ORACLE

Oracle GoldenGate Microservices Architecture on Oracle Exadata Database Service Configuration Best Practices

December 2022, Version 1.4

Public

Purpose statement

This document describes the best practices for configuring Oracle GoldenGate Microservices Architecture to work with Oracle Exadata Database Service on Dedicated Infrastructure (ExaDB-D) or Oracle Exadata Database Service on Cloud@Customer (ExaDB-C@C) and Oracle Database File System (DBFS) or Oracle ASM Cluster File System (ACFS).

Disclaimer

Table of contents

Introduction

This technical whitepaper describes best practices for configuring Oracle GoldenGate Microservices Architecture to work with Oracle Exadata Database Service on Dedicated Infrastructure (ExaDB-D) or Oracle Exadata Database Service on Cloud@Customer (ExaDB-C@C) and Oracle Database File System (DBFS) or Oracle ASM Cluster File System (ACFS). The target Oracle Exadata Database Service that hosts Oracle GoldenGate Microservices Architecture can act as the source database, the target database, or in some cases, as both source and target databases for Oracle GoldenGate. This technical whitepaper is applicable for configuring Oracle GoldenGate Microservices Architecture with Oracle Exadata Database Service on Dedicated Infrastructure or Cloud@Customer.

Oracle GoldenGate is instrumental for many reasons, including the following:

- To support Oracle Platinum MAA reference architecture resulting in zero, or near zero, database and application downtime for planned and unplanned outages. Refer to Oracle MAA Reference Architectures or Oracle Exadata MAA - Platinum Tier-Focused Presentation
- To migrate to an Oracle Database with minimal, or zero downtime
- To implement a near real-time data warehouse or consolidated database on Oracle RAC, sourced from various, possibly heterogeneous, source databases, populated by Oracle GoldenGate
- To capture data from an OLTP application running on Oracle RAC to support further downstream consumption, such as middleware integration

Configuration Overview

This section introduces Oracle GoldenGate Microservices Architecture, Oracle RAC, Oracle Clusterware, Oracle Database File System (DBFS), and Oracle ASM Cluster File System (ACFS). For more information about these features, see the References section at the end of this technical brief.

Oracle GoldenGate

Oracle GoldenGate provides real-time, log-based change data capture and delivery between homogenous and heterogeneous systems. This technology enables you to construct a cost-effective, low-impact real-time data integration and continuous availability solution.

Oracle GoldenGate replicates data from committed transactions with transaction integrity and minimal overhead on your existing infrastructure. The architecture supports multiple data replication topologies such as one-to-many, many-to-many, cascading, and bidirectional. Its wide variety of use cases includes real-time business intelligence; query offloading; zero-downtime upgrades and migrations; and active-active databases for data distribution, data synchronization, and high availability.

Oracle GoldenGate Microservices Architecture provides REST-enabled services. The REST-enabled services provide remote configuration, administration, and monitoring through HTML5 web pages, command line interfaces, and APIs. Figure 1 shows the Oracle GoldenGate Microservices Architecture referenced throughout this technical brief.

Recommended Oracle GoldenGate 21c (and higher releases) introduces unified build support, so a single software installation supports capturing and applying replicated data to multiple major Oracle Database versions (11g Release 2 to 21c). This is possible because an Oracle GoldenGate installation includes the required Oracle database client libraries without requiring a separate `ORACLE_HOME` installation.

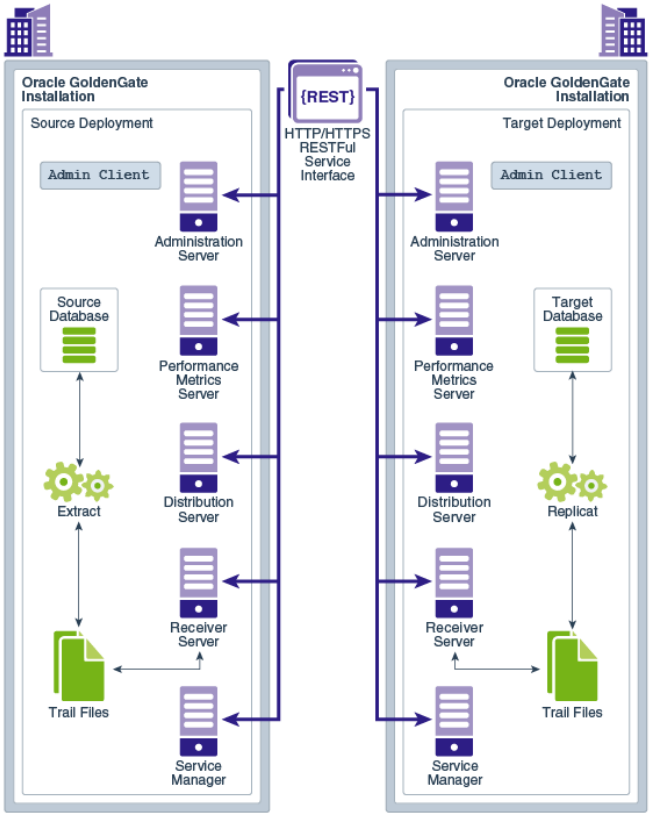

Figure 1: Oracle GoldenGate Microservices Architecture

More information about Oracle GoldenGate Microservices Architecture can be found in the Oracle GoldenGate documentation central hub referencing the highest production GoldenGate release.

Oracle Real Application Clusters

Oracle Real Application Clusters (Oracle RAC) enable multiple instances linked by an interconnect to share access to an Oracle Database. In an Oracle RAC environment, the database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database that spans multiple hardware systems, enabling Oracle RAC to provide high availability and redundancy during failures in the cluster. Oracle RAC accommodates all system types, from read-only data warehouse systems to update-intensive online transaction processing (OLTP) systems.

Oracle Real Application Clusters (RAC) and Oracle Clusterware allow the Oracle Database to run any packaged or custom application across a set of clustered servers. This capability provides continual database service uptime for node and instance failures, most planned maintenance activities, and Oracle RAC expansion. If an Oracle RAC clustered node or instance fails, the Oracle Database service runs on the surviving nodes and instances. When more processing power is needed, you can add another node without interrupting user access to the database or data.

Oracle Clusterware

Oracle Clusterware is a cluster manager explicitly designed for the Oracle Database. Oracle Clusterware monitors all Oracle resources (such as database instances and listeners) in an Oracle RAC environment. If a failure occurs, Oracle Clusterware automatically attempts to restart the failed resource. During outages, Oracle Clusterware relocates the processing performed by the inoperative resource to a backup resource. For example, if a node fails, then Oracle Clusterware relocates the database services used by the application to surviving RAC nodes and instances in the RAC cluster.

Oracle Grid Infrastructure Agents

Oracle Grid Infrastructure Agents (XAG) are Oracle Grid Infrastructure components that provide the high availability (HA) framework to application resources and resource types managed through the agent management interface, AGCTL. This framework provides a complete, ready-to-use application HA solution that contains pre-defined Oracle Grid Infrastructure resource configurations and agents to integrate applications for complete application HA.

The Oracle Grid Infrastructure Agents provide pre-defined Oracle Clusterware resources for Oracle GoldenGate, Siebel, Oracle PeopleSoft, JD Edwards, and Oracle WebLogic Server, as well as Apache and MySQL applications. Using the agent for Oracle GoldenGate simplifies the creation of dependencies on the source/target database, the application VIP, and the file system (ACFS or DBFS) mount point. The agent command line utility (AGCTL) starts and stops Oracle GoldenGate and can relocate Oracle GoldenGate between the nodes in the cluster.

Oracle Database File System (DBFS)

The Oracle Database File System (DBFS) creates a file system interface to files stored in the database. DBFS is similar to NFS in that it provides a shared network file system that looks like a local file system. Because the data is stored in the database, the file system inherits all the high availability and disaster recovery capabilities provided by Oracle Database.

With DBFS, the server is the Oracle Database. Files are stored as SecureFiles LOBs. PL/SQL procedures implement file system access primitives such as create, open, read, write, and list directories. The implementation of the file system in the database is called the DBFS SecureFiles Store. The DBFS SecureFiles Store allows users to create file systems that clients can mount. Each file system has its own dedicated tables that hold the file system content.

Oracle Automatic Storage Management Cluster File System (ACFS)

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files.

Oracle ACFS leverages Oracle Clusterware for cluster membership state transitions and resource-based high availability. Oracle ACFS is bundled into the Oracle Grid Infrastructure (GI), allowing for the integrated, optimized management of databases, resources, volumes, and file systems.

Configuration Overview

This section provides an overview of the steps that you need to follow to configure Oracle GoldenGate on Oracle Exadata Database Service on Dedicated Infrastructure (ExaDB-D) or Oracle Exadata Database Service on Cloud@Customer. The rest of the paper provides the details of these steps.

Here are the steps performed in this section:

- Step 1 - Before You Begin: To configure Oracle GoldenGate on Oracle Exadata Cloud Infrastructure or Cloud@Customer, you need a ExaDB-D or ExaDB-C@C system, CA certificates, and configure some extra software.
- Step 2 - Configure the Oracle Database for GoldenGate: Use best practices to configure the source and target databases in an Oracle GoldenGate replicated environment.
- Step 3 - Create a Shared File System to Store the GoldenGate Deployment: You must set up either Oracle DBFS or Oracle ACFS for configuring HA on Oracle Cloud Infrastructure with Oracle GoldenGate. If this architecture has GoldenGate replica database protected by a cloud physical standby database (Data Guard), use Oracle DBFS; other use ACFS.
- Step 4 - Install Oracle GoldenGate: Use best practices to install and configure Oracle GoldenGate components on Oracle Cloud Infrastructure.
- Step 5 - Create the Oracle GoldenGate Deployment
- Step 6 - Oracle Cloud Infrastructure Networking: You must configure virtual cloud network (VCN) components such as private DNS zones, VIP, bastion, security lists and firewalls for Oracle GoldenGate to function properly.
- Step 7 - Configure Oracle Grid Infrastructure Agent (XAG): You configure Oracle GoldenGate for HA on Oracle Cloud Infrastructure.
- Step 8 - Configure NGINX Reverse Proxy: Configure reverse proxy and HA by using Nginx.
- Step 9 - Create Oracle Net TNS Alias for Oracle GoldenGate Database Connections: You create a TNS alias to simplify database connectivity of the Oracle GoldenGate processes when switching between Oracle RAC nodes.
- Step 10 - Configure Oracle GoldenGate Processes: Create and configure Oracle GoldenGate Extract, Replicat, and Path processes need for data replication
- Step 11 - Configure Autostart of Extract and Replicat Processes

Configuration Best Practices

Step 1 - Before You Begin

Here are the steps performed in this section:

- Step 1.1 - Set Up the Oracle Cloud Infrastructure DB System
- Step 1.2 - Download the Required Software
- Step 1.3 - Configure Your System to Install Software from Oracle Linux Yum Server
- Step 1.4 - Secure Deployments Requirements (Certificates)

Step 1.1 - Set Up the Oracle Cloud Infrastructure DB System

To get started, you need an Oracle Exadata Database Service on Dedicated Infrastructure or Cloud@Customer for Oracle GoldenGate deployment. You can deploy Oracle GoldenGate with an existing ExaDB-D/ExaDB-C@C system or launch a new system, according to your business needs. For instructions on launching and managing an ExaDB-D system, see Oracle Exadata Database Service on Dedicated Infrastructure or for ExaDB-C@C see Oracle Exadata Database Service on Cloud@Customer.

Step 1.2 - Download the Required Software

- Create the staging directory to download all the required software

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# mkdir /u02/app_acfs/goldengate
[root@exadb-node1 ~]# chown oracle:oinstall /u02/app_acfs/goldengate
[root@exadb-node1 ~]# chmod g+w /u02/app_acfs/goldengate
```

- Download the Oracle GoldenGate 21c Microservices software, or higher, from Oracle GoldenGate Downloads.
- Download the Oracle Grid Infrastructure Standalone Agents for Oracle Clusterware 19c, version 10.2 or higher, from Oracle Grid Infrastructure Standalone Agents for Oracle Clusterware.
- Download the mount-dbfs-<version>.zip file with mount-dbfs.sh and mount-dbfs.conf from Document 1054431.1
- Download the python script (secureServices.py) from Document 2826001.1

Step 1.3 - Configure Your System to Install Software from Oracle Linux Yum Server

Oracle Linux yum server hosts software for Oracle Linux and compatible distributions. These instructions help you get started configuring your Linux system for Oracle Linux yum server and installing software via yum.

As the root OS user, create the file `/etc/yum.repos.d/oracle-public-yum-ol7.repo` with the following contents:

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]#
cat > /etc/yum.repos.d/oracle-public-yum-ol7.repo <<EOF
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://yum$ociregion.oracle.com/repo/OracleLinux/OL7/latest/\$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
EOF
```

As the root OS user, follow Doc ID 2397264.1 to modify the configuration file `/etc/yum.conf` and validate the software repositories are enabled:

```
[root@exadb-node1 ~]# yum repolist
repo id                              repo name                                          status
!public_ol7_latest                   Oracle Linux 7.9-6.0.1.el7_9 Latest (x86_64)       19,712+4,957
```

```
repolist: 19,992
```

Step 1.4 - Secure Deployments Requirements (Certificates)

A secure deployment involves making RESTful API calls and conveying trail data between the Distribution Server and Receiver Server, over SSL/TLS. You can use your own existing business certificate from your Certificate Authority (CA) or you might create your own certificates. Contact your systems administrator to follow your corporate standards to create or obtain the server certificate before proceeding. A separate certificate is required for each VIP and Service Manager pair.

Step 2 - Configure the Oracle Database for GoldenGate

The source and target Oracle GoldenGate databases should be configured using the following recommendations.

Here are the steps performed in this section:

- Step 2.1 - Database Configuration
- Step 2.2 - Create the Database Replication Administrator User
- Step 2.3 - Create the Database Services

Step 2.1 - Database Configuration

The source and target Oracle GoldenGate databases should be configured using the following recommendations.

- Enable Oracle GoldenGate replication by setting the database initialization parameter.
- Source Oracle GoldenGate Database:
  - Run the database in ARCHIVELOG mode
  - Enable FORCE LOGGING mode
  - Enable minimal supplemental logging
  - Additionally, add schema or table level logging for all replicated objects
- Configure the streams pool in the System Global Area (SGA) on the source database using the STREAMS_POOL_SIZE initialization parameter. The streams pool is only needed on the target database if integrated Replicat will be used.

For the steps on preparing the database for Oracle GoldenGate, refer to the *Using Oracle GoldenGate with Oracle Database Guide*.

