



ORACLE

---

# Oracle Maximum Availability Architecture (Oracle MAA) GoldenGate Hub

---

2020年12月4日  
Copyright © 2020, Oracle and/or its affiliates  
機密性：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	3
はじめに	3
構成の概要	5
Oracle GoldenGate	5
Oracle GoldenGate Hub	6
Oracle GoldenGate HubのOracle MAA	6
NGINXリバース・プロキシ・サーバー	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
データベース・パッチ要件	9
Oracle GoldenGate用のデータベース構成	9
Oracle GoldenGateデータベース管理者アカウント	10
データベースのロール・ベース・サービスを作成する	11
Oracle Net Servicesの接続性	11
<b>Oracle GoldenGate Hub構成</b>	<b>12</b>
手順1 - Oracle Grid Infrastructure 19cソフトウェアをインストールする	13
手順2 - Oracle Clientソフトウェアをインストールする	13
手順3 - Oracle GoldenGateソフトウェアをインストールする	14
手順4 - Oracle Standalone Agentソフトウェアをインストールする	14
手順5 - NGINXリバース・プロキシ・サーバーをインストールする	15
手順6 - アプリケーションの仮想IPアドレス（VIP）を作成する	16
手順7 - Oracle ACFSファイル・システムのレプリケーションを構成する	17
手順8 - 初期Oracle GoldenGate Microservicesデプロイメントを構成する	25
手順9 - 追加のOracle GoldenGateデプロイメントを作成する	30
手順10 - Oracle Grid Infrastructureスタンドアロン・エージェントにOracle GoldenGate デプロイメントを登録する	31
手順11 - NGINXリバース・プロキシを構成する	33
<b>Oracle GoldenGate構成</b>	<b>36</b>
Oracle Net Servicesクライアント構成ファイルの作成	36
Oracle GoldenGate Extractプロセスの推奨事項	37
Oracle Data Guardデータ損失フェイルオーバーの考慮事項	37
Oracle GoldenGate Replicatプロセスの推奨事項	39
Oracle GoldenGateの配布パスの推奨事項	40
AUTOSTARTプロファイルを作成する	41
Oracle GoldenGateプロセスの監視	41
<b>Oracle GoldenGate Hubの計画停止と計画外停止の管理</b>	<b>43</b>
計画停止	43
Oracle GoldenGate Hubの計画外停止	45
<b>結論</b>	<b>47</b>
付録A：SSHデーモン再起動アクション・スクリプトの例	48
付録B：acfs_primary CRSアクション・スクリプト	50
付録C：acfs_standby CRSアクション・スクリプト	57
付録D：配布パス・ターゲット変更スクリプト	62
付録E：Oracle GoldenGateデプロイメント作成レスポンス・ファイルの例	64
付録F：トラブルシューティング	67
Oracle ACFS Replicationのトラブルシューティング	67
Oracle GoldenGateのトラブルシューティング	72

## 本書の目的と対象読者

本書の対象読者は、Oracle GoldenGateを使用しているか、Oracle GoldenGateの使用を計画している個人です。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っても適しています。

本書の主な意図は、可用性と災害に対する許容度が高くなるようにOracle GoldenGateハブを構成するための最適な方法に関して紹介し、指針を示すことです。Oracle GoldenGateをハブとして構成することは、Oracle GoldenGateの重要な処理と管理の負荷をデータベース・サーバーからオフロードするためのアーキテクチャの最適化を意味します。ハブでOracle MAAの高可用性およびディザスタ・リカバリのベスト・プラクティスを使用することが、Oracle MAAのプラチナ・リファレンス・アーキテクチャに対するおもな前提条件です。Oracle MAAのプラチナ・ソリューションについて詳しくは、Oracle Exadata MAA - Platinum Tier Focused Presentation (<https://www.oracle.com/jp/a/tech/docs/exadata-maa-platinum-focused-ja.pdf>) を参照してください。

このホワイト・ペーパーは、Oracle Cloud Infrastructure (OCI) Marketplace環境には適用されません。

## はじめに

最高レベルの可用性を実現し、計画外停止と計画メンテナンスの両方のアクティビティにおいて停止時間をゼロまたはほぼゼロにするため、多くの場合にお客様は、Oracle Real Application Clusters (Oracle RAC)、Oracle Active Data Guard、Oracle GoldenGateを組み合わせて使用します。従来Oracle GoldenGateは、Oracle GoldenGateプロセスの接続先となるデータベース・サーバー上にローカルにインストールされて実行されます。Oracle Grid Infrastructure Standalone Agent (Oracle XAG) と併用する場合、Oracle GoldenGateプロセスは、Oracle RACノード間での再配置またはフェイルオーバーをシームレスに行い、Oracle Active Data Guardのスイッチオーバーとフェイルオーバーに従うように構成できます。

ハブ構成でOracle GoldenGateを使用することにより、Oracle GoldenGateソフトウェアとプロセスをデータベース・サーバーから移動し、ターゲット・データベースに近いネットワーク上に置かれた別個のデータベースに配置します。Oracle GoldenGateプロセスとそれらのプロセスで操作されるデータベースとの間の接続性は、Oracle Net Servicesで管理されます。

Oracle GoldenGateのハブ構成には次の利点があります。

- バージョンをまたいで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。ソース・データベースは、Oracle GoldenGateのハブとは別のプラットフォーム (IBM AIXソース・データベース・ホストやLinux GoldenGateハブなど) にすることができます
- ソースおよびターゲットのデータベース・システムからの、Oracle GoldenGateソフトウェアのインストール、構成、ライフサイクル管理のオフロード
- 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
- 複数のソースおよびターゲット・データベースをサポートすることによるデータの同時レプリケート
- 必要に応じて、またはパフォーマンス上の理由により、複数のハブを使用可能 (すなわち、異なる大陸のデータベース間の双方向レプリケーション)

Maximum Availability Architectureの原理を念頭にハブを構成することにより、サイト障害イベントの発生時にプライマリ・ハブに切り換わるスタンバイ・ハブを作成することが可能です。

## 構成の概要

この項では、Oracle GoldenGate Microservices Architectureを使ったOracle MAA構成内のOracle GoldenGate Hubの主要コンポーネントを紹介します。

## 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リアルタイム・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1に、Oracle GoldenGate Microservices Architectureを示します。

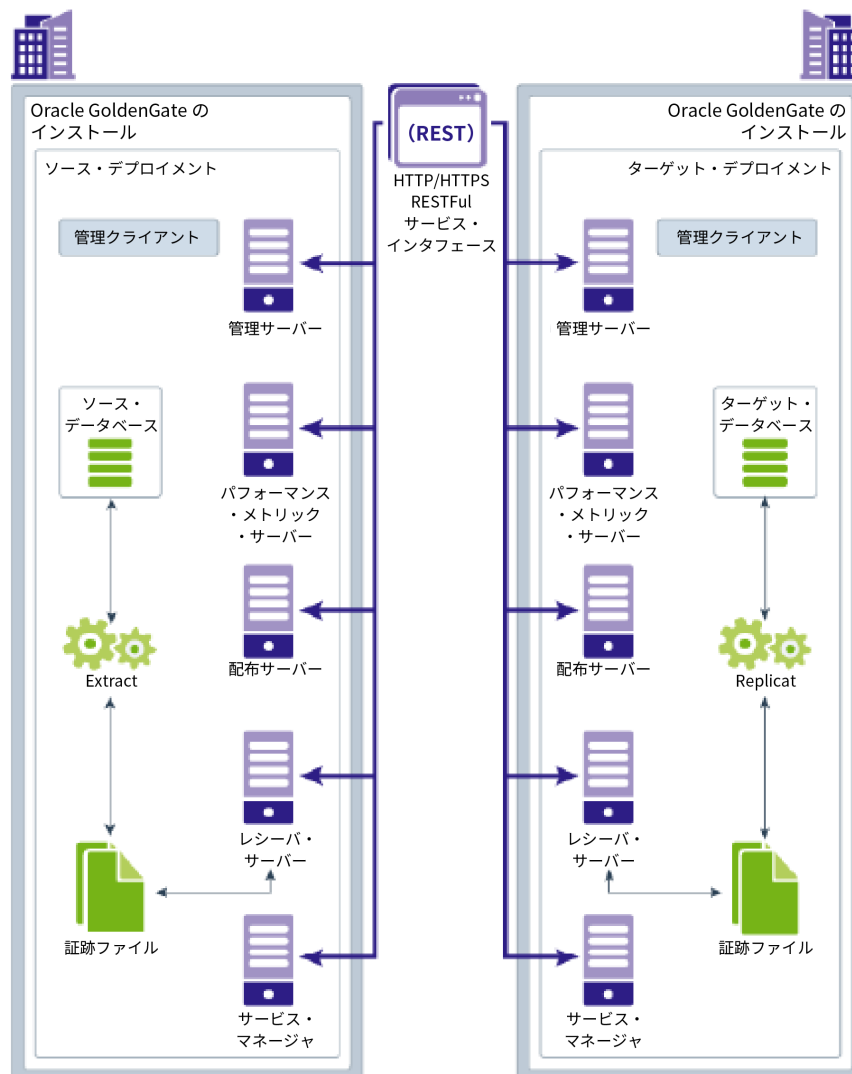


図1 : Oracle GoldenGate Microservices Architecture

Oracle GoldenGate Microservices Architectureについて詳しくは、以下のOracle GoldenGate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、図2に示すように、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概念です。ハブは、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、つまり通常は50キロメートル以内に配置する必要があります。このアーキテクチャの利点の1つは、Oracle 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、より多くのシステム・リソースをアプリケーションとデータベース処理に使用できることです。もう一つの利点は、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個のOracle GoldenGateにアクセスすることなく自己完結型の単一サーバーからOracle GoldenGateインフラストラクチャ全体を管理および監視することによって、Oracle GoldenGate構成が著しく簡素化されることです。

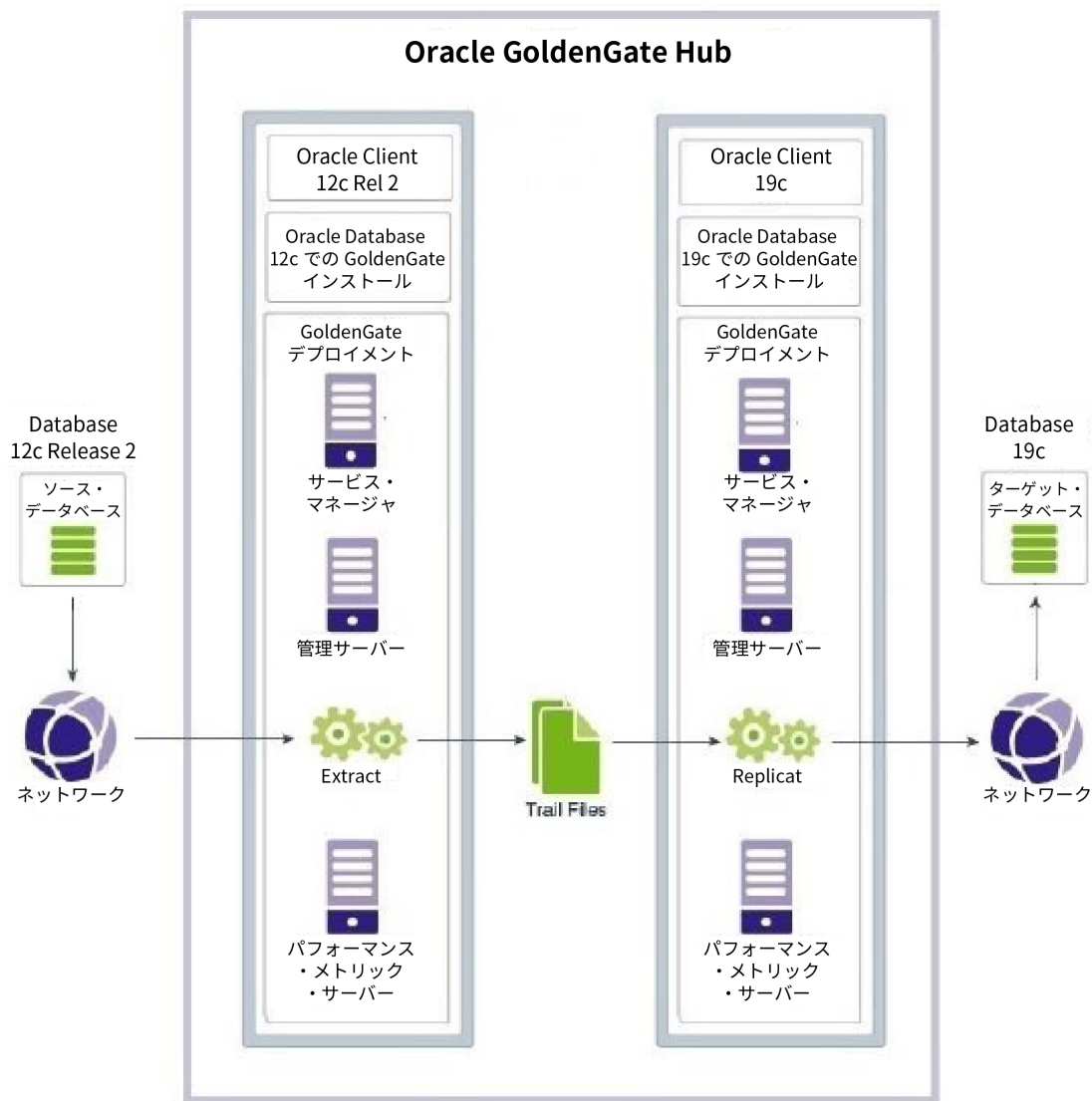


図2：Oracle GoldenGate Hubのアーキテクチャ

## GoldenGate HubのOracle MAA

災害によりローカルおよび全体的な障害が発生した場合の可用性を、ハブで最大限に高めるため、以下の図3に示すように、追加のOracleソフトウェア構成要件があります。

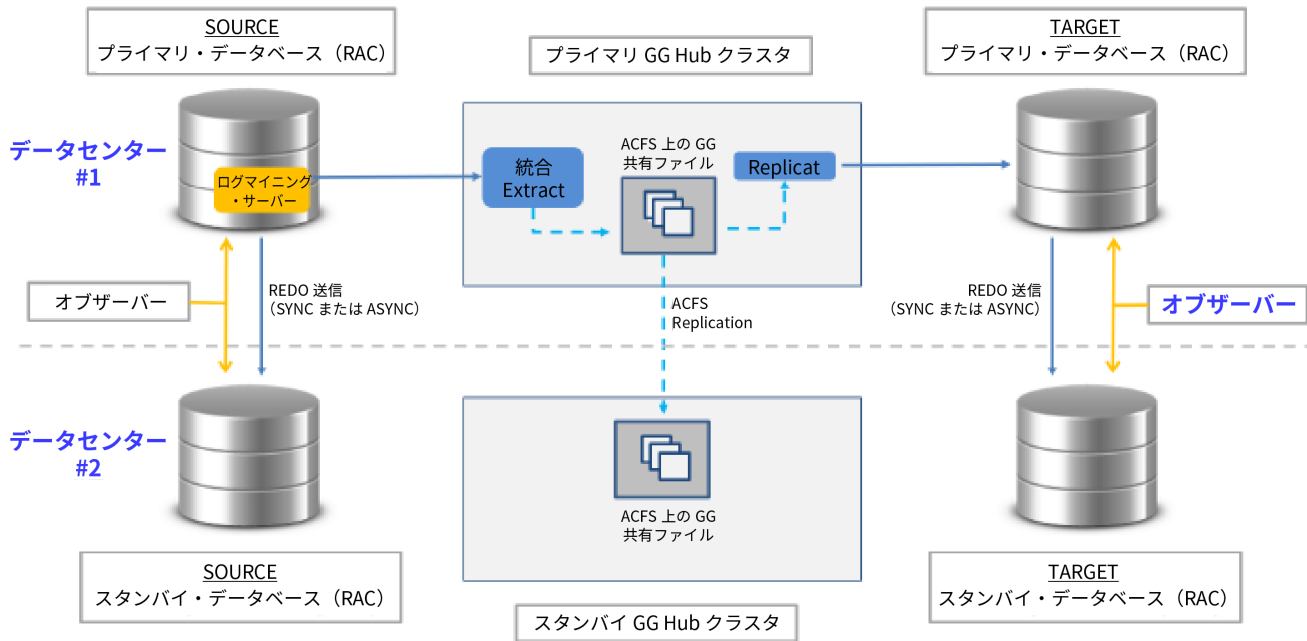


図3: Oracle MAAによるOracle GoldenGate Hub

以下の図4に、使用環境内でデータセンター障害が発生した後のOracle GoldenGateハブの変化を示します。

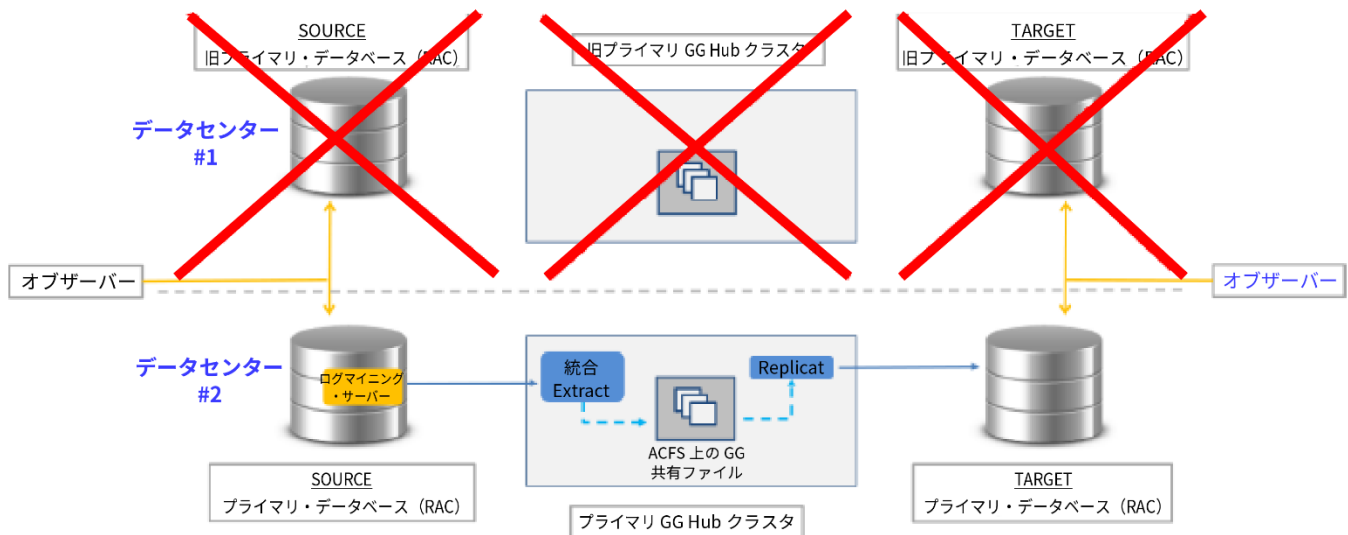


図4: 環境で障害が発生した後のOracle GoldenGate Hub

- プライマリとスタンバイのハブ・サーバーとターゲット・データベース・サーバーのラウンドトリップ待機時間は4ミリ秒未満です。双方向レプリケーションの場合、または別々のデータセンターに複数のターゲット・データベースが配置されている場合は、配布パスを持つハブを追加し、それらのハブ間で証跡ファイルを送信する必要があります。
- ハブ・サーバー上のLinux x86-64オペレーティング・システム。



- Oracle Grid Infrastructure 19cバージョン19.7以上。新規のOracle GoldenGateハブをデプロイする場合は、最新の本番 Oracle Grid Infrastructureソフトウェアの使用を推奨します。  
Oracle Grid Infrastructureにより、ビジネスに不可欠なあらゆるアプリケーションの高可用性（HA）の管理に必要なコンポーネントが提供されます。Oracle Clusterware（Oracle Grid Infrastructureのコンポーネント）ネットワークを使用することにより、データベースとOracle GoldenGateのリソースを管理して障害発生時の可用性を確保できます。この最小限のリリースは、Oracle ACFS（Oracle ASM Cluster File System）ReplicationとOracle GoldenGateをシームレスに統合する機能を提供するために必要です。
- 重要なOracle GoldenGateデプロイメント・ファイルを保管するためのOracle ACFS。
- 重要なOracle GoldenGateファイルをハブ・スタンバイ・サーバーにレプリケートするためのOracle ACFS Replication。  
Oracle ACFS Replicationにより、チェックポイント・ファイルや証跡ファイルなどのOracle GoldenGate共有ファイルに対する変更は、スタンバイ・ファイル・システムに自動的にレプリケートされます。
- Oracle Grid Infrastructure Standalone Agent（Oracle XAG）バージョン10.2以上。  
Oracle Grid Infrastructure Standalone Agentは、Oracle Grid Infrastructureのコンポーネントを利用して、Oracle GoldenGateと、データベース、ネットワーク、ファイル・システムなどのその従属リソースを統合することができます。このエージェントはまた、Oracle GoldenGateとOracle Data Guardを統合して、Oracle GoldenGateがロール移行に続いて新規のプライマリ・データベース上で再起動されるようにします。
- Oracle GoldenGate 19cバージョン19.1.0.4 Microservices以上。  
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します。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。

## NGINXリバース・プロキシ・サーバー

Oracle GoldenGateのリバース・プロキシ機能を使用すると、Oracle GoldenGateデプロイメントに関連付けられたすべてのOracle GoldenGateのマイクロサービスに対して単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、マイクロサービスには、各サービスに1つずつある、ホスト名またはIPアドレスと個別のポート番号で構成される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場合には`http://gghub.example.com:9100`を使用でき、管理サーバーの場合は`http://gghub.example.com:9101`、デプロイメントNo. 2の管理サーバーの場合は`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。上記の例で言えば、サービス・マネージャに接続する場合には、URLとして`https://gghub.example.com`を、デプロイメントNo. 1（名前：SOURCE）の管理サーバーに接続する場合には`https://gghub.example.com/SOURCE/adminsvr`を使用し、デプロイメントNo. 2（名前：TARGET）の管理サーバーの場合には`https://gghub.example.com/TARGET/adminsvr`となります。

Oracle Cloud Marketplaceからの接続では、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リバース・プロキシが推奨されており、Oracle GoldenGateではデフォルトで使用されます。

NGINXプロキシ・サーバーの構成については、このホワイト・ペーパーの後半で扱います。

## 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。自動構成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ハブにアクセス可能な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、Oracle GoldenGateのAdmin Clientコマンドを使ってハブ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・インタフェースです。Admin Clientコマンドについては、『Oracle GoldenGateコマンドライン・インタフェース・リファレンス』を参照してください。

[https://docs.oracle.com/cd/F22974\\_01/gclir/index.html](https://docs.oracle.com/cd/F22974_01/gclir/index.html)

この概要全体を通じて使用される、Oracle GoldenGateの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>/adminsrvr/services/v2/credentials/goldengate -XGET | python -m json.tool
```

- access.cfg：Oracle 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、次に示すように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推奨します。

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

- GG Hub：ハブ・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。
- デプロイメント名：これは、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。例：

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

このホワイト・ペーパーで例示されるExtract名はEXT1で、ReplicatはREP1といます。

RESTの呼び出しは、ハブから、またはHTTPSプロトコルを介してOracle GoldenGate Hubにアクセスできる任意のマシンから行うことができます。

## 構成の前提条件

### データベース・パッチ要件

Oracle GoldenGateでサポートされるデータベースの最小バージョンはOracle Database 11g Release 2 (11.2.0.4) です。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、データベース・バージョン11.2.0.4以上を対象とするMy Oracle Support Note 2193391.1に掲載されています。

ソース・データベースとハブ間のネットワーク・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、パッチ28849751もソース・データベースに適用することを推奨します。

現在のデータベース・ソフトウェアにパッチ28849751が適用されているかどうかを確認するには、次のコマンドを実行します。

