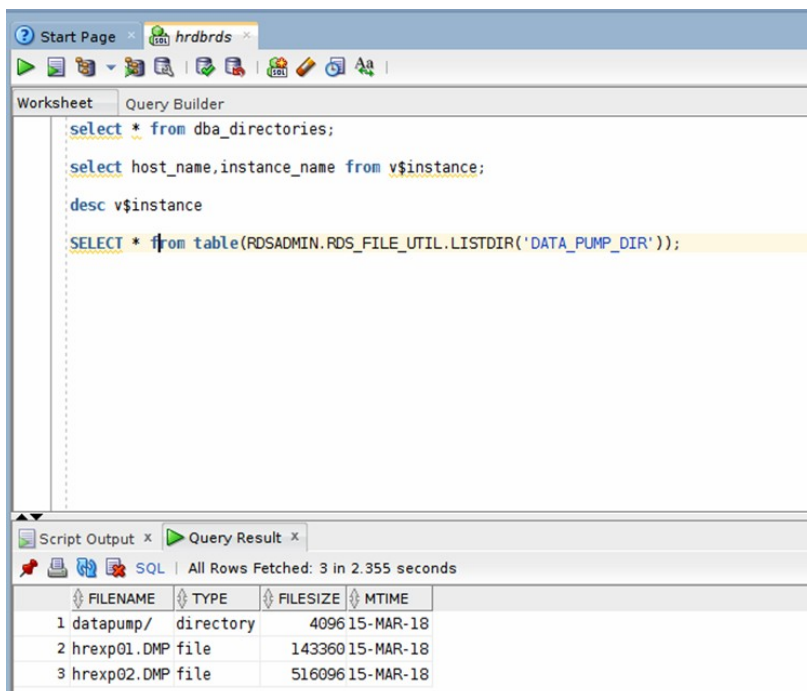
並列処理を2にした場合は、エクスポート・プロセス中、2つの並列プロセスがデータベースからデータを読み取り、それらを出カファイルに書き込みます。RDSインスタンスに、これらのプロセスを実行するための十分なCPU、メモリー、およびI/Oリソースがあることを確認してください。

7. 「出力ファイル」ページで、出力ファイル名の形式を選択します。



DATA_PUMP_DIRは、Amazon RDSインスタンスの/rdsdbdata/datapumpディレクトリを指します。

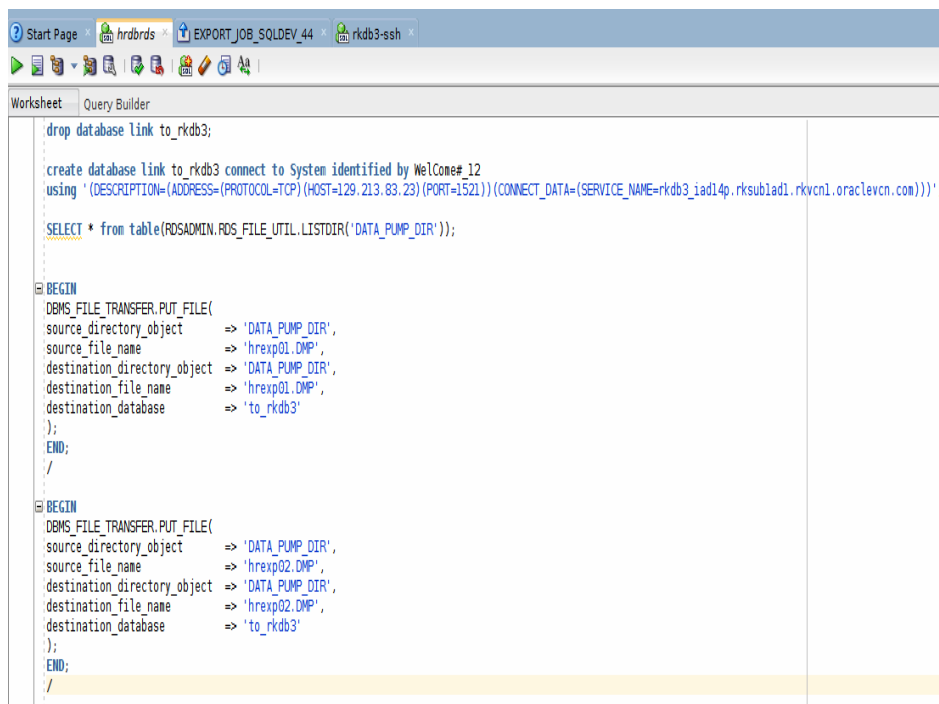
8. 次の2つ画面をそのまま進み、エクスポート・ジョブを送信します。ジョブが完了するまで、Data Pumpエクスポート・ジョブをモニタリングします。
9. RDS_FILE_UTIL.LISTDIRパッケージを実行して、DATA_PUMP_DIR内のファイルをリストします。



10. Amazon RDSのソース・データベースからOracle Cloud Infrastructureのターゲット・データベースへのデータベース・リンクを作成します。次に例を示します。

```
create database link to_tgtddb3 connect to System identified by WelCome#_12
using
' (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=129.213.83.23) (PORT=1521)) (CONNE
CT DATA=(SERVICE NAME=tgtddb3 iad14p.rsublad1.rvcn1.oraclevcn.com))) ';
```

11. dbms_file_transfer.put_fileを実行して、DATA_PUMP_DIRからファイルを読み取り、それらをターゲット・データベース・ホストに転送します。ファイル転送には、前の手順で作成されたデータベース・リンクが使用されます。



```
drop database link to_rkdb3;

create database link to_rkdb3 connect to System identified by WelCome#_12
using ' (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=129.213.83.23) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=rkdb3_iad14p.rksublad1.rkvcn1.oraclevcn.com)))';

SELECT * from table(ROSADMIN.ROS_FILE_UTIL.LISTDIR('DATA_PUMP_DIR'));

BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object => 'DATA_PUMP_DIR',
    source_file_name         => 'hrexp01.DMP',
    destination_directory_object => 'DATA_PUMP_DIR',
    destination_file_name     => 'hrexp01.DMP',
    destination_database      => 'to_rkdb3'
  );
END;
/

BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object => 'DATA_PUMP_DIR',
    source_file_name         => 'hrexp02.DMP',
    destination_directory_object => 'DATA_PUMP_DIR',
    destination_file_name     => 'hrexp02.DMP',
    destination_database      => 'to_rkdb3'
  );
END;
/
```

12. Data Pumpファイルをターゲット・データベースにインポートします。

Oracle Cloud Infrastructure上のDatabaseサービス・インスタンスにオペレーティングシステムレベルでアクセスできる場合は、Databaseサービス・インスタンスにログインし、Data Pumpインポート・プロセスを実行できます

```
[oracle@rkcomp1]$ impdp system/WelCome#_12@tgtddb3 schemas=HR
directory=DATA_PUMP_DIR dumpfile=DATA_PUMP_DIR:hrexp01.DMP,hrexp02.DMP
logfile=DATA_PUMP_DIR:hrimp01.log
```

13. データが正常にインポートされたら、オブジェクトとレコードのカウントをチェックして、すべて正しくインポートされたことを確認します。

RDS移行オプション2：SQL Developer：コピー

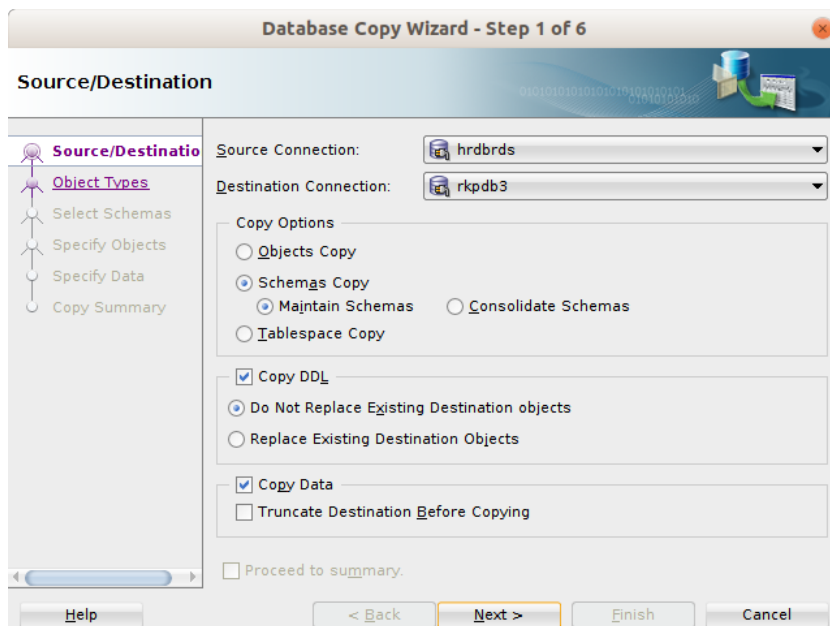
この方法では、SQL Developerから利用できるコピー・ツールを使用して、ソース・データベースからターゲット・データベースへデータのコンテンツをコピーします。このスピーディな方法は、データ量の少ない小規模なスキーマをコピーするのに適しています。

利点：この方法では、ウィザードベースの操作で、ソース・データベースのスキーマからすべてのオブジェクトをすばやく移行できます。

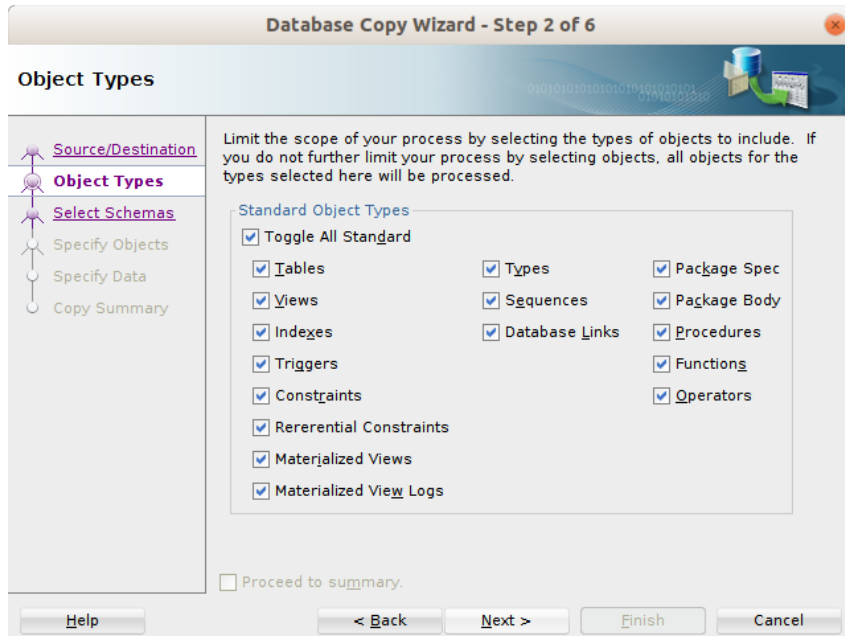
オンライン/オフライン：この方法は、オンラインで実行できます。ゼロダウンタイムのアップグレードを実行する機能は提供されません。

データベース・サイズ：200 MBまでのデータベースに使用できます。

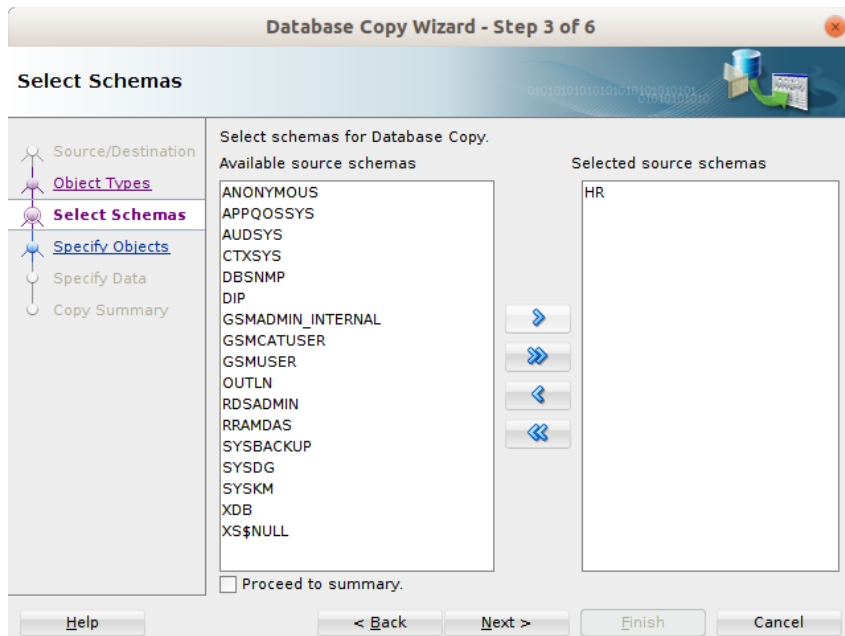
1. SQL DeveloperのUIで、「ツール」>「データベース・コピー」に移動し、データベース・コピー・ウィザードを開きます。
2. 「ソース/宛先」ページで、ソースと宛先のデータベース（およびスキーマ）を選択します。「コピー・オプション」領域で、スキーマ全体をコピーするように選択し、「DDLのコピー」および「データのコピー」チェック・ボックスをオンにします。



