



Oracle GoldenGate Hub構成での Oracle Cloudへの移行

2020年4月6日

Copyright © 2020, Oracle and/or its affiliates

機密情報：公開ドキュメント

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。

本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。

本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

免責事項	1
本書の目的と対象読者	4
はじめに	4
構成の概要	5
Oracle GoldenGate	5
Oracle GoldenGate Hub	6
Nginxリバース・プロキシ・サーバー	7
ターゲット・データベースのインスタンス化	8
このホワイト・ペーパー全体を通じて使用されるネーミング規則	8
構成の前提条件	9
Oracle Cloudネットワークを構成する	9
Oracle GoldenGateのソース・データベース・サポートを評価する	10
ソース・データベースを構成する	11
1. Oracle Netの接続性を構成する	11
2. Oracle GoldenGateデータベース管理者アカウントを作成する	13
ターゲット・データベースを構成する	14
ターゲット・クラウド・データベースを作成する	14
データベース初期化パラメータを設定する	14
Oracle GoldenGate管理者データベース・アカウントを作成する	14
Oracle GoldenGate Hubの構成	15
OCI Marketplaceのインストール	15
ネットワーク設定	15
インスタンス設定	16
Oracle GoldenGateのデプロイメントを作成する	17
Oracle GoldenGateのデプロイメントの開始と停止	18
Oracle GoldenGate Hubの受信ポートを開く	18
Oracle GoldenGateの構成	21
ソース側のGoldenGate資格証明を作成する	21
Oracle Net構成ファイルを準備する	21
Oracle GoldenGateデータベースの資格証明を作成する	22
ソース・データベース上でハートビート表を作成する	24
インスタンス化のためにソース・データベースのスキーマを準備する	25
Oracle GoldenGate Extractプロセスを作成および起動する	26
AUTOSTARTタスクを作成する	29
ソース・データベースの長時間実行トランザクションを監視する	31
ターゲット・データベースのインスタンス化	31
Oracle Data Pump (mv2oci/mv2adb)	31
Oracle RMAN複製データベース	32
Oracle GoldenGateの構成を完了する	32
ターゲット・データベースのOracle GoldenGate資格証明を作成する	32
ATP-Dターゲット・データベース	32

ExaCSターゲット・データベース	33
ターゲット・データベースのOracle GoldenGateチェックポイント表を作成する	33
ターゲット・データベースにハートビート表を作成する	35
Oracle GoldenGate Replicatを作成する	35
Oracle RMANのインスタンス化のためにReplicatを作成する	35
Data Pumpのインスタンス化のためにReplicatを作成する	36
AUTOSTARTタスクをReplicatに割り当てる	37
Oracle GoldenGateのレプリケーションを監視する	37
移行済みデータベースのテスト	38
レプリケーションを一時停止する	39
保証付きリストア・ポイント (GRP) を作成する	39
ターゲット・データベースを検証する	40
ターゲット・データベースをフラッシュバックする	40
保証付きリストア・ポイントを削除する	41
レプリケーションを再開する	41
移行済みデータベースにスイッチオーバーする	41
Oracle GoldenGateのレプリケーション・ラグが許容できる程度に小さいかどうかを判別する	41
ソース・データベースでトランザクションが開始するのを停止する	41
Extractで未処理のトランザクションが完了したことを確認する	41
Extractを停止する	42
Replicatがすべての証跡ファイル・データを適用するまで待機する	42
ターゲット・データベースにスイッチオーバーする	42
Oracle GoldenGateの構成の削除	43
Oracle GoldenGateプロセスを削除する	43
AUTOSTARTタスクを削除する	43
ターゲット・データベースのハートビート表を削除する	43
チェックポイント表を削除する	44
Oacle GoldenGate資格証明を削除する	44
Oracle GoldenGate Hubを削除する	44

本書の目的と対象読者

本書の対象読者は、OracleデータベースからOracle Cloudへの移行について調査するか、またはそれを担当する個人です。

本書の編成および内容は、Oracleデータベースの概念に精通していることを前提にしていますが、Oracle GoldenGateに良く通じている個人と、Oracle GoldenGateとレプリケーションの概念に新しく接する個人の両方にとどても適しています。

本書の主な意図は、Oracle GoldenGateを活用して、最小限の停止時間でOracleデータベースをOracle Cloudに移行するための最適な方法について教育し、指針を示すことです。これには、概念教育、ベスト・プラクティス、読者を導くことのできるステップ・バイ・ステップの例が含まれます。

はじめに

このOracle GoldenGate Maximum Availability Architecture (MAA) 論理的移行ソリューションでは、以下の特徴を持つソース・データベースをセットアップおよび移行するための手順をステップ・バイ・ステップで示します。

- » Oracle Database 11gR2以降のリリース
- » 任意のオペレーティング・システム・プラットフォーム (AIX、HP UX、Linux、Solaris、Windows)
- » 単一テナントまたはマルチテナントのデータベース
- » 非暗号化または暗号化データベース

以下に存在するクラウド認定暗号化データベースに移行することができます。

- » Oracle Cloud Infrastructure (OCI) の Exadata Cloud Service (ExaCS)
- » Autonomous Database Transaction Processingの専用デプロイメント (ATP-D)

このソリューションでは、Oracle Cloud Marketplace上のOracle GoldenGateを使用して、以下の利点をもたらすOracle GoldenGate Hub

構成を作成します。

- » 最小限からゼロの停止時間
- » バージョンをまたぐデータベースのサポート
- » クロス・プラットフォームまたはクロス・エンディアンのサポート (IBM AIXからLinuxへの移行など)
- » 単一テナントまたはマルチテナントに依存しない
- » ソースおよびターゲット・データベース・システムへのOracle GoldenGateリソースの影響を軽減
- » 複数のOracle Databaseの移行に使用可能

この移行ソリューションには、Oracle GoldenGateの利点と柔軟性のすべてが統合されています。ただし、一部のデータタイプおよび操作上の制限については、ホワイト・ペーパーに記載されているステップを行う前に考慮する必要があります。詳細については、[「Oracle GoldenGateのソース・データベース・サポートを評価する」](#)を参照してください。

以下のダイアグラムに、このホワイト・ペーパーで説明する移行フローの概要を示します。

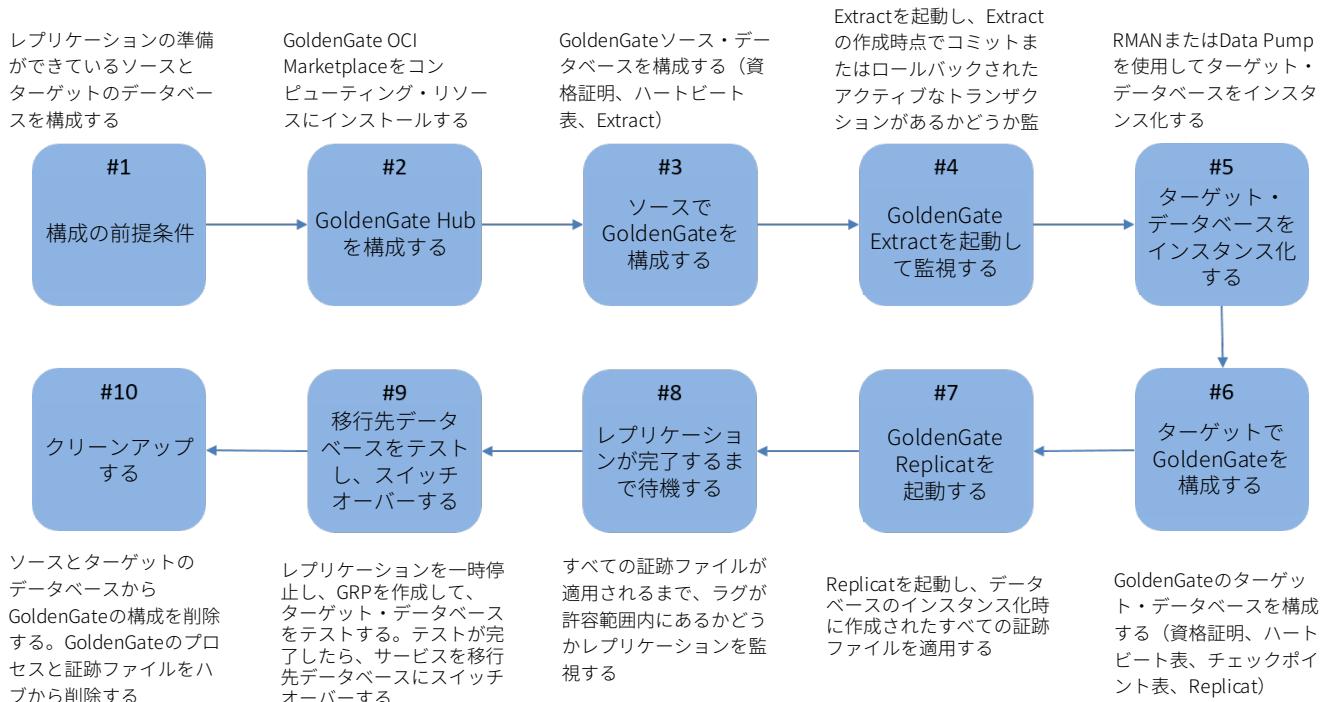


図1：移行フローの概要

構成の概要

どのような移行プロジェクトであっても、着手する前に、利用する移行ソリューションの全体的なアーキテクチャについて理解することが不可欠です。これは、ベスト・プラクティスが適用されており、そのソリューションが理想的な構成でデプロイされていることを確認するためです。このセクションでは、Oracle GoldenGateについて簡単に説明し、加えて、Oracle GoldenGate Microservices ArchitectureとそのOracle GoldenGate Hubソリューション内での構成方法について述べます。

Oracle GoldenGate

Oracle GoldenGateは、同種システムと異種システムの間で、リアルタイムのログベース・チェンジ・データ・キャプチャおよび配信機能を提供します。このテクノロジーを使用すると、低負荷で費用対効果に優れたりアルタイム・データ統合と、継続的可用性ソリューションを実現できます。

Oracle GoldenGateは、トランザクションの整合性を維持し、既存インフラストラクチャでのオーバーヘッドを最小限に抑えながら、コミットされたトランザクションからデータをレプリケートします。このアーキテクチャでは、1対多、多対多、カスケード、双方向などのさまざまなデータ・レプリケーション・トポロジがサポートされています。多様なユースケースには、リアルタイム・ビジネス・インテリジェンス、問合せのオフロード、停止時間なしのアップグレードと移行、アクティブ/アクティブ・データベースを使用したデータ分散、データ同期、高可用性が含まれます。

Oracle GoldenGate Microservices Architectureは、Oracle GoldenGate環境の一部としてREST対応サービスを提供する新しい管理アーキテクチャとして、Oracle GoldenGate Release 12.3で導入されました。REST対応サービスを使用すると、HTML5 Webページ、コマンドライン・インターフェース、APIを介してリモートでの構成、管理、監視を行うことができます。図2に、Oracle GoldenGate Microservices Architectureを示します。

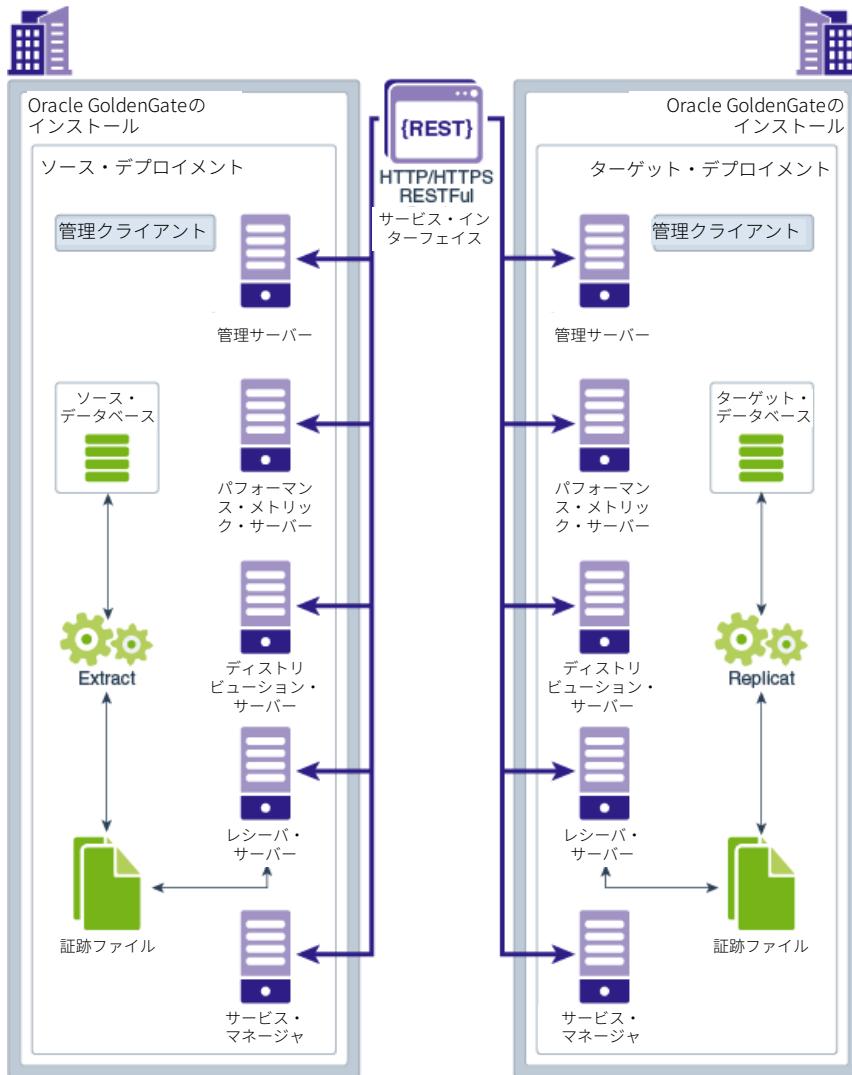


図2：Oracle GoldenGate Microservices Architecture

Oracle GoldenGate Microservices Architectureについて詳しくは、Oracle GoldenGateのドキュメントを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/using/getting-started-oracle-goldengate.html#GUID-61088509-F951-4737-AE06-29DAEAD01C0C>