```
$ $ORACLE_HOME/OPatch/opatch lsinventory|grep 28849751
28803345, 28819640, 28849751, 28852691, 28856060, 28891984, 28951382
```

## Oracle GoldenGate用のデータベース構成

ソース・データベースの前提条件は、Oracle MAAの技術概要『Oracle GoldenGate Performance Best Practices』 (<https://www.oracle.com/technetwork/database/availability/maa-gg-performance-1969630.pdf>) に記載されています。おもな前提条件は次のとおりです。

### ソース・データベースの前提条件

- 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、Oracle 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。

### ターゲット・データベースの前提条件

- 初期化パラメータのENABLE\_GOLDENGATE\_REPLICATIONを有効にすることで、Oracle GoldenGateレプリケーションを有効にする。

- Oracle GoldenGateの統合パラレルReplicatを使用している場合（「Oracle GoldenGate Replicatプロセスの推奨事項」の項を参照）、初期化パラメータSTREAMS\_POOL\_SIZEでストリーム・プールを構成する。
- 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。

## Oracle GoldenGateデータベース管理者アカウント

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。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。

Oracle GoldenGate管理者ユーザー・アカウントが存在しない場合は、管理者ユーザーを作成する必要があります。ソース・データベースの推奨されるユーザー名は、単一テナントとPDB（レプリケートされないPDBも含まれる場合）の場合にはGGADMIN、CDBデータベースとPDB（PDBが1つ存在するか、すべてのPDBがレプリケートされる場合）の場合にはC##GGADMINです。ターゲット・データベースの推奨ユーザー名は、GGADMINです。

Oracle GoldenGate管理者ユーザーがすでに存在する場合は、そのユーザー名をデータベースのOracle GoldenGate構成に使用してください。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。

この技術概要全体を通じて、データベースのGoldenGate管理者アカウントはGGADMINです。

次の例を使って、単一テナント・ソース・データベース、マルチテナント・データベースに存在するPDBのサブセット、ターゲット・データベースの新しいOracle GoldenGate管理者アカウントを作成します。

```
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');
```

マルチテナント・ソース・データベース内のすべてのPDBをレプリケートする場合は、次のコマンド例を使って、CDBとすべてのPDBでの共通ユーザーを作成します。

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

---

注：デフォルト名と一時表領域名は、CDBとすべてのPDBに必ず存在する必要があります。

---

マルチテナント・ソース・データベース内のPDBのサブセットをレプリケートする場合は、次のコマンド例を使ってCDBでユーザーを作成します。

```
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=>'<PDB name>');
```

## データベースのロール・ベース・サービスを作成する

Oracle GoldenGateのソースとターゲットのデータベースが、Oracle Data Guardを使ったOracle RACクラスタ上の推奨構成で実行されている場合は、ロール・ベースのサービスを作成して、Oracle GoldenGateのExtractまたはReplicatプロセスがOracle Data Guardの正しいプライマリ・データベース・インスタンスに接続できるようにする必要があります。プライマリとスタンバイ両方のOracle Data Guardクラスタで次のコマンドを実行して、ロール・ベースのサービスを作成します。

```
$ srvctl add service -db <database_service_name> -service oggsource -role PRIMARY  
-preferred <instance_name_1>,<instance_name_2>
```

例：

```
$ srvctl add service -db GGS2 -service oggsource -role PRIMARY -preferred GGS21,GGS22
```

---

注：Oracle Data Guardを使用していない場合はデータベース・サービスが必要になるため、`-role PRIMARY`パラメータを省略してください。

---

サービスは、以降の手順でOracle GoldenGateを構成する前に、次のコマンドを使用して開始する必要があります。

```
$ srvctl status service -d GGS2 -s oggsource  
$ srvctl start service -d GGS2 -i GGS21 -s oggsource
```

---

注：ロール・ベース・サービスは、ソースとターゲット両方のデータベースであるOracle Data GuardのプライマリとスタンバイのOracle RACクラスタで作成する必要があります。

---

## Oracle Net Servicesの接続性

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。

リモートOracle GoldenGate Extractのパフォーマンスを改善するため、より大きいサイズのSDUを使用するようにソース・データベース・リスナーを設定することを推奨します。リスナーで最大限のSDUサイズが設定されていても、クライアントがリクエストした場合には最小サイズが使用されます。たとえば、リスナーでのSDUの設定が2 MBでも、クライアントがデフォルトの8 KBのSDUをリクエストすると、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 ServicesまたはSecure Sockets Layer（SSL）の暗号化を使用することを推奨します。Oracle Net ServicesでのSSLについて詳しくは、『Oracle Databaseセキュリティ・ガイド』を参照してください。

[https://docs.oracle.com/cd/F19136\\_01/dbseg/configuring-secure-sockets-layer-authentication.html#GUID-6AD89576-526F-4D6B-A539-ADF4B840819F](https://docs.oracle.com/cd/F19136_01/dbseg/configuring-secure-sockets-layer-authentication.html#GUID-6AD89576-526F-4D6B-A539-ADF4B840819F)

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

```
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

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

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

ソース・データベースがOracle RAC上にある場合は、データベースのインスタンスを実行しているクラスター・ノードのすべてでリスナーを構成してください。これにより、1つのインスタンスが停止したとしても、サービスは稼働し続けているインスタンスに移行できます。すべてのOracle RACノード上でSSLウォレットを構成します。

リスナーを起動、停止、およびリスナー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、以下の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>
```

リスナーを起動するには、次のコマンドを実行します。

```
$ lsnrctl start GGSOURCE
```

リスナーを停止するには、次のコマンドを実行します。

```
$ lsnrctl stop GGSOURCE
```

リスナーの現在のステータスを取得するには、次のコマンドを実行します。

```
$ lsnrctl status GGSOURCE
```

## Oracle GoldenGate Hub構成

Oracle GoldenGate Hubには、以下の4つのソフトウェア・インストールと構成が必要です。

- Oracle Grid Infrastructure 19c、バージョン19.7以上
- ソースおよびターゲットのOracle Databaseバージョン（最小バージョンは11.2.0.4）を適合させるためのOracle Clientソフトウェア
- ソースおよびターゲットのOracle Databaseバージョンを適合させるためのOracle GoldenGateソフトウェア
- Oracle GoldenGate Microservicesデプロイメント接続で使用されるNGINXリバース・プロキシ・サーバー

以下の項では、必要なソフトウェアのインストールについて詳しく説明します。

---

**注：すべての命令は、プライマリとスタンバイ両方のハブ・サーバーで実行する必要があります。**

---

## 手順1 - Oracle Grid Infrastructure 19cソフトウェアをインストールする

プライマリとスタンバイ両方のハブ・サーバーにOracle Grid Infrastructure 19cをインストールしたら、最新リリースの更新をダウンロードして適用します。必須の更新バージョンの最小リリースは19.7です。

必ず、パッチ31697904をOracle Grid Infrastructureソフトウェア・ホームに適用し、ファイル・システムの計画外停止に対処するための最新のOracle ACFS Replicationの機能を確実に有効にする必要があります。この機能は、この技術概要で紹介するソリューションで必要になります。このパッチは、My Oracle Supportからダウンロードできます。

## 手順2 - Oracle Clientソフトウェアをインストールする

Oracle GoldenGateには、Oracle Client Runtimeインストールの一部としてインストールされるライブラリが必要です。インストールされるクライアント・ソフトウェア・バージョンは、Oracle GoldenGateの接続先データベースと同じリリース番号である必要があります。たとえば、ソースがOracle Database 12c Release 2で、ターゲットがOracle Database 19cである場合は、Oracle Database 12c用とOracle Database 19c用の2種類のクライアント・ソフトウェアのインストールが必要です。ソース・データベースとターゲット・データベースのリリースが同じ場合は、Oracle Clientソフトウェアのインストールが1つ必要です。

ソース・データベースがハブと同じオペレーティング・システム上で実行されていない場合は、ハブのプラットフォーム・オペレーティング・システム（Linux）向けのOracle GoldenGateソフトウェアをインストールしてください。

---

注：Oracle Clientソフトウェアは、プライマリとスタンバイ両方のハブ・サーバー上の一致するORACLE\_HOMEディレクトリにインストールする必要があります。Oracle Instant Clientソフトウェアは、必要なライブラリの一部が含まれていないため、使用しないでください。

---

Oracle Clientソフトウェアを<https://edelivery.oracle.com>からダウンロードします。Oracle Software Delivery Cloudでリリース・カテゴリを選択し、“Oracle Database Client”を検索します。

インストール時には、以下の例に示すように、「Runtime」インストール・タイプを選択します。

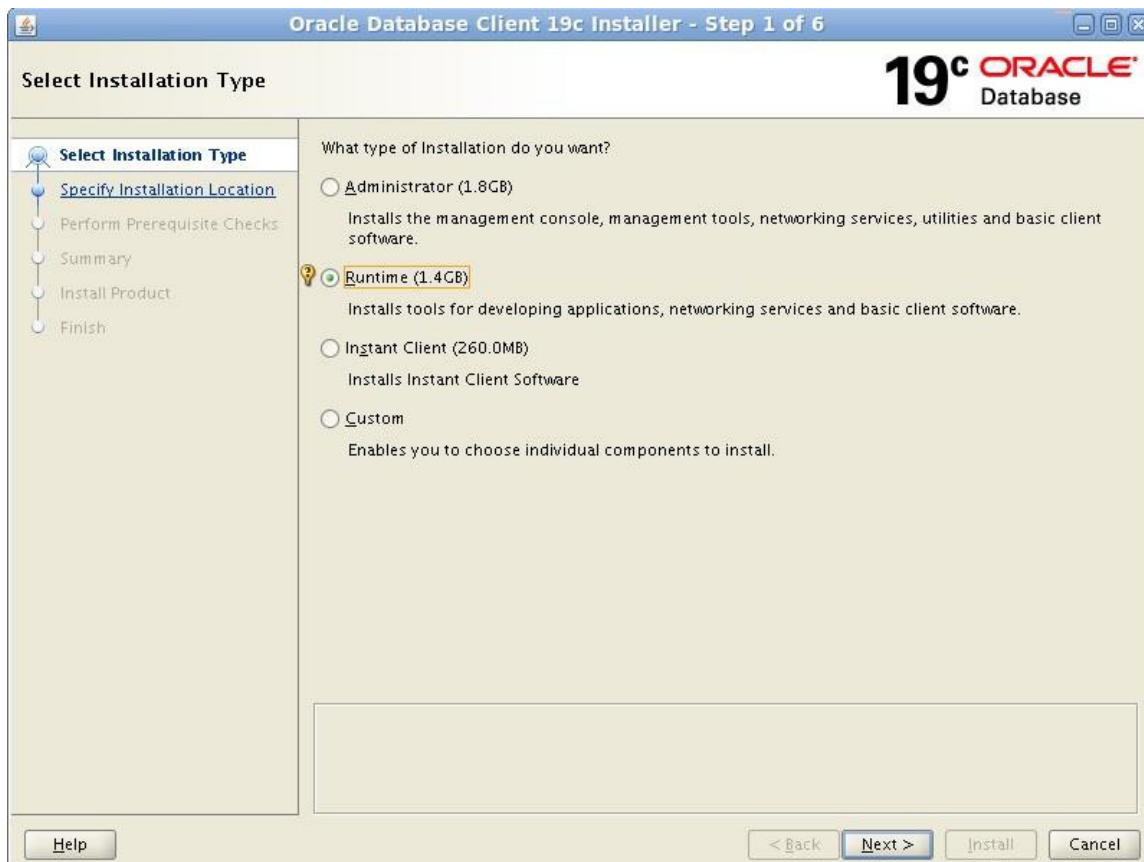


図4：Oracle Clientソフトウェアのインストール

ソースとターゲット両方のデータベース・リリースが一致するOracle Clientソフトウェアを別個のORACLE\_HOMEディレクトリにインストールします。

### 手順3 - Oracle GoldenGateソフトウェアをインストールする

ソースおよびターゲットのOracle DatabaseリリースをサポートするOracle GoldenGateソフトウェア・リリースをインストールする必要があります。Oracle GoldenGateソフトウェアの互換性マトリックスは、<https://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>に掲載されています。

最新のOracle GoldenGateソフトウェアを<https://www.oracle.com/jp/middleware/technologies/goldengate-downloads.html>からダウンロードします。入手可能な最新リリースのOracle GoldenGate（現時点では19c Release 1（19.1.0.0.4））をインストールすることを推奨します。Oracle GoldenGate 19c Release 1を使用すると、GoldenGate Extractを稼働するオペレーティング・システムのエンディアンネスがソース・データベース・プラットフォームのエンディアンネスと異なる場合に、エンディアン間抽出を行うことができます。

ソース・データベースとターゲット・データベースのリリースが異なる場合は、Oracle GoldenGateソフトウェアを2回（データベース・リリースごとに1回ずつ）インストールしてください。以下の図5に、データベース・リリースのソフトウェア・インストーラ・オプションを示します。

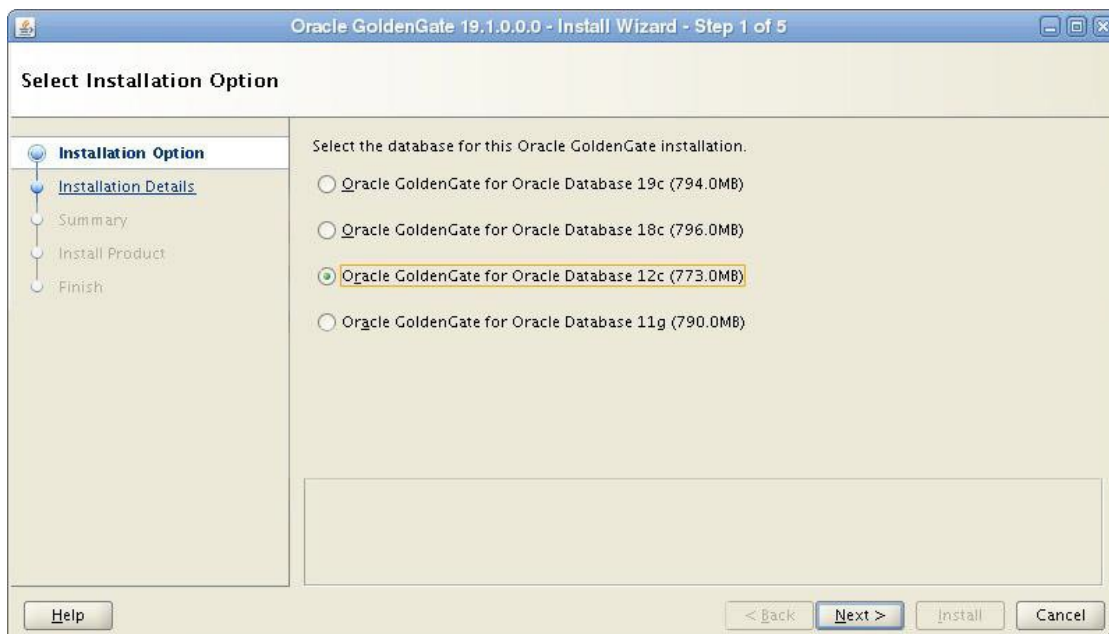


図5：Oracle Databaseバージョンに応じたOracle GoldenGateリリースを選択する

Oracle GoldenGateソフトウェアは、サポートするデータベース・バージョンに基づいて、別個のディレクトリにインストールすることが重要です。

---

注：Oracle GoldenGateの各ソフトウェア・インストールは、プライマリとスタンバイの各Oracle RACノード上の同じディレクトリ・パスに確実に配置してください。

---

### 手順4 - Oracle Standalone Agentソフトウェアをインストールする

次のステップ・バイ・ステップ手順は、Oracle Grid Infrastructure Standalone Agent（Oracle XAG）を使用してOracle GoldenGateを管理するようにOracle Clusterwareを構成する方法を示しています。Oracle XAGを使用すると、Oracle ACFSファイル・システムのマウントが自動化されると共に、Oracle RACノードとOracle Data Guardプライマリ・データベースの間での再配置の際に、Oracle GoldenGateデプロイメントの開始と停止が自動化されます。

- Oracle Grid Infrastructure Standalone Agentをインストールする

最新のOracle XAGソフトウェアを<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/xag-agents-downloads-3636484.html>からダウンロードします。

Oracle GoldenGate Microservices Architectureを使用する場合は、Oracle XAGバージョン10.2以上を使用することが要件です。

XAGソフトウェアは、ソフトウェアのパッチ適用との競合を避けるため、**必ず**、Oracle Grid Infrastructureのホーム・ディレクトリの外部にインストールする必要があります。Oracle XAGは、Oracle GoldenGateがインストールされているクラスタ内のすべてのOracle RACノードにおいて、同じディレクトリ・パスにインストールする必要があります。

たとえば、oracleユーザーとして次のように指定します。

```
$.xagsetup.sh --install --directory /u01/oracle/xag --all_nodes
```

- インストール済みのOracle XAGソフトウェアの場所をPATH変数に追加し、oracleユーザーに代わって自動的に設定されるようにします。

```
$ cat .bashrc
```

```
export PATH=/u01/oracle/xag/bin:$PATH
```

---

*注：Oracle XAGディレクトリは、正しいagctlバイナリが確実に検出されるようにするため、PATH変数で必ずOracle Grid Infrastructureディレクトリの前に指定することが重要です。*

---

## 手順5 - NGINXリバース・プロキシ・サーバーをインストールする

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に対して単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、Oracle GoldenGateデプロイメントのマイクロサービスには、各サービスに1つずつある、ホスト名またはIPアドレス、および個別のポート番号で構成される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場合には<http://gghub.example.com:9100>を使用でき、管理サーバーの場合は<http://gghub.example.com:9101>、デプロイメントNo. 2の管理サーバーの場合は<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。上記の例で言えば、サービス・マネージャに接続する場合には、URLとして<https://gghub.example.com>を、デプロイメントNo. 1 (名前：SOURCE) の管理サーバーに接続する場合には<https://gghub.example.com/SOURCE/adminsvr>を使用し、デプロイメントNo. 2 (名前：TARGET) の管理サーバーの場合には<https://gghub.example.com/Target/adminsvr>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、リバース・プロキシを推奨します。

Oracle GoldenGateのリバース・プロキシ機能では、NGINXリバース・プロキシを使用します。以下の手順は、リバース・プロキシの構成方法を示します。

1. NGINXがまだインストールされていないことを (rootとして) 確認します。

```
$ sudo rpm -qa |grep nginx
```

インストールされていない場合、何も返されません。  
NGINXがインストールされている場合、出力は次のようになります。

```
nginx-mod-http-xslt-filter-1.12.2-2.el7.x86_64
nginx-mod-http-image-filter-1.12.2-2.el7.x86_64
nginxfilesystem-1.12.2-2.el7.noarch
nginx-mod-mail-1.12.2-2.el7.x86_64
nginx-mod-http-perl-1.12.2-2.el7.x86_64
nginx-1.12.2-2.el7.x86_64
nginx-all-modules-1.12.2-2.el7.noarch
nginx-mod-http-geoip-1.12.2-2.el7.x86_64
nginx-mod-stream-1.12.2-2.el7.x86_64
```



2. NGINXがまだインストールされていない場合はインストールしてください。次の手順は、YUM (Yellowdog Updated Modified) がインストールされ、パッケージのインストールが容易になるように構成されていることを前提としています。

```
$ sudo yum install epel-release
$ sudo yum update
$ sudo yum install nginx
```

3. NGINXを起動します。

```
$ sudo systemctl start nginx
```

4. NGINXが実行されていることを確認します。

```
$ curl -I 127.0.0.1

HTTP/1.1 200 OK
Server: nginx/1.12.2
Date:Wed, 4 Apr 2019 21:48:43 GMT
Content-Type: text/html
Content-Length:3700
Last-Modified:Wed, 4 Apr 2019 05:06:50 GMT
Connection: keep-alive
ETag:"5abdc5ea-e74"
Accept-Ranges: bytes
```

5. マシンの起動時に自動起動するようNGINXを有効にします。

NGINXの自動起動を有効にするには、rootユーザーとして以下のコマンドを実行します。

```
# systemctl enable nginx
```

NGINXのインストールに関する全ドキュメントについては、<https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-open-source/>にアクセスしてください。

## 手順6 - アプリケーションの仮想IPアドレス (VIP) を作成する

各ハブ・クラスタには、プライマリOracle ACFS Replicationのプロセスで、ファイル・システムが現在マウントされている正しいハブ・スタンバイ・ノードにファイル・システム・データが確実に送信されるようにするため、専用アプリケーション仮想IPアドレス (VIP) が必要です。これは、同じノードにVIPとOracle ACFSのCRSリソースを併置することによって実現されます。VIPはOracle Clusterwareが管理するクラスタ・リソースで、ノード障害が発生すると別のクラスタ・ノードに移行されます。

アプリケーションVIPを作成する前に、次の2つの情報を確認しておく必要があります。

- 次のコマンドを使用して識別可能なネットワーク番号。

```
$ crsctl status resource -p -attr NAME,USR_ORA_SUBNET -w "TYPE = ora.network.type" |sort | uniq
NAME=ora.net1.network
USR_ORA_SUBNET=10.231.41.0
```

NAME=ora.net1.networkのnet1は、これがネットワーク1であることを示しています。

- システム管理者が指定した新しいアプリケーションVIPのIPアドレス。このIPアドレスは、上記の方法で特定されたクラスタ環境の同じサブネットの範囲内に存在する必要があります。

次のコマンド例を使用して、VIP CRSリソースを（rootとして）作成します。

```
# appvipcfg create -network=1 -ip=12.123.12.123 -vipname=gg_vip_prmy -user=oracle
```

アプリケーションVIPの作成について詳しくは、『Oracle Clusterware管理およびデプロイメント・ガイド』を参照してください。

[https://docs.oracle.com/cd/F19136\\_01/cwadd/making-applications-highly-available-using-oracle-clusterware.html#GUID-AFA2B0A1-479F-4DAB-A9D0-226C30260DD8](https://docs.oracle.com/cd/F19136_01/cwadd/making-applications-highly-available-using-oracle-clusterware.html#GUID-AFA2B0A1-479F-4DAB-A9D0-226C30260DD8)

VIPを作成したら、以下に示すようにVIPアドレスをthe /etc/ssh/sshd\_configファイルに追加してSSHデーモンを再起動することにより、Oracle ACFSのプライマリおよびスタンバイ・クラスタ上のVIPアドレスで着信接続要求をリスニングするようにSSHデーモンを構成します。

以下を/etc/ssh/sshd\_configファイルに追加します。

```
ListenAddress <VIP address>
```

SSHデーモンを再起動します。

```
# /bin/systemctl restart sshd
```

## 手順7 - Oracle ACFSファイル・システムのレプリケーションを構成する

ACFSファイル・システム・レプリケーションの構成は次のサブステップで構成されます。

- プライマリ・サーバーとスタンバイ・サーバーでファイル・システムを作成する
- Oracle ACFSとアプリケーションVIPの間でCRS依存関係を作成する
- SSHデーモンのCRSリソースを作成する
- Oracle ACFS Replicationを有効にする
- Oracle ACFS ReplicationのCRSアクション・スクリプトを作成する
- Oracle ACFSのスイッチオーバーとフェイルオーバーをテストする

### A. プライマリ・サーバーとスタンバイ・サーバーでファイル・システムを作成する

プライマリとスタンバイ両方のクラスタで同じサイズのファイル・システムを作成する必要があります。ファイル・システムは、ターゲット環境で問題が発生して新しい証跡ファイルを受信できなくなっても対応できるよう、最大12時間分の証跡ファイルを格納可能なサイズに設定し、証跡ファイル生成のための十分な領域を確保する必要があります。12時間分に必要な領域の量は、実際の本番データで証跡ファイルの生成率をテストすることによってのみ決定できます。

1. Oracle ACFSをマウント可能なプライマリとスタンバイのすべてのOracle RACノードで同じマウント・ポイントを作成します。

例（rootとして）：

```
# mkdir /mnt/acfs_rep1
# chown -R oracle:dba /mnt/acfs_rep1
```

2. ASMCMDを使用して、ファイル・システム・ボリュームを（oracleとして）作成します。

```
ASMCMD [+] > volcreate -G datac1 -s 1200G ACFS_REP1
ASMCMD [+] > volinfo --all
Diskgroup Name:DATAC1
Volume Name:ACFS_REP1
Volume Device: /dev/asm/acfs_rep1-89
State:ENABLED
Size (MB):1228800
Resize Unit (MB):64
```

```
Redundancy:MIRROR Stripe
Columns:8
Stripe Width (K):1024
```

3. ファイル・システムを（oracleとして）作成します。

```
$/sbin/mkfs -t acfs /dev/asm/acfs_rep1-89
```

4. ファイル・システムをOracle Grid Infrastructureに（rootとして）登録します。

```
# srvctl add filesystem -device /dev/asm/acfs_rep1-89 -volume ACFS_REP1 -diskgroup DATA1 -path /mnt/acfs_rep1 -user oracle -node <node1>,<node2> -autostart NEVER -mountowner oracle -mountgroup oinstall -mountperm 755
```

5. Oracle ACFSのマウントとOracle RACノード間での（oracleとしての）再配置をテストします。