3. 「オブジェクト・タイプ」ページで、コピーするスキーマ・オブジェクト・タイプを選択します。

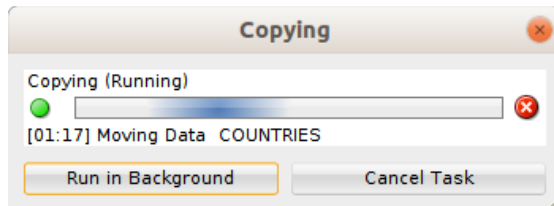


4. 「スキーマの選択」ページで、コピーするスキーマを選択します。



5. 続く3つのページはそのまま進み、ウィザードを完了してコピーを開始します。

コピーの進行中は、次のようなダイアログ・ボックスが表示されます。



コピー・プロセスがエラーなく完了すると、すべてのオブジェクトを含むスキーマとそれらのデータが、ターゲット・データベースに配置されます。

RDS移行オプション3：SQL Developer：SQL*Loader

SQL*Loaderは、外部ファイルからOracleデータベース表にデータをロードするOracleユーティリティです。これは強力なデータ・パース・エンジンで、データ・ファイル内のデータの形式についてほとんど制限がありません。データは通常、ローダー・ファイルに保存され、データのフォーマット方法をSQL*Loaderに伝える制御情報は、コントロール・ファイル内に保持されます。

SQL Developerを使用してAmazon RDS内のソース・データベース・スキーマからDDLとデータを抽出した後、SQL*Loaderを使用して、それらをOracle Cloud Infrastructure内にあるデータベースのデータベース表にロードできます。なお、このプロセスを使用してロードできるのは表だけであるという点に注意してください。その他のスキーマ・オブジェクト（パッケージ、プロシージャ、ファンクション、トリガーなど）は、ターゲット・データベース内に手動で作成する必要があります。

このプロセスを開始する前に、スキーマ（たとえば、HR）と、それに付随する表領域（たとえば、HRTS）が、ターゲット・データベースに作成されていることを確認してください。

利点：この方法では、ウィザードベースの操作で表とデータをすばやく移行できます。

オンライン/オフライン：この方法は、オンラインで実行できます。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：10 GBまでのデータベースに使用できます。

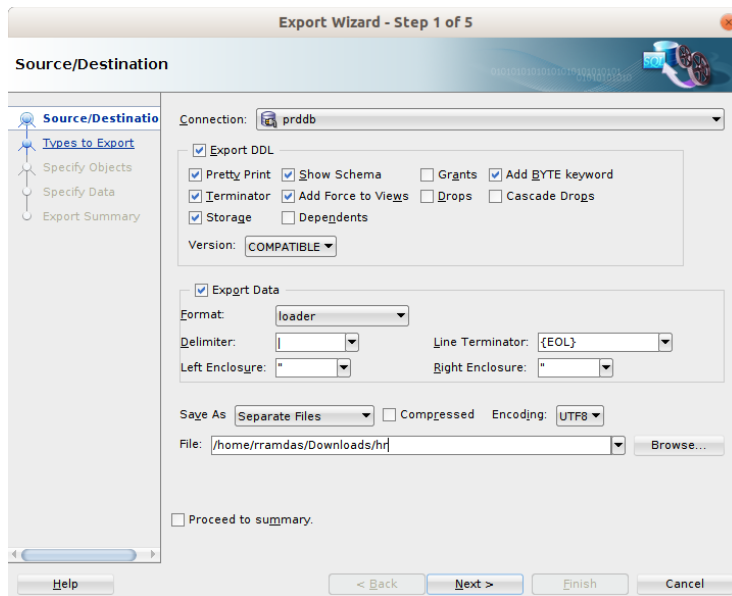
1. SQL DeveloperのUIで、「ツール」>「データベース・コピー」に移動し、データベース・コピー・ウィザードを開きます。
2. 「ソース/宛先」ページで、次の手順を実行します。

A. SQL*Loader経由で転送したいスキーマ（たとえば、HR）に接続するための接続を選択

します。

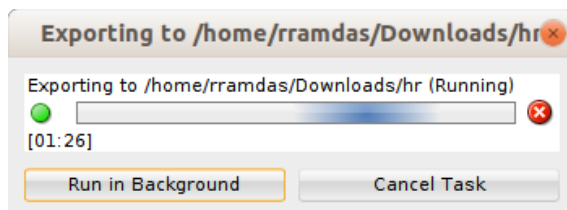
B. 「データのエクスポート」セクションで、フォーマットとして「loader」を選択します。

C. SQL*Loaderファイルを作成するディレクトリを選択します。



3. 残りの4ページはそのまま進み、要求を送信して、ファイル作成を開始します。

エクスポートの実行中は、スキーマ・エクスポートの進捗状況バーが表示されます。



エクスポートが完了すると、作成されたファイルの名前を示す画面が表示されます。エクスポートでは、次のタイプのファイルが作成されます。

ファイル名	説明
<object_name>.sql	オブジェクト作成DDL
<object_name>_SEQ.sql	シーケンス作成DDL
<object_name>_CONSTRAINT.sql	制約作成DDL
<object_name>_REFCONSTRAINT.sql	外部キー制約作成DDL
<object_name>_UK.sql	一意索引作成DDL

ファイル名	説明
<object_name>_PK.sql	プライマリ・キーの一意索引作成DDL
<object_name>_IX.sql	索引作成DDL
<object_name>_DATA_TABLE.ct1	オブジェクトのSQL*Loader制御ファイル
<object_name>_DATA_TABLE.ldr	オブジェクトのSQL*Loaderデータ・ファイル
Generated-<timestamp>.sql	オブジェクトをロードするためのコマンドを含んだファイル

4. ターゲット・データベース内のオブジェクトを作成します。
 - A. SQL*Plusを起動し、オブジェクトのロード先となる、Oracle Cloud Infrastructure内のターゲット・データベースとスキーマにログインします。
 - B. SQLスクリプトを実行して、ロードするオブジェクトを作成します。これを簡単に行うには、SQL*PlusからGenerated-*<timestamp>*.sql ファイルを実行します。これにより、データベース内にすべてのオブジェクトが作成されます。

ヒント: 後で（データが完全にロードされた後で）オブジェクトの索引を作成し、ロードのパフォーマンスを改善することを検討してください。

[illegible]

5. アンロード・ディレクトリ内の`.ctl`ファイルを使用して、オブジェクトを一度に1つつターゲット・データベースへロードします（`sqlldr`ユーティリティを使用します）。

```
rrandas@rrandas-lap18:~/Downloads/hr$ sqlldr userid=hr/hr@129.213.28.50:1521/rkbnpdb2.rksubiad1.rkvcn1.oraclevcn.com control=REGIONS_DATA_TABLE.ctl log=REGIONS_DATA_TABLE.log

SQL*Loader: Release 12.2.0.1.0 - Production on Mon Mar 26 13:16:32 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Path used:          Conventional
Commit point reached - logical record count 3
Commit point reached - logical record count 4

Table "HR"."REGIONS":
  4 Rows successfully loaded.

Check the log file:
  REGIONS_DATA_TABLE.log
For more information about the load.
rrandas@rrandas-lap18:~/Downloads/hr$ sqlldr userid=hr/hr@129.213.28.50:1521/rkbnpdb2.rksubiad1.rkvcn1.oraclevcn.com control=COUNTRIES_DATA_TABLE.ctl log=COUNTRIES_DATA_TABLE.log

SQL*Loader: Release 12.2.0.1.0 - Production on Mon Mar 26 13:16:56 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Path used:          Conventional
Commit point reached - logical record count 24
Commit point reached - logical record count 25

Table "HR"."COUNTRIES":
  25 Rows successfully loaded.

Check the log file:
  COUNTRIES_DATA_TABLE.log
For more information about the load.
rrandas@rrandas-lap18:~/Downloads/hr$
```

RDS移行オプション4：SQL Developer：マテリアライズド・ビュー

Oracleのマテリアライズド・ビューは、問合せの結果を格納するデータベース・オブジェクトです。この機能を使用すると、ターゲット・データベース内に、（定期的に更新される）ソース・データベース表のコピーを作成できます。ターゲット・データベース内に大きな表の初期コピーを作成した後に、マテリアライズド・ビューのリフレッシュを使用することで、コピーを最新に維持できます。これにより、ターゲット・データベースへのゼロダウンタイムのカットオーバーを実現することができます。この移行方法は、移行する表の数が少ない場合に適しています。

利点：この方法では、GUIを使用してマテリアライズド・ビューを作成できます。

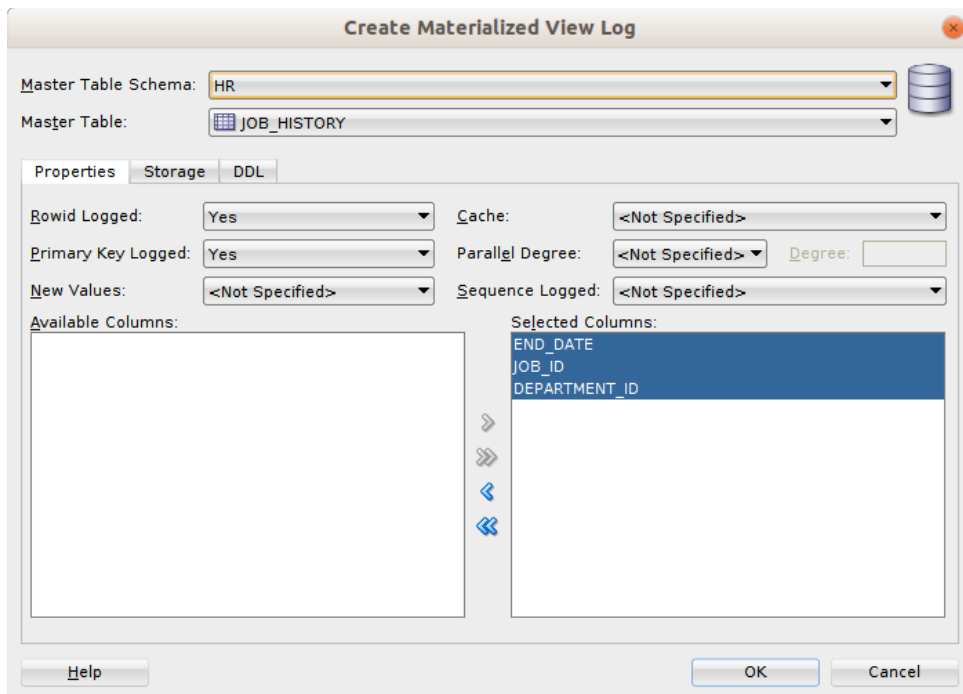
オンライン/オフライン：この方法は、オンラインで実行できます。ゼロ・ダウンタイムの移行が可能です。

データベース・サイズ：500 GBまでのデータベースに使用できます。

1. ソース表にマテリアライズド・ビュー・ログを作成します。

Oracleでは、マテリアライズド・ビュー・ログ内の表データに対するDML変更を記述する行が保存されます。これにより、増分的変更をターゲット・データベースにレプリケートし、ターゲット表に増分的に適用することが可能となっています。高速リフレッシュ（フル・リフレッシュではなく）を使用してリフレッシュできるマテリアライズド・ビューを設定するには、まずソース表にマテリアライズド・ビュー・ログを作成する必要があります。

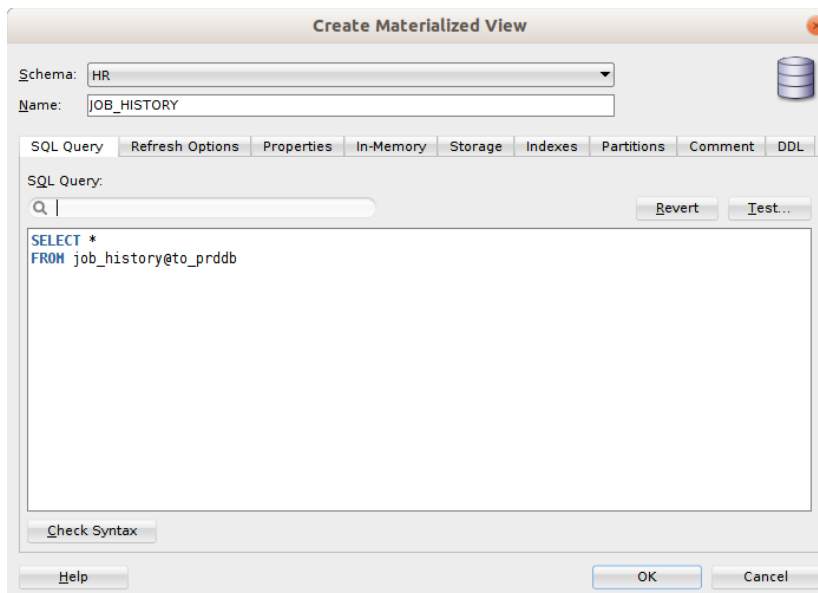
- A. SQL DeveloperのUIで、「接続」ペインを開きます。
- B. ソース・データベースの接続を展開し、「マテリアライズド・ビュー・ログ」を右クリックして、「新しいマテリアライズド・ビュー・ログ」を選択します。
- C. 「マテリアライズド・ビュー・ログの作成」ダイアログ・ボックスで、マテリアライズド・ビュー・ログを作成する表を選択し、すべての列を選択して、「OK」をクリックします。