Oracle GoldenGate Hub

Oracle GoldenGate Hubは、運用対象のデータベースとは異なるホストにOracle GoldenGateソフトウェアを配置するアーキテクチャの概念です。Oracle GoldenGate Hubは、同じOracle Cloud Infrastructure (OCI) リージョンか、可能ならターゲット・クラウド・データベースと同じOCI可用性ドメインに配置する必要があります。Oracle GoldenGate Hubは、ネットワーク待機時間が決して2~3ミリ秒を超えないようにするため、ターゲット・データベースの近接ネットワーク内に配置する必要があります。GoldenGateソフトウェアとプロセスは、図3に示すように別個のサーバーで実行されます。このアーキテクチャの利点の1つは、GoldenGateのリソース使用量の大半がソースとターゲットのデータベース・サーバーから分離されることです。もう一つの利点は、GoldenGateの構成を单一サーバーから管理および監視することができ、各データベース・サーバー上にインストールされている別々のGoldenGateにアクセスする必要がないことです。

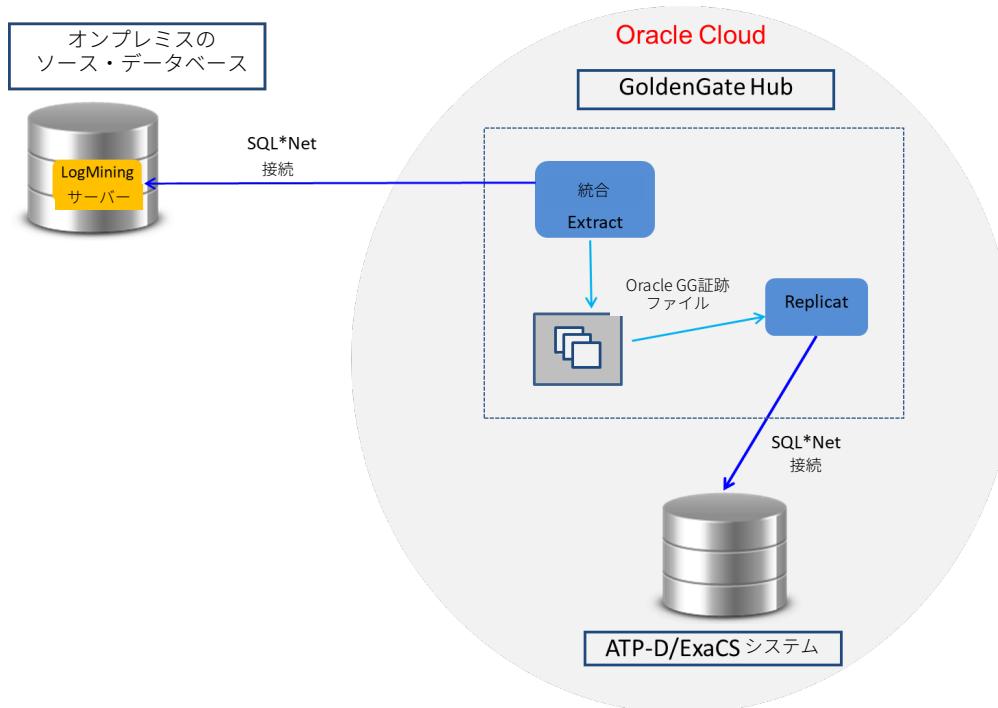


図3：Oracle Cloud内のOracle GoldenGate Hubのアーキテクチャ

Oracle GoldenGate Hubソフトウェアとデプロイメント構成は、Oracle Cloud MarketplaceからOracle GoldenGate Microservicesを使用してインストールおよび構成されます。Oracle Cloud Marketplaceには最新リリースのOracle GoldenGate Microservicesが、Oracle Databaseの全サポート・バージョンのOracle Databaseクライアント・ソフトウェアと一緒に組み込まれています。ソースとターゲットの両方のOracle Databaseで選択されているデータベース・バージョンに基づき、自動化されたGoldenGateのインストールの一部として、GoldenGateのデプロイメントが2つまで作成されます。

Oracle GoldenGate Hubのインストールについて詳しくは、後続のステップ（「[Oracle GoldenGate Hubの構成](#)」）で説明します。

Oracle Cloud MarketplaceとそのGoldenGate Hubの自動化は、Oracle Cloud Infrastructure (OCI) でのみ使用可能です。Oracle Cloud MarketplaceのOracle GoldenGate Microservicesについて詳しくは、次の場所にあるドキュメント「[Using Oracle GoldenGate on Oracle Cloud Marketplace](#)」を参照してください。<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oggmp/oracle-goldengate-microservices-oracle.html#GUID-06FC1224-DBC0-4145-A485-4C273F3199C2>

Nginxリバース・プロキシ・サーバー

Oracle GoldenGateリバース・プロキシ機能を使用すると、GoldenGateデプロイメントに関連付けられたすべてのGoldenGate Microservicesに対して单一接点を設けることができます。リバース・プロキシがない場合、GoldenGateデプロイメントのマイクロサービスには、ホスト名またはIPアドレス、および個別のポート番号（各サービスに1つずつ）からなるURLを使って接続します。たとえば、サービス・マネージャに接続するには、<http://gghub.example.com:9100>を使用できます。管理サーバーは

<http://gghub.example.com:9101>、デプロイメントNo. 2の管理サーバーは<https://gghub.example.com:9111>という具合になります。

リバース・プロキシがある場合、マイクロサービスはデプロイメント名に置き換えられているため、マイクロサービスへの接続にポート番号は不要です。前の例を使って、サービス・マネージャに接続するには<https://gghub.example.com>のURLを使用します。デプロイメントNo. 1（名前：ソース）の管理サーバーの場合は<https://gghub.example.com/Source/adminsrvr>を使用し、デプロイメントNo. 2（名前：ターゲット）の管理サーバーの場合は<https://gghub.example.com/Target/adminsrvr>となります。

マイクロサービスに容易にアクセスし、セキュリティと管理性を強化するために、Oracle GoldenGateリバース・プロキシが推奨されており、デフォルトでは、Oracle Cloud MarketplaceからGoldenGateで使用されます。

ターゲット・データベースのインスタンス化

Oracle GoldenGateを使用してソース・データベースとターゲット・データベースの間でデータをレプリケーションする場合は、その前に、Replicatプロセスが変更を適用したときにデータの競合が生じないようにするために、一貫した状態でレプリケートされるオブジェクトが両方のデータベースに存在するようにする必要があります。ターゲット・データベースのインスタンス化の方法は、ソースおよびターゲットのOracleデータベース・バージョン、データベース・サーバーのオペレーティング・システムのエンディアンネス、およびデータベース構造のおもな相違点すべて（暗号化またはマルチテナント以外のデータベースからの暗号化またはマルチテナント・データベースへの移行など）に応じて決まります。

このホワイト・ペーパーでは、次のインスタンス化方法について説明します。

- Data Pumpエクスポート/インポート** - このインスタンス化方法が必要なのは、移行元と移行先のデータベース・バージョンが異なる場合、プラットフォーム・エンディアン形式が異なる場合、またはデータベース構造に変更がある場合（単一テナントからマルチテナント・アーキテクチャまたは暗号化データベースへの移行など）です。
これは、オンプレミスのデータベースからATP-Dへの移行でサポートされる唯一の方法です、ExaCSへの移行ではmv2oci（My Oracle Support Note [2514026.1](#)）を使用し、ATP-Dへの移行ではmv2adb（My Oracle Support Note [2463574.1](#)）を使用します。
これら両方のツールでは、Data Pumpを使用してデータをターゲット・データベースに移行します。
- Oracle RMANを使用した物理データベースのインスタンス化** - ソースおよび移行先データベースのバージョンとプラットフォーム（エンディアン形式）が同じ場合は、Oracle RMANによってソース・データベースをインスタンス化することができます。このインスタンス化方法は、ターゲット・データベースをインスタンス化する最速の方法です。
この方法は、ATP-Dへの移行では使用できません。推奨されるインスタンス化方法について詳しくは、データベース・バージョンに基づいて、『Creating a Physical Standby Database in an 11.2, 12.1, 12.2 or later environment』（ドキュメントID [2275154.1](#)）を参照してください。

上記のいずれの場合でも、ソース・データベースはインスタンス化の間中オンライン状態です。

これら2種類のインスタンス化方法について詳しくは、「ターゲット・データベースのインスタンス化」セクションで説明します。

移行期間中は、高速データベース・レプリケーションに最適な環境を実現するには、特定のデータベース操作を行わないことをお奨めします。データベースの移行中は、次の機能を使用しないでください。

- データ定義言語 (DDL)** - DDLがレプリケーションされる間、Replicatは、同じオブジェクト上でDMLとDDLとの間にロック問題が発生しないようにするために、データをシリализします。
- 大規模パッチDML** - 数百万行に影響する単一のトランザクションのような大規模パッチ操作を実行すると、レプリケーション速度が遅くなる可能性があります。

このホワイト・ペーパー全体を通じて使用されるネーミング規則

このホワイト・ペーパー全体を通じて、2つのデプロイメント（SOURCEおよびTARGET）を使ってOracle GoldenGate構成を管理するREST APIエンドポイントの例を示します。自動データベース移行スクリプトにコマンドを容易に組み込めるように、REST APIを使用します。このスクリプトは、curlとpythonがインストールされたGoldenGate Hubにアクセスできる任意のサーバーからローカルまたはリモートに実行できます。

または、Admin Clientコマンドを使って、GoldenGate Hubを管理することもできます。Admin Clientは、GoldenGateレプリケーションの作成と管理に使われるスタンドアロンのコマンドライン・インターフェースです。Admin Clientのコマンドについては、『Command Line Interface Reference for Oracle GoldenGate』を参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/gclir/index.html>

REST APIの引数

このホワイト・ペーパー全体を通じて使用する、GoldenGate REST APIで指定されるおもな引数の例を以下に示します。

```
$ curl -s -K access.cfg https://<GG Hub>/<Deployment Name>/adminsvr/services/v2/credentials/goldengate -XGET | python -m json.tool
```

» **access.cfg** - GoldenGate管理者アカウントの名前とパスワードがコマンドラインに表示されないように、curlによって読み取られる構成ファイルにユーザー名とパスワードを含めることをお奨めします。たとえば、次のように指定します。

```
user = "oggadmin:password"
```

» **GG Hub** - GoldenGate Hubサーバーのホスト名またはIPアドレス。たとえば、gghub-serverのように指定します。

» **Deployment Name** - Oracle GoldenGateデプロイメントの名前です。たとえば、SOURCEまたはTARGETのように指定します。

たとえば、次のように指定します。

```
$ curl -s -K access.cfg https://gghub-server/SOURCE/adminsvr/services/v2/credentials/goldengate -XGET | python -m json.tool
```

REST呼び出し

REST呼び出しは、Oracle GoldenGate Hub、またはHTTPSプロトコルを介してOracle GoldenGate Hubにアクセス可能な任意のマシンから、オンプレミスで、またはOCIで行うことができます。

Extract名とReplicat名

このホワイト・ペーパーで使用されるExtract名の例はEXT1で、ReplicatはREP1となります。

構成の前提条件

Oracle GoldenGate Hubの構成前に、以下のセクションで説明する前提条件を満たしてください。

Oracle Cloudネットワークを構成する

GoldenGateリモートExtractがハブ上で実行され、ソースのオンプレミス・データベースからデータが抽出される場合は、システム間でアクセスできるようにするために、追加のネットワーク構成が必要になります。