As the `oracle` OS user on the source and target systems, execute the following SQL instructions to configure the database:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ source <db_name>.env
[oracle@exadb-node1 ~]$ sqlplus / as sysdba
SQL> alter system set ENABLE_GOLDENGATE_REPLICATION=true scope=both sid='*';
SQL> alter system set STREAMS_POOL_SIZE=<SIZE_IN_GB> scope=both sid='*';
```

As the `oracle` OS user on the source system, execute the following SQL instructions to configure the database:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ source <db_name>.env
[oracle@exadb-node1 ~]$ sqlplus / as sysdba
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> ARCHIVE LOG LIST
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     110
Next log sequence to archive   113
Current log sequence           113
```

Step 2.2 - Create the Database Replication Administrator User

The source and target Oracle databases need a GoldenGate Administrator user created, with appropriate privileges assigned:

- For multitenant container database (CDB):
  - » Source database, GoldenGate Extract must be configured to connect to a user in the root container database, using a c## account.
  - » Target database, a separate GoldenGate administrator user is needed for each pluggable database (PDB). For further details on creating a GoldenGate Administrator in an Oracle Multitenant Database, refer to the Using Oracle GoldenGate with Oracle Database Guide.
- For non-CDB databases, refer to the Using Oracle GoldenGate with Oracle Database Guide

As the `oracle` OS user on the source system, execute the following SQL instructions to create the database user for Oracle GoldenGate and assign the required privileges:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ source <db_name>.env
[oracle@exadb-node1 ~]$ sqlplus / as sysdba
# CDB
alter session set container=cdb$root;
create user c##ggadmin identified by "<ggadmin_password>" container=all default tablespace USERS temporary
tablespace temp;
alter user c##ggadmin quota unlimited on users;
grant set container to c##ggadmin container=all;
grant alter system to c##ggadmin container=all;
grant create session to c##ggadmin container=all;
grant alter any table to c##ggadmin container=all;
grant resource to c##ggadmin container=all;
exec dbms_goldengate_auth.grant_admin_privilege('c##ggadmin',container=>'all');

# Source PDB
alter session set container=<PDB_name>;
create user ggadmin identified by "<ggadmin_password>" container=current;
grant create session to ggadmin container=current;
grant alter any table to ggadmin container=current;
grant resource to ggadmin container=current;
exec dbms_goldengate_auth.grant_admin_privilege('ggadmin');
```

As the `oracle` OS user on the target system, execute the following SQL instructions to create the database user for Oracle GoldenGate and assign the required privileges:

```
# Target PDB
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ source <db_name>.env
[oracle@exadb-node1 ~]$ sqlplus / as sysdba

alter session set container=<PDB_name>;
create user ggadmin identified by "<ggadmin_password>" container=current;
grant alter system to ggadmin container=current;
grant create session to ggadmin container=current;
grant alter any table to ggadmin container=current;
grant resource to ggadmin container=current;
grant dv_goldengate_admin, dv_goldengate_redo_access to ggadmin container=current;
```

```
exec dbms_goldengate_auth.grant_admin_privilege('ggadmin');
```

Step 2.3 - Create the Database Services

A database service is required so that the Oracle Grid Infrastructure Agent will automatically start the Oracle GoldenGate deployment when the database is opened. When DBFS is used for the shared file system, the database service is also used to mount DBFS to the correct RAC instance.

When using a source multitenant database, a separate service is required for the root container database (CDB) and the pluggable database (PDB) that contains the schema being replicated. For a target multitenant database, a single service is required for the PDB.

As the `oracle` OS user, create and start the CDB database service using the following command:

```
[oracle@exadb-node1 ~]$ source <db_name>.env
[oracle@exadb-node1 ~]$ srvctl add service -db $ORACLE_UNQNAME -service `echo $ORACLE_UNQNAME`_ogg -
preferred <SID1> -available <SID2> -role PRIMARY
[oracle@exadb-node1 ~]$ srvctl start service -db $ORACLE_UNQNAME -service `echo $ORACLE_UNQNAME`_ogg
```

If your database is part of a multitenant environment, remember to create the service at the pluggable database (PDB).

As the `oracle` OS user, create and start the PDB database service using the following command:

```
[oracle@exadb-node1 ~]$ dbaascli database getDetails --dbname <db_name> |grep pdbName
      "pdbName" : "<PDB_NAME>",
[oracle@exadb-node1 ~]$ srvctl add service -db $ORACLE_UNQNAME -service <PDB_NAME>_ogg -preferred
<SID1>,<SID2> -pdb <PDB_NAME> -role PRIMARY
[oracle@exadb-node1 ~]$ srvctl start service -db $ORACLE_UNQNAME -service <PDB_NAME>_ogg
```

As the `oracle` OS user, verify that the services are running:

```
[oracle@exadb-node1 ~]$ srvctl status service -d $ORACLE_UNQNAME |grep _ogg
Service <ORACLE_UNQNAME>_ogg is running on instance(s) <SID1>
Service <PDB_NAME>_ogg is running on instance(s) <SID1>
```

Refer to the Real Application Clusters Administration and Deployment Guide for further details on creating a database services.

Step 3 - Create a Shared File System to Store the GoldenGate Deployment

Oracle GoldenGate Microservices Architecture is designed with a simplified installation and deployment directory structure. The installation directory: should be placed on local storage on each database node to minimize downtime during software patching. The deployment directory: which is created during deployment creation using the Oracle GoldenGate Configuration Assistant (oggca.sh), must be placed on a shared file system. The deployment directory contains configuration, security, log, parameter, trail, and checkpoint files. Placing the deployment in DBFS or Oracle Automatic Storage Management Cluster File System (ACFS) provides the best recoverability and failover capabilities in the event of a system failure. Ensuring the availability of the checkpoint files cluster-wide is essential so that the GoldenGate processes can continue running from their last known position after a failure occurs.

If Oracle GoldenGate will be configured along with Oracle Data Guard, the recommended file system is DBFS. DBFS is contained in the database protected by Data Guard and can be fully integrated with XAG. In the event of a Data Guard role transition, the file system can be automatically mounted on the new primary server, followed by the automated start-up Oracle GoldenGate. This is currently not possible with ACFS since it is not part of the Oracle Data Guard configuration.

***NOTE**: This document does not include steps to configure Oracle GoldenGate with Oracle Data Guard.*

If Oracle Data Guard is not present, the recommended file system is ACFS. ACFS is a multi-platform, scalable file system and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support customer files maintained outside the Oracle Database.

Here are the steps performed in this section; follow the instructions below to configure the chosen file system:

- Step 3a - Oracle Database File System (DBFS)
- Step 3b - Oracle ASM Cluster File System (ACFS)

Step 3a - Oracle Database File System (DBFS)

You must create the DBFS tablespace inside the same database to which the Oracle GoldenGate processes are connected. For example, if an Oracle GoldenGate integrated Extract process is extracted from a database called `GGDB`, the DBFS tablespace would be located in the same `GGDB` database.

Create a file system for storing the Oracle GoldenGate deployment files. You should allocate enough trail file disk space to permit storage of up to 12 hours of trail files. Doing this will give sufficient space for trail file generation should a problem occur with the target environment that prevents it from receiving new trail files. The amount of space needed for 12 hours can only be determined by testing trail file generation rates with real production data.

Here are the steps performed in this section:

- Step 3a.1 - Configuring DBFS on Oracle Exadata Database Service
- Step 3a.2 - Create the DBFS Repository
- Step 3a.3 - (Only for CDB) Create an Entry in TNSNAMES
- Step 3a.4 - Download and Edit the mount-dbfs Scripts
- Step 3a.5 - Register the DBFS Resource with Oracle Clusterware
- Step 3a.6 - Start the DBFS Resource

Step 3a.1 - Configuring DBFS on Oracle Exadata Database Service

As the `opc` OS user, add the grid user to the fuse group:

```
[opc@exadb-node1]$ sudo -u grid $(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)/bin/olsnodes >
~/dbs_group
[opc@exadb-node1]$ dcli -g ~/dbs_group -l opc sudo usermod -a -G fuse grid
```

As the `opc` OS user, validate that the file `/etc/fuse.conf` exists and contains the `user_allow_other` option:

```
[opc@exadb-node1]$ cat /etc/fuse.conf
# mount_max = 1000
# user_allow_other
```

As the `opc` OS user, skill this step if the option `user_allow_other` is already in the `/etc/fuse.conf` file. Otherwise, execute the following commands to add the option:

```
[opc@exadb-node1]$ dcli -g ~/dbs_group -l opc "echo user_allow_other | sudo tee -a /etc/fuse.conf"
```

As the `opc` OS user, create an empty directory that will be used as the mount point for the DBFS filesystem:

```
[opc@exadb-node1]$ dcli -g ~/dbs_group -l opc sudo mkdir -p /mnt/dbfs
```

As the `opc` OS user, change ownership on the mount point directory so the grid OS user can access it:

```
[opc@exadb-node1]$ dcli -g ~/dbs_group -l opc sudo chown oracle:oinstall /mnt/dbfs
```

Step 3a.2 - Create the DBFS Repository

Create the DBFS repository inside the target database. To create the repository, create a new tablespace within the target PDB to hold the DBFS objects and a database user that will own the objects.

***NOTE**: When using an Oracle Multitenant Database, the DBFS tablespace MUST be created in a Pluggable Database (PDB). It is recommended that you use the same PDB that the GoldenGate Extract or Replicat processes connect to, allowing DBFS to use the same database service created above in step 2 for its database dependency.*

As the `oracle` OS user, create the tablespace in the database:

```
[opc@exadb-node1]$ sudo su - oracle
[oracle@exadb-node1]$ source DB_NAME.env
[oracle@exadb-node1]$ sqlplus / as sysdba
SQL> alter session set container=<pdb_name>;
SQL> create bigfile tablespace dbfstb1 datafile size 32g autoextend on next 8g maxsize 300g NOLOGGING
EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;
SQL> create user dbfs_user identified by "<dbfs_user_password>" default tablespace dbfstb1 quota unlimited
on dbfstb1;
SQL> grant connect, create table, create view, create procedure, dbfs_role to dbfs_user;
```

As the `oracle` OS user, create the database objects that will hold DBFS. This script takes two arguments:

- dbfstb1: tablespace for the DBFS database objects
- goldengate: filesystem name; this can be any string and will appear as a directory under the mount point

```
[oracle@exadb-node1]$ sqlplus dbfs_user/"<dbfs_user_password>"@<db_name>_dbfs
SQL> start $ORACLE_HOME/rdbms/admin/dbfs_create_filesystem dbfstb1 goldengate
```

Step 3a.3 - (Only for CDB) Create an Entry in TNSNAMES

As the `oracle` OS user, find the database domain name:

```
[opc@exadb-node1]$ sudo su - oracle
[oracle@exadb-node1]$ source DB_NAME.env
[oracle@exadb-node1]$ sqlplus / as sysdba
SQL> show parameter db_domain

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_domain                            string      <db_domain_name>
```

As the `oracle` OS user, add a connect entry in `$TNS_ADMIN/tnsnames.ora` file:

```
[oracle@exadb-node1]$ vi $TNS_ADMIN/tnsnames.ora
dbfs =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY=LISTENER))
      (CONNECT_DATA =
        (SERVICE_NAME = <pdb_service_name>.<db_domain_name> )
       )
    )
```

As the `oracle` OS user, distribute the `$TNS_ADMIN/tnsnames.ora` file to the rest of the nodes

```
[oracle@exadb-node1 ~]$ /usr/local/bin/dcli -l oracle -g ~/dbs_group -f $TNS_ADMIN/tnsnames.ora -d
$TNS_ADMIN/
```

Step 3a.4 - Edit the mount-dbfs Scripts

Unzip the zip file and edit the variable settings in the file mount-dbfs.conf for your environment. Comments in the file will help you to confirm the values for these variables:

- DBNAME: echo $ORACLE_UNQNAME
- MOUNT_POINT: /mnt/dbfs/goldengate
- ORACLE_HOME (RDBMS ORACLE_HOME directory): echo $ORACLE_HOME
- GRID_HOME (GRID INFRASTRUCTURE HOME directory): echo $(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)
- DBFS_PASSWD (used only if WALLET=false)
- DBFS_PWDFILE_BASE (used only if WALET=false)
- WALLET (must be true or false)
- TNS_ADMIN (used only if WALLET=true or PDB): echo $TNS_ADMIN
- DBFS_LOCAL_TNSALIAS (used only if WALLET=true)
- IS_PDB (set to true if using PDB)
- PDB (PDB name, if applicable): PDB name
- PDB_SERVICE (the database service created in step 2.3, if applicable): <PDB_SERVICE_NAME>
- MOUNT_OPTIONS: allow_other,direct_io,failover,nolock
  - The `failover` option forces all file writes to be committed to the DBFS database in an IMMEDIATE WAIT mode. This prevents data from getting lost when it has been written into the dbfs_client cache but not yet written to the database at the time of a database or node failure.
  - The `nolock` mount option is required if you use Oracle Database 18c or later versions due to a change in the DBFS file locking, which can cause issues for GoldenGate processes after a RAC node failure when a file is currently locked.

As the `grid` OS user, unzip the `mount-dbfs-<version>.zip` and edit the configuration file `mount-dbfs.conf`:

```
[opc@exadb-node1]$ sudo su - grid
[grid@exadb-node1]$ cd /u02/app_acfs/goldengate
[grid@exadb-node1]$ unzip mount-dbfs-<version>.zip
[grid@exadb-node1]$ vi mount-dbfs.conf
```

Example of `mount-dbfs.conf`:

```
DBNAME=<DB_UNIQUE_NAME>
MOUNT_POINT=/mnt/dbfs/goldengate
DBFS_USER=dbfs_user
GRID_HOME=$(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)
if [ -z "${GRID_HOME}" ]; then
    echo "GRID_HOME is unset or set to the empty string"
fi
ORACLE_HOME=$($GRID_HOME/bin/srvctl config database -d $DBNAME |grep 'Oracle home:' | cut -d: -f2 |sed 's/
//g')
if [ -z "${ORACLE_HOME}" ]; then
    echo "ORACLE_HOME is unset or set to the empty string"
fi
LOGGER_FACILITY=user

MOUNT_OPTIONS=allow_other,direct_io,failover,nolock
PERL_ALARM_TIMEOUT=14
DBFS_PASSWD=<DBFS_USER_PASSWORD>
DBFS_PWDFILE_BASE=/tmp/.dbfs-passwd.txt
WALLET=false
```

```
TNS_ADMIN=$ORACLE_HOME/network/admin/<DB_NAME>
IS_PDB=true
PDB=<PDB_NAME>
PDB_SERVICE=<PDB_SERVICE_NAME>
```

As the `grid` OS user, modify the mount-dbfs.sh script to force unmounting of DBFS when the CRS resource is stopped:

```
[grid@exadb-node1]$ vi /u02/app_acfs/goldengate/mount-dbfs.sh

# Change two occurrences of:
$FUSERMOUNT -u $MOUNT_POINT
# To the following:
$FUSERMOUNT -uz $MOUNT_POINT
```

As the `opc` OS user, copy `mount-dbfs.conf` (rename it if desired or needed) to the directory `/etc/oracle` on database nodes and set proper permissions on it:

```
[opc@exadb-node1]$ sudo -u grid $(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)/bin/olsnodes >
~/dbs_group
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc -d /tmp -f /u02/app_acfs/goldengate/mount-
dbfs.conf
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo cp /u02/app_acfs/goldengate/mount-
dbfs.conf /etc/oracle
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo chown grid:oinstall /etc/oracle/mount-
dbfs.conf
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo chmod 660 /etc/oracle/mount-dbfs.conf
```

As the `opc` OS user, copy `mount-dbfs.sh` (rename it if desired or needed) to the proper directory ($GI_HOME/crs/script) on database nodes and set proper permissions on it:

```
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo mkdir $(grep ^crs_home
/etc/oracle/olr.loc | cut -d= -f2)/crs/script
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo chown grid:oinstall $(grep ^crs_home
/etc/oracle/olr.loc | cut -d= -f2)/crs/script
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l grid -d $(grep ^crs_home /etc/oracle/olr.loc |
cut -d= -f2)/crs/script -f /u02/app_acfs/goldengate/mount-dbfs.sh
[opc@exadb-node1]$ /usr/local/bin/dcli -g ~/dbs_group -l grid chmod 770 $(grep ^crs_home
/etc/oracle/olr.loc | cut -d= -f2)/crs/script/mount-dbfs.sh
```

Step 3a.5 - Register the DBFS Resource with Oracle Clusterware

When registering the resource with Oracle Clusterware, create it as a `cluster_resource`. The reason for using `cluster_resource` is so the file system can only be mounted on a single node at one time, preventing mounting of DBFS from concurrent nodes creating the potential of concurrent file writes, and causing file corruption problems.

As the `grid` OS user, find the resource name for the database service created in a previous step for the DBFS service dependency:

```
[opc@exadb-node1]$ sudo su - grid
[grid@exadb-node1]$ crsctl stat res |grep <PDB_NAME>
NAME=ora.<DB_UNIQUE_NAME>.<SERVICE_NAME>.svc
```

As the `oracle` OS user, register the Clusterware resource by executing the following script:

```
[opc@exadb-node1]$ sudo su - oracle
[oracle@exadb-node1]$ vi /u02/app_acfs/goldengate/add-dbfs-resource.sh
```

```
##### start script add-dbfs-resource.sh
#!/bin/bash
ACTION_SCRIPT=$(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)/crs/script/mount-dbfs.sh
RESNAME=dbfs_mount
DEPNAME=ora.<DB_UNIQUE_NAME>.<SERVICE_NAME>.svc
ORACLE_HOME=$(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)
PATH=$ORACLE_HOME/bin:$PATH
export PATH ORACLE_HOME
crsctl add resource $RESNAME \
  -type cluster_resource \
  -attr "ACTION_SCRIPT=$ACTION_SCRIPT, \
         CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \
         START_DEPENDENCIES='hard($DEPNAME)pullup($DEPNAME)',\
         STOP_DEPENDENCIES='hard($DEPNAME)',\
         SCRIPT_TIMEOUT=300"
##### end script add-dbfs-resource.sh

[oracle@exadb-node1]$ sh /u02/app_acfs/goldengate/add-dbfs-resource.sh
```

***Note**: After creating the $RESNAME resource, in order to stop the $DBNAME database when the $RESNAME resource is ONLINE, you will have to specify the force flag when using srvctl. For example: "srvctl stop database -d fsdb -f."*

Step 3a.6 - Start the DBFS Resource

As the grid OS user, start the resource:

```
[opc@exadb-node1]$ sudo su - grid
[grid@exadb-node1]$ crsctl start res dbfs_mount -n `hostname`
CRS-2672: Attempting to start 'dbfs_mount' on 'exadb-node1'
CRS-2676: Start of 'dbfs_mount' on 'exadb-node1' succeeded

[grid@exadb-node1]$ crsctl stat res dbfs_mount -t

--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
dbfs_mount
      1        ONLINE  ONLINE       exadb-node1              STABLE
--------------------------------------------------------------------------------
```

NOTE: Leave the shared file system mounted. It is required for creating the Oracle GoldenGate deployment in a later step.

Step 3b - Oracle ASM Cluster File System (ACFS)

Oracle ACFS is an alternative to DBFS for the shared Oracle GoldenGate files in an Oracle RAC configuration. Create a single ACFS file system for storing the Oracle deployment files.

It is recommended that you allocate enough trail file disk space to permit the storage of up to 12 hours of trail files. Doing this will give sufficient space for trail file generation should a problem occur with the target environment that prevents it from receiving new trail files. The amount of space needed for 12 hours can only be determined by testing trail file generation rates with real production data.

Here are the steps performed in this section:

- Step 3b.1 - Create the ASM File System
- Step 3b.2 - Make the File System
- Step 3b.3 - Create the Cluster Ready Services (CRS) Resource
- Step 3b.4 - Verify the Currently Configured ACFS File Systems
- Step 3b.5 - Start and Check the Status of the ACFS Resource
- Step 3b.6- Create GoldenGate ACFS Directory

Step 3b.1 - Create the ASM File System

As the grid OS user, use asmcmd to create the volume:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ asmcmd volcreate -G DATAC1 -s 1200G ACFS_GG
```

Note: Modify the file system size according to the determined size requirements.