2. ソース・データベースからターゲット・データベースへのデータベース・リンクを作成して、マテリアライズド・ビューをソース表に基づいて作成できるようにします。

```
create database link to_prddb connect to hr identified by hr
using ' (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=srcdb1.chyqlovvrniy.us-
east-2.rds.amazonaws.com) (PORT=1521)) (CONNECT_DATA= (SERVICE_NAME=ORCL))) ';
```

3. ターゲット・データベースにマテリアライズド・ビューを作成します。



4. マテリアライズド・ビューが作成されたら、それを次のようにリフレッシュします。

「接続」ペインでデータベースを展開し、「マテリアライズド・ビュー」を展開した後、マテリアライズド・ビュー（たとえば、JOB_HISTORY）を右クリックして、「リフレッシュ」>「今すぐ強制リフレッシュ」>「適用」を選択します。

RDS移行オプション5：AWS DMSを使用した移行

AWSによって提供されているDatabase Migration Service（DMS）は、AWSにあるOracleデータベースを、Oracle Cloud Infrastructure Databaseサービス内のOracleデータベースへと移行するために使用できます。AWS DMSでは、ソース・データベースのフル・スキーマをターゲット・データベースにコピーできます。また、ソース・データベースでのすべての変更をターゲットにレプリケートして、Oracle Cloud Infrastructure内のデータベースへとカット・オーバーするときまで、表やデータをソースと同期しておくことができます。

利点：この方法では、既存の移行サービスを使用してデータベースを移行できます。

オンライン/オフライン：この方法は、オンラインで実行できます。ゼロ・ダウンタイムの移行が可能です。

データベース・サイズ：5 TBまでのデータベースに使用できます。

この方法を使用するには、次の前提要件を満たす必要があります。

- AWS RDSデータベース・インスタンスがすでに作成されていて、ログイン資格情報がわかっている。
- Oracle Cloud Infrastructure内にDatabaseサービス・インスタンスがすでに作成されていて、ログイン資格情報がわかっている。
- AWS RDSのデータベース内にある（たとえば）HRスキーマのコンテンツを、Oracle Cloud InfrastructureのDatabaseサービス・インスタンス内のHRスキーマに移行したいと考えている。
- HRスキーマ（空のスキーマ）と、デフォルトの表領域HRTSを事前に作成済である。

次の手順を実行します。

1. レプリケーション・インスタンスを作成します。

レプリケーション・インスタンスは、ソース・データベース・インスタンスからターゲット・データベース・インスタンスへのレプリケーションを調整するためのものです。レプリケーション・インスタンスは、レプリケーション・ソフトウェアがインストールされたEC2インスタンスです。このインスタンスは、ソース・データベースへの接続、データの読み取り、ターゲット・データ・ストアによって消費されるデータのフォーマット、およびターゲット・データベース・インスタンスへのデータのロードに必要なメカニズムを提供します。

- A. AWS Management Consoleにログインします。
- B. 「**Services**」 ページの「**Database**」で、「**DMS**」をクリックします。
- C. DMSコンソールのメニュー・ペインで、「**Replication instances**」をクリックし、「**Create replication instance**」をクリックします。
- D. 「**Create replication instance**」ダイアログ・ボックスで、名前、説明、およびインスタンス・シェイプを入力します。大きなデータベースを転送する場合は、大きめのインスタンスを選択してください。その後、「**Create replication instance**」をクリックします。

Create replication instance

A replication instance initiates the connection between the source and target databases, transfers the data, and caches any changes that occur on the source database during the initial data load. Use the fields below to configure the parameters of your new replication instance including network and security information, encryption details, and performance characteristics. We suggest you shut down the replication instance once your migration is complete to prevent further usage charges.

Name*	ocirep1	?
Description*	replication	?
Instance class*	dms.t2.small	?
Replication engine version*	2.4.1	?
VPC*	vpc-699d1201	?
Multi-AZ	No	?
Publicly accessible	<input checked="" type="checkbox"/>	?

► Advanced

2. インスタンスが作成されたら、レプリケーション・エンドポイントを作成します。
 - A. DMSコンソールで、「Endpoints」をクリックし、「Create endpoint」をクリックします。
 - B. AWS RDS上のソース・インスタンスに接続するためのソース・エンドポイントを作成します。RDS上のソース・インスタンスのソース・データベース・エンジン、ホスト、ポート、およびデータベースSIDを入力します。その後、「Create endpoint」をクリックします。

Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

Endpoint type*	<input checked="" type="radio"/> Source <input type="radio"/> Target	?
Endpoint identifier*	oci-rds	?
Source engine*	oracle	?
Server name*	rkdb1.chyq1ovvmiy.us-east-2.rds.amazonaws.com	
Port*	1521	?
SSL mode*	- Select One -	?
User name*	admin	?
Password*	*****	?

SID*

▶ Advanced

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC*

Replication instance* ⓘ

☒ Refresh schemas after successful connection test ⓘ

- C. Oracle Cloud Infrastructure上のOracleデータベースに接続するためのターゲット・エンドポイントを作成します。転送するスキーマ名（たとえば、HR）を指定し、そのスキーマとベースの表領域が事前に作成済であることを確認します。

Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

Endpoint type* ☐ Source ☒ Target ⓘ

Endpoint identifier* ⓘ

Target engine* ⓘ

Server name*

Port* ⓘ

SSL mode* ⓘ

User name* ⓘ

Password* ⓘ

Port* 1521 ⓘ

SSL mode* - Select One - ⓘ

User name* HR ⓘ

Password* ⓘ

▶ Advanced

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC* vpc-699d1201 ▼

Replication instance* ocirep1 - vpc-699d1201 ⓘ

Run test

3. データ転送タスクを作成します。

A. DMSコンソールで、「Tasks」をクリックし、「Create task」をクリックします。

Create task

A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target

Task name* orcl-to-prddb ⓘ

Replication instance* ocirep1 - vpc-699d1201 ▼ ⓘ

Source endpoint* orcl-rds ▼ ⓘ

Target endpoint* orcl-oci ▼ ⓘ

Migration type* Migrate existing data ▼ ⓘ

Start task on create ☒

▼ Task Settings

Target table preparation mode* ☐ Do nothing ⓘ
☒ Drop tables on target
☐ Truncate

Include LOB columns in replication* ☐ Don't include LOB columns ⓘ

- B. 「Tasks」セクションに移動し、レプリケーション用のソース・データベースからHRスキーマを選択します。

Table mappings

Guided JSON

Selection rules ⓘ

At least one selection rule with an include action is required. Once you have one or more selection rules, you can add transformation rules.

Where ⓘ

Schema name is

Table name is like

Use % as a wildcard.

Action ⓘ

Filter ⓘ

[Add column filter](#)

[Add selection rule](#)

- C. 詳細を入力したら、「Create task」をクリックします。
- D. タスクが正常に作成されたら、「Start/Resume」をクリックしてデータ移行タスクを実行します。

以上が、AWS RDSからOracle Cloud Infrastructure Databaseへのデータベース移行に使用できる5つの移行方法です。次のセクションでは、AWS EC2インスタンスにデプロイされたOracleデータベースの移行に使用できる移行方法について説明します。

Amazon EC2からOracle Cloud Infrastructureへの移行オプション

AWSでは、Oracle DatabaseソフトウェアとOracleデータベースをAmazon EC2インスタンス上にインストールし、構成する機能が提供されています。これを実行するには、必要となるすべてのパッケージ、カーネル・パラメータ、およびOS構成をインストールして構成するとともに、Oracleソフトウェアをインストールして構成する必要があります。このオプションは、AWS RDSにOracle Database環境を設定する場合よりも、優れた柔軟性を提供するものです。ただし、このオプションを使用した場合、環境のバックアップやパッチ適用について、データベース・サービス・ツールのメリットを得ることはできません。

Oracleデータベースを実行しているホストを完全に制御できるので、Amazon EC2環境からOracle Cloud InfrastructureのDatabaseサービスに移行する際には、より多くの選択肢からオプションを選択することができます。このタイプのデプロイでは、Data PumpやRMANなどのOracleユーティリティを使用して、AWSからOracle Cloud Infrastructureに環境を移動することができます。

EC2移行オプション1：Data Pump：従来型の表領域のエクスポート/インポート

Data Pumpは、1つのデータベースから別のデータベースにデータを高速転送できる、Oracleユーティリティです。Data Pumpを使用すると、Amazon EC2内にあるソース・データベースから、Oracle Cloud InfrastructureのDatabaseサービス内で作成されたデータベースへと、スキーマ全体（または複数のスキーマ）を移動することができます。

この方法は、先に説明した「RDS移行オプション1：SQL Developer Data Pump：エクスポート/インポート」の方法と似ていますが、次の点が異なります。

- expdpユーティリティを使用してエクスポートを実行します（SQL Developerではなく）。
- ベースとなるホストにフル・アクセスできるので、scpを使用して、EC2インスタンスからOracle Cloud Infrastructureインスタンスにファイルを転送します（DBMS_FILE_TRANSFERではなく）。

利点：Data Pumpでは、大量のデータを高速にアンロード/ロードでき、データの並列処理、圧縮、暗号化が可能です。索引はエクスポートされないため、ネットワーク経由で転送しなければならないデータの量が大幅に減ります。

オンライン/オフライン：この方法は、オンラインで実行できます。Oracle GoldenGateと併用すれば、ゼロ・ダウンタイムの移行が可能になります。

データベース・サイズ：20 TBまでのデータベースに使用できます。

1. エクスポート・データ・ファイルの場所を作成します。

Data Pumpは、データベースからデータを抽出し、それらのデータを、データベース・ホストから利用可能なファイル・システムへの出力ファイルに書き込みます。大きなデータベースのエクスポート時に最善のパフォーマンスを得るには、書き込みスループットが非常に高い、大きなファイル・システムをプロビジョニングしてください。

ソース・データベースにログインし、データ・ファイルの宛先ディレクトリを作成します。

```
SQL> create directory MIG_DATA as '/u01/oradump';
Directory created.
SQL> grant all on directory MIG_DATA to public;
Grant succeeded.
SQL>
```

2. 転送するスキーマをエクスポートします。

```
expdp system/manager@prdpdb1 SCHEMAS=HR DIRECTORY=MIG_DATA  
DUMPFILE=exphr.dmp LOGFILE=MIG_DATA:exphr.log
```

3. ソース・ホストからターゲット・ホストにファイルをコピーします。

```
scp -i ~/.ssh/ociob /u01/oradump/exphr.dmp  
oracle@129.213.28.50:/u01/oradump/.
```

4. ターゲット・データベースに、Data Pumpの入力ファイル・ディレクトリを作成します。

```
SQL> create directory MIG_DATA as '/u01/oradump';  
Directory created.  
SQL> grant all on directory MIG_DATA to public;  
Grant succeeded.  
SQL>
```

5. インポートするスキーマに必要な表領域を作成します。

```
SQL> create tablespace hrts datafile size 1G autoextend on maxsize 5G;
```

6. ターゲット・データベースにデータをインポートします。

```
impdp  
system/WelCome#_12@129.213.28.50:1521/rkbmpdb2.rksublad1.rkvcn1.oraclevcn.  
com SCHEMAS=HR DIRECTORY=MIG_DATA DUMPFILE=exphr.dmp
```

これで、データベース全体がOracle Cloud InfrastructureのDatabaseサービスに移行され、使用可能な状態になります。

EC2移行オプション2：Data Pump：フル・トランスポートブル

「Data Pump：フル・トランスポートブル」の方法を使用すると、オンプレミスのホストからOracle Cloud InfrastructureのDatabaseサービスに、データベース全体をコピーすることができます。

利点：データベース全体をソースからターゲットにコピーするので、索引を再作成し、移行後に統計情報を収集する必要がありません。これにより、ターゲットでデータベースをインスタンス化するための時間が短縮されます。

オンライン/オフライン：これはオフラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

次の例では、11.2.0.4バージョンのデータベース全体をソース・ホストからコピーし、データ・ファイルをOracle Automatic Storage Management（Oracle ASM）上に配置して、12.2のデータベースにプラグブル・データベース（PDB）としてインポートします。