クラウド・ネットワークのGoldenGate Hubをオンプレミスのデータベースに接続する場合には、3つのオプションがあります。

IPSec VPN

Internet Protocol Security (IPSecまたはIP Security) は、パケットが送信元から宛先に送られる前にIPトライフィックを暗号化するプロトコル・スイートです。OCIでのIPSecの概要については、次のページを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Network/Tasks/overviewIPsec.htm>

FastConnect

OCI FastConnectを使用すると、データセンターとOCIの間に専用のプライベート接続を作成できます。FastConnectでは、より高い帯域幅のオプションが用意されており、インターネットベースの接続に比べて、より信頼性の高い一貫したネットワーク接続を体験できます。FastConnectについて詳しくは、次のページを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Network/Concepts/fastconnectoverview.htm>

インターネット・ゲートウェイ

OCIとオンプレミス・システムの間の接続は、パブリック・インターネットに加えて、インターネット・ゲートウェイを使用して確立できます。インターネット・ゲートウェイはオプションの仮想ルーターで、Oracle Virtual Cloud Network (VCN) の端をインターネットと接続します。

このゲートウェイを使用するには、接続の両端のホストにルーティング用のパブリックIPアドレスが指定されている必要があります。Oracle GoldenGate HubのVCNがインターネット・ゲートウェイを備えていない場合は、自分で追加する必要があります。インターネット・ゲートウェイの作成について詳しくは、次のページを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Network/Tasks/managingIGs.htm>

GoldenGate Hubからオンプレミス・データベースに接続するため、必ずセキュリティ・リストの受信ルールと送信ルールを編集してください。詳しくは次のページを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Network/Concepts/securitylists.htm>

このホワイト・ペーパーのこれ以降の部分では、パブリック・インターネット接続を前提に説明を続けます。

Oracle GoldenGateのソース・データベース・サポートを評価する

以下に詳述するように、Oracle GoldenGateには、ソース・データベースに対していくつかの要件があります。

データベース・パッチ要件

最新のバンドル・パッチとPSU (Patch Set Update) をソース・データベースとターゲット・データベース双方に適用することがベスト・プラクティスです。完全な推奨パッチ・リストは、11.2.0.4以上のリリースのOracle Databaseが対象のMy Oracle Support Note [2193391.1](#)に掲載されています。

ソース・データベースとGoldenGate Hub間のネットワーク・ラウンドトリップの待機時間が8ミリ秒を超過する場合、および最新のデータベース・バンドル・パッチまたはPS/CPU (Critical Patch Update) に含まれていない場合は、バグ28849751用のパッチもソース・データベースに適用することをお奨めします。パッチセットのバージョンについて詳しくは、My Oracle Support Note [28849751.8](#) を参照してください。

データタイプのサポート

ソース・データベースでは、ディクショナリ・ビューDBA_GOLDENGATE_SUPPORT_MODEを使用して、データタイプ・レプリケーションの制約のためにOracle GoldenGateによる抽出に完全に対応しないオブジェクトがあるかどうかを判別してください。

```
SQL> SELECT owner, object_name FROM DBA_GOLDENGATE_SUPPORT_MODE
WHERE support_mode NOT IN ('FULL','ID_KEY');
```

上記の問合せで挙げられている表はすべて、GoldenGate ExtractのパラメータTABLEEXCLUDE owner.object_nameを使用してキャプチャから除外する必要があります。これらのオブジェクトは、データベース移行プロセスの最後にターゲット・データベースに手動でコピーする必要があります。

統合Extractに対応していないデータタイプのリストについては、Oracle GoldenGateのドキュメントを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/1-understanding-whats-supported.html#GUID-110CD372-2F7E-4262-B8D2-DC0A80422806>

GoldenGateのTABLEEXCLUDEパラメータの詳細については、次のページを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/reference/index.html>

行の一意性

Oracle GoldenGateが、レプリケートした更新と削除の正しいターゲット行を特定するには、ソース表とターゲット表に一意の行IDが必要です。

これは通常、主キーまたは一意のキー索引で処理されます。そのようなキーを持たないことが判明した表がある場合、GoldenGateは、許容可能なすべての列を含んだ疑似キーを表内に作成する必要があります。バーチャル・カラム、UDT、関数ベースのカラム、拡張 (32K) VARCHAR2/NVARCHAR2カラム、Oracle GoldenGateユーザーがOracle GoldenGate構成から明示的に除外するカラムは除外されます。

ソース・データベースがバージョン12g Release 2以上である場合は、データ・ディクショナリ・ビューのDBA_GOLDENGATE_NOT_UNIQUEを使って、主キーまたは非NULLの一意の列を持たないすべての表を特定します。

常に一意の値を含んだ列が表にある場合は、代替キーを定義できます。この代替キーは、KEYCOLS句をExtractのTABLEパラメータとReplicatのMAPパラメータに含めることで定義します。Oracle GoldenGateが見つけた既存の主キーまたは一意のキーは、指定したキーによってオーバーライドされます。

ソース表で行の一意性を確保するための詳細については、Oracle GoldenGateのドキュメントを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/additional-oracle-goldengate-configuration-considerations.html#GUID-644099C5-8950-496C-8592-446FB1566AFD>

ソース・データベースを構成する

ソース・データベースの前提条件は、MAAのホワイト・ペーパー『Oracle GoldenGate Performance Best Practices』(<https://www.oracle.com/technetwork/database/availability/maa-gg-performance-1969630.pdf>) の「Configuring the Source Database」の項に記載されています。

おもな前提条件は次のとおりです。

- » データベースでARCHIVELOGモードを有効にする。
- » GoldenGate Extractプロセスによってすべての変更がREDO内で見つかるように、データベース強制ロギングを有効にする。
- » データベースの最小サブリメンタル・ロギングを有効にする。レプリケートされたオブジェクト用に追加のスキーマ・レベルのサブリメンタル・ロギングも必要です。
- » 初期化パラメータSTREAMS_POOL_SIZEでストリート・プールを構成する。
- » 初期化パラメータのENABLE_GOLDENGATE_REPLICATIONを有効にすることで、GoldenGateレプリケーションを有効にする。
- » 統合Extractのパフォーマンス分析用にUTL_SPADV/UTL_RPADVパッケージをインストールする。

また、Oracle GoldenGateのデータベース管理者アカウントが存在しない場合は、次の手順に従ってOracle Netの接続性を構成し、Oracle GoldenGateのデータベース管理者アカウントを作成する必要があります。

1. Oracle Netの接続性を構成する

ハブで稼働するリモートGoldenGate Extractのパフォーマンスを最大限に引き出せるように、Oracle Netの接続性を最適化する必要があります。それには、別個のOracle Net Listenerをソース・データベース・ホスト上に作成します。このOracle Net Listenerは、現在の本番リスナー・サービスへの干渉をなくす役目も果たします。

リモートGoldenGate Extractのパフォーマンスの改善のため、より大きいサイズのセッション・データ・ユニット (SDU) を使用するようソース・データベース・リスナーを設定することをお奨めします。リスナーが最大限のSDUサイズを設定していても、クライアントによってリクエストされた場合は、最小サイズが使用されます。たとえば、リスナーがSDUを2 MBに設定したが、クライアントがデフォルトの8 KBのSDUをリクエストした場合、データベースへの接続では、8 KBのSDUサイズが使用されます。Oracle Database 11g Release 2のSDUサイズの最大値は64 KB (65536バイト) で、それより後のデータベース・リリースの最大値は2 MB (2097152バイト) です。

ソース・データベースへの接続には、Oracle NetまたはSecure Sockets Layer (SSL) の暗号化を使用することをお奨めします。Oracle NetによるSSLについて詳しくは、『Oracle Database Security Guide』を参照してください。

<https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/configuring-secure-sockets-layer-authentication.html#GUID-6AD89576-526F-4D6B-A539-ADF4B840819F>

専用リスナーを作成する

データベース移行のためだけに別個のリスナーを作成することをお奨めします。

個別のTNS_ADMINディレクトリをソース・データベース・ホストに作成することで、現在の移行リスナーと新しく構成した移行リスナーが確実に分離されます。この新しいTNS_ADMINディレクトリには、sqlnet.ora/パラメータ・ファイルとlistener.ora/パラメータ・ファイルが格納されます。

ソース・データベースがOracle Real Application Clusters (RAC) 用に有効化されている場合は、そのデータベースのインスタンスが実行されているすべてのクラスタで移行リスナーを構成します。これにより、移行中に1つのインスタンスが停止したとしても、稼働し続けているインスタンスにサービスを移行できます。SSLを使用する場合は、すべてのOracle RACノードでSSLウォレットを構成してください。

次の例のsqlnet.oraファイルとlistener.oraファイルにはSSLが構成され、Oracle Database 12g Release 1以上の場合、SDUサイズが大きくなります。

sqlnet.oraファイルの例

```
SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS = TRUE
SQLNET.WALLET_OVERRIDE = FALSE SQLNET.EXPIRE_TIME =
10
WALLET_LOCATION = (SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u01/oracle/tcps_wallets)))
SSL_VERSION = 1.2

# Parameters required for Net encryption if not using SSL Authentication, replacing # the above parameters:
# SQLNET.ENCRYPTION_SERVER = accepted
# SQLNET.ENCRYPTION_TYPES_SERVER= (AES256)
DEFAULT_SDU_SIZE = 2097152
```

listener.oraファイルの例

```
Migration =
(DESCRIPTION_LIST =
(DESCRIPTION = (SDU
= 2097152)
(ADDRESS = (PROTOCOL = TCPS)(HOST = <source database host>)(PORT = 2484))
)
)

SID_LIST_Migration =
(SID_LIST =
(SID_DESC =
(SDU = 2097152)
(SID_NAME = <ORACLE_SID>) (ORACLE_HOME
=<ORACLE_HOME>)
)
)
```

環境変数を設定し、リスナーを起動する

移行リスナーを起動および停止し、そのステータスを取得するには、次の環境変数が設定されていることを確認してから、start、stop、statusコマンドを実行します。

```
export ORACLE_HOME=<oracle home directory>
export PATH=$PATH:$ORACLE_HOME/bin
export TNS_ADMIN=<TNS admin directory>
```

リスナーを起動します。

```
$ lsnrctl start Migration
```

リスナーを停止します。

```
$ lsnrctl stop Migration
```

リスナーの現在ステータスを取得します。

```
$ lsnrctl status Migration
```

2. Oracle GoldenGateデータベース管理者アカウントを作成する

ソース・データベースが現在、GoldenGate構成の一部である場合、Oracle GoldenGate管理者アカウントはすでに存在している可能性があります。GoldenGate管理者ユーザーが存在しない場合は、作成する必要があります。推奨されるユーザー名は、単一テナントとプラガブル・データベース (PDB) の場合はGGADMIN、コンテナ・データベース (CDB) の場合はC##GGADMINです。ユーザーがすでに存在する場合、権限が正しく付与されていることを確認し、まだの場合は付与する必要があります。

以下の手順に従い、GoldenGate管理者アカウントが存在することを検証し、アカウントが必要な場合は作成してください。

i. Oracle GoldenGate管理者アカウントが存在するかどうかを確認する

Oracle Multitenantを使用している場合は、以下の例のような文を使って、すべてのPDBも確認します。

```
SQL> SELECT name, username
      FROM cdb_goldengate_privileges a, v$pdbs b
      WHERE a.con_id = b.con_id
UNION
SELECT decode(a.con_id,1,'CDB ROOT'), username
      FROM cdb_goldengate_privileges a, v$pdbs b
      WHERE a.con_id=1;
```

単一テナント・データベースの場合は、次の例を使用します。

```
SQL> SELECT username FROM dba_goldengate_privileges;
```

データベース上でGoldenGateを構成したことがない場合、これらの問合せをしても行は何も返されません。

GoldenGate管理者ユーザーがすでに存在する場合は、データベース移行GoldenGate構成に使用します。以下に示す権限が現在のGoldenGate管理者アカウントに付与されていることを確認します。

このホワイト・ペーパー全体を通じて、データベースのGoldenGate管理者アカウントの名前は常にGGADMINになります。

ii. 新しいGoldenGate管理者アカウントを作成する

次の例を使って、新しいGoldenGate管理者アカウントを作成します。

注：ソース・データベースがPDBである場合は、レプリケート対象のすべてのPDB上でこのアカウントを作成する必要があります。

```
SQL> create user ggadmin identified by <password>
      default tablespace users temporary tablespace temp;
SQL> grant connect, resource to ggadmin;
SQL> grant select any dictionary to ggadmin;
SQL> grant create view to ggadmin;
SQL> grant execute on dbms_lock to ggadmin;
SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('ggadmin');
```