```
$/srvctl start filesystem -d /dev/asm/acfs_rep1-89
$/srvctl status filesystem -d /dev/asm/acfs_rep1-89
$/srvctl relocate filesystem -diskgroup DATA1 -volume acfs_rep1
$/srvctl status filesystem -d /dev/asm/acfs_rep1-89
```

## B. Oracle ACFSとアプリケーションVIPの間でCRS依存関係を作成する

ファイル・システムが確実にVIPと同じOracle RACノードにマウントされるようにするには、次のコマンド例を使用してVIP CRSリソースをOracle ACFSリソースに依存関係として追加します。

---

*注：必ず、プライマリとスタンバイ両方のハブ・クラスタで次のコマンドを実行してください。*

---

1. VIPリソースを（oracleとして）停止します。

```
$/crsctl status resource gg_vip_prmy
```

2. VIPリソースの（oracleとしての）現在の開始と停止の依存関係を判断します。

```
$/crsctl stat res gg_vip_prmy -f|grep _DEPENDENCIES
START_DEPENDENCIES=hard(ora.net1.network) pullup(ora.net1.network)
STOP_DEPENDENCIES=hard(intermediate:ora.net1.network)
```

3. VIPリソースの開始と停止の依存関係を（rootとして）変更します。

```
# crsctl modify res gg_vip_prmy -attr "START_DEPENDENCIES='hard(ora.net1.network,ora.data1.acfs_rep1.acfs) pullup(ora.net1.network) pullup:always(ora.data1.acfs_rep1.acfs)'"
# crsctl modify res gg_vip_prmy -attr "STOP_DEPENDENCIES='hard(intermediate:ora.net1.network,ora.data1.acfs_rep1.acfs)'"
```

---

*注：acfs\_rep1は、正しいOracle ACFSボリューム名で置き換えてください。*

---

4. VIPを（oracleとして）再起動します。

```
$/crsctl start resource gg_vip_prmy

CRS-2672:Attempting to start 'ora.data1.acfs_rep1.acfs' on 'node1'
CRS-2676:Start of 'ora.data1.acfs_rep1.acfs' on 'node1' succeeded
CRS-2672:Attempting to start 'gg_vip_prmy' on 'node1'
CRS-2676:Start of 'gg_vip_prmy' on 'node1' succeeded
```

```
$ crsctl status resource gg_vip_prmy
```

```
NAME=gg_vip_prmy  
TYPE=app.appvtypex2.type  
TARGET=ONLINE  
STATE=ONLINE on node1
```

5. VIPおよびOracle ACFSマウントの再配置をテストします。

```
$ crsctl relocate resource gg_vip_prmy -f  
Or  
$ srvctl relocate filesystem -diskgroup DATAC1 -volume acfs_rep1 -force  
$ crsctl status resource gg_vip_prmy
```

```
NAME=gg_vip_prmy  
TYPE=app.appvtypex2.type  
TARGET=ONLINE  
STATE=ONLINE on node2
```

```
$ srvctl status filesystem -diskgroup DATAC1 -volume acfs_rep1  
ACFS file system /mnt/acfs_rep1 is mounted on nodes node2
```

## C. SSHデーモンのCRSリソースを作成する

Oracle ACFS Replicationでは、以前に作成された仮想IPアドレスに基づき、Secure Shell (SSH) を使用してプライマリとスタンバイのファイル・システム間の通信を行います。サーバーが再起動すると、VIP CRSリソースより先にSSHデーモンが起動し、VIPを使用したクラスタへのアクセスが阻止されます。以下の命令によって、仮想IPリソースの起動後にSSHデーモンを再起動するSSH再起動CRSリソースが作成されます。

1. 付録BのCRSアクション・スクリプト例を使用してSSHデーモンを再起動します。プライマリとスタンバイのすべてのOracle ACFSファイル・システム・ノードの同じ場所に、スクリプトを配置します。
2. rootユーザーとして、次のコマンド例を使用してCRSリソースを作成します。

```
crsctl add resource sshd_restart \  
-type cluster_resource \  
-attr "ACL='owner:root:rwx,pgrp:root:rwx,other::r--,group:oinstall:r-x,user:oracle:rwx',  
ACTION_SCRIPT=/u01/oracle/scripts/sshd_restart.scr,CHECK_INTERVAL=0,AUTO_START='always',  
START_DEPENDENCIES='hard(gg_vip_prmy) pullup:always(gg_vip_prmy)  
attraction(gg_vip_prmy)',STOP_DEPENDENCIES='hard(gg_vip_prmy)'"
```

3. VIP名gg\_vip\_prmyを「[手順6 - アプリケーションの仮想IPアドレス \(VIP\) を作成する](#)」で作成したVIP名で置き換えます。
4. 次のコマンドを使用して、現行のプライマリとスタンバイのOracle ACFSノード上のOracle CRSリソースをテストします。

```
$ crsctl stat res sshd_restart
```

```
NAME=sshd_restart  
TYPE=cluster_resource  
TARGET=OFFLINE  
STATE=OFFLINE
```

```
$ crsctl start res sshd_restart
```

```
CRS-2672:Attempting to start 'sshd_restart' on 'node1'  
CRS-2676:Start of 'sshd_restart' on 'node1' succeeded
```

```
$ ls -lrt /tmp/sshd_restarted
```

```
-rw-r--r-- 1 root root 0 Jun 29 12:33 /tmp/sshd_restarted

$ crsctl stop res sshd_restart

CRS-2673:Attempting to stop 'sshd_restart' on 'node1'
CRS-2677:Stop of 'sshd_restart' on 'node1' succeeded

$ ls -lrt /tmp/sshd_restarted

ls: cannot access /tmp/sshd_restarted:No such file or directory
```

## D. Oracle ACFS Replicationを有効にする

次の3つの手順を実行してOracle ACFS Replicationを有効にします。

### 1. プライマリ・ホストとスタンバイ・ホストの間のSSH接続性を構成する

Oracle ACFSのスナップショットベースのレプリケーションでは、sshを使用してプライマリ・ホストとスタンバイ・ホストの間でスナップショットを転送します。このレプリケーション・プロセスでは、プライマリ・ノードのrootユーザーを使用して、スタンバイ・ノードの最小権限ユーザーに接続します。

プライマリ・ノードとスタンバイ・ノードの間のssh接続性を構成するには、以下のサイトに掲載されているOracle ACFSの管理者ガイドを参照してください。

<https://docs.oracle.com/en/database/oracle/oracle-database/20/acfsg/configure-acfs-replication.html#GUID-C1CB3914-E278-473A-ACCC-FD1BA7CDC0B4>

プライマリとスタンバイのすべてのOracle RACノード間で確実に構成を完了してください。Oracle ACFS Replicationのロール・リバーサルに備えて、反対方向の接続性も構成してください。

ssh構成が完了したら、すべてのプライマリ・ノードとスタンバイ・ノード間の接続性をテストし、さらに、rootユーザーとしてsshとacfsutilコマンドの両方を使用して反対方向の接続性をテストします。

例（rootとして）：

```
# ssh oracle@snode1 uname -a

Linux snode1.example.com 4.1.12-124.26.12.el7uek.x86_64 #2 SMP Wed May 8 22:25:03 PDT 2019 x86_64 x86_64 x86_64 GNU/Linux

# ssh oracle@snode1.example.com uname -a
# ssh oracle@snode2.example.com uname -a
# ssh oracle@clust2-vip1.example.com uname -n

## 上記のsshテストをoracleユーザーとして繰り返す。これにも成功する必要がある。

# acfsutil repl info -c -u oracle snode1 snode2 clust2-vip1 /mnt/acfs_rep1

A valid 'ssh' connection was detected for standby node snode1 as user oracle.
A valid 'ssh' connection was detected for standby node snode2 as user oracle.
A valid 'ssh' connection was detected for standby node clust2-vip1 as user oracle.
```

---

注：すべてのプライマリ・ファイル・システム・ノードとすべてのスタンバイ・ノードの間で、および反対方向でも、確実に接続性を確認してください。どの接続テストにおいてもエラーが発生しなかった場合にのみ次に進んでください。

---

### 2. プライマリとスタンバイのファイル・システムでReplicationを初期化する

Oracle ACFSが現在マウントされているスタンバイ・ノード（oracleとして）の場合：

```
$/sbin/acfsutil repl init standby -u oracle /mnt/acfs_rep1
```

Oracle ACFSが現在マウントされているプライマリ・ノード（oracleとして）の場合：

```
$ /sbin/acfsutil repl init primary -C -p oracle@clust1-vip1 -s oracle@clust2-vip1 -m /mnt/acfs_rep1 /mnt/acfs_rep1
```

スタンバイ・ファイル・システムの初期化時に、プライマリ・ファイル・システムの内容がスタンバイ・ノードにコピーされます。この期間中は、次のコマンドを使用して初期化の進捗状況を監視できます。

```
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1 | grep Status
```

```
Status:                               Sending initial copy
```

ステータスが'Send Completed'に変わった場合は、初期のプライマリ・ファイル・システムのコピーが終了し、プライマリ・ファイル・システムがスタンバイ・ホストに現在レプリケート中であることを意味します。

acfsutilを使用したプライマリとスタンバイのファイル・システムでのレプリケーションの初期化について詳しくは、以下のサイトに掲載されているOracle ASMの管理者ガイドを参照してください。

<https://docs.oracle.com/en/database/oracle/oracle-database/19/ostmg/acfs-commands-replication.html#GUID-E96CD9A2-00BD-4673-8FCC-DBB28A9E2DCD>

### 3. レプリケートされたファイル・システムを検証および監視する

プライマリとスタンバイ両方のファイル・システム上に存在するOracle ACFS Replicationのステータスは、次のコマンドを実行して取得できます。プライマリ・ホストの場合：

```
$ acfsutil repl util verifystandby /mnt/acfs_rep1
```

```
verifystandby return code:0
```

スタンバイ・ホストの場合：

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1
```

```
verifyprimary return code:0
```

両方のコマンドとも、問題が検出されなければ、値0を返します。0以外の値が返された場合は、続行する前に、Oracle ACFS Replicationに関連した共通の問題の監視、診断、解決について、「[Appendix A – Troubleshooting Oracle ACFS Replication](#)」を参照してください。

Oracle ACFS Replicationの現在のステータスを監視するには、プライマリ・ハブとスタンバイ・ハブの両方で次のコマンド例を使用します。

```
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1
```

```
Site:                               Primary clust17-
Primary hostname:                   vip1
Primary path:                       /mnt/acfs_rep1
Primary status:                     Running
Background Resources:              Active

Standby connect string:             oracle@clust2-vip1
Standby path:                       /mnt/acfs_rep1
Replication interval:              0 days, 0 hours, 0 minutes, 0 seconds
Sending primary as of:              Tue Aug 04 12:24:03 2020
Status:                             Sending incremental differences
Retries made:                       0
Last send started at:               Tue Aug 04 12:24:03 2020
Last send completed at:             In progress
Next send starts at:                Tue Aug 04 12:24:03 2020
Replicated tags:
```

```
Data transfer compression: ssh strict      Off
host key checking:                    On
Debug log level:                       3
Replication ID:                        0xadc39f3b
```

## E. Oracle ACFS ReplicationのCRSアクション・スクリプトを作成する

Oracle ACFSのプライマリおよびスタンバイ・ファイル・システムの健全性を確認するため、CRSアクション・スクリプトを使用します。CRSアクション・スクリプトは、プライマリとスタンバイのファイル・システム・クラスタの各ノードのOracle Grid Infrastructureトレース・ファイル・ディレクトリ（/u01/app/grid/diag/crs/<node\_name>/crs/trace）に格納されているCRSトレース・ファイル（crsd\_scriptagent\_oracle.trc）にファイル・システムの健全性を報告します。

プライマリとスタンバイ両方のファイル・システム・クラスタで、2つのスクリプトが必要になります。1つは、ローカル・プライマリ・ファイル・システムを監視し、リモート・スタンバイ・ファイル・システムが使用可能であることを検証するためのスクリプト、もう1つは、ローカル・スタンバイ・ファイル・システムを監視し、リモート・プライマリ・ファイル・システムが使用可能であることを確認するスクリプトです。Oracle ACFSの監視を実装するためのスクリプト例が[付録B](#)と[付録C](#)に掲載されていますが、ご使用の環境に適合するように編集する必要があります。

続いて、2つのCRSアクション・スクリプトについて説明します。

### acfs\_primary

acfs\_primaryリソースは、現在のOracle ACFSマウントがプライマリ・ファイル・システムかどうかを確認し、スタンバイ・ファイル・システムがアクセス可能で、レプリケートされたデータを受信していることを確認します。このリソースは、Oracle GoldenGateがプライマリのGoldenGateハブでプロセスを開始可能かどうかを自動的に確認するために使用されます。スタンバイ・ファイル・システムがプライマリからアクセス可能ではない場合、[付録B](#)のスクリプト例では、スタンバイ・ファイル・システムの検証を複数回試み、crsd\_scriptagent\_oracle.trcトレース・ファイルに次の警告メッセージを出力します。

```
2020-11-16 15:15:01.506:[acfs_primary][1:40360:26097] [check] WARNING:STANDBY not accessible (attempt 1 of 3)
2020-11-16 15:15:14.535:[acfs_primary][1:40360:26097] [check] WARNING:STANDBY not accessible (attempt 2 of 3)
2020-11-16 15:15:27.611:[acfs_primary][1:40360:26097] [check] WARNING:STANDBY not accessible (attempt 3 of 3)
2020-11-16 15:15:27.611:[acfs_primary][1:40360:26097] [check] WARNING:Problem with STANDBY file system (error: 222)
```

プライマリおよびスタンバイ・ファイル・システムがオンラインで使用可能な場合は、次のメッセージがトレース・ファイルに出力されます。

```
2020-11-04 21:35:48.588:[acfs_primary][1:20735:25070] [check] SUCCESS:Remote STANDBY file system
/mnt/acfs_rep1 is ONLINE
2020-11-04 21:35:48.588:[acfs_primary][1:20735:25070] [check] SUCCESS:PRIMARY file system
/mnt/acfs_rep1 is ONLINE
```

acfs\_primaryリソースはプライマリとスタンバイ両方のホストで実行されますが、現在のファイル・システムがプライマリ・ファイル・システムで、スタンバイ・ファイル・システムがアクセス可能になっている場合のみSUCCESSを返します。このスクリプトは、プライマリとスタンバイのすべてのファイル・システム・ノードにおいて同じ場所に配置する必要があります。

以下のパラメータでは推奨されるデフォルト設定を使用しています。値を変更する場合は、まずテストする必要があります。

```
MOUNT_POINT=/mnt/acfs_rep1    # レプリケートされたOracle ACFSマウント・ポイント
ATTEMPTS=3                    # リモート・スタンバイ・ファイル・システムの確認試行回数
INTERVAL=10                   # 試行間隔（秒）
```

プライマリとスタンバイ両方のクラスタの1つのノードで次のコマンドを使用して、プライマリおよびスタンバイ・ファイル・システムを監視するためのOracle ACFSアクション・スクリプトを登録します。

```
$ crsctl add resource acfs_primary -type cluster_resource \  
-attr "ACTION_SCRIPT=/u01/oracle/scripts/acfs_primary.scr,CHECK_INTERVAL=60,  
START_DEPENDENCIES='hard(ora.datac1.acfs_rep1.acfs)' pullup:always(ora.datac1.acfs_rep1.acfs)',  
STOP_DEPENDENCIES='hard(ora.datac1.acfs_rep1.acfs)',SCRIPT_TIMEOUT=80,OFFLINE_CHECK_INTERVAL=0,  
RESTART_ATTEMPTS=0"
```

開始し、プライマリとスタンバイ両方のクラスタでリソースのステータスを確認します。

```
$ crsctl start res acfs_primary  
$ crsctl stat res acfs_primary
```

acfs\_primary.scrアクション・スクリプト例については、[付録B](#)を参照してください。

## acfs\_standby

acfs\_standbyリソースは、ローカル・ファイル・システムがスタンバイ・ファイル・システムであることを確認し、リモート・プライマリ・ファイル・システムのステータスを検証します。プライマリ・ファイル・システムの検証（アクション・スクリプトの変数によって制御）に複数回失敗すると、Oracle Grid Infrastructureのトレース・ファイル・ディレクトリ（/u01/app/grid/diag/crs/<node\_name>/crs/trace）に格納されているCRSトレース・ファイル（crsd\_scriptagent\_oracle.trc）に警告が出力されます。

このリソースはプライマリとスタンバイ両方のホストで実行されますが、現在のファイル・システムがスタンバイ・ファイル・システムで、プライマリ・ファイル・システムがアクセス可能になっている場合にのみSUCCESSを返します。

スタンバイ・ファイル・システムとリモート・プライマリ・ファイル・システムがオンラインの場合は、次のメッセージがCRSトレース・ファイルに出力されます。

```
2020-11-04 21:39:46.752:[acfs_standby]{1:21041:20211} [check] SUCCESS:STANDBY file system  
/mnt/acfs_rep1 is ONLINE  
  
2020-11-04 21:39:47.453:[acfs_standby]{1:21041:20211} [check] SUCCESS:Remote PRIMARY file system  
/mnt/acfs_rep1 is ONLINE
```

プライマリ・ファイル・システムで問題が検出されると、次のメッセージがトレース・ファイルに出力されます。

```
2020-11-04 21:50:55.967:[acfs_standby]{1:21041:20211} [check] WARNING:Remote PRIMARY file system problem detected (attempt 1  
of 3)  
2020-11-04 21:51:08.989:[acfs_standby]{1:21041:20211} [check] WARNING:Remote PRIMARY file system problem detected (attempt 2  
of 3)  
2020-11-04 21:51:22.061:[acfs_standby]{1:21041:20211} [check] WARNING:Remote PRIMARY file system problem detected (attempt 3  
of 3)  
2020-11-04 21:51:22.061: [acfs_standby]{1:21041:20211} [check] WARNING:Problem with remote PRIMARY file system (error: 222)
```

以下のパラメータでは推奨されるデフォルト設定を使用しています。値を変更する場合は、まずテストする必要があります。

```
MOUNT_POINT=/mnt/acfs_rep1 # これはレプリケートされたOracle ACFSマウント・ポ  
ATTEMPTS=8 # リモート・プライマリ・ファイル・システムの確認試行回数  
INTERVAL=10 # 試行間隔（秒）
```



プライマリとスタンバイ両方のクラスタの1つのノードで次のコマンドを使用して、スタンバイ・ファイル・システムを監視するためのOracle ACFSアクション・スクリプトを登録します。

```
$ crsctl add resource acfs_standby \  
-type cluster_resource \  
-attr "ACTION_SCRIPT=/u01/oracle/scripts/acfs_standby.scr,CHECK_INTERVAL=150,  
CHECK_TIMEOUT=140,START_DEPENDENCIES='hard(ora.datac1.acfs_rep1.acfs,gg_vip_prmy)  
pullup:always(ora.datac1.acfs_rep1.acfs,gg_vip_prmy)',OFFLINE_CHECK_INTERVAL=300,RESTART_ATTEMPTS=0  
,INSTANCE_FAILOVER=0"
```

開始し、プライマリとスタンバイ両方のクラスタでリソースのステータスを確認します。

```
$ crsctl start res acfs_standby  
$ crsctl stat res acfs_standby
```

acfs\_standby.scrアクション・スクリプト例については、[付録C](#)を参照してください。

---

注：Oracle ACFSファイル・システムがプライマリ・ファイル・システムの場合、acfs\_primaryリソースのステータスはONLINEのみです。現時点でプライマリ・クラスタ上に存在しないノードでリソースを開始すると、スタンバイ・ファイル・システムであることによってリソースが失敗するため、エラーが報告されます。このエラーは無視して構いません。リソースは、Oracle ACFSスタンバイ・クラスタでステータスがOFFLINEとなります。acfs\_standbyリソースをプライマリ・ファイル・システム・クラスタで開始した場合にも、同じ結果になります。

---

CRSトレース・ファイル・メッセージに基づいたユースケース例については、「Oracle GoldenGate Hubの計画停止と計画外停止の管理」を参照してください。

## F. Oracle ACFS RACノードの再配置をテストする

まず計画および計画外のOracle ACFS RACノードの再配置とサーバー・ロール移行をテストし、それからOracle GoldenGateを構成することは非常に重要です。

プライマリのOracle RACノード間でOracle ACFSを再配置するには、次のコマンド例を実行します。

```
$ srvctl relocate filesystem -diskgroup DATA1 -volume acfs_rep1 -force
```

ノードの再配置後は、次のコマンド例を使用して、VIP、NGINX、sshd\_restart、2つのOracle ACFSリソース（acfs\_primaryとacfs\_standby）と共に、別のノードにファイル・システムがマウントされていることを検証してください。

```
$ srvctl status filesystem -diskgroup DATA1 -volume acfs_rep1 # ユーザーのdiskgroup名とvolume名を使用する  
$ crsctl stat res gg_vip_prmy # ユーザーのVIP名で置き換える  
$ crsctl stat res sshd_restart  
$ crsctl stat res acfs_primary  
$ crsctl stat res acfs_standby
```

プライマリとスタンバイのクラスタ間でOracle ACFSスイッチオーバー（ロール・リバーサル）を発行するには、現行スタンバイ・ノードで次のコマンド例を実行します。

```
# スタンバイ・ファイル・システム・ノードを特定する  
$ crsctl stat res gg_vip_prmy # ユーザーのVIP名で置き換える  
$ acfsutil repl failover /mnt/acfs_rep1 # Oracle ACFSマウント・ポイントを置き換え
```

プライマリ・ファイル・システムが現在使用可能であれば、このフェイルオーバー・コマンドによって、データ損失なしのOracle ACFSロール・リバーサルが正常に実行されます。プライマリ・ファイル・システムが使用不可の場合、Oracle ACFSフェイルオーバーが実行され、これによりファイル・システムのデータ損失が発生する可能性があります。Oracle GoldenGate製品にはリカバリ・メカニズムが組み込まれているため、これは問題とはなりません。

## G. Oracle ACFSトレース・ファイル管理

Oracle ACFSのレプリケーション機能により大量のトレース・ファイルを作成し、デフォルトのCRSトレース・ファイル・ディレクトリ

このトレース・ファイル・ディレクトリは、/u01/app/grid/diag/crs/<node\_name>/crs/traceとなります。定期的にトレース・ファイルを削除するcrontabタスクを作成することを推奨します。一部のトレース・ファイルはrootユーザーが生成するため、crontabタスクはrootユーザーが実行する必要があります。

たとえば、Oracle ACFSによって8時間以上に生成された古いトレース・ファイルを削除するには、次のcrontabエントリ例を12時間ごとに実行します。

```
00,12 * * * /usr/bin/find /u01/app/grid/diag/crs/<node_name>/crs/trace -maxdepth 1 -name 'sdcreate*' -mmin +480 -delete
00,12 * * * /usr/bin/find /u01/app/grid/diag/crs/<node_name>/crs/trace -maxdepth 1 -name 'sdapply*' -mmin +480 -delete
00,12 * * * /usr/bin/find /u01/app/grid/diag/crs/<node_name>/crs/trace -maxdepth 1 -name 'acfsutil*' -mmin +480 -delete
```

## 手順8 - 初期Oracle GoldenGate Microservicesデプロイメントを構成する

Oracle ClientソフトウェアとGoldenGateソフトウェアのインストールと同様に、運用対象のデータベースごとに個別のGoldenGateデプロイメントが必要です。ソース・データベースとターゲット・データベース双方のリリースが同じ場合は、1つのデプロイメントのみ必要です。

各デプロイメントは、管理サーバーと（オプションの）パフォーマンス・メトリック・サーバーで作成されます。Oracle GoldenGateの証跡ファイルを別のハブまたはOracle GoldenGate環境に移動する必要がない場合は、証跡ファイルがReplicatプロセスと同じサーバーに保存されるため、ディストリビューション・サーバーまたはレシーバ・サーバーを作成する必要はありません。

最初のデプロイメントの作成によってOracle GoldenGateサービス・マネージャが1つ作成されます。その後のデプロイメントは、同じ使用中のOracle GoldenGateホーム・ディレクトリに接続している場合、既存のサービス・マネージャを使って作成されます。

Oracle GoldenGateデプロイメントを作成する方法は次の2つです。

1. デプロイメントを作成するためのグラフィカル・ツールであるOracle GoldenGate Configuration Assistant (oggca.sh) を使用する
2. すべてのデプロイメント作成パラメータ値を含むレスポンス・ファイルをサイレント・モードで使用する

どちらのデプロイメント方法を使用しても結果は同じですが、複数のデプロイメントを作成する場合は、レスポンス・ファイルとともに選択肢2を使用するほうが簡単です。[付録E](#)には、2つのデプロイメントの作成に必要な変更をパラメータに加えたレスポンス・ファイルの例を掲載しています。デプロイメントのうちの1つは、ソース・データベースとターゲット・データベースのリリースが異なり、Oracle GoldenGate 19cを使用する場合です。Oracle GoldenGateソフトウェアのインストールには、デフォルトのレスポンス・ファイルは含まれていません。そのため、oggca.shまたは[付録E](#)にあるコマンド例を使用して初期デプロイメントを作成する必要があります。