Step 3b.2 - Make the File System

As the grid OS user, use asmcmd to confirm the “Volume Device”:

```
[grid@exadb-node1 ~]$ asmcmd volinfo -G DATAC1 ACFS_GG
```

Following is an example of the ACFS volume device output:

```
Diskgroup Name: DATAC1
         Volume Name: ACFS_GG
         Volume Device: /dev/asm/acfs_gg-151
         State: ENABLED
         Size (MB): 1228800
         Resize Unit (MB): 64
         Redundancy: MIRROR
         Stripe Columns: 8
         Stripe Width (K): 1024
         Usage:
         Mountpath:
```

As the grid OS user, make the file system with the following mkfs command:

```
[grid@exadb-node1 ~]$ /sbin/mkfs -t acfs /dev/asm/acfs_gg-151
```

Step 3b.3 - Create the Cluster Ready Services (CRS) Resource

As the opc OS user, create the ACFS mount point:

```
[opc@exadb-node1 ~]$ dcli -l opc -g ~/dbs_group sudo mkdir -p /mnt/acfs_gg
[opc@exadb-node1 ~]$ dcli -l opc -g ~/dbs_group sudo chown oracle:oinstall /mnt/acfs_gg
```

Create the file system resource as the `root` user. Due to the implementation of distributed file locking on ACFS, unlike DBFS, it is acceptable to mount ACFS on more than one RAC node at any one time.

As the root os user, create the ACFS resource for the new ACFS file system:

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# $(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)/bin/srvctl add filesystem -
device /dev/asm/acfs_gg-151 -volume ACFS_GG -diskgroup DATAC1 -path /mnt/acfs_gg -user oracle
```

Step 3b.4 - Verify the Currently Configured ACFS File Systems

As the grid OS user, use the following command to view the file system details:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ srvctl config filesystem -volume ACFS_GG -diskgroup DATAC1

Volume device: /dev/asm/acfs_gg-151
Diskgroup name: datac1
Volume name: acfs_gg
Canonical volume device: /dev/asm/acfs_gg-151
Accelerator volume devices:
Mountpoint path: /mnt/acfs_gg
Mount point owner: oracle
Mount point group: oinstall
Mount permissions: owner:oracle:rwx,pgrp:oinstall:r-x,other::r-x
Mount users: grid
Type: ACFS
Mount options:
Description:
ACFS file system is enabled
ACFS file system is individually enabled on nodes:
ACFS file system is individually disabled on nodes:
```

Step 3b.5 - Start and Check the Status of the ACFS Resource

As the grid OS user, use the following command to start and check the file system:

```
[grid@exadb-node1 ~]$ srvctl start filesystem -volume ACFS_GG -diskgroup DATAC1 -node `hostname`
[grid@exadb-node1 ~]$ srvctl status filesystem -volume ACFS_GG -diskgroup DATAC1
ACFS file system /mnt/acfs_gg is mounted on nodes exadb-node1
```

The CRS resource created is named using the format `ora.diskgroup_name.volume_name.acfs`. Using the above file system example, the CRS resource is called `ora.datac1.acfs_gg.acfs`.

To see all ACFS file system CRS resources that currently exist, use the following command.

```
[grid@exadb-node1 ~]$ crsctl stat res -w "((TYPE = ora.acfs.type) OR (TYPE = ora.acfs_cluster.type))"

NAME=ora.datac1.acfs_gg.acfs
TYPE=ora.acfs.type
TARGET=ONLINE                   , OFFLINE
STATE=ONLINE on exadb-node1, OFFLINE

NAME=ora.datac1.acfsvol01.acfs
TYPE=ora.acfs.type
TARGET=ONLINE                   , ONLINE
STATE=ONLINE on exadb-node1, ONLINE on exadb-node2
```

Step 3b.6- Create GoldenGate ACFS Directory

As the grid OS user, create the directory for storing the Oracle GoldenGate deployments.

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ mkdir -p /mnt/acfs_gg/deployments
```

Refer to the *Oracle Automatic Storage Management Cluster File System Administrator's Guide* for more information about ACFS.

NOTE: Leave the shared file system mounted. It is required for creating the Oracle GoldenGate deployment in a later step.

Step 4 - Install Oracle GoldenGate

Install the Oracle GoldenGate software **locally** on all nodes in the Oracle Exadata Database Service configuration that will be part of the GoldenGate configuration. Make sure the installation directory is **identical** on all nodes.

Here are the steps performed in this section:

- Step 4.1 - Unzip the Software and Create the Response File for the Installation
- Step 4.2 - Install Oracle GoldenGate

Step 4.1 - Unzip the Software and Create the Response File for the Installation

As the `oracle` OS user on the first database node, unzip the software:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ unzip
/u02/app_acfs/goldengate/213000_fbo_ggs_Linux_x64_Oracle_services_shiphome.zip -d /u02/app_acfs/goldengate
```

The software includes an example response file for Oracle Database 21c and lower supported versions. Copy the response file to a shared filesystem, so the same file can be used to install Oracle GoldenGate on all database nodes, and edit the following parameters:

- INSTALL_OPTION=ora21c
- SOFTWARE_LOCATION=/u02/app/oracle/goldengate/gg21c (recommended location)

As the `oracle` OS user on the first database node, copy and edit the response file for the installation.

```
[oracle@exadb-node1 ~]$ cp
/u02/app_acfs/goldengate/fbo_ggs_Linux_x64_Oracle_services_shiphome/Disk1/response/oggcore.rsp
/u02/app_acfs/goldengate
[oracle@exadb-node1 ~]$ vi /u02/app_acfs/goldengate/oggcore.rsp
# Before
INSTALL_OPTION=
SOFTWARE_LOCATION=
# After
INSTALL_OPTION=ora21c
SOFTWARE_LOCATION=/u02/app/oracle/goldengate/gg21c
```

Step 4.2 - Install Oracle GoldenGate

As the `oracle` OS user on all database nodes, install Oracle GoldenGate:

```
[oracle@exadb-node1 ~]$ cd /u02/app_acfs/goldengate/fbo_ggs_Linux_x64_Oracle_services_shiphome/Disk1/
[oracle@exadb-node1 ~]$ ./runInstaller -silent -nowait -responseFile /u02/app_acfs/goldengate/oggcore.rsp

Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 32755 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 16383 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2022-07-08_02-54-51PM. Please wait ...
You can find the log of this install session at:
```

```
 /u01/app/oraInventory/logs/installActions2022-07-08_02-54-51PM.log
Successfully Setup Software.
The installation of Oracle GoldenGate Services was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2022-07-08_02-54-51PM.log' for more details.

[oracle@exadb-node1 ~]$ cat /u01/app/oraInventory/logs/silentInstall2022-07-08_02-54-51PM.log
The installation of Oracle GoldenGate Services was successful.

[oracle@exadb-node1 ~]$ ssh exadb-node2
[oracle@exadb-node2 ~]$ cd /u02/app_acfs/goldengate/fbo_ggs_Linux_x64_Oracle_services_shiphome/Disk1
[oracle@exadb-node2 ~]$ ./runInstaller -silent -nowait -responseFile /u02/app_acfs/goldengate/oggcore.rsp

Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 32755 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 16383 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2022-07-08_03-54-51PM. Please wait ...
You can find the log of this install session at:
 /u01/app/oraInventory/logs/installActions2022-07-08_03-54-51PM.log
Successfully Setup Software.
The installation of Oracle GoldenGate Services was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2022-07-08_03-54-51PM.log' for more details.

[oracle@exadb-node1 ~]$ cat /u01/app/oraInventory/logs/silentInstall2022-07-08_03-54-51PM.log
The installation of Oracle GoldenGate Services was successful.
```

Step 5 - Create the Oracle GoldenGate Deployment

Once the Oracle GoldenGate software has been installed, the next step is to create a response file to create the GoldenGate deployment using the Oracle GoldenGate Configuration Assistant.

Here are the steps performed in this section:

- Step 5.1 - Create a Response File
- Step 5.2 - Create the GoldenGate Deployment
- Step 5.3 - (only if using DBFS) Move the GoldenGate Deployment Temp Directory

Step 5.1 - Create a Response File

For a silent configuration, please copy the following example file and paste it into any location the oracle user can access. Edit the following values appropriately:

- CONFIGURATION_OPTION
- DEPLOYMENT_NAME
- ADMINISTRATOR_USER
- SERVICEMANAGER_DEPLOYMENT_HOME
- OGG_SOFTWARE_HOME
- OGG_DEPLOYMENT_HOME
- ENV_TNS_ADMIN
- OGG_SCHEMA

Example Response File (oggca.rsp):

As the `oracle` OS user, create and edit the response file oggca.rsp to create the Oracle GoldenGate deployment:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ vi /u02/app_acfs/goldengate/oggca.rsp

oracle.install.responseFileVersion=/oracle/install/rspfmt_oggca_response_schema_v21_1_0
CONFIGURATION_OPTION=ADD
DEPLOYMENT_NAME=<ggNN>
ADMINISTRATOR_USER=oggadmin
ADMINISTRATOR_PASSWORD=<password_for_oggadmin>
SERVICEMANAGER_DEPLOYMENT_HOME=<ACFS or DBFS mount point>/deployments/<ggsmNN>
HOST_SERVICEMANAGER=localhost
PORT_SERVICEMANAGER=9100
SECURITY_ENABLED=false
STRONG_PWD_POLICY_ENABLED=true
CREATE_NEW_SERVICEMANAGER=true
REGISTER_SERVICEMANAGER_AS_A_SERVICE=false
INTEGRATE_SERVICEMANAGER_WITH_XAG=true
EXISTING_SERVICEMANAGER_IS_XAG_ENABLED=false
OGG_SOFTWARE_HOME=/u02/app/oracle/goldengate/gg21c
OGG_DEPLOYMENT_HOME=<ACFS or DBFS mount point>/deployments/<ggNN>
ENV_LD_LIBRARY_PATH=${OGG_HOME}/lib/instantclient:${OGG_HOME}/lib
ENV_TNS_ADMIN=/u02/app/oracle/goldengate/network/admin
FIPS_ENABLED=false
SHARDING_ENABLED=false
ADMINISTRATION_SERVER_ENABLED=true
PORT_ADMINSRVR=9101
DISTRIBUTION_SERVER_ENABLED=true
PORT_DISTSRVR=9102
NON_SECURE_DISTSRVR_CONNECTS_TO_SECURE_RCVRSRVR=false
RECEIVER_SERVER_ENABLED=true
PORT_RCVRSRVR=9103
METRICS_SERVER_ENABLED=true
METRICS_SERVER_IS_CRITICAL=false
PORT_PMSRVR=9104
UDP_PORT_PMSRVR=9105
PMSRVR_DATASTORE_TYPE=BDB
PMSRVR_DATASTORE_HOME=/u02/app/oracle/goldengate/datastores/<instance_name>
OGG_SCHEMA=<goldengate_database_schema>
```

Step 5.2 - Create the GoldenGate Deployment

As the `oracle` OS user on the first database node, execute oggca.sh to create the GoldenGate deployment:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ export OGG_HOME=/u02/app/oracle/goldengate/gg21c
[oracle@exadb-node1 ~]$ $OGG_HOME/bin/oggca.sh -silent -responseFile /u02/app_acfs/goldengate/oggca.rsp

Successfully Setup Software.
```

Step 5.3 - (only if using DBFS) Move the GoldenGate Deployment Temp Directory

After the deployment has been created, if you use DBFS for the shared file system, perform the following commands to move the GoldenGate deployment temp directory from DBFS to local storage.

As the `oracle` OS user on the first database node, move the GoldenGate deployment temporary directory to the local storage:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ dcli -l oracle -g ~/dbs_group mkdir -p
/u02/app/oracle/goldengate/deployments/<instance_name>
[oracle@exadb-node1 ~]$ mv /mnt/dbfs/goldengate/deployments/<instance_name>/var/temp
/u02/app/oracle/goldengate/datastores/<instance_name>
[oracle@exadb-node1 ~]$ ln -s /u02/app/oracle/goldengate/deployments/<instance_name>/temp
/mnt/dbfs/goldengate/deployments/<instance_name>/var/temp
```

As the `oracle` OS user on the rest of the database nodes, create a directory on the local storage:

```
[oracle@exadb-node2 ~]$ mkdir /u02/app/oracle/goldengate/deployments/<instance_name>
```

Step 6 - Network Configuration

In this section, you will find two different methods. The first method described in Step 6a applies to ExaDB-D only, and the second method described in Step 6b applies to ExaDB-C@C only.

Here are the steps performed in this section:

- Step 6a - (ExaDB-D only) Configure Oracle Cloud Infrastructure Networking
- Step 6b - (ExaDB-C@C only) Prepare for Application Virtual IP Address Creation

Step 6a - (ExaDB-D only) Configure Oracle Cloud Infrastructure Networking

You must configure virtual cloud network (VCN) components such as private DNS zones, VIP, bastion, security lists, and firewalls for Oracle GoldenGate to function correctly. To learn more about VCNs and security lists, including instructions for creating them, see the Oracle Cloud Infrastructure Networking documentation.

Here are the steps performed in this section:

- Step 6a.1 - Connect to GoldenGate Microservices Web Interface Using a Private IP
- Step 6a.2 - Create an Application Virtual IP Address (VIP)
- Step 6a.3 - Add Ingress Rule
- Step 6a.4 - Open Port 443 in the Firewall
- Step 6a.5 - Connecting your Source and Target VIP
- Step 6a.5 - Configuring Network Connectivity Between GoldenGate Source and Target
- Step 6a.6 - Configure Private DNS Zones Views and Resolvers

Step 6a.1 - Connect to GoldenGate Microservices Web Interface Using a Private IP

GoldenGate Microservices web interface is only accessible using a private endpoint from within the OCI network or through a bastion host that secures access to OCI resources.

If OCI Bastion is unavailable in your region, you can use your bastion on OCI Compute. This **quickstart** includes both options, so you can choose the one that works best for you. You will need one bastion for each region where Oracle GoldenGate Microservices is running.

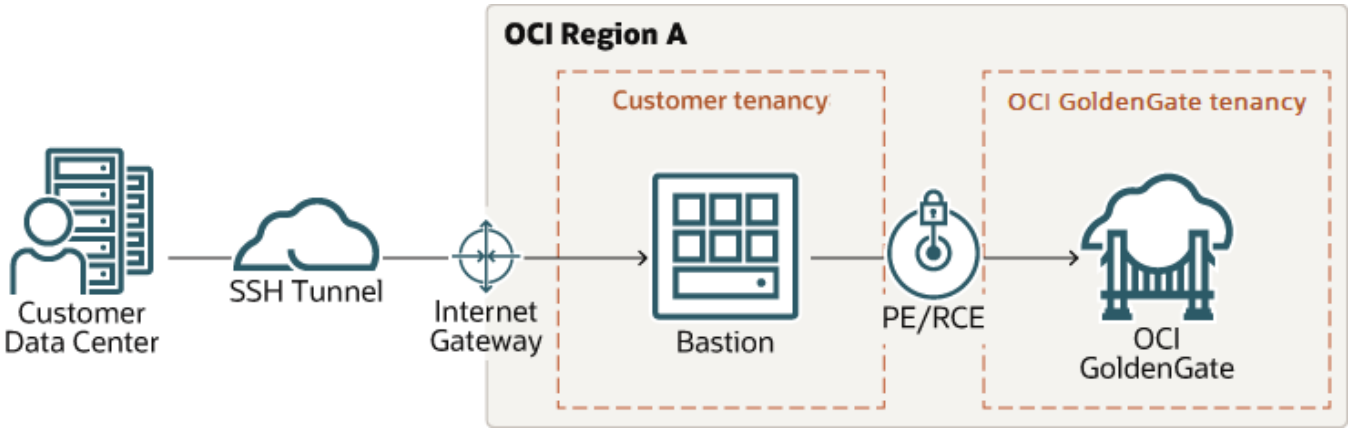

Figure 2: This illustration shows how OCI Bastion connects to the GoldenGate Microservices web interface.

***NOTE**: After creating a bastion or using a compute instance as a bastion, you need to create an SSH port forwarding session to use https://localhost:<local_port> to connect to Oracle GoldenGate Microservices.*

Step 6a.2 - Create an Application Virtual IP Address (VIP)

A dedicated application VIP is required to allow access to the GoldenGate Microservices using the same hostname, regardless of which RAC node is hosting the services. An application VIP will also ensure the Oracle GoldenGate Distribution Server can communicate with the Distribution Receiver running the current RAC node. The VIP is a cluster resource that Oracle Clusterware manages. The VIP is assigned to a database node and is automatically migrated to another node in the event of a node failure.

Using the Console, assign the VIP to the Oracle Exadata Database Service:

- Open the navigation menu. Click **Oracle Database**, then click **Exadata on Oracle Public Cloud**.
- Choose your **Compartment**.
- Click Exadata VM Cluster under Oracle Exadata Database Service on Dedicated Infrastructure.
- Navigate to the **Exadata VM Cluster** you want to create the new VIP.
- Under Resources, click **Virtual IP Address**.
- Click Attach Virtual IP Address.
- In the **Attach Virtual IP Address** dialog, enter the following mandatory information:
  - Subnet: The Client Subnet
  - Virtual IP address hostname: Use the SCAN DNS Name and replace the scan word for OGG (Example: exadb-xxxx-ggN)
- Click Create.

When the Virtual IP Address creation is complete, the status changes from Provisioning to Available, and the assigned IP will be shown in the **Virtual IP Address**. Make a note of the **Fully qualified domain name**; this is the hostname required to connect the source with the target Oracle GoldenGate deployment.

Figure 3: Application Virtual IP Address

NOTE: Adding a new VIP is available in most tenancies; log a Service Request if you have any issues.

Step 6a.3 - Add an Ingress Rule

Using the Console, open ingress port 443 to connect the Oracle GoldenGate service using Nginx as a reverse proxy. For more information, see Working with Security Lists.

After you update the security list, it will have an entry with values similar to the following ones:

- Source Type: CIDR
- Source CIDR: 0.0.0.0/0
- IP Protocol: TCP
- Source Port Range: All
- Destination Port Range: 443
- Allows: TCP traffic for ports: 443 HTTPS
- Description: Oracle GoldenGate 443

Step 6a.4 - Open Port 443 in the Firewall

As the `opc` OS user, validate if the chains are currently figured to accept traffic:

```
[opc@exadb-node1 ~]$ sudo iptables --list |grep policy

Chain INPUT (policy ACCEPT)
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

If the policy is ACCEPT, you can skip this step and proceed with Step 7. Otherwise, contact your network administrator to update the firewall to open port 443 for ingress activity.

Step 6a.5 - Configuring Network Connectivity Between GoldenGate Source and Target

You can set up your VCN to access the internet if you like. You can also privately connect your VCN to public Oracle Cloud Infrastructure services such as Object Storage, your on-premises network, or another VCN.

To learn more about whether subnets are public or private, including instructions for creating the connection, see the section **Connectivity Choices** in the Oracle Cloud Infrastructure Networking documentation.

Step 6a.6 - Configure Private DNS Zones Views and Resolvers

If the source and target Oracle GoldenGate deployments are in different regions, you must create a private DNS view in the source region with a private zone. This is required for the source Oracle GoldenGate Distribution Path to reach the target Oracle GoldenGate deployment VIP hostname.

Private DNS provides the ability to:

- Create private DNS zones with their desired names and create records for their private resources
- Private DNS resolver for DNS resolution to and from other private networks
- Resolve queries for custom private zones and system-generated zones (oraclevcn.com)
- Support for DNS views and conditional forwarding for split-horizon environments

Follow the steps in Configure private DNS zones views and resolvers to create your private DNS view and zone.

As the `opc` OS user on the source system, use the command nslookup to resolve the **Fully qualified domain name** (Step 6.2) of the target Oracle GoldenGate deployment:

```
[opc@exadb-node1 ~]$ nslookup <target_vip_fully_qualified_domain_name>
Server:         <DNS_IP>
Address:        <DNS_IP>#53
```

```
Non-authoritative answer:
Name:   <target_vip_fully_qualified_domain_name>
Address: <target_vip_ip>
```

Step 6b - (ExaDB-C@C only) Prepare for Application Virtual IP Address Creation

A dedicated application VIP is required to allow access to the GoldenGate Microservices using the same hostname, regardless of which RAC node is hosting the services. An application VIP will also ensure the Oracle GoldenGate Distribution Server can communicate with the Distribution Receiver running the current RAC node. The VIP is a cluster resource that Oracle Clusterware manages. The VIP is assigned to a database node and is automatically migrated to another node in the event of a node failure.

Your system administrator must provide the IP address for the new Application VIP. This IP address must be in the same subnet of the system environment as determined above.

The VIP will be created in the next step when configuring the Oracle Grid Infrastructure Agent.

Step 7 - Configure Oracle Grid Infrastructure Agent (XAG)

The following procedure shows how to configure Oracle Clusterware to manage Oracle GoldenGate using the Oracle Grid Infrastructure Standalone Agent (XAG). Using XAG automates the mounting of the shared file system (DBFS or ACFS) as well as the stopping and starting of the Oracle GoldenGate deployment when relocating between Oracle RAC nodes.

Here are the steps performed in this section:

- Step 7.1 - Install the Oracle Grid Infrastructure Standalone Agent
- Step 7.2 - Configure Oracle Grid Infrastructure Agent
- Step 7.2 - Start the Oracle GoldenGate Deployment

Step 7.1 - Install the Oracle Grid Infrastructure Standalone Agent

It is recommended to install the XAG software as a standalone agent outside the Grid Infrastructure `ORACLE_HOME`. This way, you can use the latest XAG release available, and the software can be updated without impact to the Grid Infrastructure.

Install the XAG standalone agent outside of the Oracle Grid Infrastructure home directory. XAG must be installed in the same directory on all RAC database nodes in the system where GoldenGate is installed.

As the `grid` OS user on the first database node, unzip the software and execute sagsetup.sh:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ unzip /u02/app_acfs/goldengate/p31215432_190000_Generic.zip -d
/u02/app_acfs/goldengate
[grid@exadb-node1 ~]$ /u02/app_acfs/goldengate/xag/xagsetup.sh --install --directory /u01/app/grid/xag --
all_nodes

Installing Oracle Grid Infrastructure Agents on: exadb-node1
Installing Oracle Grid Infrastructure Agents on: exadb-node2
Updating XAG resources.
Successfully updated XAG resources.
```

Add the location of the newly installed XAG software to the PATH variable so that the location of `agctl` is known when the `grid` user logs on to the machine.

```
[grid@exadb-node1 ~]$ grep PATH ~/.bashrc
PATH=/u01/app/grid/xag/bin:/sbin:/bin:/usr/sbin:/usr/bin:/u01/app/19.0.0.0/grid/bin:/u01/app/19.0.0.0/grid
/OPatch; export PATH
```

NOTE: It is essential to ensure that the XAG bin directory is specified BEFORE the Grid Infrastructure bin directory to ensure the correct agctl binary is found. This should be set in the grid user environment to take effect when logging on, such as in the .bashrc file when the Bash shell is in use.

Step 7.2 - Configure Oracle Grid Infrastructure Agent

The following procedure shows how to configure Oracle Clusterware to manage Oracle GoldenGate using the Oracle Grid Infrastructure Standalone Agent (XAG). Using XAG automates the mounting of the shared file system (DBFS or ACFS) as well as the stopping and starting of the Oracle GoldenGate deployment when relocating between Oracle RAC nodes.

Oracle GoldenGate must be registered with XAG so that the deployment is started and stopped automatically when the database is started and the file system is mounted.

To register Oracle GoldenGate Microservices Architecture with XAG, use the following command format.

```
agctl add goldengate <instance_name>
--gg_home <GoldenGate_Home>
--service_manager
--config_home <GoldenGate_SvcMgr_Config>
--var_home <GoldenGate_SvcMgr_Var Dir>
--port <port number>
--oracle_home <$OGG_HOME/lib/instantclient>
--adminuser <OGG admin user>
--user <GG instance user>
--group <GG instance group>
--network <network_number>
--ip <ip_address>
--vip_name <vip_name>
--filesystems <CRS_resource_name>
--db_services <service_name>
--use_local_services
--attribute START_TIMEOUT=60
--nodes <node1, node2, ... ,nodeN>
```

Where:

- `--gg_home` specifies the location of the Oracle GoldenGate software.
- `--service_manager` indicates this is a GoldenGate Microservices instance.
- `--config_home` specifies the GoldenGate Service Manager deployment configuration home directory.
- `--var_home` specifies the GoldenGate Service Manager deployment variable home directory.
- `--port` specifies the deployment Service Manager port number.
- `--oracle_home` specifies the location of the Oracle database libraries that are included as part of Oracle GoldenGate 21c and later releases. Example: `$OGG_HOME/lib/instantclient`
- `--adminuser` specifies the Oracle GoldenGate Microservices administrator account name.
- `--user` specifies the name of the operating system user that owns the GoldenGate deployment.
- `--group` specifies the name of the operating system group that owns the GoldenGate deployment.
- `--network` specifies the network subnet for the VIP.
- `--ip` specifies the IP address for the VIP. If you have already created a VIP, specify it using the `--vip_name <vip_name>` parameter in place of `--network` and `--ip`.

- `--vip_name` specifies a CRS resource name for an application VIP previously created. This parameter replaces --network and –ip (optional).
- `--filesystems` specifies the DBFS or ACFS CRS file system resource that must be mounted before the deployment is started.
- `--db_services` specifies the ora.<database>.<service_name>.svc service name created in the previous step. If using Oracle Multitenant Database, specify the PDB database service for Replicat or the CDB database service for an Extract. If using Replicat and Extract, specify both service names, separated by a comma.
- `--use_local_services` specifies that the GoldenGate instance must be co-located on the same RAC node where the db_services service is running.
- `--attribute <name>=<value>` specifies attributes that can be applied. We recommend modifying the attribute START_TIMEOUT=60 to optimize the blackout after a database crash and restart.
- `--nodes` specifies which of the RAC nodes this GoldenGate instance can run on. If GoldenGate is configured to run on any of the RAC nodes in the cluster, this parameter should still be used to determine the preferred order of nodes to run Oracle GoldenGate.

Here are the steps performed in this section:

- Step 7.2a - GoldenGate Deployments on DBFS
- Step 7.2b - GoldenGate Deployments on ACFS

Step 7.2a - GoldenGate Deployments on DBFS

As the `grid` OS user on the first database node, execute the following command to identify the network number:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ srvctl config network
Network 1 exists
Subnet IPv4: 10.1.0.0/255.255.255.0/bondeth0, static
Subnet IPv6:
Ping Targets: 10.1.0.1
Network is enabled
Network is individually enabled on nodes:
Network is individually disabled on nodes:
```

As the `root` OS user on the first database node, register Oracle GoldenGate Microservices Architecture with XAG using the following command format:

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# /u01/app/grid/xag/bin/agctl add goldengate <instance_name>          \
--gg_home /u02/app/oracle/goldengate/gg21c                                                \
--service_manager                                                                         \
--config_home /mnt/dbfs/deployments/ggsm01/etc/conf                                       \
--var_home /mnt/dbfs/deployments/ggsm01/var                                               \
--port 9100                                                                               \
--oracle_home /u02/app/oracle/goldengate/gg21c/lib/instantclient                          \
--adminuser oggadmin                                                                      \
--user oracle                                                                             \
--group oinstall                                                                          \
--network 1 --ip <virtual_IP_address>                                                     \
--filesystems <dbfs_mount_name>                                                           \
--db_services ora.<db_service_name>.svc , ora.<pdb_service_name>.svc                      \
--use_local_services                                                                      \
--attribute START_TIMEOUT=60                                                              \
--nodes <exadb-node1>, <exadb-node2>
Enter password for 'oggadmin' : <oggadmin_password>
```

Step 7.2b - GoldenGate Deployments on ACFS

As the `grid` OS user on the first database node, execute the following command to identify the network number:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ srvctl config network
Network 1 exists
Subnet IPv4: 10.1.0.0/255.255.255.0/bondeth0, static
Subnet IPv6:
Ping Targets: 10.1.0.1
Network is enabled
Network is individually enabled on nodes:
Network is individually disabled on nodes:
```

As the `root` OS user on the first database node, register Oracle GoldenGate Microservices Architecture with XAG using the following command format:

```
[root@exadb-node1 ~]# /u01/app/grid/xag/bin/agctl add goldengate <instance_name>          \
--gg_home /u02/app/oracle/goldengate/gg21c                                                \
--service_manager                                                                         \
--config_home /mnt/acfs_gg/deployments/ggsm01/etc/conf                                    \
--var_home /mnt/acfs_gg/deployments/ggsm01/var                                            \
--port 9100                                                                               \
--oracle_home /u02/app/oracle/goldengate/gg21c/lib/instantclient                          \
--adminuser oggadmin                                                                      \
--user oracle                                                                             \
--group oinstall                                                                          \
--network 1 --ip <virtual_IP_address>                                                     \
--filesystems ora.<acfs_name>.acfs                                                        \
--db_services ora.<db_service_name>.svc                                                   \
--use_local_services                                                                      \
--attribute START_TIMEOUT=60                                                              \
--nodes <exadb-node1>,<exadb-node2>
```

Step 7.3 - Start the Oracle GoldenGate Deployment

Below is some example agctl commands used to manage the GoldenGate deployment with XAG.

As the `grid` OS user, execute the following command to start the Oracle GoldenGate deployment:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ agctl start goldengate <instance_name>
```

As the `grid` OS user, execute the following command to check the status of the Oracle GoldenGate:

```
[grid@exadb-node1 ~]$ agctl status goldengate
Goldengate instance <instance_name> is running on exadb-node1
```

As the `grid` OS user, execute the following command to view the configuration parameters for the Oracle GoldenGate resource:

```
[grid@exadb-node1 ~]$ agctl config goldengate <instance_name>
Instance name: <instance_name>
Application GoldenGate location is: /u02/app/oracle/goldengate/gg21c_MS
Goldengate MicroServices Architecture environment: yes
Goldengate Service Manager configuration directory: /mnt/acfs_gg/deployments/ggsm01/etc/conf
Goldengate Service Manager var directory: /mnt/acfs_gg/deployments/ggsm01/var
```

```
Service Manager Port: 9100
Goldengate Administration User: oggadmin
Autostart on DataGuard role transition to PRIMARY: no
Configured to run on Nodes: exadb-node1 exadb-node2
ORACLE_HOME location is: /u02/app/oracle/goldengate/gg21c/lib/instantclient
Database Services needed: ora.<db_unique_name>.<service_name>.svc [use_local_services]
File System resources needed: ora.datac1.acfs_gg.acfs
Network: 1, IP:NN.NN.NN.NN, User:oracle, Group:oinstall
```

Further information on the Oracle Grid Infrastructure Bundled Agent.

Step 8 - Configure NGINX Reverse Proxy

The GoldenGate reverse proxy feature allows a single point of contact for all the GoldenGate microservices associated with a GoldenGate deployment. Without a reverse proxy, the GoldenGate deployment microservices are contacted using a URL consisting of a hostname or IP address and separate port numbers, one for each of the services. For example, to contact the Service Manager, you could use http://gghub.example.com:9100, then the Administration Server is http://gghub.example.com:9101, the second Service Manager may be accessed using http://gghub.example.com:9110, and so on.

When running Oracle GoldenGate in a High Availability (HA) configuration on Oracle Exadata Database Service with the Grid Infrastructure agent (XAG), there is a limitation preventing more than one deployment from being managed by a GoldenGate Service Manager. Because of this limitation, creating a separate virtual IP address (VIP) for each Service Manager/deployment pair is recommended. This way, the microservices can be accessed directly using the VIP.

With a reverse proxy, port numbers are not required to connect to the microservices because they are replaced with the deployment name and the hostname's VIP. For example, to connect to the console via a web browser, use the URLs:

GoldenGate Services	URL
Service Manager	https://localhost:<localPort>
Administration Server	https://localhost:<localPort>/<instance_name>/adminsrvr
Distribution Server	https://localhost:<localPort>/<instance_name>/distsrvr
Performance Metric Server	https://localhost:<localPort>/<instance_name>/pmsrvr
Receiver Server	https://localhost:<localPort>/<instance_name>/recvsrvr

NOTE: To connect to Oracle GoldenGate in OCI, you must create a bastion and an SSH port forwarding session (see Step 6.1). After this, you can connect to the Oracle GoldenGate Services using https://locahost:<localPort>.

A reverse proxy is mandatory to ensure easy access to microservices and enhance security and manageability.

Follow the instructions to install and configure NGINX Reverse Proxy with an SSL connection and ensure all external communication is secure.

NOTE: When using CA Signed Certificates with NGINX, make sure the NGINX `ssl_certificate` parameter points to a certificate file that contains the certificates in the correct order of CA signed certificate, intermediate certificate, and root certificate.

Here are the steps performed in this section:

- Step 8.1 - Install NGINX
- Step 8.2 - Configure NGINX Reverse Proxy

- Step 8.3 - Securing GoldenGate Microservices to Restrict Non-secure Direct Access
- Step 8.4 - Create a Clusterware Resource to Manage NGINX

Step 8.1 - Install NGINX Reverse Proxy Server

As the `root` OS user, set up the yum repository by creating the file /etc/yum.repos.d/nginx.repo with the following contents:

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/rhel/7/\$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
EOF
```

As the `root` OS user, run the following commands to install, enable, and start NGINX:

```
[root@exadb-node1 ~]# yum install -y python-requests python-urllib3 nginx
[root@exadb-node1 ~]# systemctl enable nginx
```

As the `root` OS user, disable the NGINX repository after the software has been installed:

```
[root@exadb-node1 ~]# yum-config-manager --disable nginx-stable
```

Step 8.2 - Configure NGINX Reverse Proxy

A separate reverse proxy configuration is required for each Oracle GoldenGate Home.