ソース・データベースがPDBである場合は、CDBで別個のアカウントを作成する必要があります。

```
SQL> create user c##ggadmin identified by <password>
      default tablespace users temporary tablespace temp;
SQL> grant connect, resource to c##ggadmin;
SQL> grant select any dictionary to ggadmin;
SQL> grant create view to c##ggadmin;
```

```
SQL> grant execute on dbms_lock to c##ggadmin;
SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('c##ggadmin',container=>'all');
```

注：単一PDBからデータをレプリケートすることだけに关心がある場合は、「all」をPDB名に置き換えてください。

ターゲット・データベースを構成する

mv2ociまたはmv2adbを内蔵するOracle Data Pumpを使用してターゲット・データベースをインスタンス化する場合は、前提条件として次の手順を実行する必要があります。

ターゲット・クラウド・データベースを作成する

最初に空のターゲット・データベースを作成する必要があります。

ATP-Dに移行する場合は、Autonomous Databaseの作成に関するドキュメントを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Database/Tasks/adbcreating.htm>

ExaCSに移行する場合は、「Creating Exadata DB Systems」を参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Database/Tasks/exacreatingDBsystem.htm>

System表領域、UNDO表領域、一時表領域、オンラインREDOログのサイズを、少なくともソース・データベースと同じにしておいてください。

データベース初期化パラメータを設定する

GoldenGate Replicatで変更をターゲット・データベースに適用できるようにするには、以下のように

ENABLE_GOLDENGATE_REPLICATIONインスタンス化パラメータを設定する必要があります。

```
SQL> ALTER SYSTEM SET enable_goldengate_replication=TRUE scope=both;
```

このパラメータはCDBで設定する必要があります。

Oracle GoldenGate管理者データベース・アカウントを作成する

GoldenGate管理者アカウントは、ATP-Dデータベースの作成の一環として作成されますが、次のコマンドを使用してロック解除する必要があります。

```
SQL> ALTER USER ggadmin IDENTIFIED BY <password> ACCOUNT UNLOCK;
```

ExaCSの場合は、ソース・データベースの移行先となるPDBでGoldenGate管理者アカウントを作成します。

GoldenGate管理者アカウントを次のコマンドで作成します。

```
SQL> create user ggadmin identified by <password>
      default tablespace users temporary tablespace temp;
SQL> grant connect, resource, dba to ggadmin;
SQL> grant select any dictionary to ggadmin;
SQL> grant create view to ggadmin;
SQL> grant execute on dbms_lock to ggadmin;
SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('ggadmin');
```

注：DDLとシーケンスのサポートには、DBAロールが必要です。DDLまたはシーケンスをレプリケートしない場合、DBAロールは必要ありません。

Oracle GoldenGate Hub構成

Oracle GoldenGateのOCI Marketplaceの搭載機能を使用し、次のソフトウェアによってGoldenGate Hubを構成します。

- » Oracle Databaseバージョン11gR2、12g、18c、19c対応のOracle Clientソフトウェア
- » Oracle Databaseバージョン11gR2、12g、18c、19c対応のOracle GoldenGateソフトウェア
- » Oracle GoldenGate Microservicesリバース・プロキシによって使用されるNGINXリバース・プロキシ・サーバー
- » 1つか2つのGoldenGateデプロイメントを作成する（ソースおよびターゲット・データベース用に1つずつ）

OCI Marketplaceを使用してOracle GoldenGate Microservicesをデプロイするには、次の前提条件を満たす必要があります。

- » Oracle Cloudアカウント
- » 割り当てられたOracle Cloud Tenantへのアクセス権
- » Oracle Cloud Tenant内のコンピュート・ノード・リソース
- » ローカルSSH/RSA鍵

OCI MarketplaceからのOracle GoldenGateのインストール方法について詳しくは、次のページを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oggmp/deploying-oracle-goldengate-microservices-oracle-cloud-marketplace.html#GUID-EF6680D7-7571-41E7-BF2D-831BD79BCC15>

OCI Marketplaceのインストール

OCI MarketplaceからOracle GoldenGateをインストールする場合は、以下のセクションで詳しく述べられている設定を使用します。

ネットワーク設定

”CREATE A NEW NETWORK”は選択しないください。代わりに、以下に示すように、既存のVirtual Cloud Network (VCN) と、ATP-DまたはExaCSターゲット環境に適合するサブネットを選択してください。

それには、Virtual Cloud Network (VCN) をあらかじめ作成しておく必要があります。

Network Settings

CREATE NEW NETWORK
Use this field to indicate whether you want to create new network resources or use existing ones

NETWORK COMPARTMENT OPTIONAL

MAA_Testing

Compartiment for new or existing network resources

VCN

MAA VCN

Existing VCN to use for new instance if not creating a new network

SUBNET (i)

AD3 MAA Public Subnet

Existing Subnet to use for new instance if not creating a new network

図4：ネットワーク設定の例

インスタンス設定

ターゲットのATP-DまたはExaCS環境の可用性ドメインに適合する可用性ドメインを選択します。コンピュート・シェイプは以下のガイドラインに基づきます。

- » VM.Standard2.4 - 1 MB/秒未満のピークREDOレート
- » VM.Standard2.8 - 15 MB/秒未満のピークREDOレート
- » VM.Standard2.16またはVM.Standard2.24 - 15 MB/秒超のピークREDOレート

必ず「ASSIGN PUBLIC IP」を選択し、オンプレミス・ネットワークからGoldenGate Hubにアクセスできるようにしてください。

インストール時のデフォルトのストレージ・サイズは以下のとおりです。

- » ブート・ボリューム・サイズ - デフォルト値は50 GB
- » スワップ・ボリューム・サイズ - デフォルト値は256 GB
- » 証跡ボリューム・サイズ - デフォルト値は512 GB
- » デプロイメント・ボリューム・サイズ - デフォルト値は128 GB

特に証跡ファイルなど、ボリューム・サイズを大きくする必要がある場合は、「CUSTOM VOLUME SIZES」を選択して大きくしてください。インスタンス設定の例：

Instance Settings

AVAILABILITY DOMAIN

gWtc:UK-1-AD-3

The availability domain for the new Oracle GoldenGate instance

COMPUTE SHAPE

VM.Standard2.4

Shape of new compute instance. Supported shapes are VM.Standard2.4, VM.Standard2.8, VM.Standard2.16 and VM.Standard2.24

ASSIGN PUBLIC IP
Indicates if the new VM should have a public IP address

CUSTOM VOLUME SIZES
Use this field to customize the size of new block storage volumes

図5：インスタンス設定の例

Oracle GoldenGateのデプロイメントを作成する

ソースとターゲットのデータベースが同じ場合は、データベース・バージョンと一緒にデプロイメント1の名前を入力します。ソースとターゲットのデータベースが異なる場合は、データベース・バージョンと一緒にデプロイメント2の名前も入力します。

デプロイメントの作成例：

Create OGG Deployments

DEPLOYMENT 1 - NAME

Source

Name for OGG deployment 1

DEPLOYMENT 1 - DATABASE

Oracle 12c

Oracle RDBMS Version for deployment 1

DEPLOYMENT 2 - NAME OPTIONAL

Target

Name for OGG deployment 2

DEPLOYMENT 2 - DATABASE OPTIONAL

Oracle 19c

Oracle RDBMS Version for deployment 2

図6：Oracle GoldenGateデプロイメントの作成例

インストールが完了すると、GoldenGateのデプロイメントが自動的に起動します。

次のページのドキュメントの説明に従って、必ずサービス・マネージャと管理サーバーのパスワードをリセットしてください。
<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oggmp/getting-started-oracle-goldengate-microservices.html#GUID-7D2C4A9C-EFEA-4BE0-8AB0-92C8FC6B6258>

このホワイト・ペーパー全体を通じて、2つのGoldenGateデプロイメントはソースおよびターゲットと呼びます。

OCI MarketplaceのGoldenGateによって作成される証跡ファイル・ストレージ・ボリュームは、/u02/trailsです。このディレクトリは、GoldenGateによって生成されるすべての証跡ファイルの場合に使用します。

Oracle GoldenGateのデプロイメントの開始と停止

OCI MarketplaceのインストールによってGoldenGateのデプロイメントが作成されると、その後のコンピュート・インスタンスのリブート時に、それらのデプロイメントが自動的に再起動されます。

同じくデプロイメントを起動するGoldenGateサービス・マネージャは、管理者アカウント・ユーザーとしてGoldenGate Hubコンピュート・インスタンスにsshで接続されている場合には、次のコマンドを使用して手動で起動および停止できます。

```
$ sudo systemctl start OracleGoldenGate  
$ sudo systemctl status OracleGoldenGate  
$ sudo systemctl stop OracleGoldenGate
```

Oracle GoldenGate Hubの受信ポートを開く

ハブでGoldenGateソフトウェアを構成したら、コンピュート・インスタンスVCNセキュリティ・リストで受信ポート443を開き、Nginxリバース・プロキシ・サーバー経由でGoldenGateのデプロイメントにアクセスできるようにする必要があります。

デフォルトのセキュリティ・ルールがすでに構成されていてNginxリバース・プロキシに接続できるようになっている場合は、新しい受信ポート・セキュリティ・ルールを追加する前にGoldenGate Hubへの接続性をテストしてください。これには、以下の図7に示すような、そのコンピュート・インスタンスのパブリックIPアドレスを取得します。

図7：GoldenGateのパブリックIPアドレスの取得

パブリックIPアドレスを使用し、以下のURLでWebブラウザからGoldenGateサービス・マネージャに接続します。

`https://<public_ip_address>`

受信ルールが追加されていないことが原因でサービス・マネージャのログイン・ページが表示されない場合は、以下の手順に従って受信ルールを作成します。

1. ナビゲーション・メニューを開きます。「Compute」で「Instances」をクリックします。
2. 自分の「Compartment」を選択します。インスタンスのリストが表示されます。
3. リストからインスタンスを選択し、そのインスタンスの「Virtual Cloud Network」をクリックします。
4. リストでサブネットを見つけ、「Security Lists」でデフォルトのセキュリティ・リストをクリックします。
5. 「Add Ingress Rules」をクリックし、次のパラメータ値と一緒に受信ルールを追加します。
 - » **Source Type** - 「CIDR」に設定
 - » **Source CIDR** - これは、名前指定ソースIPアドレス（12.34.5.67/32など、推奨）か0.0.0.0/0のどちらかに設定して、すべてのマシンがGoldenGate Hubにアクセスできるようにする必要があります。
 - » **Protocol** - 「TCP」に設定
 - » **Destination Port Range** - 「443」に設定

以下にルールの構成例を示します。

図8 受信ルールの追加例

6. パブリックIPアドレスを使用してGoldenGate Hubへの接続を再テストします。

セキュリティ・リストの作成または更新について詳しくは、次のページを参照してください。

<https://docs.cloud.oracle.com/iaas/Content/Network/Concepts/securitylists.htm>

注：GoldenGate HubのIPアドレスはDNSに登録（ドメイン名登録）されていないため、ハブ・サーバーのローカル・ホスト・ファイル・エントリを自分で作成してください。

たとえば、次のように指定します。

123.34.56.7 gghub.example.com

7. Oracle GoldenGate Microservicesの接続をテストする

簡単なテスト方法は、次のコマンドを使ってデプロイメントの状態を問い合わせる方法です。コマンドは、名前指定IPアドレスか、上記のソースCDIR用に指定した（「GoldenGateHubの受信ポートを開く」）curlおよびpythonがインストールされている任意のオンプレミス・サーバーのどちらからでも実行することができます。

```
$ curl -s -K access.cfg https://gghub.example.com/services/v2/config/health -XGET | python -m json.tool
```

サンプル出力：

```
{
  "$schema": "api:standardResponse",
  "links": [
    {
      "href": "https://gghub.example.com/services/v2/config/health",
      "mediaType": "application/json",
      "rel": "canonical"
    },
    {
      "href": "https://gghub.example.com/services/v2/config/health",
      "mediaType": "text/html",
      "rel": "alternate"
    }
  ]
}
```

```

...
  "response": {
    "$schema": "ogg:health",
    "criticalResources": [
      {
        "deploymentName": "source",
        "healthy": true,
        "name": "adminsrvr",
        "status": "running",
        "type": "service"
      },
      {
        "deploymentName": "target", "healthy": true,
        "name": "adminsrvr",
        "status": "running",
        "type": "service"
      },
    ],
    "deploymentName": "ServiceManager",
    "healthy": true,
    "serviceName": "ServiceManager", "started": "2019-09-26T21:14:08.258Z"
  }
}

```

Oracle GoldenGateの構成

この項では、ターゲット・データベースをインスタンス化する前にGoldenGate Hubで実行する必要がある、Oracle GoldenGateの構成手順について説明します。

ソース側のGoldenGate資格証明を作成する

GoldenGateプロセスがソース・データベースとターゲット・データベースに接続するには、GoldenGateデータベース資格証明が必要です。GoldenGate資格証明ストアは、クリア・テキスト・パスワードがExtractまたはReplicatパラメータ・ファイルのいずれにも保存されないようにします。ターゲット・データベースはまだ作成されていないため、この手順ではソース・データベースの資格証明のみを作成します。ターゲット・データベースの資格証明は、後の手順で作成します。