1. ソース・データベースに、Data Pumpのメタデータ・エクスポート出力ファイルを配置するためのディレクトリを作成します。

```
SQL> create directory MIG_DATA as '/u01/oradump';
Directory created.
SQL> grant all on directory MIG_DATA to public;
Grant succeeded.
SQL>
```

2. ソース表領域を読み取り専用モードで配置します。

```
SQL> alter tablespace hrts read only;
Tablespace altered.
SQL> alter tablespace oets read only;
Tablespace altered.
SQL> alter tablespace users read only;
Tablespace altered.
```

3. ソース・データベースで、Data Pumpフル・トランスポータブル・エクスポートを実行します。

```
expdp system FULL=Y TRANSPORTABLE=ALWAYS VERSION=12 DIRECTORY=MIG_DATA
DUMPFIL=MIG_DATA:expallts.dmp LOGFILE=MIG_DATA:expallts.log
```

4. データ・ファイルをコピーし、ダンプ・ファイルをソースからターゲットの一時ディレクトリにエクスポートします。
5. データ・ファイルを、一時的な場所からターゲット・ホスト上のOracle ASMへとコピーします。

グリッド・ユーザーとしてログインし、Oracle ASMの環境変数を設定します。オペレーティング・システムのコマンド・ラインから、ASMCMDコマンドライン・ユーティリティを起動してください。

```
ASMCMD> cp /u01/tmp/hrts01
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/hrts01.dbf
copying /u01/tmp/hrts01 ->
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/hrts01.dbf
ASMCMD> cp /u01/tmp/oets01
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/oets01.dbf
copying /u01/tmp/oets01 ->
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/oets01.dbf
ASMCMD> cp /u01/tmp/users02.dbf
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/users01.dbf
copying /u01/tmp/users02.dbf ->
+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/DATAFILE/users01.dbf
ASMCMD>
```

6. データ・ファイルがターゲットの場所にコピーされたら、ソース表領域を読み取り/書き込みモードに設定します。

```
SQL> alter tablespace hrts read write;
Tablespace altered.
SQL> alter tablespace oe read write;
Tablespace altered.
SQL> alter tablespace users read write;
Tablespace altered.
```

7. Data Pumpを使用して、メタデータをターゲット・データベースにインポートします。

```
impdp
system/WelCome#_12@129.213.83.23:1521/rkpdb3.rksublad1.rkvcn1.oraclevcn.co
m FULL=y DIRECTORY=MIG_DATA DUMPFILE=MIG_DATA:expallts.dmp
LOGFILE=MIG_DATA:impallts.log \
TRANSPORT_DATAFILES='+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/D
ATAFILE/hrts01.dbf','+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/D
ATAFILE/oets01.dbf','+DATA/RKDB3_IAD14P/6762A4F294A1C7B5E0534601000AB58F/D
ATAFILE/users01.dbf'
```

これで、移行が完了し、データベースが使用可能な状態になります。

EC2移行オプション3：Data Pump：トランスポートابل表領域

「Data Pump：トランスポートابل表領域」の方法を使用すると、オンプレミスのホストから Oracle Cloud InfrastructureのDatabaseサービスに、特定の表領域をコピーすることができます。

利点：表領域全体をソースからターゲットにコピーするので、索引を再作成し、移行後に統計情報を収集する必要がありません。これにより、ターゲットでデータベースをインスタンス化するための時間が短縮されます。

オンライン/オフライン：これはオフラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

次の例では、いくつかの表領域をソース・ホストからコピーし、データ・ファイルをOracle ASM上に配置して、12.2のデータベースにプラグابل・データベース（PDB）としてインポートします。

1. ソース・データベースに、Data Pumpのメタデータ・エクスポート出力ファイルを配置するためのディレクトリを作成します。

```
SQL> create directory MIG_DATA as '/u01/oradump';
Directory created.
SQL> grant all on directory MIG_DATA to public;
Grant succeeded.
SQL>
```

2. ソース表領域を読み取り専用モードで配置します。

```
SQL> alter tablespace hrts read only;
Tablespace altered.
SQL> alter tablespace oets read only;
Tablespace altered.
```

3. 移動する表領域のデータ・ファイルをリストします。

```
SQL> select file_name from dba_data_files where tablespace_name = 'HRTS';
FILE_NAME
-----
/u01/oradata/prddb/prdpdb1/hrts01.dbf
SQL> select file_name from dba_data_files where tablespace_name = 'OETS';
FILE_NAME
-----
/u01/oradata/prddb/prdpdb1/oets01.dbf
```

4. 移動する表領域を読み取り専用モードで配置します。

```
SQL> alter tablespace hrts read only;
Tablespace altered.
SQL> alter tablespace oets read only;
Tablespace altered.
SQL>
```

5. Data Pumpエクスポートを起動して、トランSPORTータブル表領域のエクスポートを実行します。

```
expdp system/manager@prdpdb1 TRANSPORT_TABLESPACES=HRTS,OETS,
TRANSPORT_FULL_CHECK=YES DIRECTORY=MIG_DATA DUMPFILE=exphrts.dmp
LOGFILE=MIG_DATA:exphrts.log
```

6. データ・ファイルをコピーし、出力ファイルをターゲット・ホストにエクスポートします。

```
[oracle@rkcomp1 122]$ scp -i ~/.ssh/ociob
/u01/oradata/prddb/prdpdb1/hrts01.dbf oracle@129.213.28.50:/u01/oradump/.
hrts01.dbf 100% 1024MB 125.5MB/s 00:08
[oracle@rkcomp1 122]$ scp -i
/u01/oradata/prddb/prdpdb1/oets01.dbf oracle@129.213.28.50:/u01/oradump/.
oets01.dbf 100% 1024MB 124.0MB/s 00:08
[oracle@rkcomp1 122]$ scp -i ~/.ssh/ociob /u01/oradump/exphrts.dmp
oracle@129.213.28.50:/u01/oradump/.
exphrts.dmp
```


7. データ・ファイルを、ターゲットのOracle ASMディスク・グループにコピーします。

```
ASMCMD> cp /u01/oradump/hrts01.dbf
+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3/DATAFILE/hrts01.dbf
copying /u01/oradump/hrts01.dbf ->
+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3/DATAFILE/hrts01.dbf
ASMCMD> cp /u01/oradump/oets01.dbf
+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3/DATAFILE/oets01.dbf
copying /u01/oradump/oets01.dbf ->
+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3/DATAFILE/oets01.dbf
```

8. ソース・データベースの表領域を、読み取り/書き込みモードに設定します。

```
SQL> alter tablespace hrts read write;
Tablespace altered.
SQL> alter tablespace oets read write;
Tablespace altered.
SQL>
```

9. ターゲット・データベースでユーザーを作成します。

```
SQL> create user hr identified by hr temporary tablespace temp account
unlock;
User created.
SQL> create user oe identified by oe temporary tablespace temp account
unlock;
User created.
```

10. Data Pumpを起動して、メタデータをターゲット・データベースにインポートします。

```
impdp
system/WelCome#_12@129.213.28.50:1521/rkbmpdb2.rksub1ad1.rkvcn1.oraclevcn.
com DIRECTORY=MIG_DATA DUMPFILE=exphrts.dmp LOGFILE=MIG_DATA:exphrts.log
\
TRANSPORT_DATAFILES='+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3
/DATAFILE/hrts01.dbf',
'+DATA/rkbmdb2_iad1r2/684206652ED426EBE0535701000AECE3/DATAFILE/oets01.dbf
'
```

11. ターゲット・データベースの表領域を、読み取り/書き込みモードに設定します。

```
SQL> alter tablespace hrts read write;
Tablespace altered.
SQL> alter tablespace oets read write;
Tablespace altered.
```

EC2移行オプション4：リモート・クローン：PDB

マルチテナント・データベースを使用していて、ソース・データベースがバージョン12.1以降である場合は、プラグブル・データベース（PDB）用のリモート・クローン機能を使用して、PDBをソース・データベースからターゲットにクローニングすることができます。

利点：PDB全体をソースからターゲットにコピーするので、索引を再作成し、移行後に統計情報を収集する必要がありません。これにより、ターゲットでデータベースをインスタンス化するための時間が短縮されます。

オンライン/オフライン：これはオンラインの方法です。開発およびテスト・データベースの移行に最適です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：サイズが5TB以下のデータベースに使用できます。

1. ソース・データベースへのデータベース・リンクを作成します。

ターゲット・データベースのコンテナ・ルートにsysとしてログインし、データベース・リンクを作成します。データベース・リンクを使用して、接続しようとしているPDBが、クローニングするPDBであることを確認してください。

```
create database link onprem_pdb connect to system identified by manager
using '72.55.128.32:1521/prdpdb1';
```

2. ソースPDBをターゲット・データベースにクローニングします。

```
SQL> create pluggable database prdpdb1 from prdpdb1@onprem_pdb;
Pluggable database created.
```

3. PDBを開きます。

```
SQL> alter pluggable database prdpdb1 open;
```

4. 次のSQLを実行して、PDBに問題がないか確認し、問題があれば、それらを修正するための必要な手順を実行します。

```
SQL> select message,time from pdb Plug_in_violations;
```

EC2移行オプション5：リモート・クローン：非CDB

マルチテナント・データベースを使用していて、ソース・データベースがバージョン12.1以降である場合は、プラグブル・データベース（PDB）用のリモート・クローン機能を使用して、非CDB（コンテナ・データベース）をソース・データベースからターゲットにクローニングすることができます。

利点：非CDBデータベース全体を、ソースからターゲットにコピーします。

オンライン/オフライン：これはオンラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：最大5TBのデータベースに使用できます。

1. ソース・データベースへのデータベース・リンクを作成します。

ターゲット・データベースのコンテナ・ルートにsysとしてログインし、データベース・リンクを作成します。データベース・リンクを使用して、接続しようとしている非CDBが、クローニングする非CDBであることを確認してください。

```
create database link onprem_noncdb connect to system identified by manager
using '72.55.128.32:1521/prdncdb';
```

2. 非CDBをソースからクローニングして、プラグブル・データベースを作成します。

```
SQL> CREATE PLUGGABLE DATABASE rkbpdb2 FROM prdncdb@onprem_noncdb;
Pluggable database created.
```

3. ターゲット・データベースに新しく作成されたrkbpdb2 PDBにsysとしてログインし、noncdb_to_pdbスクリプトを実行します。

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

4. PDBを読み取り/書き込みモードで開きます。

```
SQL> alter pluggable database rkbpdb2 open;
```

5. 次のSQLを実行して、PDBに問題がないか確認し、問題があれば、それらを修正するための必要な手順を実行します。

```
SQL> select message,time from pdb_plug_in_violations;
```

EC2移行オプション6：RMAN：トランスポートブルPDB

マルチテナント・データベースを使用していて、ソース・データベースがバージョン12.1以降である場合は、Oracle Recovery Manager (RMAN) を使用して、プラグブル・データベース (PDB) をソース・データベースから宛先に転送することができます。

利点：この方法では、マルチステップの移行が可能です。ターゲット・データベースの増分リカバリを複数回実行して、カットオーバー時に実行される最終リカバリの所要時間を最小限に短縮することができます。

オンライン/オフライン：これはオンラインの方法です。ダウンタイムは、最終バックアップをとり、それを転送し、ターゲットに適用するまでの時間になります。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

PDBを移動するためのオプションと手順は、ソース・データベースのバージョンによって若干違ってきます。次の手順は、12.2のソースPDBを、非一貫性バックアップを使用して移動する場合の手順です（つまり、ソースPDBをクローズする必要はありません）。

1. ソース・データベースのコンテナ・ルートにsysとしてログインし、現在のチェックポイント変更番号を問い合わせます。番号をメモしてください。

```
SQL> SELECT CHECKPOINT_CHANGE# FROM V$DATABASE;
CHECKPOINT_CHANGE#
-----
1446203
```

2. RMANを使用して、ソースPDBの非一貫性バックアップをとります。

```
rman target /
RMAN> BACKUP INCREMENTAL LEVEL 0
FOR TRANSPORT
ALLOW INCONSISTENT
PLUGGABLE DATABASE prdpdb1 FORMAT '/u01/backups/hr_pdb_level0.bck';
```

3. RMANを使用して、ソースPDBの最終一貫性バックアップをとります。

```
sqlplus / as sysdba
SQL> alter pluggable database prdpdb1 close immediate;
Pluggable database altered.
SQL> exit;
```

```
rman target /
connected to target database: PRDDB (DBID=3107372461)
RMAN> BACKUP INCREMENTAL FROM SCN 1446203
FOR TRANSPORT
UNPLUG INTO '/u01/backups/metadata_hr_pdb.xml'
PLUGGABLE DATABASE prdpdb1 FORMAT
```