両方のデプロイメント作成方法において、以下の推奨設定を使用してください。

- Service Manager Options
  - 上記の[手順7](#)で作成したOracle ACFSマウント・ポイントに、サービス・マネージャ・デプロイメントのホームを配置します。これは、レスポンス・ファイル・パラメータSERVICEMANAGER\_DEPLOYMENT\_HOMEによって制御されます。
  - 「Integrate with XAG」を選択します。これは、レスポンス・ファイル・パラメータINTEGRATE\_SERVICEMANAGER\_WITH\_XAGによって制御されます。
  - Ipv6構成の場合は、リスニング・ホスト名/アドレスを'localhost'または ':::1' に設定します。これは、レスポンス・ファイル・パラメータHOST\_SERVICEMANAGERによって制御されます。localhostリスニング・アドレスを使用して、サービス・マネージャがプライマリとスタンバイ両方のOracle GoldenGate Hubクラウドで起動するようになります。

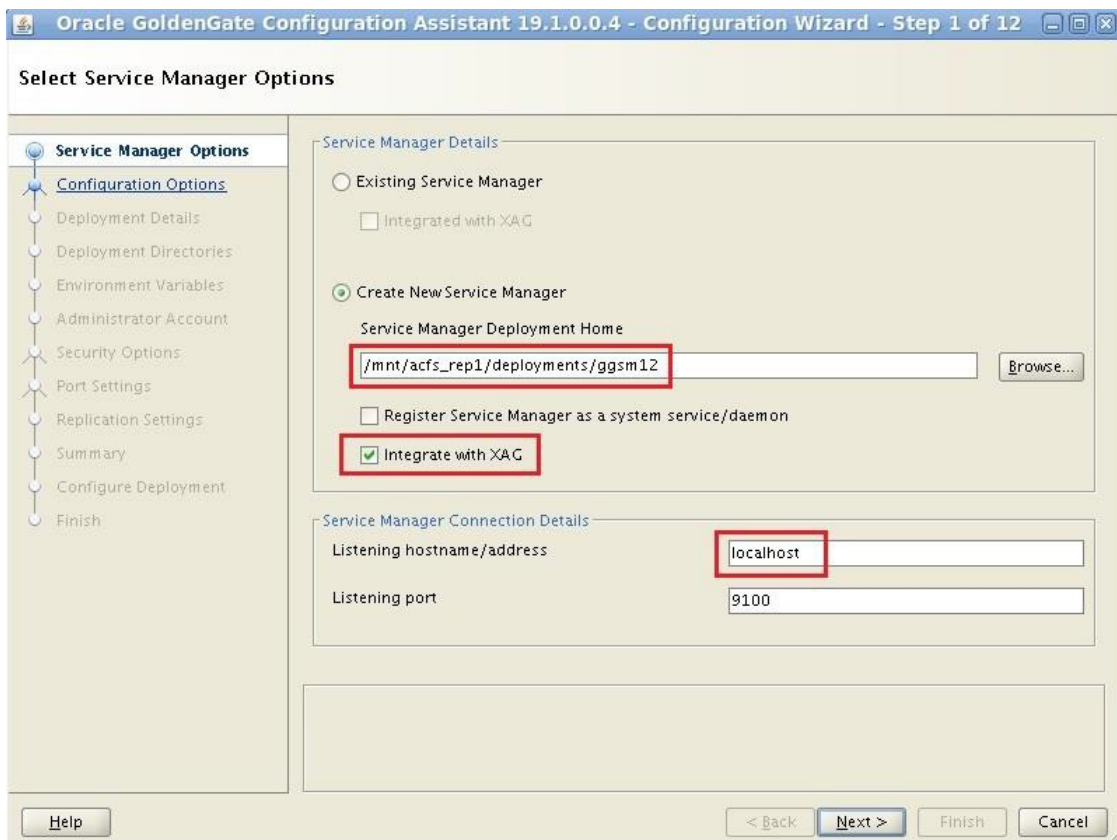


図6：サービス・マネージャ・デプロイメントの作成

- Deployment Directories (図7を参照)
  - 上記の手順7で作成したOracle ACFSマウント・ポイントに新しいデプロイメント・ホームを配置します。これは、レスポンス・ファイル・パラメータOGG\_DEPLOYMENT\_HOMEによって制御されます。

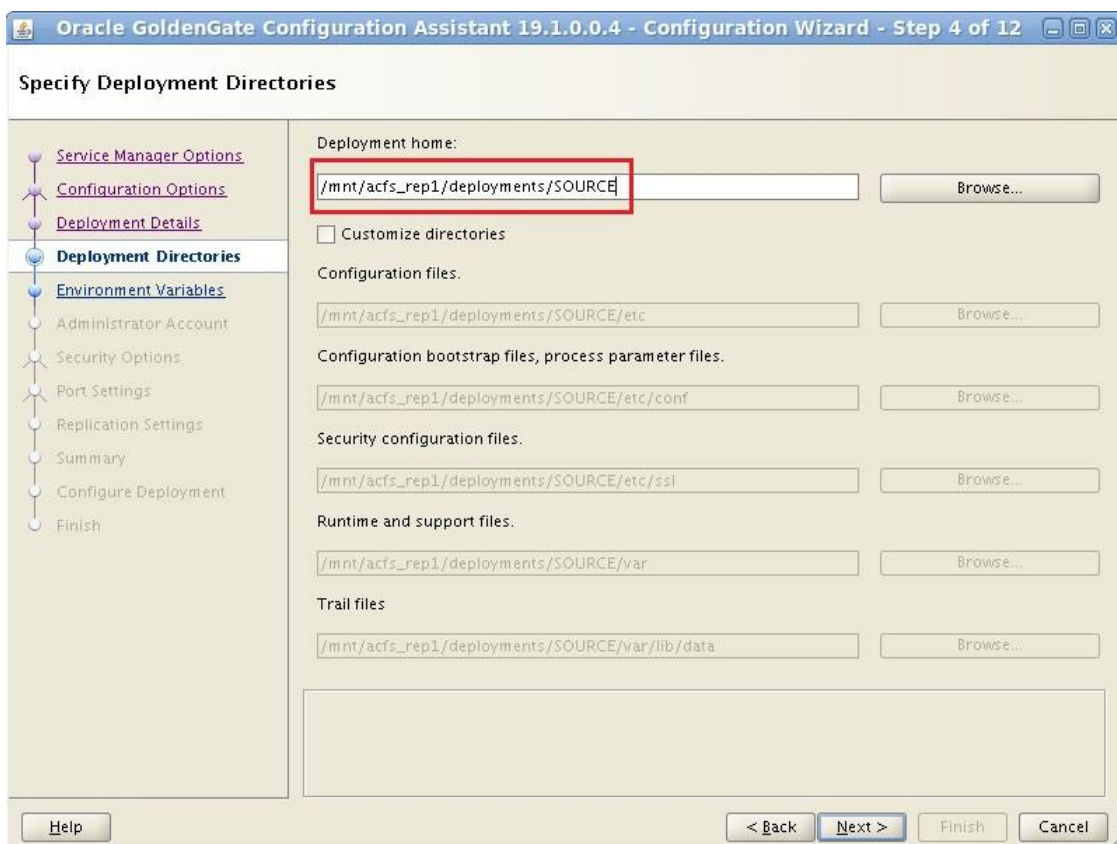


図7：デプロイメントの作成場所

- Administrator Account (図8を参照)
  - デプロイメントでのOracle XAGおよびREST呼出しで使用するため、Oracle GoldenGate管理者アカウントのユーザー名とパスワードを書き留めます。これらは、レスポンス・ファイルのパラメータADMINISTRATOR\_USERおよびADMINISTRATOR\_PASSWORDによって制御されます。

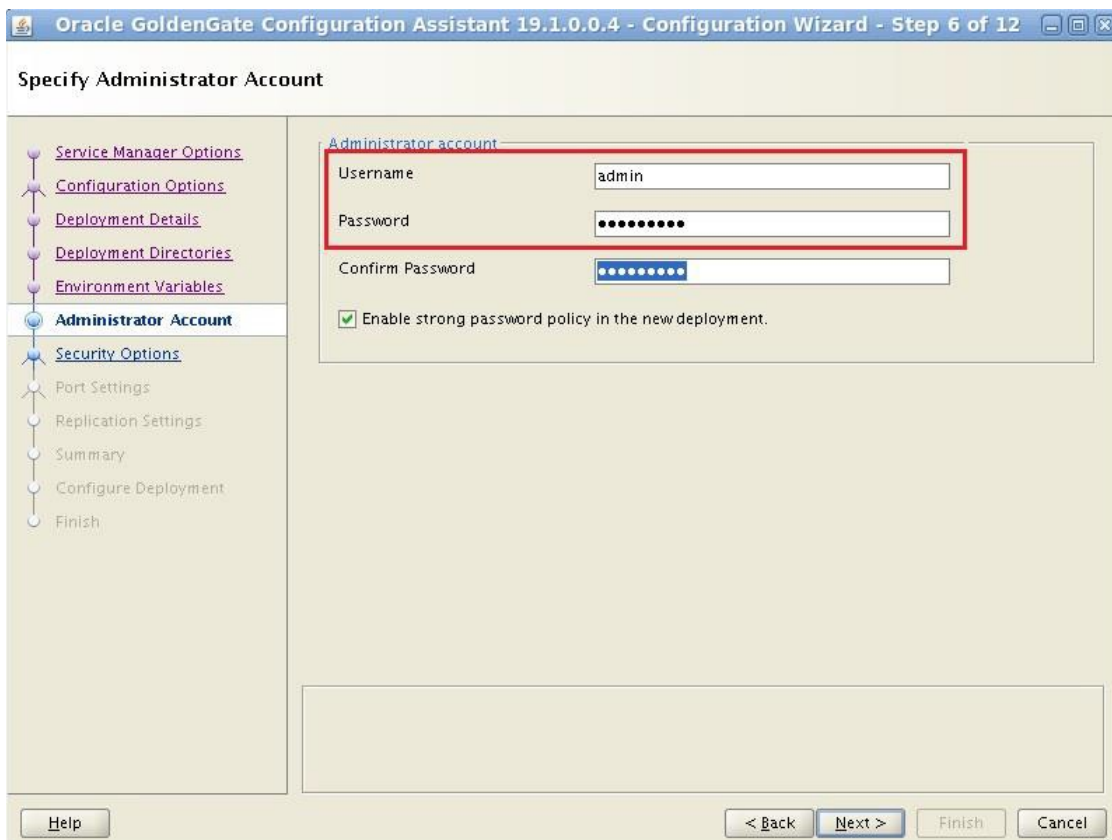


図8：デプロイメントの管理者アカウント

- Security Options
  - 「SSL/TLS Security」は選択しないでください。NGINXリバース・プロキシにより、デプロイメントのアクセス・セキュリティが提供されます。
- Port Settings（図9を参照）
  - 監視を有効にし、Berkley DB（BDB）を使用する場合は、Metrics Server DataStore homeをOracle ACFS Replicationのマウント・ポイント以外の場所に設定します。あるいは、DataStoreタイプとしてLightning Memory-Mapped Database（LMDB）を使用します。これは、レスポンス・ファイル・パラメータPMSRVR\_DATASTORE\_TYPEとPMSRVR\_DATASTORE\_HOMEによって制御されます。

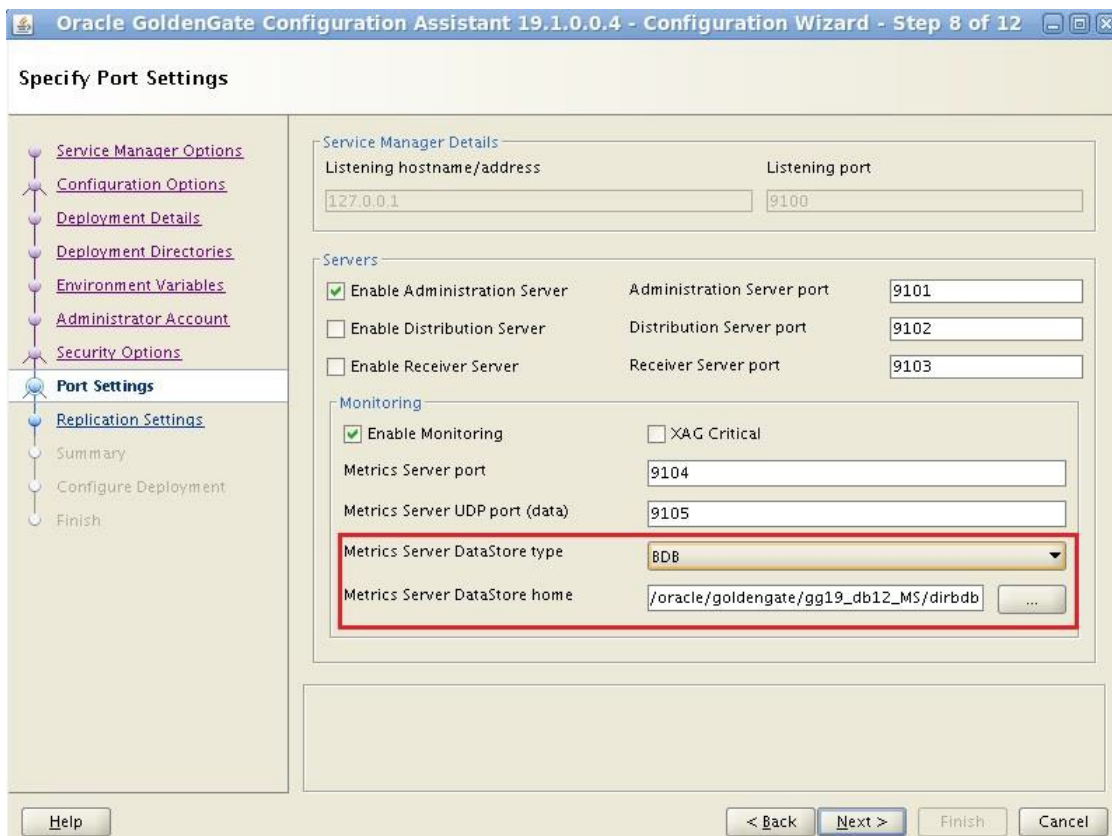


図9：Metric Server DataStoreのパラメータ

Oracle GoldenGate Configuration Assistant (oggca.sh) を使用して初期デプロイメントを作成する場合は、将来のデプロイメント作成に備えて、テンプレートとして使用するレスポンス・ファイルを保存することをお勧めします。図10は、レスポンス・ファイルを保存することができるOracle GoldenGate Configuration Assistantの最終画面を示しています。

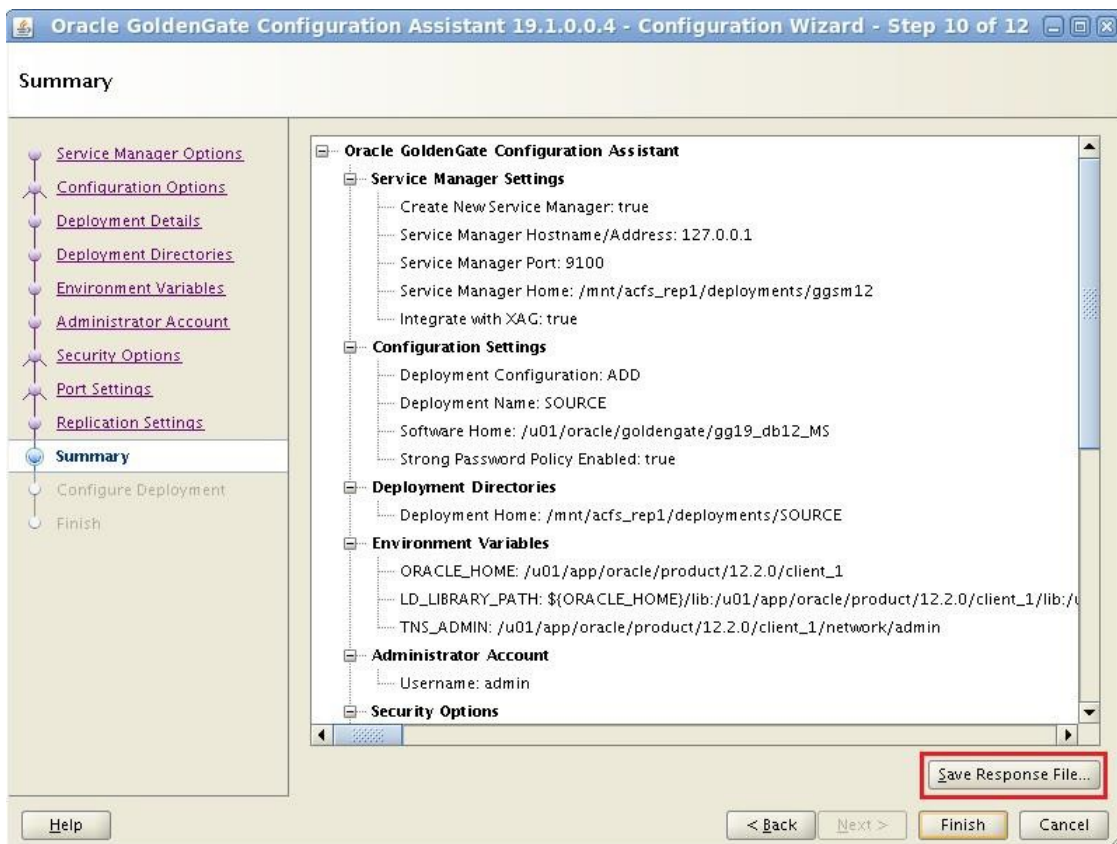


図10：デプロイメント作成レスポンス・ファイルの保存

付録Eのレスポンス・ファイル例を使用して正しいパラメータを設定する場合は、次のコマンドを使って初期デプロイメントを作成します。

```
$ cd $GG_HOME/bin
$ ./oggca.sh -silent -responseFile /u01/oracle/target_deployment.rsp
```

完了すると、次のメッセージが表示されます。

```
Linux, x64, 64bit (optimized) on Jan 01 2020 14:38:11 Operating system
character set identified as UTF-8.
Stopping...STOPPED.
Successfully Setup Software.
```

ソースとターゲットのデータベースのOracleリリースが異なる場合（12g Release 2と19cなど）は、後の手順で追加のデプロイメントを作成する必要があります（「手順9 - 追加のOracle GoldenGateデプロイメントを作成する」を参照）。

この技術概要全体では、2つのOracle GoldenGateデプロイメントをSOURCEとTARGETという名前で呼びます。

## 手順9 - 追加のOracle GoldenGate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のバージョンが異なるために追加のOracle GoldenGateデプロイメントが必要とされる場合は、上記の「手順8 - 初期Oracle GoldenGate Microservicesデプロイメントを構成する」で説明されているデプロイメント構成手順を繰り返して行ってください。異なるOGG\_HOMEディレクトリで稼働するOracle GoldenGateデプロイメントをOracle XAGで起動する場合、現行では制限があるため、別のOGG\_HOMEディレクトリの各デプロイメントは新規のサービス・マネージャで作成する必要があります。たとえば、別々のOGG\_HOMEディレクトリを使用してOracle 12cとOracle 19c両方のデータベースをサポートするデプロイメントを2つ作成する場合は、サービス・マネージャが2つ必要になります。

注：作成されるサービス・マネージャとOracle GoldenGateサーバー（管理サーバー、パフォーマンス・メトリック・サーバーなど）のすべてで、必ず別々のネットワーク・ポート番号を使用してください。

2番目のデプロイメントを作成する場合のレスポンス・ファイルの例については、[付録E](#)を参照してください。

## 手順10 - Oracle Grid Infrastructureスタンドアロン・エージェントにOracle GoldenGateデプロイメントを登録する

デプロイメントをOracle Grid Infrastructure（Oracle XAG）スタンドアロン・エージェントに登録すると、Oracle GoldenGateのサービス・マネージャ、VIP、Oracle ACFSファイル・システム・サービスの間の正しい開始と停止の依存関係が作成されます。

この技術概要で説明する構成手順の一部として作成されなかった既存のOracle GoldenGateデプロイメントを使用しており、それらのデプロイメントがオペレーティング・システムのデーモン（systemd）によって管理されている場合は、Oracle XAGに登録するためにまずそれらのデプロイメントをsystemdから削除する必要があります。

次のコマンドを使用してデプロイメントがsystemdによって管理されているかどうかを確認し、管理されている場合はsystemdから削除します。

```
# systemctl list-unit-files|grep -i goldengate
# systemctl stop OracleGoldenGate
# systemctl disable OracleGoldenGate
```

デプロイメントをOracle XAGに登録するには、次のコマンド形式を使用し、rootユーザーとして実行します。

```
agctl add goldengate <instance_name>
--gg_home <GoldenGate_Home>
--service_manager
--config_home <GoldenGate_SvcMgr_Config>
--var_home <GoldenGate_SvcMgr_VarDir>
--port <port number>
--adminuser <OGG admin user>
--user <GG instance user>
--group <GG instance group>
--filesystems <CRS_resource_name>
--filesystem_verify <yes|no>
--filesystems_always yes
--attributes TARGET_DEFAULT=online
```

コマンド説明：

--gg\_homeはOracle GoldenGateソフトウェアの場所を示します。

--service\_managerは、これがOracle GoldenGate Microservicesインスタンスであることを示します。

--config\_homeはOracle GoldenGateデプロイメント構成のホーム・ディレクトリを示します。

--var\_homeはOracle GoldenGateデプロイメント変数のホーム・ディレクトリを示します。

--portはデプロイメントのService Managerのポート番号を示します。

--adminuserはOracle GoldenGate Microservicesの管理者アカウント名を示します。

--userは、Oracle GoldenGateデプロイメントを所有するオペレーティング・システムのユーザー名を示します。

--groupは、Oracle GoldenGateデプロイメントを所有するオペレーティング・システムのグループ名を示します。

--filesystemsは、デプロイメントの開始前にONLINEになっている必要があるCRSファイル・システム・リソースを示します。これは、先の手順で作成したacfs\_primaryリソースとなります。

--filesystem\_verifyは、Oracle XAGで、config\_homeおよびvar\_homeパラメータによって指定されるディレクトリの有無を確認する必要があるかどうかを示します。これは、アクティブなOracle ACFSプライマリ・ファイル・システムの場合には‘yes’に設定する必要があります。スタンバイ・クラスタでOracle GoldenGateインスタンスを追加する場合は、‘no’と指定します。



--filesystems\_alwaysは、Oracle XAGが、--filesystemsパラメータで指定されたファイル・システムCRSリソースと同じOracle RACノードでOracle GoldenGateのサービス・マネージャを起動するように指定します。  
--attributesは、リソースのターゲット・ステータスがonlineであることを示します。これは、acfs\_primaryリソースの起動時にOracle GoldenGateデプロイメントを自動的に起動するために必要です。

Oracle GoldenGateデプロイメントは、Oracle ACFSが読み取り/書き込みまたは読み取り専用モードのいずれかでマウントされているプライマリ・ハブとスタンバイ・ハブで登録される必要があります。ファイル・システムがマウントされているクラスタのノードを調べるには、次のコマンドを実行します。

```
$ crsctl stat res acfs_standby  
  
NAME=acfs_standby  
TYPE=cluster_resource  
TARGET=ONLINE  
STATE=ONLINE on pnode01
```

以下に、Oracle GoldenGateをrootとして) Oracle XAGに登録するコマンドを示します。

```
# agctl add goldengate TARGET \  
--gg_home /u01/oracle/goldengate/gg19_db19_MS \  
--service_manager \  
--config_home /mnt/acfs_rep1/deployments/ggsm19/etc/conf \  
--var_home /mnt/acfs_rep1/deployments/ggsm19/var \  
--port 9100 \  
--adminuser admin \  
--user oracle \  
--group oinstall \  
--filesystems acfs_primary \  
--filesystem_verify yes \  
--filesystems_always yes \  
--attribute TARGET_DEFAULT=online
```

以下に、Oracle GoldenGateデプロイメントをOracle XAGで管理するために使用するagctlコマンドの例を示します。  
Oracle GoldenGateのステータスをチェックする場合：

```
% agctl status goldengate  
Goldengate instance 'TARGET' is running on pnode01
```

Oracle GoldenGateデプロイメントのサービス・マネージャを起動する場合、サービス・マネージャは、すべてのデプロイメントとOracle GoldenGateのプロセスを開始します。

```
% agctl start goldengate TARGET --node pnode02
```

サービス・マネージャが管理するOracle GoldenGateデプロイメントを停止する場合：

```
% agctl stop goldengate TARGET
```

Oracle GoldenGateデプロイメントを別のノードに手動で再配置する場合：

```
% agctl relocate goldengate TARGET --node pnode01
```