Oracle GoldenGateのtnsnames.oraファイルとsqlnet.oraファイルは、各デプロイメントのクライアント・ソフトウェア・データベース・バージョンのディレクトリに格納されます。たとえば、ソース・データベースがOracle 12cの場合、TNS_ADMINディレクトリは/u01/app/client/oracle12/network/adminです。ターゲット・データベースがOracle 19cの場合、TNS_ADMINディレクトリは/u01/app/client/oracle19/network/adminです。

Oracle Net構成ファイルを準備する

データベース資格証明はOracle Easy Connectネーミング・メソッドを使用するので、ローカルのtnsnames.oraファイルを使用する必要がありません。ただし、データベース・バージョンがRelease 19cより前で、SSL認証を使用している場合は除きます。そのような場合、Easy Connectのネーミングは使用できず、tnsnames.oraが必要です。

以下は、19cより前のOracle RACデータベースでSSL認証を使用する場合の、tnsnames.oraエントリの例です。

```
GGSOURCE =
(DESCRIPTION=
  (SDU=2097152)      # Or set to 65536 for 11.2.0.4 database
  (CONNECT_TIMEOUT=10) (RETRY_COUNT=3)
  (ADDRESS=(PROTOCOL=TCPS) (HOST=<primary db scan address>) (PORT=2484) )
  (CONNECT_DATA=(SERVICE_NAME=<db service name>))
)
```

ソース・データベースでSSL認証が使用されており、データベースのバージョンが19cより前の場合は、次のパラメータを指定したsqlnet.oraファイルが必要です。

```
SSL_CLIENT_AUTHENTICATION=TRUE
SSL_SERVER_DN_MATCH=OFF
SQLNET.EXPIRE_TIME = 10
SQLNET.WALLET_OVERRIDE = FALSE
WALLET_LOCATION = (SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY="/u01/oracle/network")))
SSL_VERSION = 1.2
DEFAULT_SDU_SIZE=2097152      # Change to 65536 for 11.2.0.4 database
```

注：ソース・データベース環境のSSLウォレットを必ず、GoldenGate HubのDIRECTORYディレクトリにコピーしてください。

最後に、ソース・データベースがSSL認証の代わりにOracle Net暗号化を使用している場合（データベースのバージョンは問わない）、次のパラメータを指定したsqlnet.oraファイルが必要です。

```
SQLNET.ENCRYPTION_CLIENT = required
SQLNET.ENCRYPTION_TYPES_CLIENT= (AES256)
# The SDU size is only required for databases earlier than 19c:
DEFAULT_SDU_SIZE=2097152      # Change to 65536 for 11.2.0.4 database
```

Oracle Database 19cはOracle Easy Connectネーミング・メソッドを拡張して、Oracle Easy Connect Plusという名称にしました。

Oracle Easy Connect Plusネーミング・メソッドについて詳しくは、『*Oracle Net Administrators Guide*』を参照してください。

<https://docs.oracle.com/en/database/oracle/oracle-database/19/netag/configuring-naming-methods.html#GUID-8C85D289-6AF3-41BC-848B-BF39D32648BA>

GoldenGateデータベースの資格証明を作成する

tnsnames.oraファイルを使用している場合は、TNSエイリアスを使ってソース・データベースへの資格証明を作成します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/credentials/goldengate/source -X POST --data
'{"userid":"ggadmin@GGSOURCE", "password":"<password>"}' | python -m json.tool
```

tnsnames.oraファイルを使用していないが、データベース・バージョンが19cより前のリリースである場合は、次のEasy Connectのネーミング例（およびSCANリスナー）を使ってソース・データベースへの資格証明を作成します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/credentials/goldengate/source -X POST --data
```

```
'{"userid":"ggadmin@database-host-scan.1521/database_service_name","password":"<password>"}' |  
python -m json.tool
```

次の例は、Oracle RACとSSL認証が構成された19cソース・データベース用の資格証明を作成する方法を示します。

```
curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source -X POST --data  
'{"userid":"ggadmin@tcp://database-host-scan:2484/db_service_name?sdu=2097152&  
SSL_SERVER_DN_MATCH=NO&ssl_server_cert_dn='cn=common_name&wallet_location='u01/oracle/network'',  
"password":"<password>"}' | python -m json.tool
```

以下は、sqlnet.oraファイルで有効にしたOracle Net暗号化を使用する資格証明をOracle RACデータベースで作成する例です。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source -X POST --data  
'{"userid":"ggadmin@tcp://database-host-scan:1521/db_service_name?sdu=2097152",  
"password":"<password>"}' | python -m json.tool
```

マルチテナント・データベースの場合は、上記のガイドラインに従って、移行対象のCDBとPDB用の資格証明を個別に作成する必要があります。

作成されたGoldenGate資格証明を表示するには、次のコマンドを使用します。

1. デプロイメントのGoldenGate資格証明の名前を表示します

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate -XGET | python -m  
json.tool | grep name  
"name": "source_cdb"  
"name": "source_pdb"
```

2. 資格証明の接続の詳細を表示します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source_cdb -XGET |  
python -m json.tool  
"response": {  
    "$schema": "ogg:credentials",  
    "userid": "ggadmin@tcp://database-host-  
scan:2484/db_service_name?sdu=2097152&SSL_SERVER_DN_MATCH=NO&ssl_server_cert_dn='cn=common_name&wa  
llet_location='u01/oracle/network'"  
}
```

3. 資格証明を削除する場合は、次のコマンドを使用します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source_cdb -X DELETE  
| python -m json.tool
```

ソース・データベース上でハートビート表を作成する

ソース・データベースとターゲット・データベース間のレプリケーションの待機時間を監視するには、GoldenGateハートビート表が必要です。ソース・データベースにすでにGoldenGateハートビート・オブジェクトが含まれている場合は、それらを使用してください。

1. ハートビート表がすでに存在するかを確認し、以下に示すように、前の手順で作成したソース・データベースの資格証明を使用します。

注：マルチテナント・データベースの場合、ハートビート表は、レプリケートされるPDBに配置する必要があります。必ずソースPDB資格証明を使用して、ハートビート表を確認してください。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/connections/goldengate.source_pdb/tables/heartbeat -XGET | python -m json.tool
```

次の出力例は、ハートビート表が現在構成されていないことを示します。

```
"messages": [
  {
    "$schema": "ogg:message",
    "code": "OGG-08100",
    "issued": "2019-09-26T23:06:38Z",
    "severity": "INFO",
    "title": "Heartbeat table ggadmin.gg_heartbeat does not exist.",
  },
  {
    "$schema": "ogg:message",
    "code": "OGG-08100",
    "issued": "2019-09-26T23:06:38Z",
    "severity": "INFO",
    "title": "Heartbeat table ggadmin.gg_heartbeat_seed does not exist.",
  },
  {
    "$schema": "ogg:message",
    "code": "OGG-08100",
    "issued": "2019-09-26T23:06:38Z",
    "severity": "INFO",
    "title": "Heartbeat table ggadmin.gg_heartbeat_history does not exist.",
  }
]
```

次の出力例は、GoldenGateハートビート表がすでに存在することを示します。

```
"messages": [
  {
    "$schema": "ogg:message",
    "code": "OGG-08100",
    "issued": "2019-09-26T23:31:08Z",
    "severity": "INFO",
    "title": "HEARTBEAT table ggadmin.gg_heartbeat exists.",
  },
  ...
]
```

```
{
  "$schema": "ogg:message",
  "code": "OGG-08100",
  "issued": "2019-09-26T23:31:08Z",
  "severity": "INFO",
  "title": "HEARTBEAT table ggadmin.gg_heartbeat_seed supplemental logging ENABLED.",
},
...
"response": {
  "$schema": "ogg:tablesHeartbeat",
  "addTrandata": true,
  "frequency": 300,
  "partitioned": false,
  "purgeFrequency": 1,
  "retentionTime": 30, "targetOnly": false
}
}
```

ハートビート表がすでに使用されている場合は、同じ表が新しく移行したGoldenGate Extractによって使用されます。

- 既存のハートビート表の頻度がデフォルトの60秒ではない場合、現在の値をメモし、移行の間60秒に変更します。

注：移行完了後、元の値に戻せるように、変更する前に頻度の値を覚えておいてください。

ハートビート更新頻度を変更します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/connections/goldengate.source_pdb/tables/he
rtbeat -X PATCH --data '{"frequency":60}'| python -m json.tool
```

- ハートビート表が存在しない場合は、ソース・データベース内にハートビート表を作成します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/connections/goldengate.source_pdb/tables/he
rtbeat -X POST
--data '{}'| python -m json.tool
```

注：ターゲット・データベースのハートビート表は、データベースのインスタンス化後に作成されます。

インスタンス化のためにソース・データベースのスキーマを準備する

データベース移行の一部であり、レプリケート対象となっているすべてのデータベース・スキーマのうちのソース・スキーマをレプリケーションできるように準備する必要があります。

注：インスタンス化のためにデータベース・スキーマを準備する作業は、GoldenGate Extractプロセスの作成前に完了する必要があります。

1. 次のコマンドを使って、各データベース・スキーマをインスタンス化します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/connections/goldengate.source_pdb/trandata/s
chema -X POST --data '{"operation":"add","schemaName":"soesmall","prepareCsnMode":"nowait"}' |
python -m json.tool
```

2. 次のコマンドを使って、ソース・データベースの準備した各スキーマを確認します。

次のコマンドで1つのスキーマを確認します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/connections/goldengate.source_pdb/trandata/s
chema -X POST --data '{"operation":"info","schemaName":"soesmall"}' | python -m json.tool
```

次のコマンドで、すべてのデータベース・スキーマ（レプリケートするスキーマはこれらのスキーマから取り出すことができます）を確認します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/connections/goldengate.source_pdb/trandata/s
chema -X POST --data '{"operation":"info","schemaName":"*"}' | python -m json.tool
```

GoldenGate Extractプロセスを作成および起動する

次の項では、Extractの存在確認、Extractの追加、プロセスの起動と停止、およびExtractの削除方法について説明します。

1. Extractが存在することを確認します（すでに作成していて、削除されていない場合）。

```
$ curl -s -K access.cfg https://gghub.example.com /SOURCE/adminsrvr/services/v2/extracts/EXT1 -X
GET | python -m json.tool
```

次の例は、Extractが存在しない場合のレスポンスを示します。

```
"messages": [
  {
    "$schema": "ogg:message",
    "code": "OGG-12029",
    "issued": "2019-09-26T04:52:41Z",
    "severity": "INFO",
    "title": "The extract with name 'EXT1' does not exist."
  }
]
```

次の例は、Extractが存在する場合のレスポンスを示します。

```
"messages": [],
"response": {
  "$schema": "ogg:extract",
  "begin": "now",
  "config": [
    "Extract EXT1",
    "ExtTrail aa",
    "UseridAlias source_cdb DOMAIN goldengate",
```

```

    "TRANLOGOPTIONS PERFORMANCEPROFILE HIGH",
    "TRANLOGOPTIONS _readaheadcount 64"
    "REPORTCOUNT EVERY 15 MINUTES, RATE",
    "STATOPTIONS REPORTFETCH",
    "DDL EXCLUDE ALL",
    "Table pdb1.soesmall.*;"  

],  

"credentials":{  

    "alias": "source_cdb",  

    "domain": "goldengate"  

},  

"registration":{  

    "containers": [  

        "PDB1"  

    ],  

    "csn":51162830  

},  

"source":{  

    "tranlogs": "integrated"  

},  

"status": "stopped",  

"targets":[],  

"type": "Integrated"  

}
}

```

2. Extractプロセスを作成します。

以下は、推奨されるExtract最小パラメータ値です。

```
-- Replace with trail file naming standard
```

```

EXTTRAIL aa
TRANLOGOPTIONS PERFORMANCEPROFILE HIGH
TRANLOGOPTIONS _readaheadcount 64      -- Required for streaming protocol (bug fix 28849751)
DISCARDFILE APPEND
REPORTCOUNT EVERY 15 MINUTES, RATE
STATOPTIONS REPORTFETCH
-- It is recommended to prevent DDL on the source database during the database
-- migration:
DDL EXCLUDE ALL

-- Repeat TABLE command for each schema being replicated:
TABLE [ container. ]<schema_name>.*;

-- Repeat TABLEEXCLUDE command for each table that was highlighted as not supported
-- for GoldenGate replication:
TABLEEXCLUDE [ container. ]<schema_name>.<tablename>;

```

ソース・データベースがPDBの場合、データベースのREDOストリーム全体を読み取る必要があるため、ExtractをCDBに接続する必要があります。PDBからレプリケートされるオブジェクトは、TABLEパラメータによって特定されます。