このコマンドは、prdpdb1というPDBの最終増分バックアップをとり、そのPDBをアンプラグした後、アンプラグされたPDBのXMLファイルを作成します。

4. ターゲット・データベース・ホストにバックアップをコピーします。

```
[oracle@rkcomp1 backups]$ scp -i ~/.ssh/ociob ./*
oracle@129.213.28.50:/u01/backups/.
hr_pdb_level0.bck          100% 486MB 127.1MB/s   00:03
hr_pdb_level1_con.bck     100%  12MB 146.2MB/s   00:00
metadata_hr_pdb.xml       100% 9138    3.9MB/s    00:00
```

- レベル0バックアップをターゲット・データベースにリストアします。

その際、FORMATスイッチを使用して、データ・ファイルがOracle Cloud Infrastructure DatabaseサービスのOracle ASMディスク・グループにリストアされるようにしてください。

```
RMAN> RESTORE
      FOREIGN PLUGGABLE DATABASE hr_pdb FORMAT '+DATA/RKDB4_IAD1TD/%u'
      FROM BACKUPSET '/u01/backups/hr_pdb_level0.bck';
```

- ソース・データベースからのレベル1バックアップを使用して、ターゲットPDBをリカバリします。

```
RMAN> RECOVER
      USING '/u01/backups/metadata_hr_pdb.xml'
      FOREIGN DATAFILECOPY
      '+DATA/RKDB4_IAD1TD/7jsublcs', '+DATA/RKDB4_IAD1TD/jdsublcs', '+DATA/RKDB4_IAD1TD/c3sublcs', '+DATA/RKDB4_IAD1TD/r7sublcs', '+DATA/RKDB4_IAD1TD/u2sublcs', '+DATA/RKDB4_IAD1TD/edsublcs'
      FROM BACKUPSET '/u01/backups/hr_pdb_level1_con.bck';
```

- リストアされたPDBの名前を確認します（RMANのリストア/リカバリ・プロセスでは、PDBに新しい一意名が付けられます）。

```
RMAN> select name from v$pdb;
```

- PDBを開きます。

```
RMAN> ALTER PLUGGABLE DATABASE RKDB4_IAD1TD_63249 open;
Statement processed
```

EC2移行オプション7：RMAN：トランスポータブル・バックアップ・セット

マルチテナント・データベースを使用していて、ソース・データベースがバージョン12.1以降である場合は、Oracle Recovery Manager（RMAN）を使用して、選択された表領域をソース・データベースからターゲット・データベースに転送することができます。

この方法を使用する場合は、ターゲット・データベースを作成し、最初のフル・バックアップを実行した後、追加の増分バックアップを実行することで、データベースを移行可能な状態に維持することができます。ソースで必要となるダウンタイムは、最後の増分バックアップをとり、その最終バックアップを使用してターゲット・データベースをリカバリするのにかかる時間のみです。

利点：この方法では、マルチステップの移行が可能です。ターゲット・データベースの増分リカバリを複数回実行して、カットオーバー時に実行される最終リカバリの所要時間を最小限に短縮することができます。

オンライン/オフライン：これはオンラインの方法です。ダウンタイムは、最終増分バックアップをとり、ファイルを転送し、ターゲットにファイルを適用するまでの時間に抑えられます。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

1. RMANを使用して、ターゲット・データベースに移動するソース・データベースから、表領域のフル・バックアップをとります。

```
rman target sys/manager@rkcomp1:1521/prdpdb1
RMAN> BACKUP
      FOR TRANSPORT
      ALLOW INCONSISTENT
      INCREMENTAL LEVEL 0
      FORMAT '/u01/backups/hrts_inconsist_10.bck'
      TABLESPACE HRTS;
```

このバックアップ・プロセスでは、処理の進行中（および完了後）に、そのソース・データベースに対する問合せやトランザクションを継続できます。

2. ソース表領域の最終増分バックアップをとります。

- A. 表領域を読み取り専用モードで配置します。

```
sqlplus sys/manager@rkcomp1:1521/prdpdb1 as sysdba
SQL> alter tablespace hrts read only;
Tablespace altered.
```

- B. 最終増分レベル1バックアップを実行して、表領域の最後のレベル1バックアップ以降に変更されたブロックのみをバックアップします。

```
oracle@rkcomp1 122]$ rman target sys/manager@rkcomp1:1521/prdpdb1
connected to target database: PRDDB:PRDPDB1 (DBID=2789311088)
RMAN> BACKUP
      FOR TRANSPORT
      INCREMENTAL LEVEL 1
      TABLESPACE HRTS
      FORMAT '/u01/backups/hrts_inconsist_11.bck'
      DATAPUMP FORMAT '/u01/backups/hrts_incr_dmp.dmp';
```

画面上のバックアップ・メッセージの一部として、次のような数行のメッセージが表示されます。このファイル名をメモしてください。リストアとリカバリの手順を実行する際に必要になります。

```
EXPDP>*****
* EXPDP> Dump file set for SYS.TRANSPORT_EXP_PRDDB_kDBs is:
EXPDP> /u01/app/oracle/product/12.2.0.1/db_1/dbs/backup_tts_PRDDB_20164.dmp
EXPDP>*****
```

3. ソース・データベースの表領域を、読み取り/書き込みモードに戻します。

```
sqlplus sys/manager@rkcomp1:1521/prdpdb1 as sysdba
SQL> alter tablespace hrts read write;
Tablespace altered.
```

4. オペレーティング・システムのセキュア・コピー・ユーティリティを使用して、Oracle Cloud Infrastructure内のターゲット・ホストにファイルをコピーします。

```
[oracle@rkcomp1 backups]$ scp -i ~/.ssh/ociob /u01/backups/*  
oracle@129.213.28.50:/u01/backups/.
```

5. ターゲット・データベースでユーザーを作成します。

```
sqlplus / as sysdba  
SQL> alter session set container=rkbpdb2;  
SQL> create user hr identified by hr temporary tablespace temp;  
SQL> GRANT CREATE SESSION, CREATE VIEW, ALTER SESSION, CREATE SEQUENCE TO  
hr;  
SQL> GRANT CREATE SYNONYM, CREATE DATABASE LINK, RESOURCE , UNLIMITED  
TABLESPACE TO hr;
```

6. 最初のフル・バックアップから表領域をリストアします。

ターゲットのデータベース・サービス・インスタンスで、表領域のバックアップをPDB内にリストアします。その際、FORMATキーワードを指定して、データ・ファイルをDATA ディスク・グループにリストアする必要があることを指定します。

```
rman target  
sys/WelCome#_12@rkbmdb2:1521/rkbpdb2.rksublad1.rkvcn1.oraclevcn.com  
connected to target database: RKBMD2:RKBMPDB2 (DBID=4247588157)  
  
RMAN> RESTORE  
FOREIGN TABLESPACE hrts  
FORMAT '+DATA/rkbmdb2_iad1r2/datafile/%u'  
FROM BACKUPSET '/u01/backups/hrts inconsist 10.bck';
```

FORMAT文字列として%uを指定したので、リストアされたデータ・ファイルには、Oracleによって生成されたランダムなファイル名が割り当てられます。

7. Oracle ASMインスタンスにログインし、ASMCMDを使用して、
+DATA/rkbmdb2_iad1r2/datafile/ directoryからデータ・ファイルをリストし、新しい
ファイル名を取得します。

8. 最終増分バックアップを使用して、表領域をリカバリします。

```
rman target  
sys/WelCome#_12@rkbmdb2:1521/rkbpdb2.rksublad1.rkvcn1.oraclevcn.com  
connected to target database: RKBMD2:RKBMPDB2 (DBID=4247588157)  
  
RMAN> RECOVER  
FOREIGN DATAFILECOPY '+DATA/rkbmdb2_iad1r2/datafile/21surg08'  
FROM BACKUPSET '/u01/backups/hrts inconsist 11.bck';
```

9. RMANバックアップから、Data Pumpエクスポートのダンプ・ファイルを一時的にリストアします。
DUMP FILE句については、手順2でメモしたファイル名を使用します。

```
rman target
sys/WelCome#_12@rkbmdb2:1521/rkbmpdb2.rksublad1.rkvcn1.oraclevcn.com

RMAN> RESTORE
  DUMP FILE 'backup_tts_PRDDB_20164.dmp'
  DATAPUMP DESTINATION '/u01/oradump'
  FROM BACKUPSET '/u01/backups/hrts_incr_dmp.dmp';
```

10. これで、データがターゲット・データベースに一時的にリストアされ、リカバリされたので、Data Pumpを使用して表領域のメタデータをインポートし、プロセスを完了します。

```
impdp
system/WelCome#_12@rkbmdb2:1521/rkbmpdb2.rksublad1.rkvcn1.oraclevcn.com
directory=MIG_DATA dumpfile=backup_tts_PRDDB_20164.dmp
transport_datafiles='+DATA/rkbmdb2_iad1r2/datafile/21surg08'
logfile=MIG DATA:hrts_incr_dmp.log
```

11. 表領域を読み取り/書き込みモードを開きます。

```
sqlplus
system/WelCome#_12@rkbmdb2:1521/rkbmpdb2.rksublad1.rkvcn1.oraclevcn.com
SQL> alter tablespace hrts read write;
Tablespace altered.
SQL> alter user hr default tablespace hrts;
User altered
```

EC2移行オプション8：RMAN：アクティブ・データベースからの複製

RMANでは、必要なすべてのファイルをデータベースの稼働中にコピーして、ターゲット・データベースにソース・データベースのコピーを作成できます。バックアップベースのRMAN複製プロセスを使用する場合は、RMAN複製プロセスを実行する前に、まずソース・システムのバックアップをとり、その後、それをターゲット・サーバー上のステージング・ロケーションにコピーする必要があります。RMAN複製プロセスをアクティブ・データベースから使用する場合は、この追加のバックアップ作成手順は不要となります。

ターゲット環境に新しいデータベース・インスタンスを作成するので、データベースを複製する前に、新しいインスタンスに必要なストレージをプロビジョニングする必要があります。

利点：これは単一ステップの移行です。一時ストレージを必要とせずに、ソース・データベースをターゲットに直接複製できます。

オンライン/オフライン：これはオンラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ : 5 TBまでのデータベースに使用できます。

1. dbcliを使用して、新しいデータベースのためのデータベース・ストレージを作成します。

```
[root@rkbmdb1 ~]# dbcli create-dbstorage --dbname prddb --dataSize 30
```

2. list-dbstorageコマンドを使用して、作成が完了したことを確認します。

```
[root@rkbmdb1 ~]# dbcli list-dbstorages
```

3. dbcliを実行し、Oracle Homeの場所を特定します。

```
[root@rkbmdb1 ~]# dbcli list-dbhomes
```

4. Oracle Homeから非デフォルトのポートを使用して、新しいリスナーを作成します。これを行うには、\$ORACLE_HOME/network/adminディレクトリにlistener.oraというファイルを作成し、そのファイルに次の行を追加します。

```
LISTENER_aux_prddb=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS= (PROTOCOL=TCP) (HOST=rkbmdb1.rksublad1.rkvcn1.oraclevcn.com) (PORT=
1528))
    )
  )
SID_LIST_LISTENER_aux_prddb=
  (SID_LIST=
    (SID_DESC=
      (GLOBAL_DBNAME=prddb)
      (ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1)
      (SID_NAME=prddb)
    )
  )
```

5. ファイアウォール・ポートを開きます。

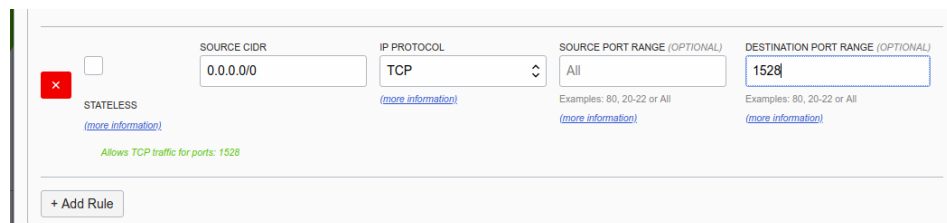
- A. データベース・サービス・ホスト上のファイアウォールから、ポート1528を開きます。

```
[root@rkbmdb1 ~]# iptables -I INPUT 5 -m state --state NEW -p tcp --
port 1528 -j ACCEPT
[root@rkbmdb1 ~]# service iptables save
[root@rkbmdb1 ~]# service iptables restart
```

B. Oracle Cloud Infrastructure内のセキュリティ・リストで、ポート1528を開きます。

Oracle Cloud Infrastructureコンソールで、Databaseサービスが属しているサブセットのセキュリティ・リストに移動します。セキュリティ・リストを編集し、次の値を使用して行を追加してください。

- ソースCIDR : 0.0.0.0/0
- IPプロトコル : TCP
- ソース・ポート範囲 : すべて
- 宛先ポート : 1528



6. 次のエントリを使用して初期化パラメータを作成し、補助インスタンスを起動します。これは、RMAN複製プロセスを開始するための一時ファイルです。このファイルは、クローニング・プロセス中に、ソース・インスタンスからのファイルによって上書きされます。