---

**注：**Oracle GoldenGateリソースは、再配置の前に必ず実行されている必要があります。

---

Oracle GoldenGateインスタンスの構成パラメータを表示する場合：

```
% agctl config goldengate TARGET  
  
Instance name:GG_MS01  
Application GoldenGate location is: /u01/oracle/goldengate/gg19_db19_MS
```

```
Goldengate MicroServices Architecture environment : yes
Goldengate Service Manager configuration directory : /mnt/acfs_rep1/deployments/ggsm19/etc/conf Goldengate
Service Manager var directory : /mnt/acfs_rep1/deployments/ggsm19/var
Service Manager Port :9100 Goldengate
Administration User : admin
File System resources needed: acfs_primary
CRS additional attributes set:TARGET_DEFAULT=online
```

より詳細なクラスタウェア構成情報を表示するには、次のコマンドを実行します。

```
% crsctl stat res -w "NAME = xag.TARGET.goldengate" -p
```

Oracle GoldenGateリソースを削除するには、プロセスを停止してからそのリソースを削除します。

```
% actl stop goldengate TARGET
```

Oracle GoldenGateインスタンスの削除は、rootとして実行する必要があります。

```
# agctl remove goldengate TARGET
```

---

注：次の手順に進む前に、Oracle XAGでOracle GoldenGateデプロイメントが起動していることを確認してください。

---

Oracle Grid Infrastructure Bundled Agentについて詳しくは、以下を参照してください。

<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/xag-agents-downloads-3636484.html>

## 手順11 - NGINXリバース・プロキシを構成する

プライマリとスタンバイ両方のクラスタのOGG\_HOMEサービス・マネージャごとに、別個のリバース・プロキシ構成が必要です。たとえば、最初のサービス・マネージャにはポート443からアクセス可能で、2番目のサービス・マネージャにはポート444からアクセス可能です。

### 1. NGINX用のサーバー証明書を作成する

NGINXがHTTPSを使ってクライアント・リクエストを認証するには、証明書が必要です。システム管理者に問い合わせ、サーバー証明書を作成または取得するための会社の基準に従ってください。

取得したら、その証明書は/etc/nginx/ogg.pemにコピーしてください。  
新しい証明書は、構成のリロード時にNGINXによって後で読み取られます。

### 2. NGINX構成ファイルを作成する

OGG\_HOME環境変数は確実に、サービス・マネージャが実行されているOracle GoldenGateソフトウェアのホーム・ディレクトリに設定します。

---

注：Oracle GoldenGateのサービス・マネージャは、次のコマンドを使用して、NGINX構成ファイルを作成するよりも前に実行されている必要があります。

---

```
export OGG_HOME=/u01/app/oracle/goldengate/gg19_db19_MS
$ $OGG_HOME/lib/utl/reverseproxy/ReverseProxySettings -u <username> -P <password> -p 443 -o ogg_db19.conf
http://localhost:9100
```

-uパラメータと-Pパラメータで指定したユーザー名とパスワードは、初期デプロイメントの作成時に (ADMINISTRATOR\_USERおよびADMINISTRATOR\_PASSWORDレスポンス・ファイル・パラメータで) 指定したGoldenGate管理者アカウントを反映している必要があります。-pパラメータによって指定されるリバース・プロキシのポート番号は、固有のサービス・マネージャが実行されているOracle GoldenGateデプロイメントごとに異なる未使用のポート番号であることが必要です。

最後に、httpポート番号（9100）も、デプロイメントの作成時に（PORT\_SERVICEMANAGERレスポンス・ファイル・パラメータで）指定したサービス・マネージャのポート番号と一致している必要があります。

固有のサービス・マネージャによって作成されたOracle GoldenGateデプロイメントごとにこの手順を繰り返し替えてください。2番目のデプロイメントのコマンド例：

```
export OGG_HOME=/u01/app/oracle/goldengate/gg19_db12_MS
export PATH=$PATH:$OGG_HOME/bin
$ $OGG_HOME/lib/utl/reverseproxy/ReverseProxySettings -u <username> -P <password> -p 444 -o ogg_db12.conf
http://localhost:9110
```

---

注：このコマンドは、同じOGG\_HOMEサービス・マネージャでデプロイメントを追加または削除する場合など、Oracle GoldenGateデプロイメントの構成に変更が加えられるたびに再実行する必要があります。

---

### 3. NGINX構成ファイルをインストールする

```
$ sudo mv ogg_db19.conf /etc/nginx/conf.d/
```

前の手順で作成されている場合は、この手順を繰り返して2番目のデプロイメントの構成ファイルをコピーします。

### 4. 新しいNGINX構成をテストする

```
$ sudo nginx -t
```

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

### 5. NGINXと新しい構成をリロードする

NGINXが現在実行されていない場合は、次のコマンドを使用して起動し、新しい構成をロードします。

```
$ sudo nginx
$ sudo nginx -s reload
```

### 6. Oracle GoldenGate Microservicesの接続性をテストする

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。

```
$ curl -s -K access.cfg https://localhost:<port#>/services/v2/config/health -XGET | python -m json.tool
```

---

注：ポート番号は、標準のHTTPSポート443を使用しない場合にのみ必要です。

---

サンプル出力：

```
{
  "$schema": "api:standardResponse", "links": [
    {
      "href": "https://localhost/services/v2/config/health",
      "mediaType": "application/json",
      "rel": "canonical"
    },
    {
```

```

...
  "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": "2019-03-28T21:52:42.835Z"
  }
}

```

## 7. Oracle GoldenGateのNGINX構成ファイルをすべての環境に配布する

Oracle ACFSファイル・システムのプライマリおよびスタンバイのすべてのOracle RACクラスタで手順3~5を繰り返し、手順2で作成した構成ファイルをコピーします。

Oracle GoldenGate Microservicesの接続性のテストは、そのデプロイメントが稼働している現在のノードでのみ実行可能です。

## 8. NGINXクラスタ・リソースを作成する

NGINXリバース・プロキシの実行ステータスは、Oracle CRSによって制御し、Oracle GoldenGateデプロイメントの開始前に自動的に起動できるようにする必要があります。

rootユーザーとして次のコマンド例を使用し、CRSリソースを作成してNGINXを管理します。

```

$ crsctl add resource nginx -type generic_application -attr
"ACL='owner:root:rwx,pgrp:root:rwx,other::r--,group:oinstall:r-
x,user:oracle:rwx',EXECUTABLE_NAMES=nginx,START_PROGRAM='/bin/systemctl start -f
nginx',STOP_PROGRAM='/bin/systemctl stop -f nginx',CHECK_PROGRAMS='/bin/systemctl status nginx'
,START_DEPENDENCIES='hard(gg_vip_prmy) pullup:always(gg_vip_prmy)
attraction(gg_vip_prmy)',STOP_DEPENDENCIES='hard(gg_vip_prmy)'"

```

---

注：上記の例はLinux 7以上の場合です。Linux 6の場合は、`/sbin/systemctl`を`/etc/rc.d/init.d/nginx start|stop|status`で置き換えてください。

---

## 9. Oracle GoldenGateのOracle XAGインスタンスをNGINX用に変更する

NGINX CRSリソースを作成したら、Oracle GoldenGate XAGリソースに変更を加え、Oracle GoldenGateデプロイメントの開始時にNGINXリバース・プロキシ・サーバーが起動するようにする必要があります。

oracleユーザーとして、次のコマンド例を使用してOracle XAGリソースに変更を加えます。

```
$ agctl modify goldengate SOURCE --filesystems acfs_primary,nginx
$ agctl modify goldengate TARGET --filesystems acfs_primary,nginx
```

## Oracle GoldenGateの構成

この項では、ハブ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。

### Oracle Net Servicesクライアント構成ファイルの作成

前の手順（データベースのロール・ベース・サービスを作成する）で作成したロール・ベースのサービス名を使用し、それぞれのOracle GoldenGateデプロイメントのTNS\_ADMINディレクトリにあるtnsnames.oraファイルで、ソースおよびターゲット・データベースのTNS別名を作成します。このデプロイメントのTNS\_ADMINディレクトリの場所は、次のコマンドを使用して確認できます。

```
$ curl -s -K access.cfg https://localhost:<port#>/services/v2/deployments/<deployment_name> -XGET | python -m json.tool
```

例：

```
curl -s -K access.cfg -k https://localhost:443/services/v2/deployments/TARGET -XGET | python -m
json.tool|grep -A1 TNS_ADMIN
      "name": "TNS_ADMIN",
      "value": "/u01/app/oracle/product/19.0.0.0/client_1/network/admin"
```

以下のsqlnet.oraファイルの例には、Oracle GoldenGateでの推奨パラメータが含まれています。

```
DEFAULT_SDU_SIZE = 2097152

# コメント解除、Oracle Net暗号化を使用する場合：
#SQLNET.ENCRYPTION_CLIENT = required
#SQLNET.ENCRYPTION_TYPES_CLIENT = (AES256)
```

Oracle GoldenGateのReplicatおよびExtractプロセスで、ハブとターゲット・データベース・ホストの間のネットワーク停止を即時検出し、それらのプロセスがTNS別名の定義に従って代替データベース・インスタンスに再接続できるようにするには、TNS別名パラメータSQLNET.RCV\_TIMEOUTを設定することを推奨します。

SQLNET.RECV\_TIMEOUTの値は、Extractで使用するTNS別名の場合には30（秒）に、ReplicatのTNS別名の場合には120（秒）に設定してください。

以下の例を使用して、Oracle Data Guardで構成されるOracle RACデータベースのOracle GoldenGateのExtract TNS別名を作成してください。

```
ggsource =
(DESCRIPTION_LIST =
  (LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=3)(RETRY_COUNT=2)
  (DESCRIPTION =
    (RECV_TIMEOUT=30)
    (ADDRESS_LIST = (LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
      (ADDRESS = (PROTOCOL = TCP)(HOST = pnode01)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = pnode02)(PORT = 1521))
    )
    (CONNECT_DATA = (SERVICE_NAME = oggsample.com))
  ) (DESCRIPTION =
    (RECV_TIMEOUT=30)
    (ADDRESS_LIST = (LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
      (ADDRESS = (PROTOCOL = TCP)(HOST = snode01)(PORT = 1521))
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = snode02)(PORT = 1521))
)
(CONNECT_DATA = (SERVICE_NAME = oggsource.example.com))
)
)
```

注：複数のOracle GoldenGateデプロイメントを使用する場合は、TNS別名が確実にデプロイメントの正しいTNS\_ADMIN tnsnames.oraファイルに作成されるようにしてください。

## Oracle GoldenGate Extractプロセスの推奨事項

Oracle GoldenGateの統合Extractは、Oracle Data Guardのロール移行を介して抽出を継続するため、Oracle Data Guardで構成されるソース・データベースからデータを抽出する際の要件です。

### Oracle Data Guardデータ損失フェイルオーバーの考慮事項

Oracle Data Guard構成では、次のパラメータにより、ソース・データベースでのOracle Data Guardのフェイルオーバーに続く統合Extractプロセスの動作を制御します。

#### TRANLOGOPTIONS HANDLEDLFAILOVER

このパラメータにより、Extractが、REDOデータを抽出したり、Oracle Data Guardスタンバイ・データベースにまだ適用されていない証跡ファイル・データに書き込んだりできなくなります。このパラメータが指定されていない場合は、データ損失フェイルオーバーの後に、ソース・データベースに存在しないデータがターゲット・データベースに存在し、データの相違と論理データの不整合が発生する可能性があります。

Oracle Data Guardのフェイルオーバー後、古いプライマリ（新しいスタンバイ）データベースが使用できない場合、ExtractはDLFAILOVER\_TIMEOUTの値（秒）が過ぎるまで（デフォルトでは300秒）新しいプライマリ・データベースからデータを抽出し続けます。Extractは、次のメッセージをレポート・ファイルに報告して、タイムアウト期間が始まったことを示します。

```
2020-05-15 14:32:59      INFO      OGG-25052   The primary database has undergone a role transition.
Temporarily suspend the HANDLEDLFAILOVER behavior for 300 seconds to allow time for standby database reinstatement. Note
that for the duration of this timeout period, Extract will not throttle trail generation. Therefore, there is a potential for data
divergence between the Oracle
GoldenGate target database and the reinstating Oracle Data Guard standby database.
```

新しいスタンバイ・データベースのタイムアウト期限が切れるまでに使用可能にならなかった場合、Extractは次のようなエラーを出力して異常終了します。

```
2020-05-15 14:38:00      INFO      OGG-25053   Timeout waiting for 300 seconds for standby database
reinstatement. Now enforcing HANDLEDLFAILOVER.
2020-05-15 14:38:30      ERROR      OGG-06219   Unable to extract data from the Logmining server
OGG$CAP_EXT1.
```

この時点で、開始時の自動再試行回数がExtractでの自動再起動タスク制限数（「Autostartプロファイルを作成する」を参照）に到達した場合、Extractは、古いプライマリ・データベースがスタンバイ・データベースとして復帰した際には、手動で再起動する必要があります。

古いプライマリ・データベースが長時間使用不可になるか、または完全に使用不能になった場合は、Oracle GoldenGateレプリケーションを継続するために、Extractパラメータ・ファイルからHANDLEDLFAILOVERパラメータを削除する必要があります。Extractは、データを抽出する前に、REDOがスタンバイ・データベースに適用されるまで待つということがなくなります。スタンバイ・データベースがオンラインに戻り、プライマリ・データベースからのすべてのREDOを適用するのに要する時間中は、スタンバイ・データベースとOracle GoldenGateターゲット・データベースの間でデータの相違が発生します。この問題は、スタンバイ・データベースが更新されるとすぐに解決します。その時に、HANDLEDLFAILOVERパラメータを元のExtractパラメータ・ファイルに追加して戻してください。

注：ソースのOracle Data Guard構成の通常の操作時に、スタンバイ・データベースが使用不可になった場合、Extractは、HANDLEDLFAILOVERパラメータによるOracle GoldenGateターゲット・データベースとの間で発生する可能性があるデータの相違を阻止するため、ソース・データベースからのデータの抽出を中止します。DLFAILOVER\_TIMEOUTパラメータは、Oracle Data Guardのフェイルオーバーが発生しなかった場合には有効にならず、Extractレポート・ファイルにメッセージは出力されません。

## TRANLOGOPTIONS DLFAILOVER\_TIMEOUT

このパラメータは、スタンバイ・データベースの復帰後フェイルオーバーを許可するための構成可能なタイムアウト値（秒）です。統合Extractがロール移行の直後に起動できるよう、HANDLEDLFAILOVERと併用されます。タイムアウト期間の終わりに、スタンバイ・データベースがまだ使用不可の場合、Extractは終了します。デフォルトのタイムアウト値は300秒（5分）で、この値はほとんどの構成の場合に適切です。

## TRANLOGOPTIONS FAILOVERTARGETDESTID

オラクルでは、ファスト・スタート・フェイルオーバー（FSFO）を構成し、プライマリ・データベースの喪失イベントにおいて、Oracle Data Guardブローカーが、以前に選択されたスタンバイ・データベースに自動的にフェイルオーバーできるようにすることを推奨しています。FSFOなしの場合、スタンバイ・データベースへのフェイルオーバーは手動プロセスとなります。

FSFOが無効になっている場合は、Extractパラメータ（FAILOVERTARGETDESTID）を指定する必要があります。

このパラメータは、スタンバイ・データベースにまだ適用されていないREDOデータを抽出しないようにするためにOracle GoldenGateのExtractプロセスが後に残る必要があるOracle Data Guardスタンバイ・データベースを特定します。FAILOVERTARGETDESTIDの正しい値を確認するには、データベース初期化パラメータarchive\_log\_destを使用します。nの値を正しいログのアーカイブ先識別子で置き換えてください。

たとえば次のとおりです。

```
SQL> show parameters log_archive_dest
```

NAME	TYPE	VALUE
log_archive_dest_1	string	location=USE_DB_RECOVERY_FILE_DEST, valid_for=(ALL_LOGFILES, ALL_ROLES)
log_archive_dest_2	string	service="ggs2d", ASYNC NOAFFIR M delay=0 optional compression =disable max_failure=0 max_con nections=1 reopen=300 db_uniqu e_name="GGS2D" net_timeout=30, valid_for=(online_logfile,all _roles)

この例では、Extractパラメータが次のように設定されます。

```
TRANLOGOPTIONS FAILOVERTARGETDESTID 2
```

以下に、必要なパラメータを使用したExtractの作成コマンド例を示します。

```
curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X POST
--data '{"config":["Extract EXT1", "ExtTrail
/mnt/acfs_rep1/deployments/SOURCE/var/lib/data/aa", "UseridAlias source_db DOMAIN goldengate", "ENCRYPTTRAIL
AES256", "TRANLOGOPTIONS PERFORMANCEPROFILE HIGH", "TRANLOGOPTIONS
_readaheadcount 64", "TRANLOGOPTIONS HANDLEDLFAILOVER", "TRANLOGOPTIONS FAILOVERTARGETDESTID 2", "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":"/mnt/ac
fs_rep1/deployments/SOURCE/var/lib/data", "name":"aa", "sizeMB":500}], "registration":"default"}|
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。

注：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。

Extractプロセスが作成されたことを確認するには、次のコマンド例を使用します。

```
curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsrvr/services/v2/extracts/EXT1 -X GET | python -m json.tool
# サンプル出力：
```

```
"config": [
  "Extract EXT1",
  "ExtTrail /mnt/acfs_rep1/deployments/SOURCE/var/lib/data/aa",
  "UseridAlias source_db DOMAIN goldengate",
  "ENCRYPTTRAIL AES256",
  "TRANLOGOPTIONS PERFORMANCEPROFILE HIGH",
  "TRANLOGOPTIONS _readaheadcount 64",
  "TRANLOGOPTIONS HANDLEDLFAILOVER",
  "TRANLOGOPTIONS FAILOVERTARGETDESTID 2",
  "REPORTCOUNT EVERY 15 MINUTES, RATE",
  "STATOPTIONS REPORTFETCH",
  "DDL EXCLUDE ALL",
  "Table soesmall.*;"
],
"credentials": {
  "alias": "source_db",
  "domain": "goldengate"
},
"encryptionProfile": "LocalWallet",
"registration": {
  "csn": 1109250433
},
"source": {
  "tranlogs": "integrated"
},
"status": "stopped"
```

## Oracle GoldenGate Replicatプロセスの推奨事項

通常Oracleでは、ハブとターゲット・データベースの間のネットワーク待機時間の値が低い（4ミリ秒以下）場合にほとんどのワークロードの場合に良好な適用パフォーマンスを示す非統合パラレルReplicatの使用を推奨しています。



次の状況では、統合パラレルReplicatを使用してください。

- 以下のいずれかのReplicat機能を使用する場合：
  - データ操作言語（DML）の適用ハンドラまたはエラー・ハンドラ。
  - 自動競合検出および解決（CDR）。
  - プロシージャ・レプリケーション。
- オラクルのハブとターゲット・データベースの間のネットワーク待機時間は高く（4ミリ秒超）なります。

最大限の適用パフォーマンスは、ハブとターゲット・データベースの間のネットワーク待機時間の値が可能な限り低い場合に達成可能です。オラクルのハブで実行されているリモートReplicatの場合は、次の構成を推奨します。

- APPLY\_PARALLELISM – MAX\_APPLY\_PARALLELISMとMIN\_APPLY\_PARALLELISMを使用する代わりに自動並列処理を無効にし、ターゲット・データベースへの最大限の同時実行性を許可します。このパラメータは、ハブおよびターゲット・データベース・サーバーの使用可能なCPUに基づいて、可能な限り高い値に設定することを推奨します。
- MAP\_PARALLELISM – 2~5の値を設定してください。アプライアの数が多い場合は、マップパーの数を増やすと、アプライアに作業を渡す能力が高くなります。
- BATCHSQL – 待機時間が長いネットワークでのネットワーク・オーバーヘッド量を削減する配列処理を使用して、DMLを適用します。データ競合が多く発生している場合は、BATCHSQLを使用すると、バッチ操作のロールバックに続いて非バッチ・モードで適用する証跡ファイルからの再読み取りがあるため、パフォーマンスが低下することを念頭においてください。

Oracle GoldenGateのパラレルReplicatについて詳しくは、『Using GoldenGate for Oracle database』ガイド (<https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/index.html>) をご覧ください。

## Oracle GoldenGateの配布パスの推奨事項

証跡ファイルをハブに送信するOracle GoldenGateの配布パスが存在し、異なるサーバー間での再配置が可能なワイド・エリア・ネットワークの仮想IPアドレスが使用中ではない場合は、ハブのロール移行の後、その配布パスを変更して証跡ファイルを新しいプライマリ・ハブ・サーバーに送信する必要があります。これには、次のREST呼出し例を使用します。

```
curl -s -K src_access.cfg https://<source host>/<source deployment name>/distsrvr/services/v2/sources/<distribution path name> -X PATCH --data '{"target":{"uri":"ogg://scao04client07-vip1:9103/services/v2/targets?trail=dd"}}' | python -m
```

必要に応じて、ソース配布パスのターゲット・アドレスの変更は、[付録D](#)に掲載されているシェル・スクリプト例を使用してハブのロール移行後に自動化することが可能です。このスクリプトは、ファイル・システムのスイッチオーバー/フェイルオーバーが発生したときにCRSアクション・スクリプトacfs\_standby ([付録C](#)を参照) によって呼び出されます。

ソースの配布パスは、障害発生後に自動で再開するように構成し、Oracle RACノード間での、またはスタンバイ・ハブへのターゲットOracle GoldenGateデプロイメントの再配置イベントで、配布パスが再開できるようにする必要があります。自動再開なしで配布パスを作成した場合は、配布サーバーのWeb UIまたはREST呼出しから有効にすることが可能です。たとえば次のとおりです。

```
$ curl -s -K access.cfg https://ggsources.example.com/GG01/distsrvr/services/v2/sources/ggs_to_gghub -X PATCH --data '{"options":{"autoRestart":{"delay": 2,"retries": 10}}}' | python -m json.tool
```

配布パスの現在の構成をチェックするには、次のコマンド例を使用します。

```
$ curl -s -K access.cfg https://ggsources.example.com/GG01/distsrvr/services/v2/sources/ggs_to_gghub -X GET | python -m json.tool
```

# サンプル出力：

```
"name": "scam_to_github",
  "options": {
    "autoRestart": {
      "delay": 2,
      "retries": 10
    }
  },
}
```

## AUTOSTARTプロファイルを作成する

サービス・マネージャによってデプロイメントが起動されたときにOracle 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AUORESTARTプロファイルが必要です。デフォルトでは、デプロイメントの起動時に自動的に起動させるため、新規作成のOracle GoldenGateのプロセスはアクティブ・プロファイルの一部になっていません。

以下の手順に従ってAUTOSTARTプロファイルを構成し、Oracle GoldenGateのプロセスに割り当てます。

1. ソース・デプロイメントとターゲット・デプロイメントの両方で、次のAUTOSTARTプロファイルとAUORESTARTプロファイルを作成します。

```
# SOURCEデプロイメント
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:AUTOSOURCE01 -XPOST --data '{"autoStart": {"enabled": true, "delay": 10}, "autoRestart": {"delay": 30, "disableOnFailure": true, "enabled": true, "onSuccess": false, "retries": 5, "window": 1200}}' | python -m json.tool

# TARGETデプロイメント
$ curl -s -K access.cfg
https://gghub.example.com/TARGET/adminsvr/services/v2/config/types/ogg:managedProcessSettings/values/ogg:managedProcessSettings:AUTOTARGET01 -XPOST --data '{"autoStart": {"enabled": true, "delay": 10}, "autoRestart": {"delay": 30, "disableOnFailure": true, "enabled": true, "onSuccess": false, "retries": 5, "window": 1200}}' | python -m json.tool
```

2. AUTOSTARTプロファイルをExtractプロセスとReplicatプロセスに割り当てます。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1 -X PATCH --data '{"managedProcessSettings": "AUTOSOURCE01"}' | python -m json.tool
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsvr/services/v2/replicats/REP1 -X PATCH --data '{"managedProcessSettings": "AUTOTARGET01"}' | python -m json.tool
```

Oracle GoldenGateのプロセスが新しいプロファイルを使用して構成されていることを確認します（出力の一部を表示）。

```
$ curl -s -K access.cfg https://gghub.example.com/SOURCE/adminsvr/services/v2/extracts/EXT1 -X GET | python -m json.tool | grep managedProcessSettings
  "managedProcessSettings": "AUTOSOURCE01",
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsvr/services/v2/replicats/REP1 -X GET | python -m json.tool | grep managedProcessSettings
  "managedProcessSettings": "AUTOTARGET01",
```