次のコマンドを使って、Extractを作成します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1 -X POST --data '{"config": ["Extract EXT1", "ExtTrail /u02/trails/aa", "UseridAlias source_cdb DOMAIN goldengate", "TRANLOGOPTIONS PERFORMANCEPROFILE HIGH", "TRANLOGOPTIONS _readaheadcount 64", "REPORTCOUNT EVERY 15 MINUTES, RATE", "STATOPTIONS REPORTFETCH", "DDL EXCLUDE ALL", "Table pdb1.soesmall.*;"], "source": {"tranlogs": "integrated"}, "credentials": {"alias": "source_cdb", "domain": "goldengate"}, "begin": "now", "targets": [{"path": "/u02/trails", "name": "aa", "sizeMB": 500}], "registration": {"containers": ["pdb1"]}}' | python -m json.tool
```

注：バグ28849751のパッチをソースのオンプレミス・データベースに適用した後は、_readaheadcount パラメータを指定する必要があります。

複数のスキーマを抽出する場合は、複数のTable <Source PDB>.<Schema Name>.*; パラメータを使用します。

次のコマンドを使って、非CDBデータベースのExtractを作成します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1 -X POST --data '{"config": ["Extract EXT1", "ExtTrail /u02/trails/aa", "UseridAlias source_db DOMAIN goldengate", "TRANLOGOPTIONS PERFORMANCEPROFILE HIGH", "TRANLOGOPTIONS _readaheadcount 64", "REPORTCOUNT EVERY 15 MINUTES, RATE", "STATOPTIONS REPORTFETCH", "DDL EXCLUDE ALL", "Table soesmall.*;"], "source": {"tranlogs": "integrated"}, "credentials": {"alias": "source_db", "domain": "goldengate"}, "begin": "now", "targets": [{"path": "/u02/trails", "name": "aa", "sizeMB": 500}], "registration": "default"}' | python -m json.tool
```

3. Extract構成を確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1 -X GET | python -m json.tool
```

4. Extractのステータスを確認します。

Extractは自動的に起動しないので、ステータスは停止になっています。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1/info/status -X GET | python -m json.tool
```

次の例は、Extractが起動しない場合に予想される結果を示します。

```
"messages": [],
"response": {
    "$schema": "ogg:extractStatus",
    "lag": 0,
    "lastStarted": null,
    "position": "0.0",
    "sinceLagReported": 208,
    "status": "stopped"
}
```

5. Extractを起動します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/commands/execute -X POST --data '{"name":"start","processName":"EXT1"}' | python -m json.tool
```

1~2分待ってから、Extractのステータスを確認します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1/info/status -X GET | python -m  
json.tool
```

適切に起動した後の出力は次のようにになります。

```
"response": {  
    "$schema": "ogg:extractStatus",  
    "lag": 1,  
    "lastStarted": "2019-09-26T19:09:45.524Z",  
    "position": "0.51251093",  
    "processId": 4702,  
    "sinceLagReported": 2,  
    "status": "running"  
}
```

AUTOSTARTタスクを作成する

デプロイメントが起動したときに、GoldenGateプロセスを自動的に起動させるには、AUTOSTARTプロファイルとAUTORESTARTプロファイルが必要です。GoldenGateプロセスは、これらのプロセスを自動的に起動させるアクティブ・プロファイルの一部としてデフォルト設定されていません。

1. AUTOSTARTプロファイルが存在するかどうかを確認し、存在する場合は、削除または更新することができます。

AUTOSTARTプロファイルが存在するかどうかを確認します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:MIGRATE01 -XGET | python -m json.tool
```

次の出力は、プロファイルが存在しない場合の例を示しています。

```
"messages": [  
    {  
        "$schema": "ogg:message",  
        "code": "OGG-12029",  
        "issued": "2019-09-26T17:22:43Z",  
        "severity": "INFO",  
        "title": "The value with name 'ogg:managedProcessSettings:MIGRATE01' does not exist."  
    }  
]
```

プロファイルが存在する場合は、次の出力が表示されます。

```
"response": {  
    "autoRestart": {  
        "delay": 60,  
        "disableOnFailure": true,  
        "enabled": true,  
        "interval": 1000000000000000000,  
        "maxAttempts": 10,  
        "maxInterval": 1000000000000000000,  
        "minInterval": 1000000000000000000,  
        "process": "MIGRATE01",  
        "startOnFailure": true  
    }  
}
```

```
        "enabled": true,
        "onSuccess": false,
        "retries": 5,
        "window": 1200
    },
    "autoStart": {
        "delay": 60, "enabled": true
    }
}
```

プロファイルがすでに存在する場合は、削除するか、次の推奨設定値で更新できます。プロファイルを削除する場合：

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:MIGRATE01 -XDELETE | python -m json.tool
```

推奨設定値で現在のプロファイルを更新する場合：

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:MIGRATE01 -XPUT --data '{"autoStart": {"enabled": true, "delay": 10}, "autoRestart": {"delay": 30, "disableOnFailure": true, "enabled": true, "onSuccess": false, "retries": 5, "window": 1200}}' | python -m json.tool
```

2. ソース・デプロイメントおよびターゲット・デプロイメント双方の次のAUTOSTARTプロファイルとAUTORESTARTプロファイルを作成します。

プロファイルが現在存在しない場合、または前の手順で削除した場合は、プロファイルを作成します。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:MIGRATE01 -XPOST --data '{"autoStart": {"enabled": true, "delay": 60}, "autoRestart": {"delay": 60, "disableOnFailure": true, "enabled": true, "onSuccess": false, "retries": 5, "window": 1200}}' | python -m json.tool
```

3. AUTOSTARTプロファイルをExtractに割り当てます。

注：GoldenGate Replicatはまだ作成していないので、この時点ではプロファイルを割り当てるにはできません。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X
PATCH --data '{"managedProcessSettings": "MIGRATE01"}' | python -m json.tool
```

Extractに新しいプロファイルが構成されていることを確認します（部分的な出力を示します）。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X GET
| python -m json.tool | grep managedProcessSettings
"managedProcessSettings": "MIGRATE01",
```

ソース・データベースの長時間実行トランザクションを監視する

データベースを使ってターゲット・データベースをインスタンス化するには、GoldenGateですべてのトランザクション・データが確実にレプリケートされるよう、GoldenGate Extractの作成時点でアクティブだったトランザクションを確実にコミットさせることが重要です。

たとえば、次のタイムラインについて考えてみます。

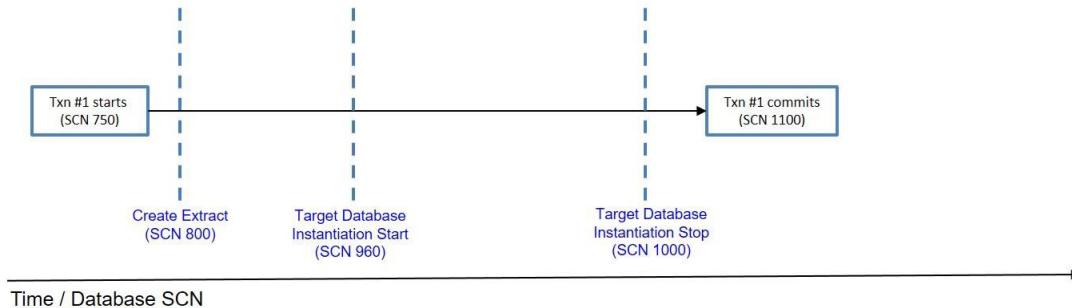


図9：長時間実行トランザクションのタイムライン

トランザクション1はソース・データベースでSCN 1100の時にコミットされますが、Extractでは取得されません。それは、Extractでは完全なトランザクションのみを処理し、プロセスはトランザクション開始後に作成されたからです。SCN 750は800より前なので、このトランザクションは無視されます。

次の問合せをソース・データベースに対して実行して、Extractの作成前に開始されたトランザクションすべてがコミットまたはロールバックされた時点を特定し、ターゲット・データベースのインスタンス化を安全に開始できるようにします。

```
SQL> SELECT capture_name, c.start_scn, t.start_scn
  FROM dba_capture c, v$transaction t WHERE t.start_scn < c.start_scn;
```

問合せをしても行が返されない場合は、ターゲット・データベースのインスタンス化を開始しても安全です。

ターゲット・データベースのインスタンス化

Oracle Data Pump (mv2oci/mv2adb)

Oracle Data Pumpにより、ソース・データベースとターゲット・データベースのバージョンが異なる場合、ハードウェア・プラットフォームが異なる場合、またはデータベース構造が異なる場合に、ターゲット・データベースをインスタンス化することができます。たとえば、非暗号化データベースや暗号化データベースに移行する場合です。

ソース・データベース・スキーマがレプリケーション用に用意されているときに、GoldenGate Extractが作成されると（「[Prepare the Source Database Schemas for Instantiation](#)」を参照）、その後のData Pumpのエクスポート・ダンプ・ファイルには表インスタンス化SCNが含まれるようになり、各表が存在するSCNが一貫性を保つようにされます。GoldenGate Replicatによりインスタンス化ACNの前に抽出された証跡ファイルのデータが適用されても、それは無視されます。インスタンスがSCNに到達すると、レプリケートされた表ごとに、Replicatがレプリケートされたトランザクションを適用します。この方法により確実にゼロ・データ損失を達成し、複製データの適用を阻止します。

mv2ociツールとmv2adbツールを使用したターゲット・データベースへのデータの移行について詳しくは、以下のMy Oracle Support (MOS) Noteを参照してください。

- » ATP-Dターゲット・データベースについて詳しくは、MOS Note 2463574.1 (mv2adbを使用する場合) を参照してください。
- » ExaCSターゲット・データベースについて詳しくは、MOS Note 2514026.1 (mv2ociを使用する場合) を参照してください。

オンプレミス・データベースの移行時にmv2adb/mv2ociによるData Pumpエクスポート/インポートを使用する場合は、以下の点に注意してください。

- » Data Pumpエクスポート/インポートの進捗状況をV\$SESSION_LONGOPSデータ・ディクショナリ・ビューで監視します。
- » Oracle GoldenGateでレプリケートするオブジェクトは前の手順すでに準備してあるので、オブジェクトがターゲット・データベースにインポートされた後、Replicatは複製データを適用しなくとも、適用すべき証跡ファイルのトランザクションを判別できます。このため、「auto -netlink」オプションを使用しない場合は、mv2adb/mv2ociのパラメータ--goldengateを使用してください。

Oracle RMAN複製データベース

Oracle RMANを使用したソース・データベースの複製は、ソースと移行先のプラットフォーム・エンディアンネスが適合し、Oracle Databaseのバージョンが同じ場合にデータベース全体をコピーするための代替方法です。このデータベースのインスタンス化方法は、ATP-Dへの移行時には使用できません。Oracle RMANを使用してソース・データベースを複製してExacCSに移行する方法について詳しくは、次のページのOracle Cloud Infrastructureのドキュメントで説明されています。

<https://docs.cloud.oracle.com/iaas/Content/Database/Tasks/mig-rman-duplicate-active-database.htm>

複製後、ターゲット・データベースを開いたら、さらにいくつかの構成手順を実行する必要があります。

1. 複製後、ターゲット・データベースを開いたときに、データベースがリカバリされた時点のSCNを記録する必要があります。このSCNは、Oracle GoldenGate Replicatプロセスを起動する後の手順で使用します。SCNはターゲット・データベースのalert.logまたはデータ・ディクショナリから得ることができます。両方の例を以下に示します。
SCNをalert.logエントリから取得します。

```
RESETLOGS after incomplete recovery UNTIL CHANGE 681543060 time 05/01/2019 15:05:20
```

データ・ディクショナリの問合せを使ってSCNを取得します。

```
SQL> select RESETLOGS_CHANGE# - 1 from v$database;
```

注：このSCNを記録しておいてください。GoldenGate Replicatプロセスの起動時に後で使用します。

2. データベースのインスタンス化パラメータENABLE_GOLDENGATE_REPLICATIONがTRUEに設定されていることを確認します。設定されていない場合は、次のコマンドを使って設定してください。

```
SQL> alter system set enable_goldengate_replication=TRUE scope=both;
```

Oracle GoldenGateの構成を完了する

この項では、ターゲット・データベースのインスタンス化後に必要なOracle GoldenGateの構成について説明します。

ターゲット・データベースのGoldenGate資格証明を作成する

Oracle Data Pumpを使ってターゲット・データベースをインスタンス化した場合、前の手順「[Oracle GoldenGate管理者データベース・アカウントを作成する](#)」でGoldenGateデータベース管理者アカウントをすでに作成しています。

GoldenGateでターゲット・データベースへの接続を処理できるようにするには、GoldenGateのデータベース資格証明が必要です。

ターゲット・データベースのGoldenGate資格証明は、CDBではなくPDBに接続する必要があります。

ターゲットのGoldenGate資格証明は、先の「[ソースのGoldenGate資格証明を作成する](#)」で指定したソース・データベースの資格証明と同様の方法で作成しますが、次の追加の注意点があります。