When running multiple Service Managers, the following instructions will provide configuration using a separate VIP for each Service Manager. NGINX uses the VIP to determine which Service Manager an HTTPS connection request is routed to.

An SSL certificate is required for clients to authenticate the server they connect to through NGINX. Contact your systems administrator to follow your corporate standards to create or obtain the server certificate before proceeding. A separate certificate is required for each VIP and Service Manager pair.

***NOTE**: The common name in the CA-signed certificate must match the target hostname/VIP used by NGINX.*

Here are the steps performed in this section:

- Step 8.2.1 - Create the NGINX Configuration File
- Step 8.2.2 - Modify NGINX Configuration Files
- Step 8.2.3 - Install Server Certificates for NGINX
- Step 8.2.4 - Install the NGINX Configuration File
- Step 8.2.5 - Test the New NGINX Configuration
- Step 8.2.6 - Reload NGINX and the New Configuration
- Step 8.2.7 - Test GoldenGate Microservices Connectivity
- Step 8.2.8 - Distribute the GoldenGate NGINX Configuration Files

Step 8.2.1 - Create the NGINX Configuration File

You can configure Oracle GoldenGate Microservices Architecture to use a reverse proxy. Oracle GoldenGate MA includes a script called `ReverseProxySettings` that generates a configuration file for only the NGINX reverse proxy server.

The script requires the following parameters:

- The --user parameter should mirror the GoldenGate administrator account specified with the initial deployment creation.
- The GoldenGate administrator password will be prompted.
- The reverse proxy port number specified by the --port parameter should be the default HTTPS port number (443) unless you are running multiple GoldenGate Service Managers using the same --host. In this case, specify an HTTPS port number that does not conflict with previous Service Manager reverse proxy configurations. For example, if running two Service Managers using the same hostname/VIP, the first reverse proxy configuration is created with '--port 443 --host hostvip01', and the second is created with '--port 444 --host hostvip01'. If using separate hostnames/VIPs, the two Service Manager reverse proxy configurations would be created with '--port 443 --host hostvip01' and '--port 443 --host hostvip02'.
- Lastly, the HTTP port number (9100) should match the Service Manager port number specified during the deployment creation.

Repeat this step for each additional GoldenGate Service Manager.

As the `oracle` OS user, use the following command to create the Oracle GoldenGate NGINX configuration file:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ export OGG_HOME=/u02/app/oracle/goldengate/gg21c
[oracle@exadb-node1 ~]$ export PATH=$PATH:$OGG_HOME/bin

[oracle@exadb-node1 ~]$ cd /u02/app_acfs/goldengate
[oracle@exadb-node1 ~]$ $OGG_HOME/lib/utl/reverseproxy/ReverseProxySettings --user oggadmin --port 443 --
output ogg_<instance_name>.conf http://localhost:9100 --host <VIP hostname/IP>

Password: <oggadmin_password>
```

Step 8.2.2 - Modify NGINX Configuration Files

When multiple GoldenGate Service Managers are configured to use their IP/VIPs with the same HTTPS 443 port, some small changes are required to the NGINX reverse proxy configuration files generated in the previous step. With all Service Managers sharing the same port number, they are independently accessed using their VIP/IP specified by the --host parameter.

As the `oracle` OS user, determine the deployment name managed by this Service Manager. If not already known, the deployment name is listed in the reverse proxy configuration file:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ cd /u02/app_acfs/goldengate
[oracle@exadb-node1 ~]$ grep "Upstream Servers" ogg_<instance_name>.conf
## Upstream Servers for Deployment '<instance_name>'
```

In this example, the deployment is called SOURCE.

As the `oracle` OS user, change all occurrences of "_ServiceManager" by prepending the deployment name before the underscore.

```
$ sed -i 's/_ServiceManager/<instance_name>_ServiceManager/' ogg_<instance_name>.conf
```

Step 8.2.3 - Install Server Certificates for NGINX

As the `root` OS user, copy the server certificates and key files in the `/etc/nginx/ssl` directory, owned by root with file permissions 400 (-r--------):

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# mkdir /etc/nginx/ssl
[root@exadb-node1 ~]# chmod 400 /etc/nginx/ssl
```

As the root OS user, set the correct filenames for the certificate and key files for each reverse proxy configuration file generated in Step 8.2.1:

```
[oracle@exadb-node1 ~]$ vi /u02/app_acfs/goldengate/ogg_<instance_name>.conf

# Before
    ssl_certificate         /etc/nginx/ogg.pem;
    ssl_certificate_key     /etc/nginx/ogg.pem;

# After
    ssl_certificate         /etc/nginx/ssl/server.chained.crt;
    ssl_certificate_key     /etc/nginx/ssl/server.key;
```

When using CA-signed certificates, the certificate named with the ssl_certificate NGINX parameter must include the 1) CA signed, 2) intermediate, and 3) root certificates in a single file. The order is significant; otherwise, NGINX fails to start and displays the error message:

```
(SSL: error:0B080074:x509 certificate routines:  X509_check_private_key:key values mismatch)
```

The root and intermediate certificates can be downloaded from the CA-signed certificate provider.

The SSL certificate single file can be generated using the following example command:

```
[root@exadb-node1 ~]# cat CA_signed_cert.crt intermediate.crt root.crt > server.chained.crt
```

The ssl_certificate_key file is generated when creating the Certificate Signing Request (CSR), which is required when requesting a CA-signed certificate.

Step 8.2.4 - Install the NGINX Configuration File

As the root OS user, copy the deployment configuration file (or files if multiple files were created in Step 8.2.1) to /etc/nginx/conf.d directory:

```
[root@exadb-node1 ~]# mv /u02/app_acfs/goldengate/ogg_<instance_name>.conf /etc/nginx/conf.d
```

Step 8.2.5 - Test the New NGINX Configuration

As the root OS user, validate the NGINX configuration file. If there are errors in the file, they will be reported with the following command:

```
[root@exadb-node1 ~]# nginx -t

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

Step 8.2.6 - Reload NGINX and the New Configuration

As the root OS user, restart NGINX to load the new configuration:

```
[root@exadb-node1 ~]# systemctl restart nginx
```

Step 8.2.7 - Test GoldenGate Microservices Connectivity

As the root OS user, create a curl configuration file (access.cfg) that contains the deployment username and password:

```
[root@exadb-node1 ~]# vi access.cfg
user = "oggadmin:<password>"
```

As the root OS user, query the health of the deployments using the following command:

```
[root@exadb-node1 ~]# curl -svf -K access.cfg https://<VIP hostname/IP>:<port#>/services/v2/config/health
-XGET && echo -e "\n*** Success"
```

Sample output:

```
{"$schema":"api:standardResponse","links":[{"rel":"canonical","href":"https://gg-prmy-
vip1/services/v2/config/health","mediaType":"application/json"},
{"rel":"self","href":"https://gg-prmy-
vip1/services/v2/config/health","mediaType":"application/json"},{"rel":"describedby","href":"https://gg-
prmy-vip1/services/ServiceManager/v2/metadata-catalog/health",
"mediaType":"application/schema+json"}],"messages":[],"response":{"$schema":"ogg:health","deploymentName":
"ServiceManager","serviceName":"ServiceManager","started":"2021-12-09T23:33:03.425Z","healthy":true,
"criticalResources":[{"deploymentName":"SOURCE","name":"adminsrvr","type":"service","status":"running","he
althy":true},{"deploymentName":"SOURCE","name":"distsrvr","type":"service","status":"running","healthy":tr
ue},
{"deploymentName":"SOURCE","name":"recvsrvr","type":"service","status":"running","healthy":true}]}}
*** Success ***
```

As the root OS user, remove the curl configuration file (access.cfg) that contains the deployment username and password:

```
[root@exadb-node1 ~]# rm access.cfg
rm: remove regular file ‘access.cfg’? y
```

Step 8.2.8 - Distribute the GoldenGate NGINX Configuration Files

Once all the reverse proxy configuration files have been created for the GoldenGate Service Managers, they must be copied to all the database nodes.

As the `opc` OS user, distribute the NGINX configuration files to all database nodes:

```
[opc@exadb-node1 ~]$ sudo tar fczP nginx_conf.tar /etc/nginx/conf.d/ /etc/nginx/ssl/
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc -d /tmp -f nginx_conf.tar
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo tar fxzP /tmp/nginx_conf.tar
```

As the opc OS user, test the new NGINX configuration on all nodes the new configuration files were copied to:

```
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo nginx -t

exadb-node1: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
exadb-node1: nginx: configuration file /etc/nginx/nginx.conf test is successful
exadb-node2: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
exadb-node2: nginx: configuration file /etc/nginx/nginx.conf test is successful
```

As the `opc` OS user, restart NGINX to load the new configuration on all nodes:

```
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo systemctl restart nginx
```

Step 8.3 - Securing GoldenGate Microservices to Restrict Non-secure Direct Access

After configuring the NGINX reverse proxy with an unsecured Oracle GoldenGate Microservices deployment, the microservices can continue accessing HTTP (non-secure) using the configured microservices port numbers. For example, the following non-secure URL could be used to access the Administration Server: http://<vip-name>:9101.

Oracle GoldenGate Microservices' default behavior for each server (Service Manager, adminserver, pmsrvr. distsrvr, and recsrvr) is to listen using a configured port number on all network interfaces. This is undesirable for more secure installations, where direct access using HTTP to the Microservices needs to be disabled and only permitted using NGINX HTTPS.

Use the following commands to alter the Service Manager and deployment services listener address to use only the localhost address. Access to the Oracle GoldenGate Microservices will only be permitted from the localhost, and any access outside of the localhost will only succeed using the NGINX HTTPS port.

Step 8.3.1 - Stop the Service Manager

As the `grid` OS user, stop the service manager:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ agctl stop goldengate <instance_name>
[grid@exadb-node1 ~]$ agctl status goldengate
Goldengate  instance '<instance_name>' is not running
```

Step 8.3.2 - Modify the Service Manager Listener Address

As the `oracle` OS user, modify the listener address with the following commands. Use the correct port number for the Service Manager being altered. The server will fail to start, ignore the error, and proceed with the next step:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ export OGG_HOME=/u02/app/oracle/goldengate/gg21c
[oracle@exadb-node1 ~]$ export OGG_VAR_HOME=<acfs or dbfs mount point>/deployments/ggsm01/var
[oracle@exadb-node1 ~]$ export OGG_ETC_HOME=<acfs or dbfs mount point>/deployments/ggsm01/etc

[oracle@exadb-node1 ~]$ $OGG_HOME/bin/ServiceManager --prop=/config/network/serviceListeningPort --
value='{"port":9100,"address":"127.0.0.1"}' --type=array --persist --exit

[oracle@exadb-node1 ~]$
```

Step 8.3.3 - Restart the Service Manager and Deployment

As the `grid` OS user, restart the Service Manager and deployment:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ agctl start goldengate <instance_name>
[grid@exadb-node1 ~]$ agctl status goldengate
Goldengate  instance '<instance_name>' is running on exadb-node1
```

Step 8.3.4 - Modify the GoldenGate Microservices listener address

As the `oracle` OS user, modify all the GoldenGate microservices (adminsrvr, pmsrvr, distsrvr, recvsrvr) listening address to localhost for the deployments managed by the Service Manager using the following command:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ cd /u02/app_acfs/goldengate
[oracle@exadb-node1 ~]$ chmod u+x secureServices.py
[oracle@exadb-node1 ~]$ ./secureServices.py http://localhost:9100 --user oggadmin

Password for 'oggadmin': <oggadmin_password>
*** Securing deployment - ogg_deployment
Current value of "/network/serviceListeningPort" for "<instance_name>/adminsrvr" is
{
    "address": "127.0.0.1",
    "port": 9101
}
Current value of "/network/serviceListeningPort" for "<instance_name>/distsrvr" is
```

```
{
    "address": "127.0.0.1",
    "port": 9102
}
Current value of "/network/serviceListeningPort" for "<instance_name>/pmsrvr" is
{
    "address": "127.0.0.1",
    "port": 9104
}
Current value of "/network/serviceListeningPort" for "<instance_name>/recvsrvr" is
{
    "address": "127.0.0.1",
    "port": 9103
}
```

NOTE: To modify a single deployment (adminsrvr, pmsrvr, distsrvr, recvsrvr), add the flag "--deployment <instance_name>"

Step 8.3.5 - Remove NGINX default.conf Configuration File

As the `opc` OS user, remove the default configuration file (default.conf) created in /etc/nginx/conf.d:

```
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo rm -f /etc/nginx/conf.d/default.conf
[opc@exadb-node1 ~]$ /usr/local/bin/dcli -g ~/dbs_group -l opc sudo nginx -s reload
```

Step 8.4 - Create a Clusterware Resource to Manage NGINX

Oracle Clusterware needs to have control over starting the NGINX reverse proxy so that it can be started automatically before the GoldenGate deployments are started.

As the `grid` OS user, use the following command to get the network CRS resource name required to create the NGINX resource with a dependency on the underlying network CRS resource:

```
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ crsctl stat res -w "TYPE == ora.network.type"|grep NAME
NAME=ora.net1.network
```

As the `root` OS user, use the following example command to create a Clusterware resource to manage NGINX. Replace the `HOSTING_MEMBERS` and `CARDINALITY` to match your environment:

```
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# $(grep ^crs_home /etc/oracle/olr.loc | cut -d= -f2)/bin/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(ora.net1.network) pullup(ora.net1.network)',
STOP_DEPENDENCIES='hard(intermediate:ora.net1.network)', RESTART_ATTEMPTS=0, HOSTING_MEMBERS='<exadb-
node1, exadb-node2>', CARDINALITY=2"
```

The NGINX resource created in this example will run on the named database nodes simultaneously, specified by HOSTING_MEMBERS. This is recommended when multiple GoldenGate Service Manager deployments are configured and can independently move between database nodes.

Once the NGINX Clusterware resource is created, the GoldenGate XAG resources need to be altered so that NGINX must be started before the GoldenGate deployments are started.

As the `root` OS user, modify the XAG resources using the following example commands.

```
# Determine the current --filesystems parameter:
[opc@exadb-node1 ~]$ sudo su - grid
[grid@exadb-node1 ~]$ agctl config goldengate <instance_name> |grep "File System"
File System resources needed: <file_system_resource_name>

# Modify the --filesystems parameter:
[opc@exadb-node1 ~]$ sudo su -
[root@exadb-node1 ~]# /u01/app/grid/xag/bin/agctl modify goldengate <instance_name> --filesystems
<file_system_resource_name>,nginx
```

Repeat the above commands for each XAG GoldenGate registration relying on NGINX.

Step 9 - Create Oracle Net TNS Alias for Oracle GoldenGate Database Connections

To provide local database connections for the Oracle GoldenGate processes when switching between Oracle RAC nodes, create a TNS alias on **all** the RAC nodes where Oracle GoldenGate may be started. Create the TNS alias in the `tnsnames.ora` file in the TNS_ADMIN directory specified in the deployment creation.