## Oracle GoldenGateプロセスの監視

エンド・ツー・エンドのOracle GoldenGateレプリケーションの待機時間は、自動ハートビート機能を使用して監視する必要があります。自動ハートビートは、ハートビート・シード表とハートビート表のレコードを更新し、ハートビート履歴表を構成することによって、各ソース・データベースからレプリケーション・ストリームに送信されます。レプリケーション・パスの各レプリケーション・プロセスでは、これらのハートビート・レコードを処理し、含まれる情報を更新します。これらのハートビート・レコードは、ターゲット・データベースのハートビート表で挿入または更新されます。

ハートビート・オブジェクトは、前に作成した正しいデータベース資格証明を使用して、ソースとターゲット両方のデータベースにインストールする必要があります。それがマルチテナント・データベースの場合は、ハートビート表をPDB内に作成する必要があります。

次のコマンド例を使用してハートビート・オブジェクトを作成し、作成されたことを確認します。

```
$ curl -s -K access.cfg
https://ggub.example.com/SOURCE/adminsvr/services/v2/connections/goldengate.source_db/tables/heart beat -X POST --data
{"frequency":5} | python -m json.tool
$ curl -s -K access.cfg -k
https://gghub.example.com/SOURCE/adminsvr/services/v2/connections/goldengate.source_db/tables/heart
beat -XGET | python -m json.tool
```

プロセスのステータスと待機時間のOracle GoldenGateによる基本的な監視動作は、次のコマンド例を使用して実行可能です。

### レプリケーションの待機時間の監視

次のコマンドを使用してターゲット・デプロイメントのハートビート表を問い合わせます。これにより、ソース・データベースでの変更からターゲット・データベースでのレプリケーションまでの待機時間が表示されます。

```
$ curl -s -K access.cfg https://gghub.example.com/TARGET/adminsvr/services/v2/connections/goldengate.target_db/tables/heart
tbeat/REP1 -XGET | python -m json.tool | grep -e ageSeconds -e lagSeconds
# サンプル出力：
  "response": {
    "$schema": "ogg:processHeartbeat",
    "heartbeats": [
      {
        "ageSeconds":2.02,
        "lagSeconds":0.63,
        "path":"EXT1 ==> REP1",
        "source":"GGS2",
        "target":"GGT"
      }
    ]
  }
```

### ExtractプロセスとReplicatプロセスのステータスの表示

Oracle GoldenGateプロセスのステータスは、次のコマンドを使用して表示できます。

```
$ curl -s -K access.cfg
https://gghub.example.com/SOURCE/adminsvr/services/v2/replicats/EXT/info/status -X GET | python -m json.tool
# サンプルの部分出力：

  "response": {
    "$schema": "ogg:extractStatus",
    "lag":1,
```

```
"lastStarted":"2020-08-06T06:01:34.605Z",
"position":"0.1118127854",
"processId":98612,
"sinceLagReported": 6,
"status": "running"
}
```

## ExtractとReplicatのレポート・ファイルの表示

ExtractまたはReplicatプロセスが異常終了した場合は、次のコマンドを使用してプロセス・レポート・ファイルを表示できます。

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

# サンプルの部分出力：

```
"response": {
  "$schema": "ogg:report",
  "lines": [
    "",
    "*****",
    "          Oracle GoldenGate Delivery for Oracle",
    "    Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO",
    "    Linux, x64, 64bit (optimized), Oracle 19c on Oct 17 2019 21:17:25",
    "",
    "Copyright (C) 1995, 2019, Oracle and/or its affiliates.All rights reserved.",
    "",
    "          Starting at 2020-08-05 23:32:26",
    "*****",
    "Operating System Version:",
    "Linux",
    "Version #2 SMP Wed May 8 22:25:03 PDT 2019, Release 4.1.12-124.26.12.el7uek.x86_64",
```

Oracle GoldenGate管理パックを使用してOracle GoldenGateプロセスを監視することを推奨します。Oracle GoldenGateの管理パックには、エンタープライズ全体のコンテキストでOracle GoldenGateを監視および管理するためのOracle GoldenGate Enterprise Manager Plug-in、Oracle GoldenGateソリューション限定の監視および管理に特化したOracle GoldenGate Monitor、レガシーOracle GoldenGateコンポーネントを管理および監視するためのOracle GoldenGate Directorからなる、3つの補完製品が組み込まれています。これらの製品を併用することで、すべてのOracle GoldenGateユーザーは、経験のあるなしを問わず同様に、Oracle GoldenGateのソリューションに対して、エンタープライズ・レベルの監視、アラート、管理業務を行うことができます。

## Oracle GoldenGate Hubの計画停止と計画外停止の管理

プライマリまたはスタンバイいずれかのファイル・システム・クラスタのハブで計画または計画外停止が実施される場合には、いくつかの考慮事項について検討する必要があります。

### 計画停止

Oracle GoldenGate Hubで計画メンテナンスを実行する必要がある場合は、一部のCRSリソースを停止して無効にし、再起動したり、誤ってファイル・システムのフェイルオーバーが引き起こされたときに好ましくない結果に至ったり、またはOracle GoldenGateが稼働できなくなったりすることのないようにする必要があります。プライマリまたはスタンバイのハブ・クラスタの計画停止イベントでは、以下の推奨事項に従ってください。

## プライマリOracle GoldenGate Hub

プライマリ・ハブ・クラスタでメンテナンスを実施可能にするため、実施前に以下の手順に従います。

1. Oracle ACFSロール・リバーサルを実施して、スタンバイが新しいプライマリになるようにします。プライマリとスタンバイ両方のファイル・システムがオンラインの場合は、repl failoverコマンドにより、プライマリ・ファイル・システムのすべての未処理の変更がロール・リバーサルの完了前にスタンバイに確実に転送および適用されるようにします。

現行スタンバイ・ファイル・システム・サーバーで、次のコマンドを実行します。

```
$ /sbin/acfsutil repl failover /mnt/acfs_rep1 # 正しいマウント・ポイントを指定する
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1 # プライマリとスタンバイ両方のサーバーで実行し
# Oracle ACFS Replicationのロール・リバーサルを確認する
$ agctl start goldengate SOURCE # Oracle GoldenGateのインスタンス名を指定する
```

2. 新しいスタンバイ・クラスタでCRSのトレース・ファイル・メッセージを無効にする場合は、新しいスタンバイ・クラスタでacfs\_standbyリソースを停止し、無効にしてください。

現行スタンバイ・ハブ・クラスタで次のコマンドを実行してacfs\_standbyリソースを無効にします。

```
$ crsctl stop res acfs_standby
$ crsctl modify resource acfs_standby -attr "ENABLED=0"
```

メンテナンスの完了後、手順を逆行行って構成をメンテナンス前の状態に戻します。

1. スタンバイ・クラスタでacfs\_standbyリソースを有効にし、開始します

```
$ crsctl modify resource acfs_standby -attr "ENABLED=1"
$ crsctl start res acfs_standby
```

2. 必要に応じて、Oracle ACFSファイル・システムをスイッチオーバーして元のOracle ACFS Replication方向に戻します。

現行スタンバイ・ファイル・システム・サーバーで、次のコマンドを実行します。

```
$ /sbin/acfsutil repl failover /mnt/acfs_rep1 # 正しいマウント・ポイントを指定する
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1 # プライマリとスタンバイ両方のサーバーで実行し
# Oracle ACFS Replicationのロール・リバーサルを確認する
$ agctl start goldengate SOURCE # Oracle GoldenGateのインスタンス名を指定する
```

## スタンバイOracle GoldenGate Hub

スタンバイ・ハブ・クラスタでメンテナンスを実行可能にするには、実施前にスタンバイ・クラスタでacfs\_standbyリソースを停止し、無効にして、CRSトレース・ファイルに警告を書き出し続けないようにします。

```
$ crsctl stop res acfs_standby
$ crsctl modify resource acfs_standby -attr "ENABLED=0"
```

メンテナンスの完了後、手順を逆行行って構成をメンテナンス前の状態に戻します。スタンバイ・クラスタでacfs\_standbyリソースを有効にし、開始します。

```
$ crsctl modify resource acfs_standby -attr "ENABLED=1"
$ crsctl start res acfs_standby
```

## Oracle GoldenGate Hubの計画外停止

プライマリまたはスタンバイいずれかのOracle GoldenGate Hubクラスタで計画外停止が発生した場合は、Oracle GoldenGateが動作を継続するようにするためにいくつかの指示に従います。プライマリおよびスタンバイのハブ・クラスタの計画外停止イベントでは、以下のハブ障害のユースケースを使用して対処してください。

### ユースケース1：スタンバイ・ハブの障害またはプライマリ・ハブがスタンバイ・ハブと通信できない

プライマリ・ハブがスタンバイ・ハブと通信できない場合は、アクティブなクラスタ・ノードのプライマリCRSトレース・ファイル (crsd\_scriptagent\_oracle.trc) に次のメッセージが出力されます。

```
2020-11-16 15:15:01.506:[acfs_primary]{1:40360:26097} [check] WARNING:STANDBY not accessible (attempt 1 of 3)
2020-11-16 15:15:14.535:[acfs_primary]{1:40360:26097} [check] WARNING:STANDBY not accessible (attempt 2 of 3)
2020-11-16 15:15:27.611:[acfs_primary]{1:40360:26097} [check] WARNING:STANDBY not accessible (attempt 3 of 3)
2020-11-16 15:15:27.611:[acfs_primary]{1:40360:26097} [check] WARNING:Problem with STANDBY file system (error: 222)
```

この時点で、スタンバイ・ファイル・システムはプライマリ・ファイル・システムでの変更を受信しなくなります。プライマリ・ファイル・システムとOracle GoldenGateは妨げられることなく機能し続けます。

このシナリオでは、次の対処法を使用してください。

- '/sbin/acfsutil repl util verifystandby/mnt/acfs\_rep1 -v'コマンドを使用してスタンバイ・ファイル・システムの状態をチェックし、スタンバイ・ハブにアクセスできない理由を突き止めます。一般的なOracle ACFS Replicationのエラーとそのソリューションについては、[付録F](#)を参照してください。
- 通信エラーの原因を修正すると、スタンバイは自動的に未処理のプライマリ・ファイル・システムの変更の適用を開始します。CRSトレース・ファイルに警告メッセージが報告されることはなくなり、代わりに次のメッセージが出力されます。

```
2020-11-16 15:28:56.097: [acfs_primary]{1:40360:26097} [check] SUCCESS:STANDBY file system /mnt/acfs_rep1 is ONLINE
```

### ユースケース2：プライマリ・ハブの障害またはスタンバイ・ハブがプライマリ・ハブと通信できない

スタンバイ・ハブがプライマリ・ハブと通信できない場合は、アクティブなクラスタ・ノードのスタンバイCRSトレース・ファイル (crsd\_scriptagent\_oracle.trc) に次のメッセージが出力されます。

```
2020-11-16 15:31:42.198:[acfs_standby]{1:25821:51519} [check] WARNING:PRIMARY not accessible (attempt 1 of 3)
2020-11-16 15:31:55.270:[acfs_standby]{1:25821:51519} [check] WARNING:PRIMARY not accessible (attempt 2 of 3)
2020-11-16 15:32:08.291:[acfs_standby]{1:25821:51519} [check] WARNING:PRIMARY not accessible (attempt 3 of 3)
2020-11-16 15:32:08.291: [acfs_standby]{1:25821:51519} [check] WARNING:Problem with PRIMARY file system (error: 222)
```

この時点で、スタンバイ・ファイル・システムはプライマリ・ファイル・システムからファイル・システムでの変更を受信する可能性は低くなります。このシナリオでは、次の対処法を使用してください。

- '/sbin/acfsutil repl util verifyprimary/mnt/acfs\_rep1 -v'コマンドを使用してプライマリ・ファイル・システムの状態をチェックし、プライマリ・ハブにアクセスできない理由を突き止めます。一般的なOracle ACFS Replicationのエラーとそのソリューションについては、[付録F](#)を参照してください。

- プライマリ・ファイル・システム・クラスタが停止しており、再起動できない場合は、スタンバイ・ハブで次のOracle ACFSフェイルオーバー・コマンドを実行します。

```
$ /sbin/acfsutil repl failover /mnt/acfs_rep1 # 正しいマウント・ポイントを指定する
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1
```

- 次のコマンドを実行して、新しいプライマリ・ハブで開始するようacfs\_primaryリソースを準備し、Oracle GoldenGateを再起動します。

```
$ echo "RESTART" > $MOUNT_POINT/acfs_primary
$ agctl start goldengate SOURCE # Oracle GoldenGateのインスタンス名を指定する
```

- 古いプライマリ・ファイル・システムがオンラインに復帰した際、新しいプライマリと古いプライマリの間の接続が再開されると、古いプライマリ・ファイル・システムは自動的にスタンバイに変換されます。
- 古いプライマリ・ファイル・システムがオンラインに復帰しても、プライマリとスタンバイのファイル・システムの間で接続を確立できない場合は、acfs\_primaryリソースがノードのクラッシュを検出します。また、スタンバイへの接続を確認できないため、Oracle GoldenGateは起動されません。これにより、相互に通信できないことによって発生する'スプリット・ブレイン' (2つのファイル・システムの両方がプライマリであると認識している状態) を防止します。

### ユースケース3 - 二重障害ケース：プライマリ・ハブ障害とスタンバイ接続障害

プライマリ・ハブがクラッシュし、オンラインに復帰してもスタンバイ・ファイル・システムとの通信を確立できない場合は、アクティブなクラスタ・ノードのプライマリCRSトレース・ファイル (crsd\_scriptagent\_oracle.trc) に次のメッセージが出力されます。

```
2020-11-16 17:08:52.621:[acfs_primary][1:40360:36312] [start] WARNING:PRIMARY file system
/mnt/acfs_rep1 previously crashed
2020-11-16 17:08:55.678:[acfs_primary][1:40360:36312] [start] WARNING:STANDBY not accessible -
disabling acfs_primary
```

プライマリ・ファイル・システムの手動での再起動が試みられると、CRSトレース・ファイルに次のメッセージがさらに出力されます。

```
2020-11-16 17:25:54.224:[acfs_primary][1:40360:37687] [start] WARNING:PRIMARY /mnt/acfs_rep1
disabled to prevent split brain
```

このシナリオでは、次の対処法を使用してください。

- '/sbin/acfsutil repl util verifystandby/mnt/acfs\_rep1 -v'コマンドを使用してスタンバイ・ファイル・システムの状態をチェックし、スタンバイ・ハブにアクセスできない理由を突き止めます。一般的なOracle ACFS Replicationのエラーとそのソリューションについては、[付録F](#)を参照してください。
- スタンバイ・ファイル・システムとの通信を再確立できる場合は、プライマリ・ハブでOracle GoldenGateを再起動します。

```
$ agctl start goldengate SOURCE # Oracle GoldenGateのインスタンス名を指定する
```

- スタンバイ・ファイル・システムとの通信を再確立できない場合は、次のコマンドを使用してプライマリ・ハブでOracle GoldenGateを再起動します。

```
$ echo "RESTART" > $MOUNT_POINT/acfs_primary
$ agctl start goldengate SOURCE # Oracle GoldenGateのインスタンス名を指定する
```

- スタンバイ・ファイル・システムとの通信が復旧すると、Oracle ACFS Replicationはプライマリ・ファイル・システムでの変更のレプリケーションを続行します。

## 結論

要約すると、Oracle GoldenGateではハブ構成に容易にデプロイ可能なMicroservicesアーキテクチャが採用されており、最高レベルのMaximum Availability Architecture（MAA）を達成しつつ、データベース・サーバーにデプロイされるコンポーネント数が最小限に抑えられています。

この技術概要で説明されているガイドラインとベスト・プラクティスに従うことにより、Oracle Grid InfrastructureとOracle ACFSを利用することによって可能になるもっとも堅牢で信頼性の高いHA構成でハブ構成を容易にデプロイし、プライマリ・ハブとスタンバイ・ハブの両方が災害から保護されるようにすることができます。ソースおよびターゲットのデータベースではOracle RACとOracle Active Data Guardがすでに構成されていることが想定されるため、保護、冗長性、ディザスタ・リカバリの各レイヤーが追加で提供されます。



## 付録A：SSHデーモン再起動アクション・スクリプト例

CRSアクション・スクリプト例を使用してsshデーモンを再起動し、スクリプトがOracle Grid Infrastructureユーザーによって確実に所有されるようにします。

```
#!/bin/bash

LOGGER="/bin/logger -t sshd_restart"
PATH_NAME="/tmp/sshd_restarted"
LOGGER_FACILITY=user
PERL_ALARM_TIMEOUT=14
GREP=/bin/grep
ECHO=/bin/echo
BASENAME=/bin/basename
TOUCH=/bin/touch
RM="/bin/rm -rf"
### パターン・マッチングのためのメッセージが確実に英語で表示されるようにする
LANG=en_US.UTF-8
NLS_LANG=American_America.AL32UTF8

export PERL_ALARM_TIMEOUT
export STATUS_TIMEOUT
export LANG NLS_LANG
if [ -z "$STATUS_TIMEOUT" ]; then STATUS_TIMEOUT=0; fi

logit () {
    ### type: info, error, debug
    type=$1
    msg=$2
    if [ "$type" = "info" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.info "$msg"
    elif [ "$type" = "error" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.error "$msg"
    elif [ "$type" = "debug" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.debug "$msg"
    fi
}

### 呼び出された方法を特定し、場所を導き出す
SCRIPTPATH=$0
SCRIPTNAME=`$BASENAME $SCRIPTPATH`

$ECHO $SCRIPTPATH | $GREP ^/ > /dev/null 2>&1

if [ $? -ne 0 ]; then
    MYDIR=`pwd`
    SCRIPTPATH=${MYDIR}/${SCRIPTPATH}
fi

case "$1" in 'start')
    logit info "Restarting sshd..."
    $RM $PATH_NAME
```

```

/bin/systemctl restart sshd
$ECHO "STARTED" > $PATH_NAME
exit 0
;;

'check'|'status')
STARTED="$(grep STARTED ${PATH_NAME}|wc -l 2>&1)";
if [ $STARTED -ne 1 ]; then
    logit info "sshd restart is stopped"
    exit 1
fi
$SCRIPTPATH start
logit info "sshd restarted"
;;

'restart')
logit info "Restarting sshd restart"
$SCRIPTPATH start
;;

'stop')
logit info "Stopping sshd restart..."
$ECHO "STOPPED" > $PATH_NAME
exit 0
;;

'clean'|'abort')
logit info "Stopping sshd restart..."
$ECHO "STOPPED" > $PATH_NAME
exit 0
;;

*)
$ECHO "Usage: $SCRIPTNAME { start | stop | check | status | restart | clean | abort }"
;;

esac

```

## 付録B：acfs\_primary CRSアクション・スクリプト

次のCRSアクション・スクリプト例では、Oracle ACFSファイル・システムがマウントされていること、それがOracle ACFSプライマリ・ファイル・システムであること、スタンバイ・ファイル・システムの検証に成功していることをチェックします。スタンバイ・ファイル・システムがそれ自体の検証チェックに失敗すると、CRSトレース・ファイル (crsd\_scriptagent\_oracle.trc) に警告メッセージが出力されます。

Oracle Grid Infrastructureユーザーが所有し、Oracle ACFSとOracle GoldenGateの環境の一部となっている、プライマリおよびスタンバイのOracle RACノードのすべてにスクリプトをコピーします。ファイル権限は755に設定してください。

```
#!/bin/bash
# acfs_action.scr - Oracle ACFSファイル・システムのタイプがPRIMARY (0 -
#                  成功/オンライン) かSTANDBY (1 - 失敗/オフライン) かをチェックする
#                  Oracle XAGが管理するOracle GoldenGateはこれが0を返す場合にのみ起動できる。
# レプリケーションされた正しいOracle ACFSマウント・ポイントで置き換える
MOUNT_POINT=/mnt/acfs_rep1
LOGGER="/bin/logger -t ACFS_GG01"
PATH_NAME=$MOUNT_POINT/acfs_primary

# 注：CRSリソースの作成時には、CHECK_TIMEOUTと
#      CHECK_INTERVAL > (ATTEMPTS*INTERVAL)を設定する
ATTEMPTS=3    # リモートのスタンバイ・ファイル・システムのチェック試行回数
INTERVAL=10   # 試行間隔 (秒)

LOGGER_FACILITY=user
PERL_ALARM_TIMEOUT=14
GREP=/bin/grep
ECHO=/bin/echo SLEEP=/bin/sleep
BASENAME=/bin/basename
STAT=/usr/bin/stat
PERL=/usr/bin/perl
TOUCH=/bin/touch
RM="/bin/rm -rf"

### パターン・マッチングのためのメッセージが確実に英語で表示されるようにする
LANG=en_US.UTF-8
NLS_LANG=American_America.AL32UTF8
export STAT MOUNT_POINT PERL_ALARM_TIMEOUT
export STATUS_TIMEOUT export LANG NLS_LANG
if [ -z "$STATUS_TIMEOUT" ]; then STATUS_TIMEOUT=0; fi
logit () {
    ### type: info, error, debug
    type=$1
    msg=$2
    if [ "$type" = "info" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.info "$msg"
    elif [ "$type" = "error" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.error "$msg"
    elif [ "$type" = "debug" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.debug "$msg"
    fi
}
```

```

fi
}
### 呼び出された方法を特定し、場所を導き出す
SCRIPTPATH=$0
SCRIPTNAME=`$BASENAME $SCRIPTPATH`

$ECHO $SCRIPTPATH | $GREP ^/ > /dev/null 2>&1
if [ $? -ne 0 ]; then
    MYDIR=`pwd`
    SCRIPTPATH=${MYDIR}/${SCRIPTPATH}
fi

function check_for_standby()
{
$PERL <<'TOT'
    eval {
        $STANDBY=`/sbin/acfsutil repl info -c -v /mnt/acfs_rep1|grep "Site"|grep Standby|wc -l 2>&1`;
        if ( $STANDBY == 1 ) {
            exit 1;
        } else {
            exit 0;
        }
    };
TOT
    RC=$?

    ### perlブロックからのプロセス・リターン・コード
    if [ $RC -eq 1 ]; then
        echo 1
    else
        echo 0
    fi
}

# standby_status - スタンバイDSにアクセス可能で動作状態が良好であることを検証する
function standby_status()
{
    OUTPUT="$(ls -l)"
    s_status="$(/sbin/acfsutil repl util verifystandby ${MOUNT_POINT}|cut -d ' ' -f4 2>&1)"

    if [ $s_status -ne 0 ]; then
        ### FSのステータスがOKAYではない
        echo $s_status
    else # FSはOKAY
        echo 0;
    fi
}

# verify_standby - スタンバイの現在ステータスの調べる
function verify_standby()
{
    for ((i=1;i<=$ATTEMPTS;i++)); do

```

```

sresult=$(standby_status) #結果が0 (SUCCESS)であることをチェックする

if [ $sresult -eq 0 ]; then
    break
else
    logit debug "WARNING:STANDBY not accessible (attempt $i of $ATTEMPTS)"
    if [ $i -lt $ATTEMPTS ]; then
        $SLEEP $INTERVAL
    fi
fi
done

if [ $i -lt $ATTEMPTS ]; then
    logit debug "SUCCESS:Remote STANDBY file system $MOUNT_POINT is ONLINE"
    presult=0
else
    logit debug "WARNING:Problem with STANDBY file system (error: $sresult)"
    presult=0
fi
}

function verify_primary()
{
    ### Oracle ACFSがマウントされているかどうかをチェックする
    $PERL <<'TOT'
    $timeout = $ENV{'PERL_ALARM_TIMEOUT'};
    $SIG{ALRM} = sub {
        ### 問題があるのでクリーンアップする必要がある
        exit 3;
        die "timeout" ;
    };
    alarm $timeout;
    eval {
        $STATUSOUT=`$ENV{'STAT'} -f -c "%T" $ENV{'MOUNT_POINT'} 2>&1 `;

        chomp($STATUSOUT);
        ### Linux 6出力のfuseblkチェックを追加した
        if ( ($STATUSOUT eq 'acfs') ) {
            ### FSがマウントされている
            exit 0
        } else {
            ### ファイル・システムはオフライン
            exit 1;
        }
    };
}
TOT

RC=$?
presult=0

### perlブロックからのプロセス・リターン・コード
if [ $RC -eq 3 ]; then