注：ターゲット・データベースがマルチテナントの場合、GoldenGate資格証明は以前にインスタンス化されているPDBに接続します。

ATP-Dターゲット・データベース

ATP-Dの場合、Oracleクライアント資格証明（ウォレット・ファイル）は、Oracle Cloud Infrastructureコンソールからダウンロードする必要があります。資格証明のダウンロードについて詳しくは、次のページのATP-Dのドキュメントを参照してください。

<https://docs.oracle.com/en/cloud/paas/atp-cloud/atpud/download-client-credentials.html#GUID-9F5AD1CB-5DAD-44C4-8978-C485575E23C2>

ウォレット・ファイルはGoldenGate Hub上のディレクトリに格納されており、以下のwallet_locationパラメータで参照されます。

事前定義のデータベース・サービス名tp_tlsを使用するようお奨めします。たとえば、Oracle Easy Connect Plusネーミング・メソッド別名を使用してターゲット・データベースの資格証明を作成します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/credentials/goldengate/target -X POST --data
'{"userid":"ggadmin@tcp://<public_ip_address>:2484/atpd_tp_tls.atp.oraclecloud.com?sdu=2097152&
SSL_SERVER_DN_MATCH=NO&ssl_server_cert_dn='cn=common_name'&wallet_location='u01/oracle/network",
"password":"<password>"}' | python -m json.tool
```

ExaCSターゲット・データベース

ExaCSの場合、データベースへの接続性は、データベース・ホストのパブリックIPアドレスか、DNS管理者に登録されているSCANリスナー名のいずれかを使用するように構成されます。ExaCSで構成されているデフォルトのリスナー・ポートはポート1521で、TCPプロトコルに従います。

GoldenGate HubとExaCSのターゲット・データベースとの間では、Oracle Net暗号化を使用することをお奨めします。GoldenGate Hubのsqlnet.oraファイルで次のパラメータを使用してOracle Net暗号化を有効にしてください。

```
SQLNET.ENCRYPTION_CLIENT = required
SQLNET.ENCRYPTION_TYPES_CLIENT= (AES256)
```

ExaCSデータベースへの接続について詳しくは、次のページを参照してください。<https://docs.oracle.com/en/cloud/paas/exadata-cloud/csexa/connect-db-using-net-services.html>

以下は、ExaCS Oracleデータベースの最初のノードに対して、sqlnet.oraファイルで有効にしたOracle Net暗号化を使用する資格証明を作成する場合の例です。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/credentials/goldengate/target -X POST --data
'{"userid":"ggadmin@tcp://<public_ip_address>:1521/db_service_name", "password":"<password>"}' |
python -m json.tool
```

作成されたGoldenGate資格証明を表示するには、次のコマンドを使用します。

1. デプロイメントのGoldenGate資格証明の名前を表示します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/credentials/goldengate -XGET | python -m
json.tool | grep name
"name": "target"
```

2. 資格証明の接続の詳細を表示します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/credentials/goldengate/target -XGET | python
-m json.tool
```

ターゲット・データベースのGoldenGateチェックポイント表を作成する

チェックポイント表は、このホワイト・ペーパーで使用が推奨されている、統合されていないパラレルReplicatの必須コンポーネントです。

統合されていないReplicatは自身のリカバリ・チェックポイントを、ターゲット・データベースに格納されたチェックポイント表に保持しています。チェックポイントは、Replicatトランザクション内のチェックポイント表に書き込まれます。チェックポイントはトランザクションとともに成功または失敗するので、プロセスまたはデータベースの失敗があっても、Replicatは、トランザクションが1回しか適用されないようにします。

1. RMAN Duplicateを使用してターゲット・データベースを作成した場合は、次のコマンドを使用してチェックポイント表が存在することを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X POST --data '{"name": "report", "reportType": "checkpointTables", "credentials": {"alias": "target_atp1", "domain": "goldengate"}, "specification": "ggadmin.gg_checkpoints"}' | python -m json.tool
```

注：チェックポイント表のエイリアス、ドメイン、仕様は、ターゲット・データベースに適合するように置き換えてください。

チェックポイント表がすでに存在する場合、出力例は次のようになります。

```
"response": {  
    "$schema": "ogg:commandResult",  
    "tables": [  
        "GGADMIN.GG_CHECKPOINTS"  
    ]  
}
```

チェックポイント表が存在しない場合、出力例は次のようになります。

```
"response": {  
    "$schema": "ogg:commandResult",  
    "tables": []  
}
```

2. チェックポイント表がすでに存在する場合は、削除してから新しいチェックポイント表を作成します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/checkpo  
int -X POST --data '{"operation": "delete", "name": "ggadmin.gg_checkpoints"}' | python -m json.tool
```

3. チェックポイント表を作成します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/checkpo  
int -X POST --data '{"operation": "add", "name": "ggadmin.gg_checkpoints"}' | python -m json.tool
```

4. 次に、表が作成されたことを確認します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/checkpo  
int -X POST --data '{"operation": "info", "name": "ggadmin.gg_checkpoints"}' | python -m json.tool
```

返されるメッセージの例：

```
"$schema": "ogg:message",  
"code": "OGG-08100",  
"issued": "2019-06-26T17:18:09Z",
```

```
"severity":"INFO",
"title":"Checkpoint table ggadmin.gg_checkpoints created 2019-06-26 17:17:36.",
```

ターゲット・データベースにハートビート表を作成する

ソース・データベースとターゲット・データベース間のレプリケーションの待機時間を監視するには、GoldenGateハートビート表が必要です。RMAN DUPLICATEを使ってインスタンス化した後、ターゲット・データベースにGoldenGateハートビート・オブジェクトがすでに含まれている場合、オブジェクトをいったん削除してから再作成する必要があります。

1. 現時点でターゲット・データベースにハートビート・オブジェクトが存在するかどうかを確認します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/heartbe
at -XGET | python -m json.tool
```

2. ハートビート・オブジェクトが存在する場合は、次のコマンドを使用してターゲットのハートビート表オブジェクトを削除します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/heartbe
at -X DELETE| python -m json.tool
```

3. ターゲットのハートビート表オブジェクトを作成します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/heartbe
at -X POST --data '{"targetOnly":true}'| python -m json.tool
```

Oracle GoldenGate Replicatを作成する

RMANを使ってターゲット・データベースをインスタンス化した場合は、Replicatを作成する前に、データベース移行用にすでに作成したGoldenGate Extractを登録解除し、削除する必要があります。

次のコマンドを使用してExtractを登録解除し、データベースから削除します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/extracts/EXT1 -X
DELETE | python -m json.tool
```

Replicatプロセスの作成は、ターゲット・データベースをどのようにインスタンス化したかによって異なります。RMANまたはData Pumpによるインスタンス化に応じて、次の手順に従います。

REST APIを使ってReplicatを作成する場合、プロセス・パラメータ・ファイルが自動的に作成されるため、手動で作成する必要はありません。

Oracle RMANのインスタンス化のためにReplicatを作成する

次のサンプル・コマンドでは、統合されていないパラレルReplicatを作成します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X
POST --data '{"credentials":{"alias":"target","domain": "goldengate"}, "source": {"path":"/u02/trails","name":"aa"}, "begin": {"sequence":0,"offset":0}, "checkpoint": {"table":"ggadmin.gg_checkpoints"}, "mode": {"type": "nonintegrated","parallel":true}, "config": ["Replicat REP1", "UseridAlias target"]}
```

```
DOMAIN goldengate","MAP_PARALLELISM 4","MIN_APPLY_PARALLELISM 2", "MAX_APPLY_PARALLELISM 50", "REPORTCOUNT EVERY 15 MINUTES, RATE", "BATCHSQL", "Map pdb1.*.*", Target PDB1.*.*;"]' | python -m json.tool
```

Replicatが作成されたことを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
GET | python -m json.tool
```

Replicatを起動するには、前述の「[Oracle RMAN複製データベース](#)」の項で記録したSCNが必要です。前述の例のSCN値、681543060と以下のサンプル・コマンドを使用して、Replicatを起動します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X  
POST --data '{"name":"start", "processName":"REP1", "AT":681543060}' | python -m json.tool
```

Replicatのステータスを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1/info/status -X GET | python -m json.tool
```

出力例：

```
"response": {  
    "$schema": "ogg:replicatStatus",  
    "lag": 0,  
    "lastStarted": "2019-09-26T19:40:10.419Z",  
    "position": {  
        "name": "aa",  
        "offset": 190639772,  
        "path": "/u02/trails",  
        "sequence": 5  
    },  
    "processId": 55184,  
    "sinceLagReported": 8,  
    "status": "running"  
}
```

Data Pumpのインスタンス化のためにReplicatを作成する

ターゲット・データベースをOracle Data Pumpでインスタンス化する場合、Replicatプロセスの作成時に追加のパラメータが必要になります。ENABLE_INSTANTIATION_FILTERINGパラメータは、Data Pumpエクスポート・ファイルに記録されている、各表のインスタンス化SCNより前にコミットされた証跡ファイル・データを除外し適用しないようにReplicatに指示します。インスタンス化SCNは、表のインポート時にターゲット・データベースのデータ・ディクショナリに保存されます。

Replicatを作成します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
POST --data '{"credentials": {"alias": "target", "domain": "goldengate"}, "source": {"path":  
"/u02/trails", "name": "aa"}, "begin": {"sequence": 0, "offset": 0}, "checkpoint":  
{"table": "ggadmin.gg_checkpoints"}, "mode": {"type": "nonintegrated", "parallel": 1}}
```

```
4","MIN_APPLY_PARALLELISM 2","MAX_APPLY_PARALLELISM 50","REPORTCOUNT EVERY 15 MINUTES, RATE","BATCHSQL",
"DBOPTIONS ENABLE_INSTANTIATION_FILTERING","Map pdb1.*.*","Target PDB1.*.*"]'|  
python -m json.tool
```

Replicatが作成されたことを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
GET | python -m json.tool
```

Replicatを起動します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X  
POST --data '{"name":"start","processName":"REP1"}' | python -m json.tool
```

起動したReplicatのステータスを確認します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1/info/status -X GET | python -m json.tool
```

AUTOSTARTタスクをReplicatに割り当てる

Replicatプロセスが実行されるようになったので、次のサンプル・コマンドを使って、前述の「[AUTOSTARTタスクを作成する](#)」のAUTOSTARTタスクをReplicatに割り当てます。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
PATCH --data '{"managedProcessSettings":"MIGRATE01"}' | python -m json.tool
```

Replicatが新しいプロファイルで構成されていることを確認します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
GET | python -m json.tool | grep managedProcessSettings  
"managedProcessSettings":"MIGRATE01",
```

GoldenGateのレプリケーションを監視する

ExtractとReplicatが両方とも実行されている状態で、ターゲット・データベースのインスタンス化の間に生成された、未処理のすべてのトランザクション・データが適用されるまで、GoldenGateのエンド・ツー・エンドの待機時間を監視し、適用後、ユーザーとアプリケーションを移行先データベースに切り替えることができます。

まず、ExtractプロセスとReplicatプロセスの証跡ファイルの進捗状況を比較します。

Extractチェックポイントの進捗状況を表示します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1/info/checkpoints -XGET |  
python -m json.tool
```

出力例：

```
"current": {  
  "name": "aa",  
  "offset": 388314032,
```

```
"path": "/u02/trails/",  
"sequence": 311,  
"sequenceLength": 9, "sequenceLengthFlip": false,  
"timestamp": "2019-10-07T21:43:36.660Z"  
}
```

Replicatチェックポイントの進捗状況を表示します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1/info/checkpoints -XGET |  
python -m json.tool
```

出力例：

```
"current": { "name": "aa",  
    "offset": 346955548,  
    "path": "/u02/trails/",  
    "sequence": 311,  
    "sequenceLength": 9,  
    "sequenceLengthFlip": false,  
    "timestamp": "2019-10-07T21:48:12.906Z"  
}
```

ReplicatがExtractの書き込み先と同じ証跡ファイルから読み取りを行う際（チェックポイントのオフセットは類似）、Replicatは、ターゲット・データベースのインスタンス化の間に生成されたすべての証跡ファイルを適用します。

次に、ハートビート表を使ってOracle GoldenGateのエンド・ツー・エンドの待機時間を監視します。ラグ時間が許容できる程度に短くなるまで（2秒未満など）、次のコマンドを実行します。

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/heartbe  
at/REP1 -XGET | python -m json.tool
```

出力例：