\$ORACLE_HOME/dbs/initprddb.oraファイルを作成し、次のエントリを追加します。

```
db_name=prddb
control_files='+RECO/PRDDB/CONTROLFILE/'
db_block_size=8192
db_file_name_convert=('/u01/oradata/prddb','+DATA/PRDDB/DATAFILE')
log_file_name_convert=('/u01/oradata/prddb','+DATA/PRDDB/DATAFILE')
db_create_file_dest='+DATA'
db_create_online_log_dest_1='+RECO'
db_recovery_file_dest='+RECO'
db_recovery_file_dest_size=50G
```

7. データベース・リスナーを起動します。

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1
export ORACLE_SID=prddb
export ORACLE_UNQNAME=prddb
export NLS_DATE_FORMAT="mm/dd/yyyy hh24:mi:ss"
export PATH=$ORACLE_HOME/bin:$PATH

oracle@rkbmdb1 ~]$ lsnrctl start LISTENER_aux_prddb
```

8. インスタンス・パスワード・ファイルを、ソース・データベースからターゲット・データベース・ホストにコピーします。

パスワード・ファイルは、インスタンスにリモート接続してクローニング・プロセスを開始するために必要です。

```
scp -i ~/.ssh/ociob /u01/app/oracle/product/12.2.0.1/db_1/dbs/orapwprddb  
oracle@rkbmdb1:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/.
```

9. ターゲット・ノードで、補助インスタンスをnomountモードで起動します。

```
SQL> sqlplus / as sysdba  
SQL> startup nomount;
```

10. ソース・データベースにログインし、rmanを実行してRMANアクティブ複製を実行します。このプロセスは、ソース・データベースのフル・コピーをとり、それをターゲット・データベース・サーバーにクローニングします。

```
rman target sys/manager@129.213.89.242:1521/prddb auxiliary  
sys/manager@129.213.18.112:1528/prddb  
RMAN> uplciate target database to prddb from active database  
spfile  
PARAMETER_VALUE_CONVERT prddb prddb PRDDB PRDDB  
set cluster_database='false'  
set db_name='prddb'  
set db_unique_name='prddb'  
set db_create_file_dest='+DATA'  
set db_create_online_log_dest_1='+RECO'  
set db_recovery_file_dest='+RECO'  
set db_recovery_file_dest_size='50G'  
set audit_file_dest =  
'/u01/app/oracle/product/12.2.0.1/dbhome_1/rdbms/audit'  
set log_archive_dest_1='LOCATION=+DATA'  
reset control files;
```

11. ターゲット・データベース・サーバーで、rootとしてログインし、dbcliを実行してデータベースを登録します。

```
[root@rkbmdb1 ~]# dbcli register-database --dbclass OLTP --dbshape odb1 --  
servicename prddb -syspassword
```

これで、データベースが使用可能な状態になります。

EC2移行オプション9：PDBのアンプラグとプラグ

Oracleデータベースのマルチテナント・オプションを使用している場合は、プラグブル・データベース（PDB）全体をソース・コンテナ・データベースからアンプラグし、それをOracle Cloud Infrastructure Databaseサービスのターゲット・コンテナ・データベースにプラグすることができます。

利点：これは、ダウンタイムを必要とする、単一ステップの移行方法です。

オンライン/オフライン：これはオフラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

1. ソース・データベースでPDBをクローズします。

```
sqlplus / as sysdba
SQL> Alter pluggable database prdpdb1 close;
```

2. PDBをアンプラグします。

```
SQL> ALTER PLUGGABLE DATABASE prdpdb1 UNPLUG INTO '/u01/tmp/prdpdb1.xml';
```

3. セキュア・コピー・ユーティリティを使用して、XMLファイルとデータ・ファイルを Databaseサービスのコンピュート・ノードに転送します。

```
scp -i ~/.ssh/ociob /u01/tmp/prdpdb1.xml oracle@129.213.18.112:/u01/tmp/.
scp -i ~/.ssh/ociob /u01/oradata/prddb/prdpdb1/*
oracle@129.213.18.112:/u01/tmp/.
```

4. ターゲット・データベース・インスタンスで、CREATE PLUGGABLE DATABASEコマンドを実行して、prdpd1 PDBを作成します。

```
SQL> CREATE PLUGGABLE DATABASE prdpdb1 USING '/u01/tmp/prdpdb1.xml'
SOURCE_FILE_DIRECTORY = '/u01/tmp'
COPY
FILE_NAME_CONVERT = ('/u01/tmp', '+DATA') ;
```

FILE_NAME_CONVERTを指定すると、データ・ファイルが/u01/tmpからDATAディスク・グループにコピーされます。

5. PDBを開きます。

```
SQL> alter pluggable database prdpdb1 open;
```

6. 次のビューを問い合わせ、PDBにエラーがないかどうかを特定し、エラーがあれば、それらを修正します。

```
SQL> select message,time from pdb_plug_in_violations;
```

EC2移行オプション10：非CDBのアンプラグとプラグ

ソース・データベースが12cの非CDB（コンテナ・データベース）である場合は、次の手順を使用して、非CDB全体をプラグブル・データベース（PDB）としてOracle Cloud Infrastructure Databaseサービス上のCDBに移動することができます。

利点：これは単一ステップの移行です

1. ソースの非CDBデータベースを読み取り専用モードで起動します。

```
sqlplus / as sysdba
SQL> startup mount;
SQL> alter database open read only;
```

2. ソース・データベース・ホストで、DBMS_PDB.DESCRIBEコマンドを実行して、ターゲット・データベースでプラグインされるデータ・ファイルのリストを含んだXMLファイルを生成します。

```
sqlplus / as sysdba
SQL> BEGIN
  DBMS_PDB.DESCRIBE(
    pdb_descr_file => '/u01/oradump/ncdb.xml');
END;
```

3. セキュア・コピー・ユーティリティを使用して、DBMS_PDB.DESCRIBEコマンドの出力とデータ・ファイルをターゲット・データベース・ホストにコピーします。

```
scp -i ~/.ssh/ociob /u01/oradump/ncdb.xml oracle@129.213.18.112:/u01/tmp/.
scp -i ~/.ssh/ociob /u01/oradata/prdncdb/*
oracle@129.213.18.112:/u01/tmp/.
```

4. Oracle Cloud Infrastructure上のCDBにPDBを作成します。

```
SQL> CREATE PLUGGABLE DATABASE rkbmpdb2 USING '/u01/tmp/ncdb.xml'
  SOURCE_FILE_DIRECTORY = '/u01/tmp'
  COPY
  FILE_NAME_CONVERT = ('/u01/tmp', '+DATA') ;
```

5. rkbmpdb2 PDB（先ほどターゲットCDBに作成したPDB）に接続してクリーンアップ・スクリプトを実行し、\$ORACLE_HOME/rdbms/admin/noncdb_to_pdbスクリプトを実行します。

```
SQL> alter session set container=rkbmpdb2;
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb
```

6. CDBを開きます。

```
SQL> alter pluggable database rkbmpdb2 open;
```

7. 次のビューを問い合わせ、PDBにエラーがないかどうかを特定し、エラーがあれば、それらを修正します。

```
SQL> select message,time from pdb plug in violations;
```

EC2移行オプション11：Data Guardを使用した移行

Oracle Data Guardでは、スタンバイ・データベースを作成、保守、管理する一連のサービスが提供されます。スタンバイ・データベースはプライマリ・データベースのフル・コピーで、Oracle Cloud Infrastructure Databaseサービスへの移行に伴うダウンタイムを減らすために使用できます。スタンバイ・データベースでは、REDO転送およびREDO Applyサービスを使用して、プライマリ・データベースとの同期を維持することができます。Oracle Cloud Infrastructureにカット・オーバーする準備ができたなら、Data Guardフェイルオーバーを実行するだけで、Oracle Cloud Infrastructure上に最新かつフル機能のデータベースが作成されます。

利点：Data Guardを使用すれば、Oracle Cloud Infrastructure上のターゲット・データベースを設定し、必要であれば、ターゲット・データベースへのカットオーバー前に、（アクティブなData Guardやスナップショット・スタンバイ・データベースを使用して）いくつかのテストを実行することができます。

オンライン/オフライン：これはオンラインの方法です。ゼロ・ダウンタイムの移行が可能です。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

次の手順は、RMAN複製を使用してスタンバイ・データベースを作成する方法と、スタンバイ・データベースを移行用にフェイルオーバーする方法について説明したものです。

ターゲット環境に新しいデータベース・インスタンスを作成するので、データベースを複製する前に、新しいインスタンスに必要なストレージをプロビジョニングする必要があります。

1. dbcliを使用して、新しいデータベースのためのデータベース・ストレージを作成します。

```
[root@rkbmdb2 ~]# dbcli create-dbstorage --dbname prddb --dataSize 30
```

2. list-dbstorageコマンドを使用して、作成が完了したことを確認します。

```
[root@rkbmdb2 ~]# dbcli list-dbstorages
```

3. dbcliを実行し、Oracle Homeの場所を特定します。

```
[root@rkbmdb2 ~]# dbcli list-dbhomes
```

4. ソース・データベースで強制ロギングを設定します。

```
SQL> alter database force logging;
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;
```

5. REDOログ・データを受信するようにプライマリ・データベースを構成します。

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/oradata/prddb/sbredolog1.rdo') SIZE 100M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/oradata/prddb/sbredolog2.rdo') SIZE 100M;
```

6. tnsnames.ora ファイルをサービス名エントリで更新します。

ソースとターゲットの両方のデータベース・システムで、

\$ORACLE_HOME/rdbms/admin/tnsnames.ora ファイルに次のエントリを追加します。
これらのエントリで、18.218.43.72 はソース・データベースが現在実行されているホストの IP アドレスです。rkbmdb2.rksublad1.rkvcn1.oraclevcn.com は、Oracle Cloud Infrastructure 内のターゲット・データベース・ホストのホスト名です。

```
prddb =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 18.218.43.72) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = prddb)
    )
  )
prddb_stby =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST =
rkbmdb2.rksublad1.rkvcn1.oraclevcn.com) (PORT = 1528))
    )
    (CONNECT_DATA =
      (SID = prddb)
    )
  )
```

7. Oracle Home から非デフォルトのポートを使用して、新しいリスナーを作成します。ターゲット・データベースで、\$ORACLE_HOME/network/admin ディレクトリに listener.ora というファイルを作成し、そのファイルに次の行を追加します。

```
LISTENER_aux_prddb=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP) (HOST=rkbmdb2.rksublad1.rkvcn1.oraclevcn.com) (PORT=
1528))
    )
  )
SID_LIST_LISTENER_aux_prddb=
  (SID_LIST=
    (SID_DESC=
      (GLOBAL_DBNAME=prddb_stby)
      (ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1)
      (SID_NAME=prddb)
    )
  )
```

8. ファイアウォール・ポートを開きます。

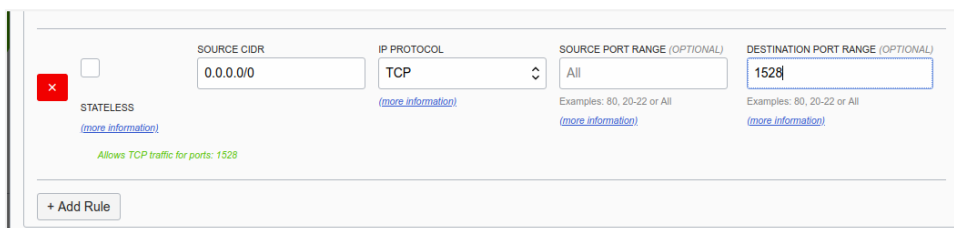
A. Databaseサービス・ホスト上のファイアウォールから、ポート1528を開きます。

```
[root@rkbmdb2 ~]# iptables -I INPUT 5 -m state --state NEW -p tcp --dport 1528 -j ACCEPT
[root@rkbmdb2 ~]# service iptables save
[root@rkbmdb2 ~]# service iptables restart
```

B. Oracle Cloud Infrastructure内のセキュリティ・リストで、ポート1528を開きます。

Oracle Cloud Infrastructureコンソールで、Databaseサービスが属しているサブセットのセキュリティ・リストに移動します。セキュリティ・リストを編集し、次の値を使用して行を追加してください。

- **ソースCIDR** : 0.0.0.0/0
- **IPプロトコル** : TCP
- **ソース・ポート範囲** : すべて
- **宛先ポート** : 1528



9. ターゲット・データベースで、データベース・リスナーを起動します。

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1
export ORACLE_SID=prddb
export ORACLE_UNQNAME=prddb
export NLS_DATE_FORMAT="mm/dd/yyyy hh24:mi:ss"
export PATH=$ORACLE_HOME/bin:$PATH

oracle@rkbmdb2 ~]$ lsnrctl start LISTENER_aux_prddb
```

10. インスタンス・パスワード・ファイルを、ソース・データベースからターゲット・データベース・ホストにコピーします。

パスワード・ファイルは、インスタンスにリモート接続してクローニング・プロセスを開始するために必要です。