```

```

STATUS_TIMEOUT=$(( $STATUS_TIMEOUT + 1 ))
    logit error "Found timeout while checking status, cleaning mount automatically"
    $SCRIPTPATH clean
    logit debug "Filesystem is OFFLINE" presult=1
elif [ $SRC -eq 2 ]; then
    STATUS_TIMEOUT=$(( $STATUS_TIMEOUT + 1 ))
    logit error "Found error while checking status, cleaning mount automatically"
    $SCRIPTPATH clean
    logit debug "WARNING:PRIMARY file system $MOUNT_POINT OFFLINE"
    presult=1
elif [ $SRC -eq 1 ]; then
    logit debug "WARNING:PRIMARY file System is not mounted"
    presult=2
elif [ $SRC -eq 0 ]; then
    # ファイル・システムがマウントされている
    result=$(check_for_standby) # FSプライマリまたはスタンバイかどうかを確認する
    if [ $result -eq 1 ]; then
        presult=1
    fi
fi # ファイル・システムがマウントされていれば終了する
}

case "$1" in
'start')
    logit info "$SCRIPTNAME starting at $MOUNT_POINT"

    result=$(check_for_standby) # スタンバイ・ファイル・システムでなければ0を返す

    if [ $result -eq 0 ]; then
        verify_primary
    else
        logit info "Detected local standby file system" exit 1
    fi

    if [[ $presult == 0 ]]; then
# プライマリ・ファイル・システムがマウントされるR/W
# $PATH_NAMEが"STARTED"でノードの以前のCRASHを示すかどうかをチェックする

        STARTED="$( ${GREP} STARTED ${PATH_NAME} | wc -l 2>&1)";

        if [ $STARTED -eq 1 ]; then
            # ファイル・システムは以前にクラッシュした
            logit info "WARNING:PRIMARY file system $MOUNT_POINT previously crashed"

# 起動前にスタンバイ・ファイル・システムがアクセス可能なことをチェックする
            rresult=$(standby_status) # STANDBY結果が0 (SUCCESS) であることをチェックする

            if [ $rresult -ne 0 ]; then
                logit debug "WARNING:STANDBY not accessible - disabling acfs_primary"
                $ECHO "DISABLED" > $PATH_NAME
                exit 0
            fi
        fi
    fi
}

```

```

    fi
fi

DISABLED="${GREP} DISABLED ${PATH_NAME}|wc -l 2>&1";
if [ $DISABLED -eq 1 ]; then
# まだスタンバイにアクセス可能かどうかをチェックする
    sresult=$(standby_status) # STANDBY結果が0 (SUCCESS)であることをチェックする

    if [ $sresult -ne 0 ]; then # スタンバイはまだアクセス可能ではない
        logit info "WARNING:PRIMARY $MOUNT_POINT disabled to prevent split brain"
        exit 0
    fi # スタンバイが戻っていないため、elseによりプライマリが再起動できるようにする
fi

$RM $PATH_NAME
$ECHO "STARTED" > $PATH_NAME

# プライマリのステータス・ファイルが存在するかどうかをチェックする
if [ -f "$PATH_NAME" ]
then # リソースのSTARTでのみ呼び出される
    logit info "acfs_primary started"

# Oracle GoldenGate Distribution Serverがハブのレシーバ・サーバーに証跡を送信している場合は、パス・ターゲットを変更する :
# /u01/oracle/scripts/change_path_hub.sh scam09db05:443 scam09db06:443 scam_to_gghub gg01
/u01/oracle/scripts/scam.cfg
# RET=?
# if [[ $RET == 0 ]]; then
#     logit info "Changed distribution path scam_to_gghub target URI"
# else
#     logit info "ERROR:Unable to change distribution path target URI"
# fi
# exit 0
else
    logit info "WARNING:Unable to create $PATH_NAME"
    exit 1
fi
else
    logit info "WARNING:Verify of PRIMARY failed"
    exit 1
fi
;;

'check'|'status')
    result=$(check_for_standby) # スタンバイ・ファイル・システムでなければ0を返す

    if [ $result -ne 0 ]; then
        logit info "Detected local standby file system" exit 1
    fi

```

```

verify_primary    # プライマリFSのステータスをチェックする

logit info "DEBUG: presult $presult"

if [[ $presult == 0 ]]; then                                # プライマリ・ファイル・システムは良好

# まず手動で停止されたのではないことをチェックする：
STOPPED="$( ${GREP} STOPPED ${PATH_NAME}|wc -l 2>&1)";
if [ $STOPPED -eq 1 ]; then
    logit info "SUCCESS:PRIMARY file system $MOUNT_POINT is STOPPED"
    exit 1
fi

DISABLED="$( ${GREP} DISABLED ${PATH_NAME}|wc -l 2>&1)";
if [ $DISABLED -eq 1 ]; then
    logit info "WARNING:Primary file system $MOUNT_POINT DISABLED"
    exit 1
fi

RESTART="$( ${GREP} RESTART ${PATH_NAME}|wc -l 2>&1)";
if [ $RESTART -eq 1 ]; then
    logit info "Restarting PRIMARY file system $MOUNT_POINT"
    $SCRIPTPATH start
fi

sresult=$(standby_status)                                # スタンバイ・ステータスをチェックする (0=GOOD)

if [ $sresult -eq 0 ]; then                                # スタンバイはすべて良好
    if [ -f "$PATH_NAME" ]
    then
        STARTED="$( ${GREP} STARTED ${PATH_NAME}|wc -l 2>&1)";
        if [ $STARTED -eq 1 ]; then
            logit info "SUCCESS:STANDBY file system $MOUNT_POINT is ONLINE"
            exit 0
        fi
    else # $PATH_NAMEにアクセスできない
        logit info "WARNING:PRIMARY file $PATH_NAME does NOT exist "
        exit 1
    fi
else # $sresult != 0 (BAD)
    verify_standby # スタンバイ・ファイル・システムの現在ステータスを取得し、結果をCRSトレースで報告する
fi
elif [ $presult -eq 2 ]; then                                # プライマリ・ファイル・システムがマウントされていない、おそらくスタンバイFS
    exit 1
else
    logit info "WARNING:Problem with local PRIMARY file system."
    exit 1
fi
;;

'restart')
logit info "Restart -- ACFS file system type checking..."

```



```

$SCRIPTPATH start
;;

'stop')
logit info "Stop -- Stopping ACFS file system type checking..."
$ECHO "STOPPED" > $PATH_NAME
exit 0
;;

'clean'|'abort')
logit info "Clean/Abort -- Stopping ACFS file system type checking..."
$ECHO "STOPPED" > $PATH_NAME
exit 0
;;

*)
$ECHO "Usage: $SCRIPTNAME { start | stop | check | status | restart | clean | abort }"
;;

esac
```

## 付録C：acfs\_standby CRSアクション・スクリプト

次のacfs\_standbyアクション・スクリプト例では、Oracle ACFSファイル・システムがマウントされていること、それがOracle ACFSスタンバイ・ファイル・システムであること、プライマリ・ファイル・システムの検証に成功していることをチェックします。

Oracle Grid Infrastructureユーザーが所有し、Oracle ACFSとOracle GoldenGateの環境の一部となっている、プライマリおよびスタンバイのOracle RACノードのすべてにスクリプトをコピーします。ファイル権限は755に設定してください。

```
#!/bin/bash

# Oracle ACFSレプリケーションの環境に合わせてこのセクションのパラメータ
# を変更する
MOUNT_POINT=/mnt/acfs_rep1
LOGGER="/bin/logger -t ACFS_GG01"
PATH_NAME=/tmp/acfs_standby

# 注：CRSリソースの作成時には、CHECK_TIMEOUTとCHECK_INTERVAL >
# (ATTEMPTS*INTERVAL)を設定する
ATTEMPTS=3      # リモートのプライマリ・ファイル・システムのチェック試行回数
INTERVAL=10     # 試行間隔（秒）

LOGGER_FACILITY=user
PERL_ALARM_TIMEOUT=14
ECHO=/bin/echo
GREP=/bin/grep
SLEEP=/bin/sleep
BASENAME=/bin/basename
STAT=/usr/bin/stat
PERL=/usr/bin/perl
TOUCH=/bin/touch
RM="/bin/rm -rf"

### パターン・マッチングのためのメッセージが確実に英語で表示されるようにする
LANG=en_US.UTF-8
NLS_LANG=American_America.AL32UTF8

export STAT MOUNT_POINT PERL_ALARM_TIMEOUT
export STATUS_TIMEOUT ATTEMPTS INTERVAL
export LANG NLS_LANG

if [ -z "$STATUS_TIMEOUT" ]; then STATUS_TIMEOUT=0; fi

logit () {
    ### type: info, error, debug
    type=$1
    msg=$2
    if [ "$type" = "info" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.info "$msg"
    elif [ "$type" = "error" ]; then
        $ECHO $msg
        $LOGGER -p ${LOGGER_FACILITY}.error "$msg"
    elif [ "$type" = "debug" ]; then
        $ECHO $msg
    fi
}
```

```

    $LOGGER -p ${LOGGER_FACILITY}.debug "$msg"
  fi
}

### 呼び出された方法を特定し、場所を導き出す
SCRIPTPATH=$0
SCRIPTNAME=`basename $SCRIPTPATH`

$ECHO $SCRIPTPATH | $GREP ^/ > /dev/null 2>&1
if [ $? -ne 0 ]; then
  MYDIR=`pwd`
  SCRIPTPATH=${MYDIR}/${SCRIPTPATH}
fi

# check_for_standby - Oracle ACFSマウント・ポイントがSTANDBY (1を返す) であることをチェックする
function check_for_standby()
{
  $PERL <<'TOT'
  eval {
    $STANDBY=`/sbin/acfsutil repl info -c -v $ENV{'MOUNT_POINT'}|grep "Site"|grep Standby|wc -l 2>&1`;

    if ( $STANDBY == 1 ) {
      ### FS がSTANDBY
      exit 1;
    } else {      # FSがPRIMARY
      exit 0;
    }
  };
  TOT
  RC=$?

  ### perlブロックからのプロセス・リターン・コード
  if [ $RC -eq 1 ]; then
    echo 1
  else
    echo 0
  fi
}

# verify_primary - プライマリFSのステータスをチェックする
function verify_primary()
{
  $PERL <<'TOT'
  eval {
    $p_status=`/sbin/acfsutil repl util verifyprimary $ENV{'MOUNT_POINT'}|cut -d ' ' -f4 2>&1`;

    if ( $p_status > 0 ) {          # プライマリはOkayではない
      ### FSのステータスはOKAYではない
      exit $p_status;
    } else {      # FSはOKAY
      echo 0;
    }
  };
  TOT
  RC=$?
}

```

```

    }
};
TOT
RC=$?

### perlブロックからのプロセス・リターン・コード
if [ $RC -ne 1 ]; then
    echo $RC
else
    echo 0
fi
}

case "$1" in
'start')
    logit info "$SCRIPTNAME starting to check ACFS remote primary at $MOUNT_POINT"

    result=$(check_for_standby)

    if [ $result -ne 1 ]; then                # STANDBYファイル・システムを検出
        logit info "local file system ($MOUNT_POINT) is a PRIMARY"
    fi

    SRM $PATH_NAME
    SECHO "STARTED" > $PATH_NAME
    if [ -f "$PATH_NAME" ]
    then                                     # リソースを起動する場合にのみ呼び出され、オフラインからオンラインに切り換わらない
        exit 0
    else
        logit info "ERROR:Unable to create standby file system status file ($PATH_NAME)"
        exit 1
    fi
;;

'check'|'status')
    ### Oracle ACFSがSTANDBYファイル・システムとしてマウントされているかどうかをチェックする
    $PERL <<'TOT'
        $timeout = $ENV{'PERL_ALARM_TIMEOUT'};
        $SIG{ALRM} = sub {
            ### 問題があるのでクリーンアップする必要がある
            exit 3;
            die "timeout" ;
        };
        alarm $timeout;
        eval {
            $STATUSOUT=` $ENV{'STAT'} -f-c "%T" $ENV{'MOUNT_POINT'} 2>&1 `;
            chomp($STATUSOUT);

            if ( ( $STATUSOUT eq 'acfs' ) ) {
                ### ファイル・システムがマウントされている
                exit 0
            } else {

```

```

    ### ファイル・システムはオフライン
    exit 1;
}
];

TOT

RC=$?
### perlブロックからのプロセス・リターン・コード
if [ $SRC -eq 3 ]; then
    STATUS_TIMEOUT=$(( $STATUS_TIMEOUT + 1 ))
    logit error "Found timeout while checking status, cleaning mount automatically"
    $SCRIPTPATH clean
    logit debug "File system check encountered a problem"
    exit 1
elif [ $SRC -eq 1 ]; then
    logit debug "File system is OFFLINE"
    exit 1
elif [ $SRC -eq 0 ]; then
    # ファイル・システムがマウントされている

    result=$(check_for_standby)          # 0：プライマリ 1：スタンバイ

    if [ $result -eq 1 ]; then          # STANDBYファイル・システムが検出された

# プライマリFSがアクセス可能なことをチェックする。これは任意であり、プライマリFS
# で問題が発生した場合に動作を継続する場合はスキップしても構わない - 上記警告を参照

    for ((i=1;i<=$ATTEMPTS;i++)); do
        result=$(verify_primary)          # 結果が0 (SUCCESS) であることをチェックする

        if [ $result == 0 ]; then          # プライマリ・ファイル・システムはOKAY
            logit debug "SUCCESS:PRIMARY file system $MOUNT_POINT is ONLINE"
            exit 0
        elif [ $result -eq 222 ] || [ $result -eq 255 ] || [ $result -eq 237 ]; then
            logit debug "WARNING:PRIMARY not accessible (attempt $i of $ATTEMPTS)"
            if [ $i -lt $ATTEMPTS ]; then
                $SLEEP $INTERVAL
            fi
        fi
    done
    logit debug "WARNING:Problem with PRIMARY file system (error: $result)"
    exit 0
else
    logit debug "Local PRIMARY file system $MOUNT_POINT"
    exit 1
fi # if [ $result -eq 1 ]; then          # STANDBYファイル・システムが検出された
fi # if [ $SRC -eq 3 ];
;;

'restart')
logit info "Restarting ACFS remote primary checking..."
$SCRIPTPATH start
;;

```

```
'stop')
  logit info "Stopping ACFS remote primary checking..."
  $ECHO "STOPPED" > $PATH_NAME
  exit 0
;;

'clean'|'abort')
  logit info "Stopping ACFS file system type checking..."
  $ECHO "STOPPED" > $PATH_NAME
  exit 0
;;

*)
  $ECHO "Usage: $SCRIPTNAME { start | stop | check | status | restart | clean | abort }"
  ;;
esac
```

## 付録D：配布パス・ターゲット変更スクリプト

ハブ・ロール移行後は、次のスクリプト例を使用して、レシーバ・サーバーの新規の場所を反映するように、ソース Oracle GoldenGateデプロイメントの配布パス・ターゲット・アドレスを変更できます。この例では、ソース Oracle GoldenGateデプロイメントがMAAアーキテクチャで構成されると想定しているため、配布サーバーの現在の場所を特定する必要があります。

```
#!/bin/bash

# change_path_gghub.sh - ターゲットがGG Hubスタンバイに移動するときに
# GG配布パスのターゲット・ホストを変更する。

# 使用例： ./change_path_hub.sh scam09db05:443 scam09db06:443 scam_to_gghub gg01
/u01/oracle/scripts/gghub/scam.cfg

SOURCE1=$1      # scam09db05:443
SOURCE2=$2      # scam09db06:443
DPATH=$3        # scam_to_gghub
DEP=$4          # gg01
ACCESS=$5       # scam.cfg
CONNECT=0

result=$(curl -i -s -K $ACCESS -k https://$SOURCE1/$DEP/distsrvr/services/v2/sources/$DPATH -X GET
| grep HTTP | awk '{print $2}')

# NGINXが実行されていない場合はNULL、サーバーと接触できない場合は502、サーバーとの接触が良好な場合は200、その他の望
# まれない理由の場合はその他の値（404）を返す。

if [[ -z $result || $result -ne 200 ]]; then                                # 配布サーバーへのアクセスを管理
    echo "**** Couldn't contact Distribution Server at $SOURCE1 ****"

# 他のソース・ホストを試す：
else
    echo "**** Got status of Distribution Server at $SOURCE1 ****"
    SOURCE=$SOURCE1
    CONNECT=1
fi

if [ $CONNECT -eq 0 ]; then

    result=$(curl -i -s -K $ACCESS -k https://$SOURCE2/$DEP/distsrvr/services/v2/sources/$DPATH -X
GET| grep HTTP | awk '{print $2}')
    if [[ -z $result || $result -ne 200 ]]; then                            # 配布サーバーへのアクセスを管理
        echo "**** Couldn't contact Distribution Server at $SOURCE2 ****"
# 他のソース・ホストを試す：
    else
        echo "**** Got status of Distribution Server at $SOURCE2 ****"
        SOURCE=$SOURCE2
        CONNECT=1
    fi
fi

if [ $CONNECT -eq 1 ]; then
    curl -s -K $ACCESS -k https://$SOURCE/$DEP/distsrvr/services/v2/sources/$DPATH -X PATCH --
data '{"status": "stopped"}' | python -m json.tool
```

```
curl -s -K $ACCESS -k https://$SOURCE/$DEP/distsrvr/services/v2/sources/$DPATH -X PATCH --data
'{"target":{"uri":"ogg://scao04client07-vip1:9103/services/v2/targets?trail=dd"}}' | python -m json.tool

curl -s -K $ACCESS -k https://$SOURCE/$DEP/distsrvr/services/v2/sources/$DPATH -X GET | python -m json.tool

curl -s -K $ACCESS -k https://$SOURCE/$DEP/distsrvr/services/v2/sources/$DPATH -X PATCH --data '{"status": "running"}' | python
-m json.tool

exit 0

fi

# この場合は、どちらの配布サーバーにも接続できなかったことを意味する

exit 1
```



## 付録E：Oracle GoldenGateデプロイメント作成レスポンス・ファイルの例

ほとんどの構成変数は、デプロイメントとOracle GoldenGateのプロセスロ・ファイルが格納されるファイル・システムのディレクトリに関連付けられているため、ディレクトリがそれらの作成済みのディレクトリと一致していることが重要です。

最新のOracle GoldenGateインストール・ソフトウェアの適切なレスポンス・ファイルを使用して、パラメータが正しく指定されていることを確認してください。レスポンス・ファイルのパラメータ変更が必要になる例：

### Section A - General

DEPLOYMENT\_NAME=TARGET # 確実に一意のデプロイメント名となるようにします

### Section B - Administrator Account

ADMINISTRATOR\_USER=admin # GGデプロイメント管理アカウント、データベース・アカウントではありません。

ADMINISTRATOR\_PASSWORD=<password> # 注：パスワードを指定する必要があります

### Section C - Service Manager

SERVICEMANAGER\_DEPLOYMENT\_HOME=/mnt/acfs\_rep1/deployments/ggsm # サービス・マネージャ・デプロイメントのファイル・システム・ディレクトリです。これは、最初のデプロイメント作成でのみ設定されます。後続のデプロイメント作成では空白のままにします。

HOST\_SERVICEMANAGER=127.0.0.1 # localhost (127.0.0.1) のままにします

PORT\_SERVICEMANAGER=9100 # このデプロイメントに固有の未使用ポート番号に設定します。2番目のデプロイメント作成の場合、これは初期デプロイメント作成で使用したポート番号に設定します。

SECURITY\_ENABLED=false # セキュリティはNGINXリバース・プロキシによって処理されます。

CREATE\_NEW\_SERVICEMANAGER=true # 最初のデプロイメント作成ではtrueにのみ設定します。後続のデプロイメント作成では、falseに設定します。

INTEGRATE\_SERVICEMANAGER\_WITH\_XAG=true

### Section D - Software Home

OGG\_SOFTWARE\_HOME=/u01/app/goldengate/db19c # Oracle GoldenGateソフトウェアの場所

### Section E - Deployment Directories

OGG\_DEPLOYMENT\_HOME=/mnt/acfs\_rep1/deployments/target # Oracle ACFSのレプリケートされたファイル・システムに格納する必要があります

### Section F - Environment Variables

ENV\_ORACLE\_HOME=/u01/app/oracle/product/19.1/client

ENV\_LD\_LIBRARY\_PATH=\${ORACLE\_HOME}/lib:/u01/app/oracle/product/19.1/client/lib:/u01/app/oracle/product/19.1/client/rdbms/lib:/u01/app/oracle/product/19.1/client/jdbc/lib:/u01/oracle/goldengate/target/lib

ENV\_TNS\_ADMIN=/u01/app/oracle/network/admin # sqlnet.oraおよびtnsnames.oraの管理を容易にするために、すべてのクライアント・インストールで単一のTNS\_ADMINを使用します

### Section I - Services

PORT\_ADMINSRVR=9101 # このデプロイメントに固有の未使用ポート番号に設定します。

DISTRIBUTION\_SERVER\_ENABLED=false # ローカルの証跡ファイルがあるので使用しません。

PORT\_DISTSRVR=9102 # このデプロイメントに固有の未使用ポート番号に設定します。

RECEIVER\_SERVER\_ENABLED=false # ローカルの証跡ファイルがあるので使用しません。

PORT\_RCVRSRVR=9103 # このデプロイメントに固有の未使用ポート番号に設定します。

PORT\_PMSRVR=9104 # このデプロイメントに固有の未使用ポート番号に設定します。

UDP\_PORT\_PMSRVR=9105 # このデプロイメントに固有の未使用ポート番号に設定します。  
PMSRVR\_DATASTORE\_TYPE=BDB # またはLMDBを使用します  
PMSRVR\_DATASTORE\_HOME=# BDBを使用する場合は、Oracle ACFSマウント・ポイントの外部のディレクトリを設定します。

## Section J - REPLICATION OPTIONS

OGG\_SCHEMA=ggadmin # データベースのGG管理者が使用するスキーマに設定します。ソースとターゲットのデータベース・スキーマ名が確実に一致するようにしてください。

後続のデプロイメント作成で同じOGG\_HOMEディレクトリを使用する場合は、次のレスポンス・ファイルのパラメータが異なります。

## Section A - General

DEPLOYMENT\_NAME=SOURCE # これが確実に一意のデプロイメント名となるようにし、SOURCEまたはTARGETを使用してください

## Section C - Service Manager

SERVICEMANAGER\_DEPLOYMENT\_HOME=# 後続のデプロイメント作成では空白のままにします。  
HOST\_SERVICEMANAGER=127.0.0.1 # localhost (127.0.0.1) のままにする必要があります  
PORT\_SERVICEMANAGER=9100 # これは、初期デプロイメント作成で使用したポート番号に設定します。  
CREATE\_NEW\_SERVICEMANAGER=false # 初期デプロイメントではサービス・マネージャがすでに作成されています。  
EXISTING\_SERVICEMANAGER\_IS\_XAG\_ENABLED=true

## Section D - Software Home

OGG\_SOFTWARE\_HOME=/u01/app/oracle/goldengate/db12c # データベースの互換性を正しく保つため上記の手順2で指定したOracle GoldenGateソフトウェアの場所です。

## Section E - Deployment Directories

OGG\_DEPLOYMENT\_HOME=/mnt/acfs\_rep1/deployments/source # レプリケートされたOracle ACFSファイル・システムに格納します。

## Section F - Environment Variables

ENV\_ORACLE\_HOME=/u01/app/oracle/product/12.2/client # ソース・データベースが12.2であると想定していますが、データベース・バージョンに応じて異なる場合があります  
ENV\_LD\_LIBRARY\_PATH=\${ORACLE\_HOME}/lib:/u01/app/oracle/product/19.1/client/lib:/u01/app/oracle/product/19.1/client/rdbms/lib:/u01/app/oracle/product/19.1/client/jdbc/lib:/u01/oracle/goldengate/source/lib  
ENV\_TNS\_ADMIN=/u01/app/oracle/network/admin # sqlnet.oraおよびtnsnames.oraの管理を容易にするために、すべてのクライアント・インストールで単一のTNS\_ADMINを使用します

## Section I - Services

PORT\_ADMINSRVR=9111 # このデプロイメントに固有の未使用ポート番号に設定します  
DISTRIBUTION\_SERVER\_ENABLED=false  
PORT\_DISTSRVR=9112 # このデプロイメントに固有の未使用ポート番号に設定します  
RECEIVER\_SERVER\_ENABLED=false  
PORT\_RCVSRVR=9113 # このデプロイメントに固有の未使用ポート番号に設定します  
PORT\_PMSRVR=9114 # このデプロイメントに固有の未使用ポート番号に設定します  
UDP\_PORT\_PMSRVR=9115 # このデプロイメントに固有の未使用ポート番号に設定します

PMSRVR\_DATASTORE\_TYPE=BDB

# またはLMDBを使用します

PMSRVR\_DATASTORE\_HOME=

# BDBを使用する場合は、Oracle ACFSマウント・ポイントの外部のディレクトリを設定します。

## 付録F：トラブルシューティング

### Oracle ACFS Replicationのトラブルシューティング

Oracle ACFS Replicationの健全性は、acfsutil repl util verifyprimary/verifystandbyコマンドによって判断します。これらのコマンドは、付録Bおよび付録Cに掲載されているCRSアクション・スクリプト例によって呼び出されますが、ファイル・システムのロール移行中にも暗黙的に呼び出されます。