```
"heartbeats": [  
    {  
        "ageSeconds": 5.02,  
        "lagSeconds": 1.27,  
        "path": "EXT1 ==> REP1",  
        "source": "DSGG:PDB1",  
        "target": "GGT:PDB1"  
    }  
]
```

移行済みデータベースのテスト

GoldenGateの適用を一時停止した後、ターゲット・データベースをテストできるように、保証付きリストア・ポイント（GRP）を作成します。テストが終わり、データベースをGRPにラッシュバックすると、GoldenGate Replicatが再起動し、レプリケーションが再開します。すべてのテストが完了したら、GRPを削除できます。レプリケーションの一時停止、GRPの作成、テスト、データベースのラッシュバック、レプリケーションの再開のサイクルは、何度でも実行できます。

ATP-Dでの権限の制限のため、保証付きリストア・ポイントを作成することはできません。そのため、以下に示す移行済みデータベースの読み書きテストの手順は、ExaCSの場合にのみ有効です。ATP-Dへの移行時には、読み取り専用問合せによってデータベースを検証してください。

レプリケーションを一時停止する

GoldenGate Replicatのプロセスを停止してレプリケーションを一時停止します。Extractは実行したままなので、ソース・データベースのデータを抽出し続けて、新しい証跡ファイル・データを作成します。

1. Replicatを停止します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X POST --data '{"name":"stop", "processName":"REP1", "force":false}' | python -m json.tool
```

2. Replicatのステータスを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X GET | python -m json.tool
```

Replicatのステータスが完全に停止するまでは、続行するのは危険です。出力例：

```
"sinceLagReported": 14, "status":  
"stopped"
```

保証付きリストア・ポイント (GRP) を作成する

ターゲット・データベース・データベースがマルチテナント・データベースである場合は、CDBからGRPを作成します。

CDBにSYSDBAユーザーとして接続し、次の文を実行します。

```
SQL> CREATE RESTORE POINT <GRP name> FOR PLUGGABLE DATABASE <PDB name> GUARANTEE FLASHBACK DATABASE;
```

単一テナント・データベースの場合は、次のコマンドを使用してリストア・ポイントを作成します。

```
SQL> CREATE RESTORE POINT <GRP name> GUARANTEE FLASHBACK DATABASE;
```

GRPの作成を確認します。

```
SQL> set linesize 140 pages 1000  
SQL> col con_id format 9999  
SQL> col pdb_restore_point format a20  
SQL> col name format a20  
SQL> col guarantee_flashback_database format a5  
SQL> col scn format 999999999999  
SQL> col time format a48  
SQL> alter session set nls_date_format='dd-Mon-yyyy hh:mi:ss';  
SQL> select con_id, name, scn, time, guarantee_flashback_database, pdb_restore_point  
from v$restore_point;
```

ターゲット・データベースを検証する

ターゲット・データベースをアプリケーションの検証やデータベース表に対するDML/DDLなど健全性チェックに使用できるようになりました。テスト期間は通常、2時間もかかるないはずです。

移行を開始する前に、包括的なパフォーマンスと機能のテストを完了させておくことをお奨めします。そのような包括的なテストに現在の移行期間を使うことには、次のリスクが伴います。

1. 移行期間が長引く。
2. GoldenGateに必要な時間が長引いて、ソース・データベース上で絶えず生成されるトランザクションのレプリケートに追い付かなくなる可能性がある。
3. ターゲット・システムと移行全体に悪影響を及ぼす問題が生じる。

テスト中、「[GoldenGateのレプリケーションを監視する](#)」の項に詳述されているように、GoldenGateのラグとチェックポイントの相違（現在のExtractの証跡ファイルと最後に適用されたReplicatの証跡ファイル）を監視し続けることをお奨めします

ターゲット・データベースをフラッシュバックする

テストが完了したら、ターゲット・データベースを閉じて、GRPにフラッシュバックし、再び開くことができます。

マルチテナント・データベースを実行している場合は、SYSDBAユーザーとしてCDBに接続している間、次のすべてのコマンドを実行する必要があります。

1. データベースをシャットダウンします。単一テナントの場合：

```
SQL> shutdown immediate
```

マルチテナントの場合：

```
SQL> ALTER PLUGGABLE DATABASE <PDB name> CLOSE IMMEDIATE;
```

2. データベースをフラッシュバックします。

単一テナントの場合：

```
SQL> FLASHBACK DATABASE TO RESTORE POINT <GRP name>;
```

マルチテナントの場合：

```
SQL> FLASHBACK PLUGGABLE DATABASE <PDB name> TO RESTORE POINT <GRP name>;
```

3. データベースを起動します。

単一テナントの場合：

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

マルチテナントの場合：

```
SQL> ALTER PLUGGABLE DATABASE <PDB name> OPEN RESETLOGS;
```

データベースをフラッシュバックしたら、テストおよび同じGRPへのフラッシュバックのサイクルを複数回繰り返すことができます。ただし、レプリケーションの一時停止が長ければ長いほど、レプリケーションに追い付くまでにかかる時間が長くなります。

保証付きリストア・ポイントを削除する

データベースを開いて、テストが完了したら、GRPを削除します。テスト・サイクルを複数回実行する場合は、テスト・サイクルを再び繰り返す前に、レプリケーションを再開し、GoldenGateが追い付くようにすることをお奨めします。

単一テナントの場合：

```
SQL> DROP RESTORE POINT <GRP name>;
```

マルチテナントの場合：

```
SQL> DROP RESTORE POINT <GRP name> FOR PLUGGABLE DATABASE <PDB name>;
```

レプリケーションを再開する

GoldenGate Replicatのプロセスを再起動して、レプリケーションを再開します。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X  
POST --data '{"name":"start","processName":"REP1"}' | python -m json.tool
```

GoldenGateのチェックポイントとラグの監視を続け、テストが完了したら、最終的なスイッチオーバーに進みます。

移行済みデータベースにスイッチオーバーする

テストが完了したら、ターゲット・データベースにスイッチオーバーできます。

GoldenGateのレプリケーション・ラグが許容できる程度に小さいかどうかを判別する

許容可能な低レプリケーション・ラグの長さは、ソース・ワークロードとネットワーク待機時間に応じて大きく異なります。

GoldenGate ExtractおよびReplicatの現在のチェックポイント位置とともにラグを監視して、両方のプロセスが同じ証跡ファイル内で動作していることを確認する必要があります。通常、同じ証跡ファイル内で動作している場合、ExtractとReplicat間の待機時間が少ないことを意味します。「[GoldenGateのレプリケーションを監視する](#)」の項で指定されているコマンドを使って、ExtractとReplicatが許容可能なラグに達したときを判別します。

ソース・データベースでのトランザクションの開始を停止する

データ損失ゼロのスイッチオーバーを完了するには、SRVCTLを使ってソース・データベースのサービスを停止することで、ソース・データベース上のトランザクションの発生を阻止します。

ソース・データベース上で新しい接続とトランザクションを禁止した後、データベース上のすべてのアクティブなトランザクションが完了するまで監視します。

次に例を示します。

```
SQL> select XIDUSN, XIDSLOT, XIDSQN, STATUS, USED_UBLK, USED_UREC from v$transaction where RECURSIVE='NO';
```

行が返らない場合、ソース・データベースがアイドル状態であることを推察できます。

Extractで未処理のトランザクションが完了したことを確認する

ソース・データベース上ですべてのトランザクションが停止したら、Extractを監視して、未処理の全トランザクションの抽出に確実に追いつくようにします。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1/info/checkpoints -XGET |
python -m json.tool
```

Extractの出力のチェックポイント位置はあまり変わらないはずで、GoldenGateハートビート表によってのみ60秒ごとに更新されます。

Extractを停止する

Extractがこれ以上前に進まないことがわかったら、Extractを停止できます。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/commands/execute -X
POST --data '{"name":"stop","processName":"EXT1","force":true}' | python -m json.tool
```

ステータスが停止したことを確認します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X GET
| python -m json.tool | grep status
```

Replicatがすべての証跡ファイル・データを適用するまで待機する

Extractが停止したら、Replicatが未処理の全証跡ファイル・データを適用するまで待つ必要があります。それには、LOGENDコマンドでReplicatを監視します。

```
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1/command -X POST --data '{"command":"LOGEND"}' |
python -m json.tool
```

出力例：

```
"replyData": {
    "$schema": "er:logEndResult",
    "allRecordsProcessed": true
}
```

Replicatが証跡ファイル・データをすべて適用すると、“true”的値のallRecordsProcessedが返されます。

ターゲット・データベースにスイッチオーバーする

レプリケートされたすべてのデータがターゲット・データベースに適用されたので、Replicatを停止することができます。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/commands/execute -X
POST --data '{"name":"stop","processName":"REP1","force":true}' | python -m json.tool
```

Replicatが停止したら、データベース・クライアントをターゲット・データベースにスイッチオーバーします。

移行プロセス中、ソース・データベースとターゲット・データベースの両方で自由に読み取り/書き込みを実行できるので、アプリケーション・データベース・サービスを有効にし、ターゲット・サービスを使用するようにアプリケーションを切り替える作業は、システム管理者またはデータベース管理者の判断で行われます。

読み取り専用アプリケーションの場合は、GoldenGate Replicatが未処理のソース・トランザクションすべてを適用して停止すると即座にスイッチオーバーが実行されるので、これらのサービスでアプリケーションの停止時間をゼロにすることができます。読み取り/書き込み可能なアプリケーションについては、最新データが確実に操作されるよう、アプリケーションをスイッチオーバーする前に、すべてのトランザクションがターゲット・データベースに適用済みであることを確認しなければならない場合があります。

必要な場合は、Oracle GoldenGate Veridataを使って、移行先データベースにスイッチオーバーする前に、同期外れのデータを見つけて修復することができます。詳しくは、Oracle GoldenGate Veridataのドキュメントを参照してください。

<https://docs.oracle.com/en/middleware/goldengate/veridata/12.2.1.2.0/books.html>

GoldenGateの構成の削除

Oracleデータベースの移行後は、次のコマンドを使用してGoldenGateの構成を削除できます。

GoldenGateプロセスを削除する

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/replicats/REP1 -X  
DELETE | python -m json.tool  
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X  
DELETE | python -m json.tool
```

次のコマンドを使用して証跡ファイルを削除します。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/commands/execute -X  
POST --data '{"name": "purge", "purgeType": "trails", "trails": [{"name": "aa"}], "useCheckpoints":  
false, "keep": [{"type": "min", "units": "files", "value": 0}]} | python -m json.tool
```

AUTOSTARTタスクを削除する

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/valu  
es/ogg:managedProcessSettings:MIGRATE01 -XDELETE | python -m json.tool  
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/config/types/ogg:managedProcessSettings/valu  
es/ogg:managedProcessSettings:MIGRATE01 -XDELETE | python -m json.tool
```

ターゲット・データベースのハートビート表を削除する

```
$ curl -s -K access.cfg  
https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/heartbe  
at -X DELETE | python -m json.tool
```

移行の一環としてハートビート表をソース・データベース内に作成した場合は、そのハートビート表も削除する必要があります。

```
$ curl -s -K access.cfg  
https://gghub.example.com/SOURCE/adminsrvr/services/v2/connections/goldengate.source_pdb/tables/he  
artbeat -X DELETE | python -m json.tool
```

チェックポイント表を削除する

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/connections/goldengate.target/tables/checkpoint -X POST --data '{"operation": "delete", "name": "ggadmin.gg_checkpoints"}' | python -m json.tool
```

Oracle GoldenGate資格証明を削除する

Oracle GoldenGate資格証明を削除しても、ユーザー・アカウントはデータベースから削除されません。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsrvr/services/v2/credentials/goldengate/target -X DELETE | python -m json.tool
```

次の例は、マルチテナント・ソース・データベースの場合です。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source_cdb -X DELETE | python -m json.tool
```



```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/credentials/goldengate/source_pdb -X DELETE | python -m json.tool
```

GoldenGateの資格証明が削除されたら、次のSQLコマンド例を使用してデータベースからユーザーを削除できます。

```
SQL> drop user ggadmin cascade;
```

Oracle GoldenGate Hubを削除する

GoldenGate Hubとそれに伴って作成されたデプロイメントが追加のデータベース移行でも不要になった場合は、GoldenGate InstanceをOracle Cloud環境から削除してください。次の手順に従ってOracle GoldenGate Instanceを削除します。

1. Oracle Cloudアカウントにログインします。
2. メニューから、リソース・マネージャ「Stack」を選択します。
自分のコンパートメントで構築されているスタックのリストを表示できます。
3. 削除するスタックのリンクをクリックします。
4. 「Stack」の「TerraForm Action」ドロップダウンリストで、「Destroy」を選択します。
これにより、Oracle GoldenGate Instanceが完全に削除されます。
5. スタックを削除します（任意）

オラクルの情報を発信しています

+1.800.ORACLE1までご連絡いただくなか、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 © 2020, Oracle and/or its affiliates. All rights reserved.本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による默示的保証を含め、商品性ないし特定目的適合性に関する默示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

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

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

Oracle GoldenGate Hub構成でのOracle Cloudへの移行 2020年4月

著者：Stephan Haisley