```
scp -i ~/.ssh/ociob /u01/app/oracle/product/12.2.0.1/db_1/dbs/orapwprddb
oracle@rkbmdb2:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/.
```


11. 次のエントリを使用して初期化パラメータを作成し、補助インスタンスを起動します。これは、RMAN複製プロセスを開始するための一時ファイルです。このファイルは、クローニング・プロセス中に、ソース・インスタンスからのファイルによって上書きされます。

\$ORACLE_HOME/dbs/initprddb.oraファイルを作成し、次のエントリを追加します。

```
db_name=prddb
control_files='+RECO/PRDDB/CONTROLFILE/'
db_block_size=8192
db_file_name_convert=('/u01/oradata/prddb','+DATA/PRDDB/DATAFILE')
log_file_name_convert=('/u01/oradata/prddb','+DATA/PRDDB/DATAFILE')
db_create_file_dest='+DATA'
db_create_online_log_dest_1='+RECO'
db_recovery_file_dest='+RECO'
db_recovery_file_dest_size=50G
```

12. ターゲット・ノードで、補助インスタンスをnomountモードで起動します。

```
SQL> sqlplus / as sysdba
SQL> startup nomount;
```

13. Oracle ASMインスタンスにログインし、PRDDB用のディレクトリを作成します。

```
asmcmd
ASMCMD> mkdir +DATA/PRDDB
ASMCMD> mkdir +DATA/PRDDB/DATAFILE
```

14. ソース・データベースにログインし、rmanを実行してRMANアクティブ複製を実行します。このプロセスは、ソース・データベースのフル・コピーをとり、それをターゲット・データベース・サーバーにクローニングします。

```
rman target sys/manager@18.218.43.72:1521/prddb auxiliary
sys/manager@129.213.28.50:1528/prddb
RMAN> duplicate target database for standby from active database
dorecover
spfile
set cluster_database='false'
set db_name='prddb'
set db_unique_name='prddb_stby'
set
control_files='+DATA/PRDDB/DATAFILE/control01.ctl','+DATA/PRDDB/DATAFILE/contr
ol02.ctl'
set db_create_file_dest='+DATA'
set db_create_online_log_dest_1='+RECO'
set db_recovery_file_dest='+RECO'
set db_recovery_file_dest_size='50G'
set audit_file_dest = '/u01/app/oracle/product/12.2.0.1/dbhome_1/rdbms/audit'
set log_archive_dest_1='LOCATION=+DATA'
nofilenamecheck
;
```

15. ターゲット・データベースでマネージド・リカバリを起動します。

```
sqlplus / as sysdba
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM
SESSION;
```

16. Data Guardブローカを設定します。

- A. 両方のデータベースから、次のコマンドを実行します。

```
sqlplus / as sysdba
SQL> ALTER SYSTEM SET dg_broker_start=true;
```

- B. プライマリ・データベースから、次のコマンドを実行します。

```
dgmgrl
sys/manager@rkcomp1:1521/prddb
Connected to "prddb"
Connected as SYSDBA.
DGMGRL> CREATE CONFIGURATION prddb_dg_config AS PRIMARY DATABASE IS
prddb CONNECT IDENTIFIER IS prddb;
Configuration "prddb_dg_config" created with primary database "prddb"
DGMGRL> ADD DATABASE prddb_stby AS CONNECT IDENTIFIER IS prddb_stby
MAINTAINED AS PHYSICAL;
Database "prddb_stby" added
DGMGRL> Enable Configuration;
Enabled.
```

17. 新しいスタンバイ・データベースにフェイル・オーバーします。新しいスタンバイ・データベース・ホストで、次のコマンドを実行します。

```
dgmgrl sys/manager@prddb_stby
Connected to "prddb_stby"
Connected as SYSDBA.
DGMGRL> failover to prddb_stby;
Performing failover NOW, please wait...
Failover succeeded, new primary is "prddb_stby"
DGMGRL>
```

この時点で、Oracle Cloud Infrastructure内のデータベース・コピーへのカット・オーバーが完了します。

EC2移行オプション12：Recovery Managerを使用した移行

Oracle Recovery Manager（RMAN）では、データベースを効率的にバックアップしてリストアするための必要なプログラムとプロセスがすべて提供されます。ソース・データベースがリトルエンディアン・プラットフォームで実行されている場合は、RMANを使用してデータベースをソース・インスタンスからバックアップし、それをOracle Cloud Infrastructure上のDatabaseサービス・インスタンスにリストアすることができます。RMANには、ソース・インスタンス上のバックアップを並列処理、圧縮、および暗号化するための必要なメカニズムが備わっています。

ソース環境からターゲット環境にバックアップを転送するには、2つの方法があります。前のオプションで示したセキュア・コピー・メカニズム（たとえば、scp）を使用するか、データベースを Oracle Cloud Infrastructure内のObject Storageに直接バックアップして、Object Storageから Databaseサービス・インスタンスに直接リストアすることができます。次の手順は、セキュア・コピーを使用する方法について説明したものです。Object Storageを使用する方法については、その後のセクションで説明します。

利点：この方法では、マルチステップの移行が可能です。ターゲット・データベースの増分リカバリを複数回実行して、カットオーバー時に実行される最終リカバリの所要時間を最小限に短縮することができます。

オンライン/オフライン：これはオンラインの方法です。ゼロダウンタイムの移行を実行する機能は提供されません。

データベース・サイズ：すべてのサイズのデータベースに使用できます。

1. RMANを使用して、ソース・データベースのフル・バックアップとアーカイブ・ログを作成します。RMANによって作成されたバックアップ・ファイルは、バックアップ時に指定されたファイル・システムに配置されます。

次のコマンドは、バックアップに対してパスワードベースの暗号化を実行します（encryptitは、例として使用されているパスワードです）。この例の場合、RMANは、/u01/backupsに保存されたファイルにデータベースをバックアップします。

```
rman target /
RMAN> set encryption identified by encryptit only
executing command: SET encryption
using target database control file instead of recovery catalog
RMAN> run {
    allocate channel c1 device type disk format '/u01/backups/%U' ;
    backup as compressed backupset incremental level 0 SECTION SIZE=512M
    DATABASE PLUS ARCHIVELOG TAG='fullbackup';
    release channel c1;
}
```

2. OSレベルのセキュア・コピー・ユーティリティを使用して、RMANバックアップ・ファイルをソースからターゲット・インスタンスに転送します。controlfileおよびspfile autobackupもコピーするようにしてください。

```
scp -i ~/.ssh/ociob /u01/backups/* oracle@129.213.28.50:/u01/backups/.
scp -i ~/.ssh/ociob /u01/app/oracle/product/12.2.0.1/db_1/dbs/c-
3107896219-20180327-02 oracle@129.213.28.50:/u01/backups/.
```

3. ソース・データベースで次の問合せを実行し、それらの出力をメモします。

```
SQL> Select dbid from v$database;
SQL> Select member from v$logfile;
```

4. dbcliを使用して、新しいデータベースのためのデータベース・ストレージを作成します。

```
[root@rkbmdb2 ~]# dbcli create-dbstorage --dbname prddb --dataSize 30
```

5. list-dbstorageコマンドを使用して、作成が完了したことを確認します。

```
[root@rkbmdb2 ~]# dbcli list-dbstorages
```

6. dbcliを実行し、Oracle Homeの場所を特定します。

```
root@rkbmdb2 ~]# dbcli list-dbhomes
```

spfileをバックアップからターゲット・ホストにリストアします。このコマンドでは、手順3でソース・データベースから問い合わせたDBIDを使用します。

```
export ORACLE_SID=prddb
rman target /
RMAN> startup force nomount;
RMAN> set dbid 3107896219;
RMAN> set DECRYPTION identified by encryptit;
RMAN> allocate channel c1 device type disk format '/u01/backups/%U' ;
RMAN> RESTORE SPFILE to
'/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/spfileprddb.ora' FROM
'/u01/backups/c-3107896219-20180327-02';
RMAN> shutdown immediate;
```

7. \$ORACLE_HOME/dbsディレクトリにinitprddb.oraという新しいテキスト・ファイルを作成し、次のエントリ（先ほどリストアしたspfileを指すエントリ）を追加します。

```
SPFILE=/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/spfileprddb.ora
```

8. ターゲットの初期化パラメータを変更します。

```
sqlplus / as sysdba
SQL> alter system set db_create_file_dest='+DATA' scope=spfile;
SQL> alter system set db_recovery_file_dest='+RECO' scope=spfile;
SQL> alter system set db_recovery_file_dest_size=4385144832 scope=spfile;
SQL> alter system set
control_files='+DATA/prddb/controlfile/control01.ctl','+RECO/prddb/control
file/control02.ctl' scope=spfile;
shutdown immediate;
startup nomount;
```

9. controlfileをリストアします。このコマンドでは、手順3でソース・データベースから問い合わせたDBIDを使用します。

```
rman target /
RMAN> set dbid 3107896219
RMAN> set DECRYPTION identified by encryptit;
RMAN> SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F';
RMAN> RESTORE CONTROLFILE FROM '/u01/backups/c-3107896219-20180327-02';
RMAN> shutdown immediate;
RMAN> startup mount;
```

これで、先ほどリストアしたcontrolfileがデータベースにマウントされます。

10. データベースをリストアしてリカバリします。

このコマンドでは、手順3で問い合わせたオンラインREDOログ・ファイル名を使用します。リストアとリカバリが完了したら、Oracle Databaseサービス上のデータベースが使用可能な状態になります。

```
rman target /
set DECRYPTION identified by encryptit;
run {
set ARCHIVELOG DESTINATION to '+RECO';
set NEWNAME for database to '+DATA';
allocate channel c1 device type disk format '/u01/backups/%U' ;
SQL "ALTER DATABASE RENAME FILE '/u01/oradata/prddb/redo01.log'
    TO '"+DATA/PRDDB/ONLINELOG/redo01.log'" ";
SQL "ALTER DATABASE RENAME FILE '/u01/oradata/prddb/redo02.log'
    TO '"+DATA/PRDDB/ONLINELOG/redo02.log'" ";
SQL "ALTER DATABASE RENAME FILE '/u01/oradata/prddb/redo03.log'
    TO '"+DATA/PRDDB/ONLINELOG/redo03.log'" ";
restore database;
switch datafile all;
restore archivelog all;
recover database;
alter database open resetlogs;
}
```

リストアとリカバリが完了したら、Oracle Cloud Infrastructure Databaseサービス上のデータベースが使用可能な状態になります。

Oracle RMANには、Amazon EC2インスタンスからOracle Cloud Infrastructure Object Storageサービスに直接バックアップできる機能があります。Object Storageにバックアップすると、次のメリットがあります。

- RMANバックアップを保存するために、ソース・インスタンスでAmazon Elastic Block Storage (EBS) をプロビジョニングする必要がありません。
- Object Storageでは、無制限のストレージが提供されます。

- RMANバックアップ・ファイルを個別の手順で宛先に転送する必要がありません。
- Object StorageはOracle Cloud Infrastructure内のターゲット・データベース環境と同じ環境内にあるので、リストアが迅速です。

Object Storageにバックアップするには、次の手順を実行する必要があります。

1. ソースのAmazon EC2インスタンスで、Oracle Cloudバックアップ・モジュールをインストールします。
2. Oracle Cloudバックアップ・モジュールの構成を行って、バックアップがObject Storageに対して実行されるようにします。
3. SBT_TAPEパラメータを使用してObject Storageをポイントし、RMANバックアップの宛先を設定します。

レプリケーションのためのOracle GoldenGate

Oracle GoldenGateを使用すると、異なるOracleデータベース間でリアルタイムなデータ統合を実行できます。Oracle GoldenGateでは、ソース・データで発生する変更をキャプチャして変換し、ターゲット・データベースにプロパゲートして適用するために必要なプログラムがすべて提供されます。お客様は、Oracle GoldenGateを使用して、ソース・データベースの初期コピーをAmazon RDSからOracle Cloud Infrastructure上のOracle Databaseインスタンスへと移し、実際のカットオーバーが実行されるまで、それらの同期を維持することができます。これにより、新しい移行先データベースへのカット・オーバーを最小限のダウンタイムで達成することができます。

この例では、Oracle GoldenGateがOracle Cloud Infrastructureインスタンスにインストールされ、ハブとして構成されています。このインスタンスは、Oracle GoldenGateの抽出プロセスを使用して、ソース・データベースからすべての変更を収集します。変更はハブ上の証跡ファイルに保存され、Replicatプロセスを使用して、これらの変更がターゲット・データベースに転送、適用されます。

次の手順では、Oracle GoldenGate 12.3.0.1.2をOracle Cloud Infrastructure上のハブ・インスタンスとして使用し、HRスキーマを、12.2.0.1の非CDBデータベースから、Oracle Cloud Infrastructure Databaseサービスのコンテナ・データベースにあるPDBへと移行します。

1. Oracle Cloud Infrastructure ComputeインスタンスにOracle GoldenGateをインストールし、構成します。
 - A. ハブ・インスタンスとして機能するための適切なCPUとメモリーを使用して、Computeインスタンスを作成します。オペレーティング・システムとしてOracle Linux 7.5を選択し、必要な量のブロック・ストレージを作成して、インスタンスにアタッチします。

インスタンスには、レプリケートするトランザクション量に対応できるだけのCPUパワーを搭載し、すべての証跡ファイルを保存できるだけのブロック・ストレージをアタッチする必要があります。

- B. <http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html> から、Oracle GoldenGate 12.3.0.1 for Oracle on Linux x86-64をインストールします。
- C. <http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html>から、Oracle 12cインスタント・クライアントをダウンロードします。**Instant Client Package - Basic**と**Instant Client Package - SQL*Plus**をダウンロードします。
- D. ホストに**oracle**というユーザーを作成し、複雑なパスワードを設定します。

2. Computeインスタンスに、Oracleインスタント・クライアントをインストールします。

- A. Computeインスタンスに、**oracle**としてログインします。
- B. ダウンロードしたインスタント・クライアントのコンテンツを、
/u01/app/oracle/product/12.2.0.1/db_1ディレクトリに解凍します（すべてのファイルを、instantclient_12_2ディレクトリからdb_1に移動します）。

3. ComputeインスタンスにGoldenGateをインストールします。

- A. Computeインスタンスに、**oracle**としてログインします。
- B. 123012_fbo_ggs_Linux_x64_shiphome.zipファイルを、ステージング・ディレクトリ（たとえば、/u01/tmp）に解凍します。
- C. 次のエントリを使用して、レスポンス・ファイル（たとえば、/u01/tmp/gg.rsp）を作成します。