両方のコマンドとも、問題が検出されなければ、値'0'を返します。0以外の値が返された場合は、verboseフラグを指定して同じコマンドを実行して、検証テストの出力全体を確認します。

次に例を示します。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1 -v

- Attempting to ping clust1-vip1
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3          -x oracle@clust1-vip1 true
  2>&1'
- ssh output:Host key verification failed.
- ssh output:Host key verification failed.
- ssh attempt failed, ret=255
verifyprimary return code:255
```

verifyコマンドによって報告されるエラー"Host key verification failed"には、失敗の理由が明確に示されます。この例では、スタンバイとプライマリのファイル・システムのホスト間のSSH構成に問題がありました。問題が解決されたら、verifyコマンドを再び実行して、これ以上問題がないことを確認します。

付録Cの例のようにして自動化ファイル・システムのフェイルオーバー機能を使用する場合は、フェイルオーバーの完了後に、acfsutilトレース・ファイルでフェイルオーバーが発生した理由を確認することを推奨します。acfsutilトレース・ファイルは、CRSトレース・ファイル・ディレクトリ (/u01/app/grid/diag/crs/<nodename>/crs/trace) に格納されます。

以下に、Oracle ACFS Replicationの誤った構成で発生する可能性がある一般的な障害をいくつか示します。

#### 1.SSHデーモンがシャットダウンしたか、VIP上で実行されるように構成されていない

Oracle ACFSのプライマリおよびスタンバイ・クラスタでアプリケーションVIPを使用する場合は、SSHデーモンを構成してVIPアドレスで着信接要求続をリスニングする必要があります。この構成が行われていない場合、またはSSHデーモンが現行のプライマリ/スタンバイのどちらのホストでも実行されていない場合は、verifyprimaryまたはverifystandbyコマンドが次のエラーを出力して失敗します：sbin/acfsutil repl util verifystandby /mnt/acfs\_rep1 -v

```
- Attempting to ping scaqaa08client01-avip
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true          -o Ciphers=aes128-ctr -o ConnectTimeout=3  -x
oracle@scaqaa08client01-avip true 2>&1'
- ssh output: ssh: connect to host scaqaa08client01-avip port 22:Connection refused
- ssh output: ssh: connect to host scaqaa08client01-avip port 22:Connection refused
- ssh attempt failed, ret=255
verifystandby return code:255
```

verifyprimaryおよびverifystandbyコマンドが実行されるたびに、出力はCRSトレース・ファイル・ディレクトリのトレース・ファイルにも書き込まれます。たとえば次のとおりです。

```

2020-06-11 17:08:58.905 :REPLCMP:3313712320: [246432]:0000:ofsrUtil: entered, isAdmin=1,
numargs=4, ac=5
2020-06-11 17:08:58.905 :REPLCMP:3313712320: [246432]:0001:ofsrGetSnapMount:Checking mount path
/mnt/acfs_rep1
2020-06-11 17:08:58.906 :REPLCMP:3313712320: [246432]:0002:uscgGetReplicationState: repl state = 3
2020-06-11 17:08:58.906 :REPLCMP:3313712320: [246432]:0003:uscgGetRacoonV2:Interesting racoon flags.USC_RACFLAGS=0x5
2020-06-11 17:08:58.906 :REPLCMP:3313712320: [246432]:0004:uscgMakeSshPath:
USC_SSHPATH=/usr/bin/ssh
2020-06-11 17:08:58.906 :REPLCMP:3313712320: [246432]:0005:uscgMakeAcfsutilPaths:Using ssh:TRUE
2020-06-11 17:08:58.906 :REPLCMP:3313712320: [246432]:0006:uscgInvokeCommand: invoking '/bin/ping
-c 1 -w 3 scaqaa08client01-avip > /dev/null 2>&1'
2020-06-11 17:08:58.911 :REPLCMP:3313712320: [246432]:0007:uscgVerifyStandbyV2: ping OK (/bin/ping
-c 1 -w 3 scaqaa08client01-avip > /dev/null 2>&1)
2020-06-11 17:08:58.921 :REPLCMP:3313712320: [246432]:0008:uscgVerifyStandbyV2: ssh produced
cmdOutput: ssh: connect to host scaqaa08client01-avip port 22:Connection refused

2020-06-11 17:09:00.932 :REPLCMP:3313712320: [246432]:0010:ERROR: uscgVerifyStandbyV2: could not
ssh to the standby (/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x
oracle@scaqaa08client01-avip true 2>&1), ret=255

```

'connection refused'エラーが返された場合は、まずVIPアドレスが/etc/ssh/sshd\_configファイルに追加されているかどうかをチェックしてください。

```

# VIPアドレスを特定する
$ crsctl stat res <VIP name> -f | grep USR_ORA_VIP |grep -v GEN
# sshd_configファイルでListenAddressが正しいことをチェックする：
ListenAddress <VIP network address>

```

SSHデーモンを（rootとして）再起動します。

```

# /bin/systemctl restart sshd
# /bin/systemctl status sshd

```

acfsutil repl verifystandby/veryprimaryにより、プライマリとスタンバイ両方のホストから'0'が結果として返されることをチェックします。

## 2. プライマリOracle ACFSのバックグラウンド・リソースが実行されていない

プライマリOracle ACFSのバックグラウンド・リソースが実行されていない場合は、verifyprimaryコマンドが次のようにして失敗します。

```

$ acfsutil repl util verifyprimary /mnt/acfs_rep1 -v
- Attempting to ping scao04client07-vip1
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x
oracle@scao04client07-vip1 true 2>&1'
- ssh to primary was successful
- Checking primary file system via
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x
oracle@scao04client07-vip1 /sbin/acfsutil info fs '/mnt/acfs_rep1' > /dev/null 2>&1'
- Primary file system is mounted
- Checking primary file system availability
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x
oracle@scao04client07-vip1 /sbin/acfsutil info fs '/mnt/acfs_rep1' | grep state | grep Available >
/dev/null 2>&1'

```

```

- Primary file system is available
- Checking if the daemon is running via
  '/usr/bin/ssh -o BatchMode=true          -o Ciphers=aes128-ctr -o ConnectTimeout=3          -x
oracle@scao04client07-vip1 /sbin/acfsutil repl bg info '/mnt/acfs_rep1' | grep Current | grep ONLINE > /dev/null 2>&1'

- Could not detect a running daemon verifyprimary return
code:238

```

このエラーにより、ファイル・システムのフェイルオーバーもできなくなり、次のメッセージがスタンバイ・ホストのacfsutilトレース・ファイルに報告されます。

```

2020-05-28 09:34:01.062 :REPLFAIL:3331837120: [62774] Start: /sbin/acfsutil.bin repl failover
/mnt/acfs_rep1
2020-05-28 09:34:01.062 :REPLFAIL:3331837120: [62774]:0000:ofsrfailover: entered, isAdmin=0,
numargs=3, ac=3
2020-05-28 09:34:01.062 :REPLFAIL:3331837120: [62774]:0001:ofsrfailover:Checking mount path
/mnt/acfs_rep1
...
2020-05-28 09:34:01.620 :REPLFAIL:3331837120: [62774]:0016:uscgVerifyPrimaryV2: file system is available
2020-05-28 09:34:01.620 :REPLFAIL:3331837120: [62774]:0017:uscgInvokeCommand: invoking
'/usr/bin/ssh -o BatchMode=true          -o Ciphers=aes128-ctr -o ConnectTimeout=3          -x
oracle@scao04client07-vip1 /sbin/acfsutil repl bg info '/mnt/acfs_rep1' | grep Current | grep ONLINE > /dev/null 2>&1'
2020-05-28 09:34:01.910 :REPLFAIL:3331837120: [62774]:0018:ERROR: uscgVerifyPrimaryV2: /usr/bin/ssh
-o BatchMode=true          -o Ciphers=aes128-ctr -o ConnectTimeout=3          -x oracle@scao04client07-vip1
/sbin/acfsutil repl bg info '/mnt/acfs_rep1' | grep Current | grep ONLINE > /dev/null 2>&1 did not detect a running daemon
2020-05-28 09:34:01.910 :REPLFAIL:3331837120: [62774]:0019:ofsrfailover: using reverse, isAdmin=0 2020-05-28 09:34:01.910
:REPLFAIL:3331837120: [62774]:0020:uscgInvokeCommand: invoking
'/usr/bin/ssh -o BatchMode=true          -o Ciphers=aes128-ctr          -x oracle@scao04client07-vip1
/sbin/acfsutil repl reverse primary '/mnt/acfs_rep1' > /dev/null' 2020-05-28
09:34:07.087 :REPLFAIL:3331837120: [62774] Returned:1

```

プライマリ・ホストでOracle ACFSバックグラウンド・プロセスのステータスをチェックします。

```

$ /sbin/acfsutil repl bg info /mnt/acfs_rep1

Resource:          ora.repl.dupd.datac1.acfs_rep1.acfs
Target State:      ONLINE , OFFLINE
Current State:OFFLINE on pnode1, OFFLINE on pnode2

```

バックグラウンド・サービスを再起動します。

```

$ /sbin/acfsutil repl bg start /mnt/acfs_rep1

$ /sbin/acfsutil repl bg info /mnt/acfs_rep1
Resource:          ora.repl.dupd.datac1.acfs_rep1.acfs
Target State:      OFFLINE          , ONLINE
Current State:OFFLINE on pnode1, ONLINE on pnode2
$ /sbin/acfsutil repl info -c -v /mnt/acfs_rep1

Site:              Primary
Primary hostname:  pnode-vip1
Primary path:      /mnt/acfs_rep1

```

```
Primary status: Running
Background Resources: Active
```

スタンバイ・サーバーから、プライマリ・ファイル・システムのステータスをチェックします。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1

verifyprimary return code:0
```

### 3. プライマリまたはスタンバイのOracle ACFSサーバーにアクセスできない

プライマリ・クラスタで稼働中のOracle GoldenGateデプロイメントが明白な理由もなく動作を停止した場合は、まず、CRS トレース・ディレクトリに格納されているcrsd\_scriptagent\_oracle.trcトレース・ファイルをチェックします。それらのデプロイメントは、スタンバイ・ファイル・クラスタがプライマリからアクセス不能になった場合に停止する可能性があります。このケースが当てはまる場合は、プライマリ・サーバーのトレース・ファイルに次のメッセージが出力されます。

```
2020-07-19 23:56:00.000 :CLSDYNAM:358348544: [acfs_primary]{1:17701:16727} [check] Standby not
accessible (attempt 1 of 3)
2020-07-19 23:56:13.233 :CLSDYNAM:358348544: [acfs_primary]{1:17701:16727} [check] Standby not
accessible (attempt 2 of 3)
2020-07-19 23:56:26.463 :CLSDYNAM:358348544: [acfs_primary]{1:17701:16727} [check] Standby not
accessible (attempt 3 of 3)
2020-07-19 23:56:26.463 :CLSDYNAM:358348544: [acfs_primary]{1:17701:16727} [check] Disabling
primary FS access via acfs_primary service
2020-07-19 23:56:26.463 :CLSDYNAM:358348544: [acfs_primary]{1:17701:16727} [check] ACFS primary
file system problem - manual intervention required
2020-07-19 23:56:26.464 : AGFW:362551040: [ INFO] {1:17701:16727} acfs_primary 1 1 state
changed from:ONLINE to:OFFLINE
```

From current primary server, verify the standby file system:

```
$ acfsutil repl util verifystandby /mnt/acfs_rep1

verifystandby return code:1
```

acfsutil repl util verifystandbyコマンドの出力は、画面だけでなく、CRSトレース・ディレクトリに格納されているacfsutilトレース・ファイルにも表示されます。

```
2020-07-19 23:56:10.110 :REPLCMP:3508260032: [333747]:0006:uscgInvokeCommand: invoking '/bin/ping
-c 1 -w 3 scao04client07-vip1 > /dev/null 2>&1'
2020-07-19 23:56:13.117 :REPLCMP:3508260032: [333747]:0007:ERROR: uscgVerifyStandbyV2: could not
ping (/bin/ping -c 1 -w 3 snode-vip > /dev/null 2>&1), ret=1
```

何らかの理由でスタンバイ・サーバーへの接続が遮断されていることは明らかです。pingコマンドを手動実行すると、これにも失敗します。

```
$/bin/ping -c 1 -w 3 snode-vip

PING snode-vip.example.com (1.234.56.789) 56(84) bytes of data.

--- snode-vip.example.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 2999ms
```

スタンバイ・サーバーが稼働しており、プライマリとスタンバイ両方のサーバーでVIPが実行されていることを確認します。次のコマンドを使用して、ステータスをチェックし、VIPを起動します。

```
$ crsctl stat res gg_vip_primy
```

```
$ crsctl start res gg_vip_prmy
```

acfs\_primaryリソースはハブのプライマリ・クラスタで自動的に再起動しますが、手動で再起動することも可能で、その場合は続いてOracle GoldenGateデプロイメントを再起動します。

```
$ crsctl start res acfs_primary
```

## 4. Oracle ACFS ReplicationユーザーによるSSHの問題

Oracle ACFS Replicationでは、sshを使用してプライマリとスタンバイのファイル・システム間の通信を行います。プライマリまたはスタンバイ・サーバーのいずれかのOracle ACFS Replicationユーザーで問題が発生すると、acfsutil repl util verifyprimary/verifystandbyコマンドが次のエラーを出力して失敗します。このエラーは、crsd\_scriptagent\_oracle.trcトレース・ファイルでも報告されます。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1 -v
```

```
- Attempting to ping pnode-vip
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip true 2>&1'
- ssh output: WARNING: Your password has expired.
- ssh output:Password change required but no TTY available.
- ssh output: WARNING: Your password has expired.
- ssh output:Password change required but no TTY available.
- ssh attempt failed, ret=1
verifyprimary return code:255
```

ユーザーによる問題は、プライマリ・ホストへのSSH接続を試すことによって確認可能です。

```
$ ssh oracle@pnode-vip
```

```
You are required to change your password immediately (password aged)
Last login:Thu Jun 11 20:24:13 PDT 2020
Last login:Thu Jun 11 20:26:06 2020 from pnode1.example.com
WARNING:Your password has expired.
You must change your password now and login again!
Changing password for user oracle.
Changing password for oracle.
```

プライマリ・ホストでユーザー・パスワードをリセットして、verifyコマンドを再試行します。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1 -v
```

```
- Attempting to ping pnode-vip
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip true 2>&1'
- ssh to primary was successful
- Checking primary file system via
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip /sbin/acfsutil info fs '/mnt/acfs_rep1' > /dev/null 2>&1'
- Primary file system is mounted
- Checking primary file system availability
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip
```



```
/sbin/acfsutil info fs '/mnt/acfs_rep1' | grep state | grep Available > /dev/null 2>&1'
- Primary file system is available
- Checking if the daemon is running via
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip
/sbin/acfsutil repl bg info '/mnt/acfs_rep1' | grep Current | grep ONLINE > /dev/null 2>&1'
- Daemon is ONLINE
verifyprimary return code:0
```

## 5. SSHホスト鍵認証問題

プライマリまたはスタンバイのファイル・システム・サーバーでのSSH鍵の変更で問題が発生すると、Oracle ACFS Replicationが次のエラーを出力して停止します。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1 -v

- Attempting to ping pnode-vip
- ping successful
- Attempting to ssh
  '/usr/bin/ssh -o BatchMode=true -o Ciphers=aes128-ctr -o ConnectTimeout=3 -x oracle@pnode-vip true 2>&1'
- ssh output:Host key verification failed.
- ssh attempt failed, ret=255
verifyprimary return code:255
```

このエラーは、次のように障害を報告した現行のプライマリまたはスタンバイ・サーバーに対してsshコマンドを手動で発行することによって解決可能です。

```
$ ssh oracle@pnode-vip

The authenticity of host 'pnode-vip (1.234.567.89)' can't be established.
ECDSA key fingerprint is SHA256:itKt1FeSjPESSfi4LVboJAljXAHwVdQHh+8racY/SNE.
ECDSA key fingerprint is MD5:c0:29:ea:db:f0:d2:7f:81:07:a1:7a:df:22:e3:24:b6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pnode-vip,1..234.567.89' (ECDSA) to the list of known hosts.
Last login: Tue Jun 30 11:44:59 PDT 2020 from client01.example.com on pts/2
Last login: Tue Jun 30 11:54:04 2020 from pnode1.example.com
```

sshが正しく機能するようになったことを確認します。

```
$ acfsutil repl util verifyprimary /mnt/acfs_rep1

verifyprimary return code:0
```

## Oracle GoldenGateのトラブルシューティング

場合によっては、Oracle RACノードでOracle GoldenGateのプロセスが正常に開始されないこともあります。その場合は、Oracle GoldenGate、Oracle XAG、およびOracle CRSによって生成されるいくつかのファイルを確認して、問題の原因を特定します。

以下に、重要なログ・ファイルとトレース・ファイルの一覧、その場所および出力の例を示します。

### 1. Oracle XAGのログ・ファイル

場所：<XAG installation directory>/log/<hostname>

場所の例：/u01/oracle/xag/log/pnode1

ファイル名：agctl\_goldengate\_oracle.trc

agctlによって実行されるすべてのコマンド、およびOracle CRSによって実行されるコマンドを含むコマンドによる出力が記録されています。

例：

```
2020-07-16 20:26:48: agctl start goldengate TARGET
2020-07-16 20:26:48: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl status res xag.TARGET.goldengate
2020-07-16 20:26:48: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl status res xag.TARGET.goldengate -f
2020-07-16 20:26:49: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl start resource xag.TARGET.goldengate -f
2020-07-16 20:26:55: Command output:
> CRS-2672:Attempting to start 'xag.TARGET.goldengate' on 'pnode1'
> CRS-2676:Start of 'xag.TARGET.goldengate' on 'pnode1' succeeded
>End Command output
```

## 2. Oracle XAG GoldenGateインスタンスのトレース・ファイル

場所：<XAG installation directory>/log/<hostname>

場所の例：/u01/oracle/xag/log/pnode1

ファイル名：<GoldenGate\_instance\_name>\_agent\_goldengate.trc

agctlによって実行されたコマンドからの出力と使用された環境変数、および基礎となるコマンドで有効化されたデバッグ出力が記録されています。

抜粋例：

```
2020-07-19 23:02:37:Getting attributes for xag.TARGET.goldengate 1 1

2020-07-19 23:02:37:GG_HOME = /u01/oracle/goldengate/gg19_db19_MS
2020-07-19 23:02:37:GG_CONF_HOME = /mnt/acfs_rep1/deployments/ggsm19/etc/conf
2020-07-19 23:02:37:GG_VAR_HOME = /mnt/acfs_rep1/deployments/ggsm19/var
2020-07-19 23:02:37:GG_SVC_MGR_PORT = 9100
2020-07-19 23:02:37:GG_ADMIN_USERNAME = admin
2020-07-19 23:02:37:GG_OGG_CONTAINER =
2020-07-19 23:02:37:ENVIRONMENT_VARS =
2020-07-19 23:02:37:DATAGUARD_AUTOSTART = no
2020-07-19 23:02:37:VIP_NAME =
2020-07-19 23:02:37:START_TIMEOUT = 300
2020-07-19 23:02:37: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl get credmaint -path xag_ogg_user_TARGET -credtype
userpass -id 0 -attr passwd
2020-07-19 23:02:37: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl status res xag.TARGET.goldengate -f
2020-07-19 23:02:38:Exported GGS_HOME /u01/oracle/goldengate/gg19_db19_MS
2020-07-19 23:02:38:Exported OGG_CONF_HOME /mnt/acfs_rep1/deployments/ggsm19/etc/conf
2020-07-19 23:02:38:Exported LD_LIBRARY_PATH
/u01/oracle/goldengate/gg19_db19_MS:/u01/app/19.0.0.0/grid/lib:/etc/ORCLcluster/lib 2020-07-19 23:02:38:Exported
LD_LIBRARY_PATH_64 /u01/oracle/goldengate/gg19_db19_MS
2020-07-19 23:02:38:Exported LIBPATH /u01/oracle/goldengate/gg19_db19_MS

2020-07-19 23:02:38: ogg input =
{"oggHome":"/u01/oracle/goldengate/gg19_db19_MS","serviceManager":{"oggConfHome":"/mnt/acfs_rep1/de
ployments/ggsm19/etc/conf","portNumber":9100},"username":"","credential":""}
```

2020-07-19 23:02:38>About to exec /u01/oracle/goldengate/gg19\_db19\_MS/bin/XAGTask HealthCheck

2020-07-19 23:02:39:XAGTask retcode = 0

### 3. CRSトレース・ファイル

場所：/u01/app/oracle/diag/crs/<hostname>/crs/trace

場所の例：/u01/app/oracle/diag/crs/pnode1/crs/trace

ファイル名：crsd\_scriptagent\_oracle.trc

任意のOracle CRSリソース・アクション・スクリプト（XAGまたはdbfs\_mountなど）によって作成される出力が記録されています。このトレース・ファイルは、Oracle RACノードでOracle DBFSまたはGoldenGateが起動しなかった理由を判断するのに不可欠です。

例：

```
2020-07-16 20:26:49.152 :          AGFW:2208286464: [          INFO] {1:9722:45224} Agent received the
message:RESOURCE_START[xag.TARGET.goldengate 1 1] ID 4098:299415

2020-07-16 20:26:49.152 :          AGFW:2208286464: [          INFO] {1:9722:45224} Preparing START command
for: xag.TARGET.goldengate 1 1

2020-07-16 20:26:49.152 :          AGFW:2208286464: [          INFO] {1:9722:45224} xag.TARGET.goldengate 1 1
state changed from:OFFLINE to:STARTING

2020-07-16 20:26:49.153 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start]
Executing action script: /u01/oracle/XAG/bin/agggoldengatescaas[start]

2020-07-16 20:26:49.305 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] GG
agent running command 'start' on xag.TARGET.goldengate

2020-07-16 20:26:49.605 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start]
Starting OGG SCA instance

2020-07-16 20:26:52.661 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start]
ServiceManager fork pid = 258069

2020-07-16 20:26:52.661 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] Waiting
for /mnt/acfs_rep1/deployments/ggsm19/var/run/ServiceManager.pid

2020-07-16 20:26:52.661 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] Waiting
for SM to start

2020-07-16 20:26:52.661 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start]
ServiceManager PID = 258071

2020-07-16 20:26:52.762 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] execute
XAGTask HealthCheck

2020-07-16 20:26:53.814 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] XAGTask
retcode = 1

2020-07-16 20:26:53.814 :CLSDYNAM:2197780224: [xag.TARGET.goldengate]{1:9722:45224} [start] XAG
HealthCheck after start returned 1

2020-07-16 20:26:53.814 :          AGFW:2197780224: [          INFO] {1:9722:45224} Command: start for
resource: xag.TARGET.goldengate 1 1 completed with status:SUCCESS
```

## 4. Oracle GoldenGateデプロイメントのログ・ファイル

場所：<Goldengate\_deployment\_directory>/<deployment\_name>/var/log

場所の例：/mnt/acfs\_rep1/deployments/TARGET/var/log

ファイル名：adminsrvr.log, recvsrvr.log, pmsrvr.log, distsrvr.log

Oracle GoldenGateデプロイメント・プロセス（管理サーバー、配布サーバー、レシーバ・サーバー、およびパフォーマンス・メトリック・サーバー）の開始、停止、およびステータス・チェックの出力が記録されています。

## 5. Oracle GoldenGateレポート・ファイル

場所：<Goldengate\_deployment\_directory>/<deployment\_name>/var/lib/report

場所の例：/mnt/acfs\_rep1/deployments/TARGET/var/lib/report

GoldenGateレポート・ファイルには、Managerプロセスを含むすべてのGoldenGateプロセスの重要な情報、警告メッセージ、およびエラーが記録されています。GoldenGateプロセスのいずれかが開始しなかったり、実行中に異常終了したりすると、そのプロセスのレポート・ファイルに、障害の原因を特定するのに役立つ重要な情報が記録されます。

## CONNECT WITH US

+1.800.ORACLE1までご連絡いただくか、[oracle.com](http://oracle.com)をご覧ください。  
北米以外の地域では、[oracle.com/contact](http://oracle.com/contact)で最寄りの営業所をご確認いただけます。

 [blogs.oracle.com](http://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

[twitter.com/oracle](https://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证や法律による黙示的保证を含め、商品性ないし特定目的適合性に関する黙示的保证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 Maximum Availability Architecture (Oracle MAA) GoldenGate Hub  
2020年12月

著者：Stephan Haisley