```
oracle.install.responseFileVersion=/oracle/install/rspfmt_ogginstall_r  
esponse_schema_v12_1_2  
INSTALL_OPTION=ORA12c  
SOFTWARE_LOCATION=/u01/app/oracle/product/gg  
START_MANAGER=false  
INVENTORY_LOCATION=/u01/app/oraInventory  
UNIX_GROUP_NAME=oracle
```

- D. ディレクトリを/u01/tmp/fbo_ggs_Linux_x64_shiphome/Disk1に変更します。
- E. 次のコマンドを実行します。

```
runInstaller -silent -nowait -responseFile /u01/tmp/gg.rsp
```

プロンプトが表示されたら、**orainstRoot.sh**を実行します。

- F. **.bash_profile**で、次のパラメータを設定します。

```
export GGHOME=/u01/app/oracle/product/gg  
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/db_1  
export LD_LIBRARY_PATH=$ORACLE_HOME:$LD_LIBRARY_PATH  
export TNS_ADMIN=/u01/app/oracle/product/12.2.0.1/db_1/network/admin  
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$GGHOME
```

4. Oracle GoldenGateハブを構成します。

- A. \$ORACLE_HOME/network/admin/tnsnames.oraファイルにエントリを追加して、Amazon RDSのソース・データベース・インスタンスに接続します。

```
HRcomp =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = rdsdb.chyqlovvrniy.us-east-
2.rds.amazonaws.com) (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = prddb3)
    )
  )

HRDbaaS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.62) (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = prdpdb3.sub04231321440.prdrvcl1.oraclevcn.com)
    )
  )
```

- B. GGSCIを実行し、必要なディレクトリを作成します。

```
cd $GGHOME
./ggsci
GGSCI> create subdirs
GGSCI> edit param mgr

PORT 7865

GGSCI> start mgr
GGSCI> add credentialstore
GGSCI> alter credentialstore add user ggadmin@hrcomp password ggadmin
alias ggadminc
GGSCI> alter credentialstore add user ggadmin@hrdbaas password
WelCome#_12 alias ggadmincl
GGSCI> quit
```

5. AWSにソース・データベースを準備します。

```
sqlplus sys/manager@hrcomp as sysdba
alter database add supplemental log data;
alter system set enable_goldengate_replication=true scope=both;
```

6. ソース・データベースに、Oracle GoldenGateユーザーを作成します。

```
create tablespace ggs_data datafile '/u01/oradata/prddb3/ggs_data01.dbf'
size 1024m autoextend on;
create user ggadmin identified by ggadmin default tablespace ggs_data
temporary tablespace temp;
```


7. ユーザーに必要な権限を付与します。

```
grant connect,resource,create session, alter session to ggadmin;
grant select any dictionary, select any table,create table to ggadmin;
grant alter any table to ggadmin;
grant execute on utl_file to ggadmin;
@marker_setup.sql (Specify ggadmin as the user)
@ddl_setup.sql (Specify ggadmin as the user)
@role_setup.sql (Specify ggadmin as the user)
@ddl_enable
@sequence (Specify ggadmin as the user)
exec dbms_goldengate_auth.grant_admin_privilege('GGADMIN');
grant ggs_ggsuser_role to ggadmin;
@ddl_enable

Log in to the Compute instance in AWS where the source database instance
resides.

sqlplus / as sysdba
shutdown immediate
startup mount
alter system set db_recovery_file_dest_size=20G scope=both;
alter system set db_recovery_file_dest='/u01/orareco' scope=both;
alter database flashback on;
alter database open;
alter database add supplemental log data;
alter database force logging;
grant execute on dbms_logmnr_d to GGADMIN;
grant select on sys.logmnr_buildlog to GGADMIN;
grant execute on utl_file to GGADMIN;
grant exempt access policy to GGADMIN;
```

8. Oracle Cloud Infrastructureのターゲット・データベース・インスタンスを構成します。

```
sqlplus sys/manager@hrdbaas as sysdba
alter database add supplemental log data;
alter system set enable_goldengate_replication=true scope=both; (Run this
as cdb)
create tablespace ggs_data datafile '+DATA' size 1024m autoextend on;
create user ggadmin identified by WelCome#_12 default tablespace ggs_data
temporary tablespace temp;
grant connect,resource,create session, alter session to ggadmin;
grant select any dictionary, select any table,create table to ggadmin;
grant alter any table to ggadmin;
grant execute on utl_file to ggadmin;
@marker_setup.sql (Specify ggadmin as the user)
@ddl_setup.sql (Specify ggadmin as the user) -- Does not work in pdb
@role_setup.sql (Specify ggadmin as the user)
@ddl_enable -- Does not work in pdb
@sequence (Specify ggadmin as the user)
```

```

exec dbms_goldengate_auth.grant_admin_privilege('GGADMIN');
grant ggs_ggsuser_role to ggadmin;
@ddl_enable

Log in to the host where the target database service instance resides.

export ORACLE_UNQNAME=prddb3_iad16p
sqlplus / as sysdba
shutdown immediate
startup mount
alter system set db_recovery_file_dest_size=20G scope=both;
alter system set db_recovery_file_dest='/u01/orareco' scope=both;
alter database flashback on;
alter database open;
alter database add supplemental log data;
alter database force logging;
alter session set container=prdpdb3
grant EXECUTE on dbms_logmnr_d to GGADMIN;
grant SELECT on sys.logmnr_buildlog to GGADMIN;
grant execute on utl_file to GGADMIN;
grant exempt access policy to GGADMIN;

```

- ターゲット・データベースから、GoldenGateの抽出を構成し、開始します。Oracle Cloud Infrastructure Computeインスタンスに、**oracle**ユーザーとしてログインします。

```

cd $GGHOME

./ggsci

GGSCI> dblogin useridalias ggadmin
GGSCI> add schematransdata hrcomp ALLCOLS
GGSCI> edit params extehrc

extract extehrc
exttrail ./dirdat/ee
tranlogoptions IntegratedParams (max_sga_size 256)
discardfile ./dirrpt/silext01.dsc, append megabytes 50
logallsupcols
updaterecordformat compact
reportcount every 2 hours, rate
useridalias ggadmin
table HR.*;

GGSCI> register extract extehrc database
GGSCI> add extract extehrc, integrated tranlog, begin now
GGSCI> add exttrail ./dirdat/ee, extract extehrc
GGSCI> start extract extehrc

```

register extractステップで、システム変更番号（SCN）が表示されます。この番号をメモしてください。

10. ターゲット・データベースをData Pumpインポート用に準備します。Oracle Cloud Infrastructure Databaseサービス・ホスト（ターゲット・データベースがある場所）に、オペレーティング・システム・ユーザー**oracle**としてログインします。

```
sqlplus / as sysdba
SQL> create user hr identified by rimjhimgire default tablespace hrts
temporary tablespace temp;
SQL> GRANT CREATE SESSION, CREATE VIEW, ALTER SESSION, CREATE SEQUENCE TO
hr;
SQL> GRANT CREATE SYNONYM, CREATE DATABASE LINK, RESOURCE , UNLIMITED
TABLESPACE TO hr;
```

11. AWS内のソース・データベースから、HRスキーマをエクスポートします。

```
expdp system/manager schemas=HR directory=dpdir dumpfile=hr.dmp
logfile=dpdir:hr.log flashback_scn=1625157
```

ここで、flashback_scnは、register extractステップで取得したSCNから1を引いたものです。

12. Oracle Cloud Infrastructure内のターゲット・データベースに、HRスキーマをインポートします。

```
impdp
'system/WelCome#_12@prddb3:1521/prdpdb3.sub04231321440.prdrvcl1.oraclevcn.c
om' directory=dpdir dumpfile=hr.dmp schemas=hr table_exists_action=replace
logfile=dpdir:himport.log
```

13. ターゲット・データベースで、Replicatプロセスを開始します。Oracle Cloud Infrastructure Computeインスタンスに、**oracle**ユーザーとしてログインします。

```
cd $GGHOME
./ggsci

GGSCI> dblogin userdalias ggadmind
GGSCI> add schematransdata hr allcols
GGSCI> add replicat rephrd integrated exttrail ./dirdat/ee
GGSCI> edit params rephrd

replicat rephrd
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rephrd01.dsc
DDL INCLUDE ALL
USERIDALIAS ggadmind
REPORTCOUNT EVERY 1 HOURS, RATE
MAP HR.*, TARGET HR.*;

GGSCI> start replicat rephrd
```

GoldenGateは、ソース・データベースで実行されたすべてのトランザクションを、ターゲット・データベースにレプリケートします。Oracle Cloud Infrastructure内のデータベースにカット・オーバーする準備ができたなら、ソース・データベースを停止し、GoldenGateレプリケーションを停止して、ターゲット・データベースでのトランザクションを開始できます。

移行後の手順：透過的データベース暗号化

Oracle Cloud Infrastructureにデプロイされたすべてのデータベースでは、透過的データベース暗号化（TDE）を使用して、保存データを暗号化する必要があります。AWSから移行された表領域については、それらがソースで暗号化されていた場合、TDEを使用するように変換してください。


- Data Pumpなどの論理移行を使用する場合は、データがインポートされる前に、表領域をTDE用に設定することができます。
- RMANバックアップ・リストアなどの物理移行を実行する場合は、2つの方法でTDEへの移行を達成できます。TDEへのオフライン変換を実行するか、（Data Guard環境があれば）TDEへのオンライン変換を実行できます。このプロセスについての詳細は、このドキュメントでは対象外なので、「[高速オンライン変換を使用したOracle Data Guardでの透過的データベース暗号化への変換](#)」ホワイト・ペーパーで確認してください。

結論

OracleデータベースをOracle Cloud Infrastructure Databaseサービスに移行すれば、世界クラスのデータベースを、エンタープライズクラスのクラウド・プラットフォームで実行することができます。お客様は、最善のコスト・パフォーマンスを享受し、管理機能を自動化できるだけでなく、すべてのデータベース機能やオプションを柔軟に使用することができます。このホワイト・ペーパーでは、OracleデータベースをAWSからOracle Cloud Infrastructureに移行するための17種類の移行方法と、それらをどのような場合に使用するかについて説明しました。このホワイト・ペーパーで説明した手順に従うことで、大規模なデータベースを、最小限のダウンタイムで、AWSからOracle Cloud Infrastructureへと移行することができます。

参考資料

- [Oracle Cloud Infrastructureの概要](#)
- [Oracle Cloud Infrastructure Networkingの概要](#)
- [Oracle Cloud Infrastructureのセキュリティ・リストの概要](#)
- [Oracle Cloud Infrastructureのルート表の概要](#)
- [Oracle Cloud Infrastructure Load Balancingの概要](#)
- [Oracle Cloud Infrastructure Databaseサービスの概要](#)

- 
- [Oracle Cloud Infrastructure Computeサービスの概要](#)
 - [Oracle Cloud Infrastructure Virtual Cloud Network Overview and Deployment Guide](#)
 - [Amazon RDS for Oracle Database](#)



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US



blogs.oracle.com/oracle



facebook.com/oracle



twitter.com/oracle



oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2018, 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。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社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。本文書はオラクル社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はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。0518

Oracle DatabaseのAmazon Web ServicesからOracle Cloud Infrastructure Databaseへの移行
2018年5月

著者：Adeel Amin、Nikhil Marrapu、Rajeev Ramdas、Shan Gupta



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