Here are the steps performed in this section:

- Step 9.1 - Create the TNS Alias Definitions
- Step 9.2 - Create the Database Credentials

Step 9.1 - Create the TNS Alias Definitions

If the source database is a multitenant database, two TNS alias entries are required, one for the container database (CDB) and one for the pluggable database (PDB) that is being replicated. For a target Multitenant database, the TNS alias connects the PDB to where replicated data is being applied. The pluggable database SERVICE_NAME should be set to the database service created in an earlier step (refer to Step 2.3: Create the Database Services).

As the `oracle` OS user, find the database domain name:

```
[opc@exadb-node1]$ sudo su - oracle
[oracle@exadb-node1]$ source DB_NAME.env
[oracle@exadb-node1]$ sqlplus / as sysdba
SQL> show parameter db_domain

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_domain                            string      <db_domain_name>
```

As the `oracle` OS user on the first database node, follow the steps to create the TNS alias definitions and distribute them to all database nodes:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ dcli -l oracle -g ~/dbs_group mkdir -p /u02/app/oracle/goldengate/network/admin
[oracle@exadb-node1 ~]$ vi /u02/app/oracle/goldengate/network/admin/tnsnames.ora

OGGSRV_CDB =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL=IPC)(KEY=LISTENER))
      (CONNECT_DATA =
        (SERVICE_NAME = <cdb_service_name>.<db_domain_name>)
```

```
        )
    )
OGGSRV_<PDB_NAME> =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL=IPC)(KEY=LISTENER))
      (CONNECT_DATA =
        (SERVICE_NAME = <pdb_service_name>.<db_domain_name>)
       )
    )
[oracle@exadb-node1 ~]$ /usr/local/bin/dcli -l oracle -g ~/dbs_group -f
/u02/app/oracle/goldengate/network/admin/*.ora -d /u02/app/oracle/goldengate/network/admin
```

NOTE: When the tnsnames.ora or sqlnet.ora located in the TNS_ADMIN directory for the Oracle GoldenGate deployment are modified; the deployment needs to be restarted to pick up the changes.

Step 9.2 - Create the Database Credentials

With the Oracle GoldenGate deployment created, use the Oracle GoldenGate Administration Service home page to create the database credentials using the above TNS alias names. See figure 4 below for an example of the database credential creation using the TNS alias.

As the oggadmin user, create the database credentials:

- Log in into the Administration Service: https://localhost:<localPort>/<instance_name>/adminsrvr
- Click in Configuration under Administration Service
- Click the plus button to **Add Credentials**
- Add the required information as follows:

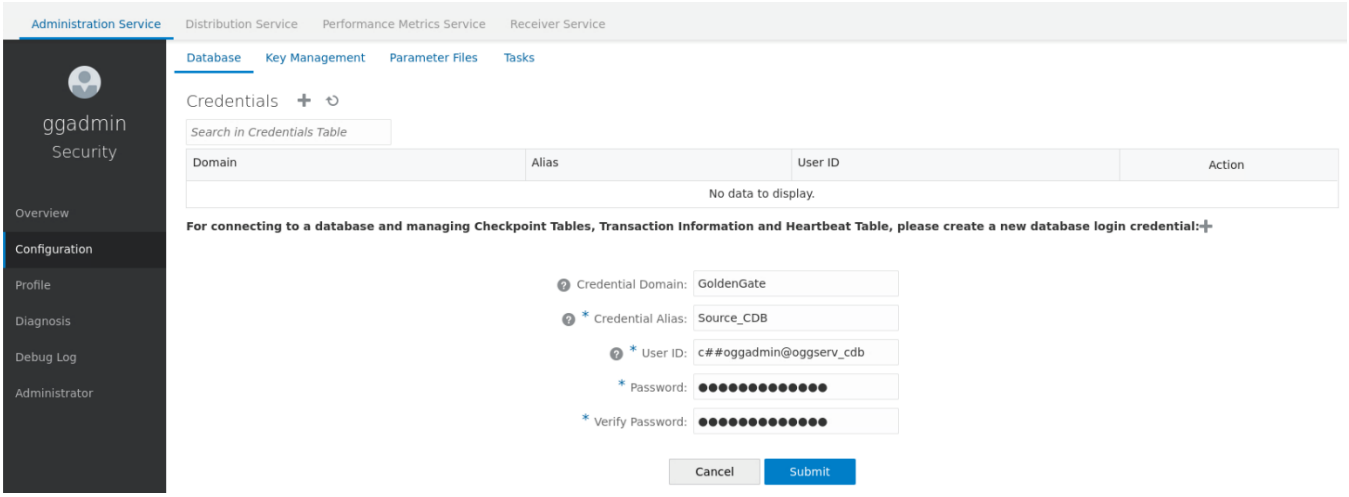

Figure 4: Creating a database credential

If the source database is a Multitenant Database, create database credentials for the CDB and PDB. If the target database is a Multitenant Database, create a single credential for the PDB.

The following picture shows an example of a source multitenant database:

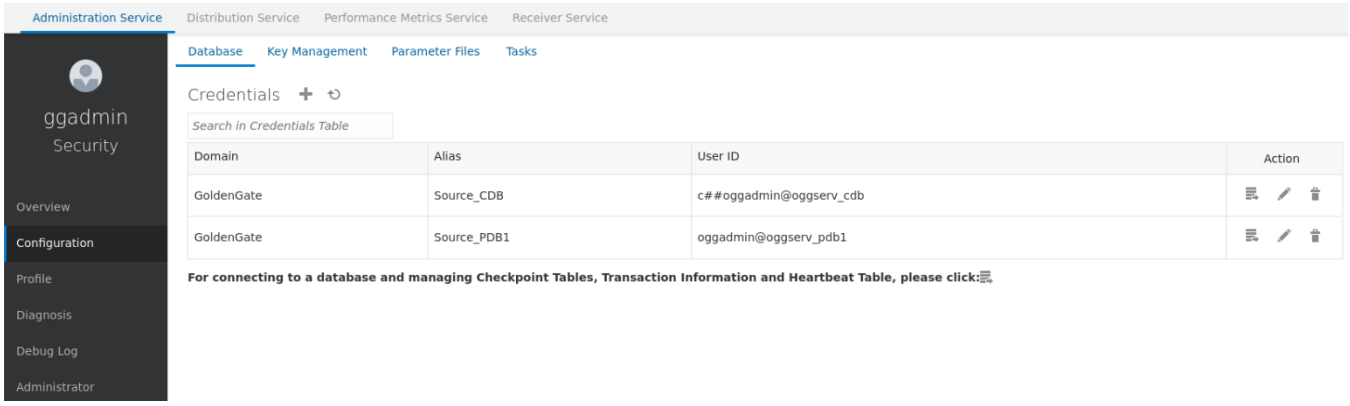

Figure 5: Database credential for source multitenant database

Step 10 - Create a New Profile

Create a new profile to automatically start the Extract and Replicat processes when the Oracle GoldenGate Administration Server is started. Then, restart if any Extract or Replicat processes are abandoned. With GoldenGate Microservices, auto start and restart is managed by Profiles.

Using the Oracle GoldenGate Administration Server GUI, create a new profile that can be assigned to each of the Oracle GoldenGate processes:

- Log in to the **Administration Service** on the Source and Target GoldenGate.
- Click on **Profile** under Administration Service.
- Click the plus (+) sign next to Profiles on the Managed Process Settings home page. The Add Profile page is displayed.
- Enter the details as follows:

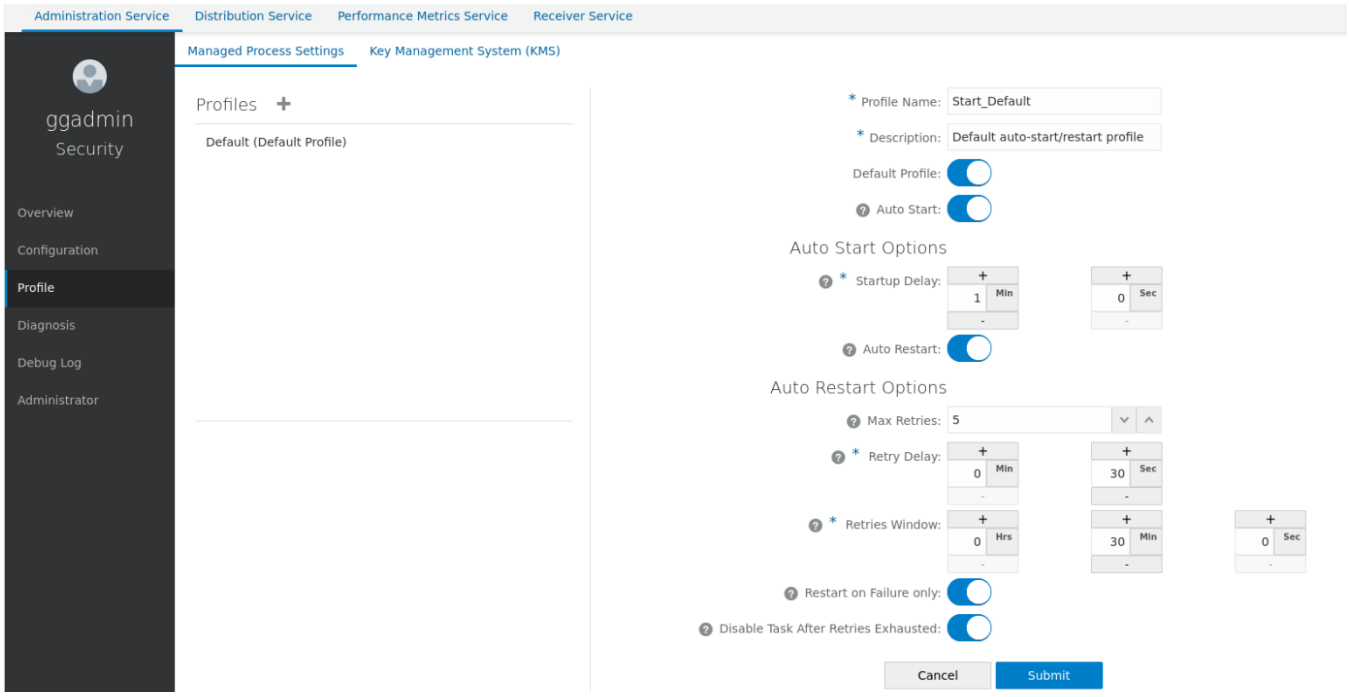

Figure 6: Creating an auto-start profile using the Oracle GoldenGate Administration Server GUI

- Click **Submit**.

Step 11 - Configure Oracle GoldenGate Processes

When creating Extract, Distribution Paths, and Replicat processes with Oracle GoldenGate Microservices Architecture, all files that need to be shared between Oracle RAC nodes are already shared with the deployment files stored on a shared file system (DBFS or ACFS).

Listed below are essential configuration details recommended for running Oracle GoldenGate Microservices on RAC for Extract, Distribution Paths, and Replicat processes.

Here are the steps performed in this section:

- Step 11.1 - Extract Configuration
- Step 11.2 - (DBFS only) Place the Temporary Cache Files on the Shared Storage
- Step 11.3 - Distribution Path Configuration
- Step 11.4 - Replicat Configuration

Step 11.1 - Extract Configuration

When creating an Extract using the Oracle GoldenGate Administration Service GUI interface, leave the **Trail SubDirectory** parameter blank so that the trail files are automatically created in the deployment directories stored on the shared file system. The default location for trail files is the `/<deployment directory>/var/lib/data` directory.

NOTE: To capture from a multitenant database, you must use an Extract configured at the root level using a c## account. To apply data into a multitenant database, a separate Replicat is needed for each PDB because a Replicat connects at the PDB level and doesn't have access to objects outside of that PDB

Create the database credentials:

- Log in to the Oracle GoldenGate **Administration Server** in the Source Oracle GoldenGate
- Click in Overview under Administration Service
- Click the plus button to **Add Extract**
- Select Integrated Extract
- Add the required information as follows:

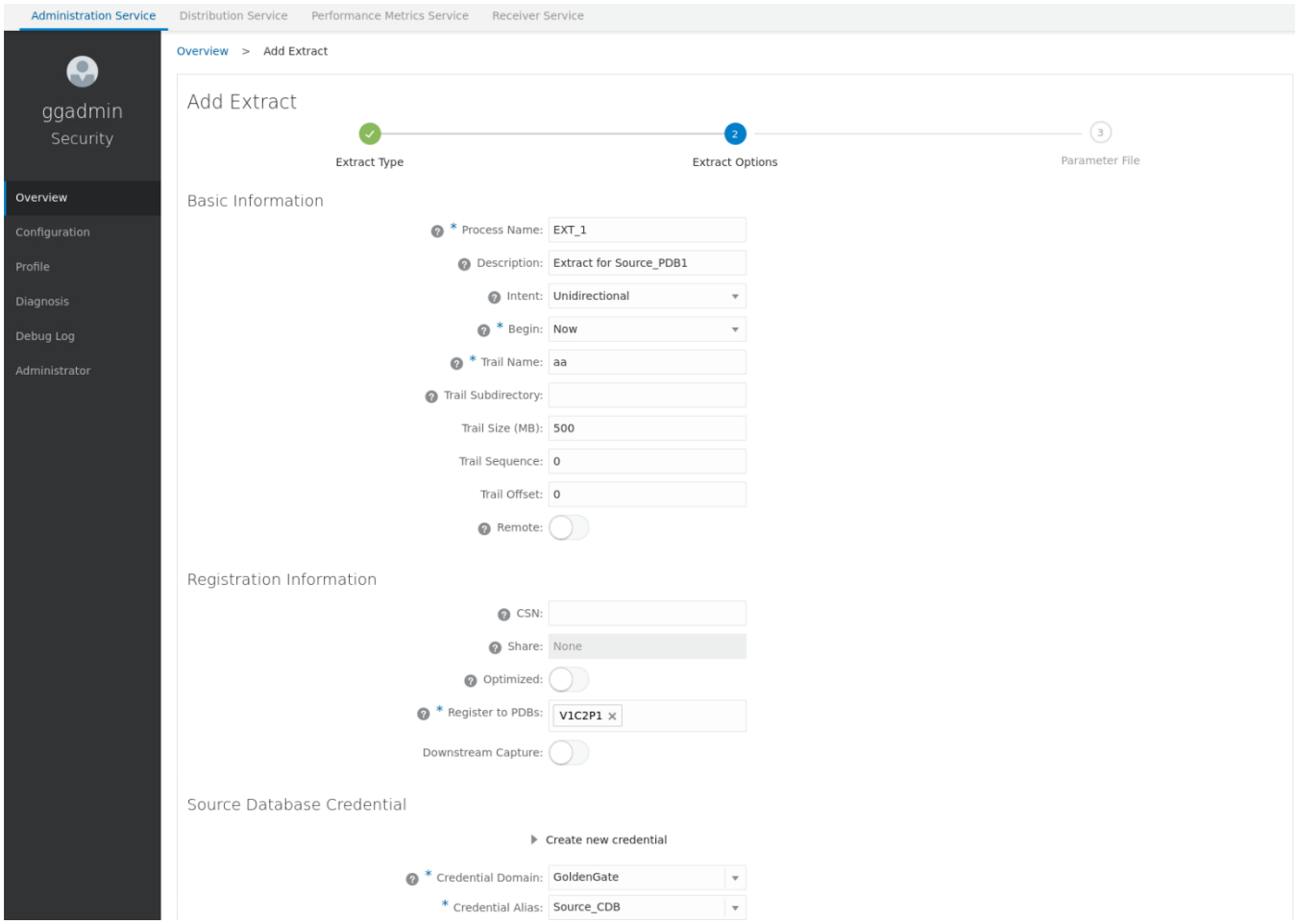

Figure 7: Adding an Extract process with the Administration Server.

- Click **Next**
- If using CDB Root Capture from PDB, add the SOURCATALOG parameter with the PDB Name:

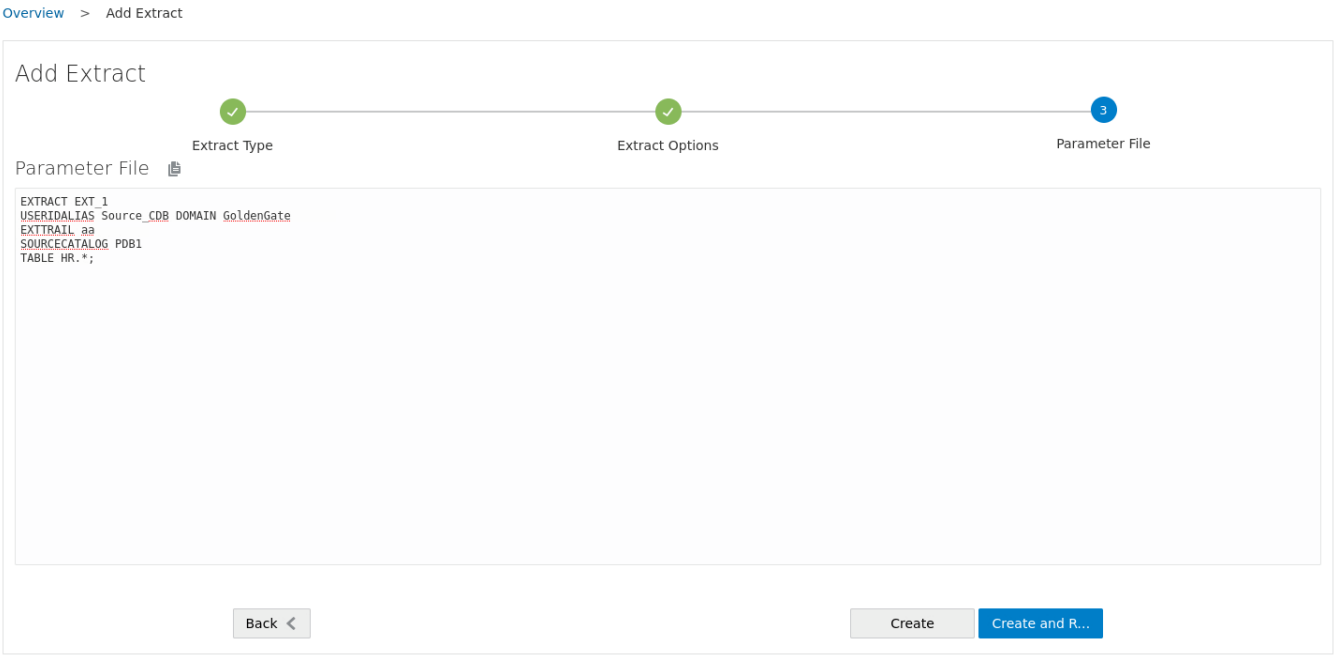

Figure 8: Adding an Extract with CDB root capture from PDB

- Click **Create**

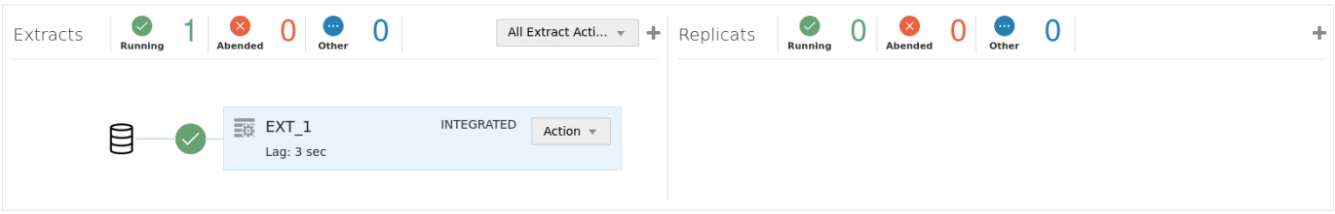

Figure 9: Oracle GoldenGate Extracts

Step 11.2 - (DBFS only) Place the Temporary Cache Files on the Shared Storage

If using DBFS for shared storage, and the deployment `var/temp` directory was moved to local storage as described in Step 6: Create the Oracle GoldenGate Deployment, it is recommended to use the Extract `CACHEMGR` parameter to place the temporary cache files on the shared storage.

As the oracle OS user, create a new directory under the DBFS deployment mount point.:

```
[opc@exadb-node1 ~]$ sudo su - oracle
[oracle@exadb-node1 ~]$ mkdir /mnt/dbfs/goldengate/deployments/<instance_name>/temp_cache
```

Set the Extract parameter to the new directory:

```
CACHEMGR CACHEDIRECTORY /mnt/dbfs/goldengate/deployments/<instance_name>/temp_cache
```

Here is an example of the parameters specified for an integrated Extract with the Oracle GoldenGate Administration Server GUI.

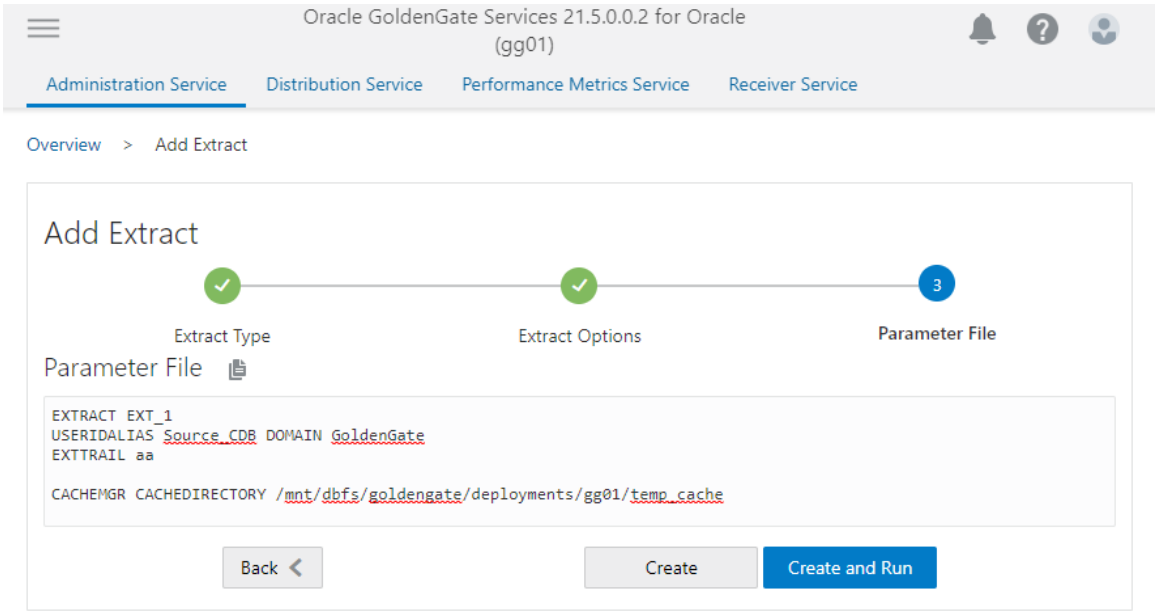

Figure 10: Extract parameters for defining the temporary cache files.

More instructions about creating an Extract process are available in the *Using Oracle GoldenGate Microservices Architecture* Guide.

Step 11.3 - Distribution Path Configuration

When using Oracle GoldenGate Distribution paths with the NGINX Reverse Proxy, additional steps must be carried out to ensure the path client and server certificates are configured.

More instructions about creating distribution paths are available in *Using Oracle GoldenGate Microservices Architecture*. A step-by-step example is in the following video, "Connect an on-premises Oracle GoldenGate to OCI GoldenGate using NGINX," to correctly configure the certificates.

Here are the steps performed in this section:

- Step 11.3.1 - Download the Target Server's Root Certificate, and then upload it to the source Oracle GoldenGate
- Step 11.3.2 - Create a user in the Target Deployment for the Source Oracle GoldenGate to use
- Step 11.3.3 - Create a Credential in the Source Oracle GoldenGate
- Step 11.3.4 - Create a Distribution Path on the Source Oracle GoldenGate to the Target Deployment
- Step 11.3.5 - Verify the Connection in the Target Deployment Console Receiver Service

Step 11.3.1 - Download the Target Server's Root Certificate, and then upload it to the source Oracle GoldenGate

Download the target deployment server's root certificate and add the CA certificate to the source deployment Service Manager.

- Log in to the **Administration Service** on the Target GoldenGate.
- Follow "Step 2 - Download the target server's root certificate" in the video "Connect an on-premises Oracle GoldenGate to OCI GoldenGate using NGINX."

Step 11.3.2 - Create a user in the Target Deployment for the Source Oracle GoldenGate to use

Create a user in the target deployment for the distribution path to connect to:

- Log in to the **Administration Service** on the Target GoldenGate.
- Click on Administrator under Administration Service.
- Click the plus (+) sign next to Users.
- Enter the details as follows:

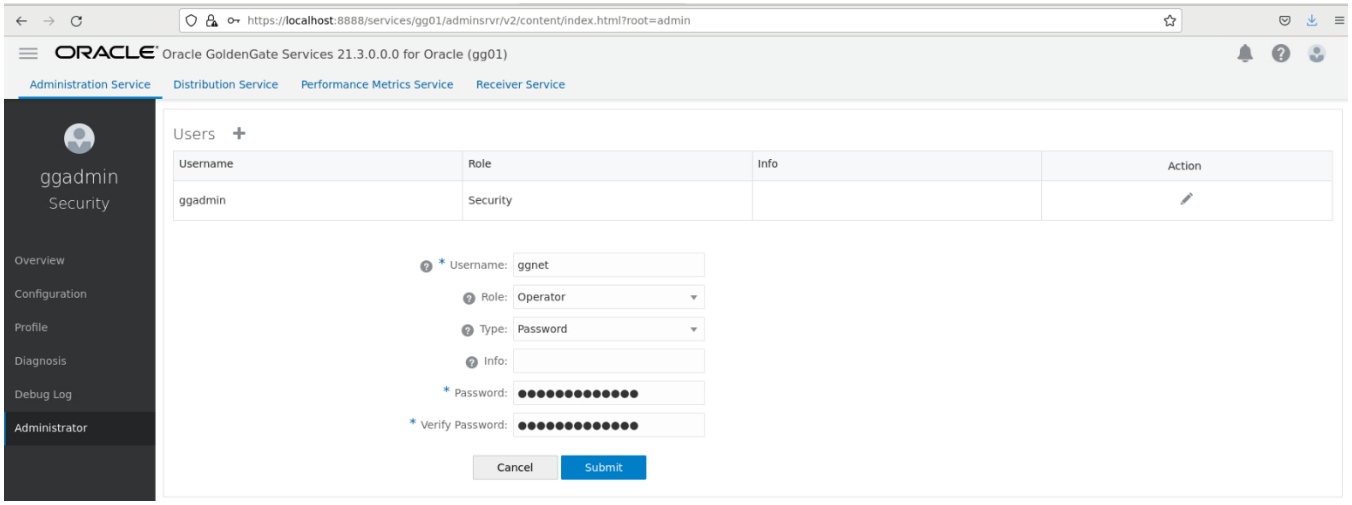

Figure 11: User creation in the target deployment for the source Oracle GodenGate to use

Step 11.3.3 - Create a Credential in the Source Oracle GoldenGate

Create a credential in the source deployment connecting the target deployment with the user created in the previous step. For example, a domain of OP2C and an alias of WSSNET.

- Log in to the **Administration Service** on the Source Oracle GoldenGate.
- Click in Configuration under Administration Service.
- Click the plus (+) sign next to Credentials on the Database home page. The Add Credentials page is displayed.
- Enter the details as follows:

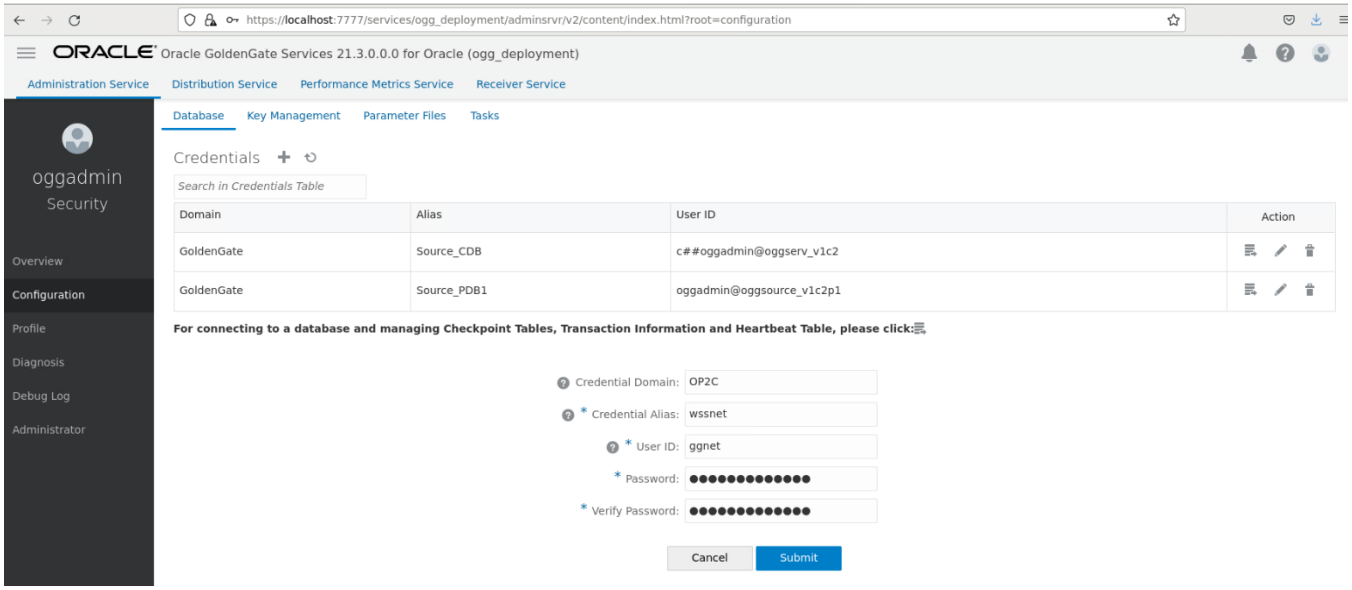

Figure 12: User creation in the source deployment

Step 11.3.4 - Create a Distribution Path on the Source Oracle GoldenGate to the Target Deployment

A path is created to send trail files from the Distribution Server to the Receiver Server. You can create a path from the Distribution Service. To add a path for the source deployment:

- Log in to the **Distribution Service** on the Source Oracle Goldengate.
- Click the plus (+) sign next to Path on the Distribution Service home page. The Add Path page is displayed.
- Enter the details as follows:

Options	Description
Path Name	Select a name for the path.
Source: *Trail Name*	Select the Extract name from the drop-down list, which populates the trail name automatically. If it doesn't, enter the trail name you provided while adding the Extract.
Generated Source URI	Specify localhost for the server's name; this allows the distribution path to be started on any of the Oracle RAC nodes.
Target Authentication Method	Use 'UserID Alias'
Target	Set the **Target** transfer protocol to wss (secure web socket). Set the **Target Host** to the target hostname/VIP that will be used for connecting to the target system along with the **Port Number** that NGINX was configured with (default is 443).
Domain	Set the **Domain** to the credential domain created above in Step 11.3.3, for example, OP2C.
Alias	The **Alias** is set to the credential alias wssnet, also created in Step 11.3.3.
Auto Restart Options	Set the distribution path to restart when the Distribution Server starts automatically. This is required, so that manual intervention is not required after a RAC node relocation of the Distribution Server. It is recommended to set the number of **Retries** to 10. Set the **Delay**, which is the time in minutes to pause between restart attempts, to 1.

- Click **Create Path.**
- From the Action Menu, click **Start**.
- Verify the Distribution Service is running

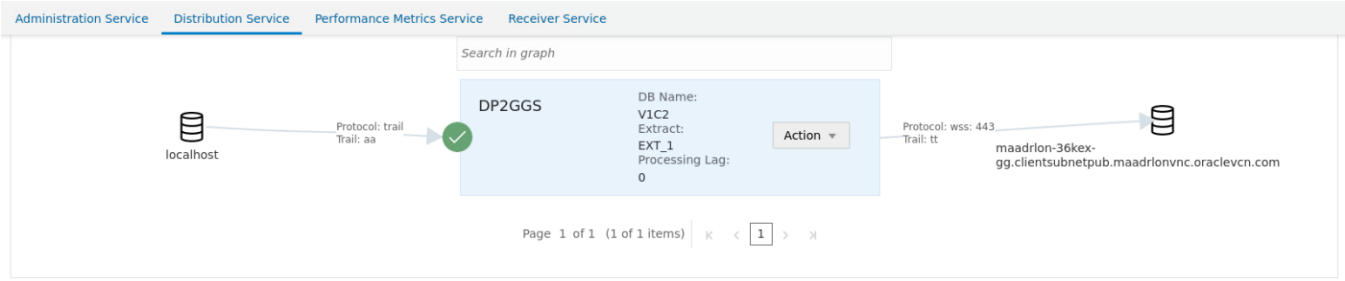

Figure 13: Oracle GoldenGate Distribution Service

Step 11.3.5 - Verify the Connection in the Target Deployment Console Receiver Service

- Log in to the **Administration Service** on the Target Deployment Console.
- Click on Receiver Service.

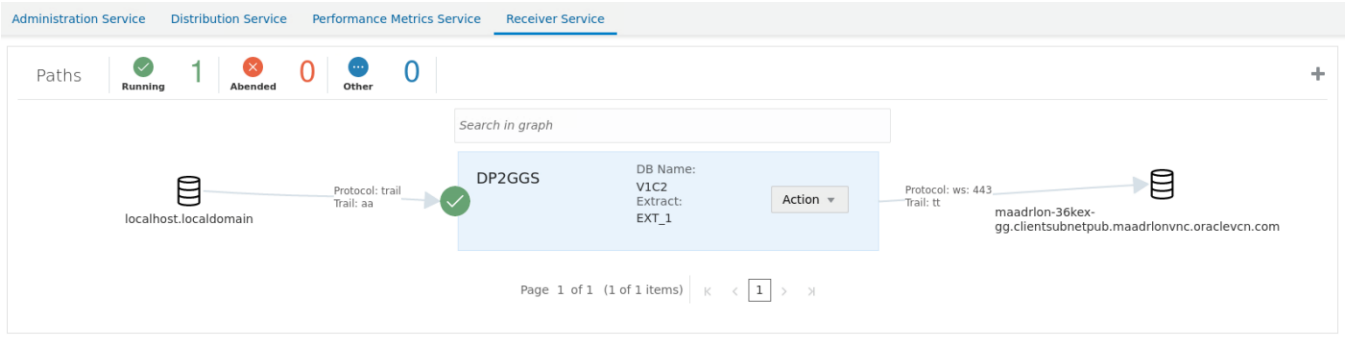

Figure 14: Oracle GoldenGate Receiver Service

Step 11.4 - Replicat Configuration

The Replicat process receives the trail data and applies it to the database.

Step 11.4.1 - Create the Checkpoint Table

The checkpoint table is a required component for Oracle GoldenGate Replicat processes. After connecting to the database from the Credentials page of the Administration Service, you can create the checkpoint table.

Create the checkpoint table in the target deployment:

- Log in to the **Administration Service** on the Target GoldenGate.
- Click in Configuration under Administration Service.
- Click on **Database** and **Connect** to the target database or PDB.

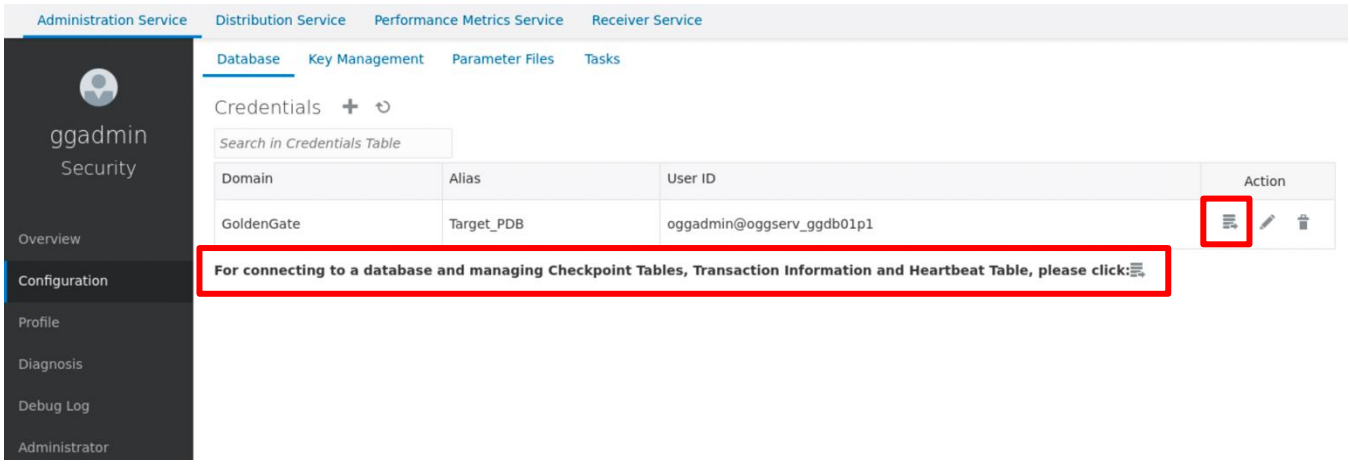

Figure 15: Oracle GoldenGate database connection

- Click the plus (+) sign next to Checkpoint. The Add Checkpoint page is displayed.
- Enter the details as follows:

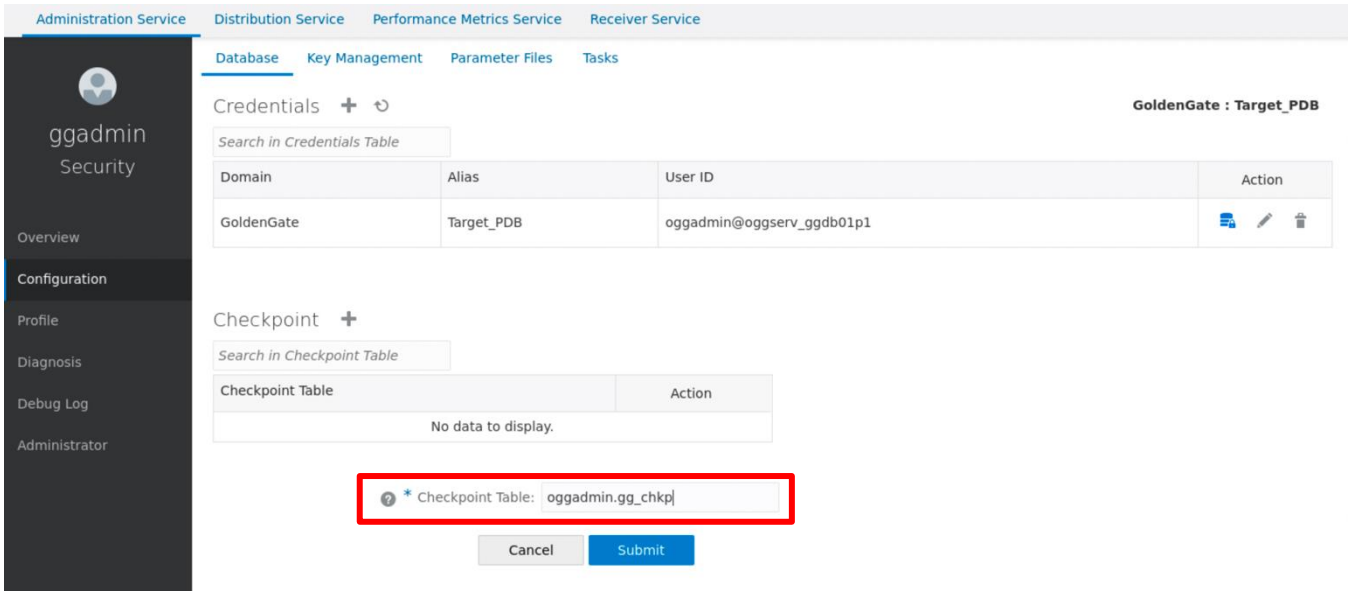

Figure 16: Creating the checkpoint table for Replicat processes

Refer to the Using Oracle GoldenGate with Oracle Database Guide for more information on the checkpoint table.

Step 11.4.2 - Add a Replicat

After you've set up your database connections and verified them, you can add a Replicat for the deployment by following these steps:

- Log in to the **Administration Service** on the Target GoldenGate.
- Click the plus (+) sign next to Replicats on the Administration Service home page. The Add Replicat page is displayed.
- Select a Replicat type and click **Next**.
- Enter the details as follows:

Options	Description
Process Name	The name of the Replicat process
Credential Domain	Credential domain created in Step 9.2. In our example is GoldenGate
Credential Alias	Credential alias created in Step 9.2. Our example is Target_PDB
Source	Select the source to use Trail.
Trail Name	A two-character trail name.
Trail Subdirectory	
Checkpoint Table	Set the use of an existing checkpoint table.

- Click **Create Path.**

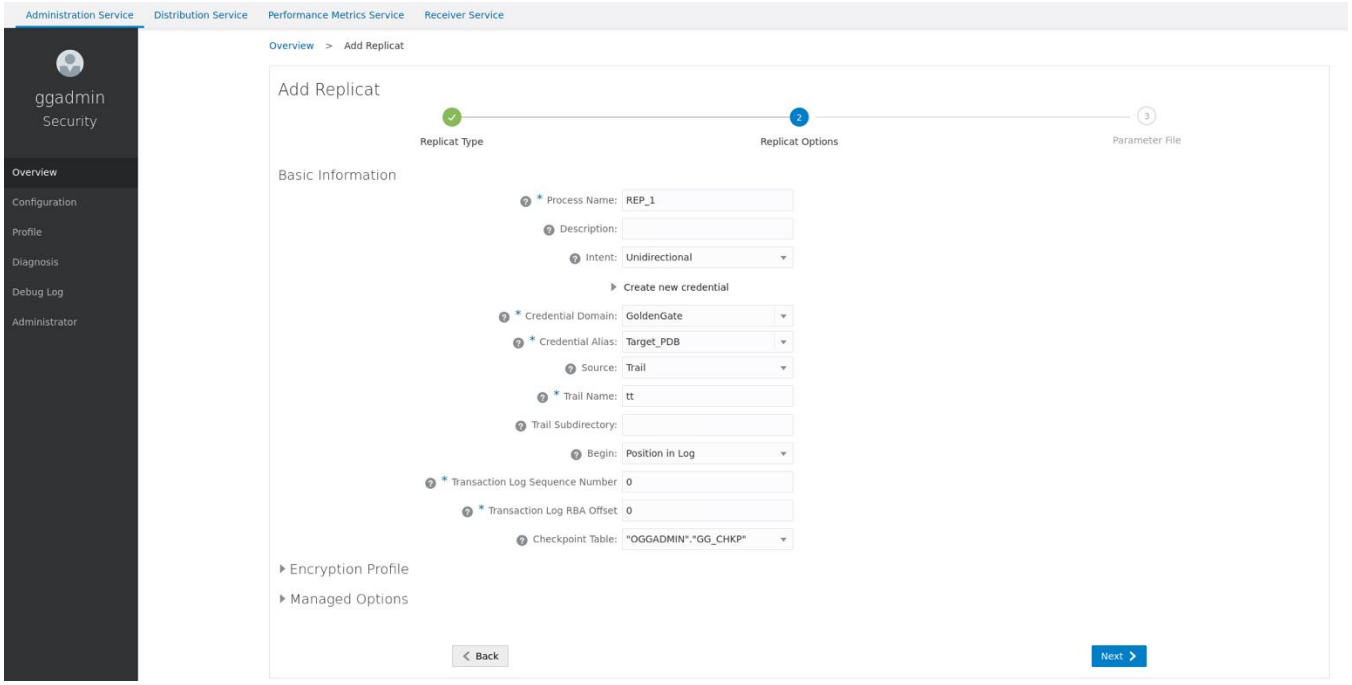

Figure 17: Oracle GoldenGate add replicat

- From the Action Menu, click **Start**.
- Verify the Replicat is running

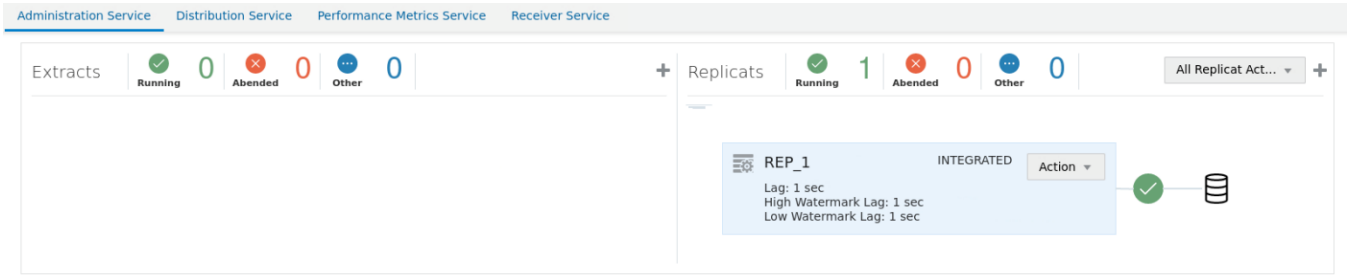

Figure 18: Verify replicats via Oracle GoldenGate Administration Service

References

- Oracle GoldenGate 21c Documentation
- Oracle Database SecureFiles and Large Object Developer's Guide 19c (DBFS)
- Oracle Automatic Storage Management Cluster File System Administrator's Guide 19c (ACFS)
- Oracle Clusterware Administration and Deployment Guide 19c
- Oracle Maximum Availability Architecture Website
  http://www.otn.oracle.com/goto/maa

Appendix A: Troubleshooting Oracle GoldenGate on Oracle RAC

There may be occasions when Oracle GoldenGate processes are not successfully started on an Oracle RAC node. Several files generated by Oracle GoldenGate, XAG, and CRS should be reviewed to determine the cause of the problem.

Below is a list of important log and trace files, their example locations, and some examples of output.

A.1 - XAG log file

Location: <XAG installation directory>/log/<hostname>
Example location: /u01/app/grid/xag/log/`hostname`

File name: `agctl_goldengate_grid.trc`
Contains all commands executed with agctl along with the output from the commands, including those that CRS executes.

Example:

```
2022-04-18 11:52:21: stop resource success
2022-04-18 11:52:38: agctl start goldengate <instance_name>
2022-04-18 11:52:38: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl status res
xag.<INSTANCE_NAME>.goldengate
2022-04-18 11:52:38: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl status res
xag.<INSTANCE_NAME>.goldengate -f
2022-04-18 11:52:38: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl start resource
xag.<INSTANCE_NAME>.goldengate -f
2022-04-18 11:52:45: Command output:
>  CRS-2672: Attempting to start 'xag.<INSTANCE_NAME>.goldengate' on 'exadb-node1'
>  CRS-2676: Start of 'xag.<INSTANCE_NAME>.goldengate' on 'exadb-node1' succeeded
>End Command output
2022-04-18 11:52:45: start resource success
```

A.2 - XAG GoldenGate instance trace file

Location: <XAG installation directory>/log/<hostname>
Example location: /u01/app/grid/xag/log/`hostname`

File name: <GoldenGate_instance_name>_agent_goldengate.trc
It contains the output from the commands executed by agctl, the environment variables used, and any debug output enabled for the underlying commands.

Example:

```
2022-04-18 12:14:46: Exported ORACLE_SID ggdg1
2022-04-18 12:14:46: Exported GGS_HOME /u01/oracle/goldengate/gg21c_MS
2022-04-18 12:14:46: Exported OGG_CONF_HOME /mnt/dbfs/goldengate/deployments/ggsm01/etc/conf
2022-04-18 12:14:46: Exported LD_LIBRARY_PATH
/u01/oracle/goldengate/gg21c_MS:/u01/app/19.0.0.0/grid/lib:/etc/ORCLcluster/lib
2022-04-18 12:14:46: Exported LD_LIBRARY_PATH_64 /u01/oracle/goldengate/gg21c_MS
2022-04-18 12:14:46: Exported LIBPATH /u01/oracle/goldengate/gg21c_MS
2022-04-18 12:14:46: ogg input =
{"oggHome":"/u01/oracle/goldengate/gg21c_MS","serviceManager":{"oggConfHome":"/mnt/dbfs/goldengate/deploym
ents/ggsm01/etc/conf","portNumber":9100},"username":"admin","credential":"xyz"}
2022-04-18 12:14:46: About to exec /u01/oracle/goldengate/gg21c_MS/bin/XAGTask HealthCheck
2022-04-18 12:14:47: XAGTask retcode = 0
```

A.3 - CRS trace file

Location: /u01/app/grid/diag/crs/<hostname>/crs/trace
Example location: /u01/app/grid/diag/crs/`hostname`/crs/trace

File name: `crsd_scriptagent_oracle.trc`
Contains the output created by any CRS resource action scripts, like XAG or dbfs_mount. This trace file is crucial to determining why DBFS or GoldenGate did not start on a RAC node.

Example:

```
2022-04-18 11:52:38.634 :    AGFW:549631744: {1:30281:59063} Agent received the message:
RESOURCE_START[xag.<INSTANCE_NAME>.goldengate 1 1] ID 4098:4125749
2022-04-18 11:52:38.634 :    AGFW:549631744: {1:30281:59063} Preparing START command for:
xag.<INSTANCE_NAME>.goldengate 1 1
2022-04-18 11:52:38.634 :    AGFW:549631744: {1:30281:59063} xag.<INSTANCE_NAME>.goldengate 1 1 state
changed from: OFFLINE to: STARTING
2022-04-18 11:52:38.634 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
Executing action script: /u01/oracle/XAG_MA/bin/aggoldengatescaas[start]
2022-04-18 11:52:38.786 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start] GG
agent running command 'start' on xag.<INSTANCE_NAME>.goldengate
2022-04-18 11:52:42.140 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
ServiceManager fork pid = 265747
2022-04-18 11:52:42.140 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
Waiting for /mnt/dbfs/goldengate/deployments/ggsm01/var/run/ServiceManager.pid
2022-04-18 11:52:42.140 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
Waiting for SM to start
2022-04-18 11:52:42.140 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
ServiceManager PID = 265749
2022-04-18 11:52:43.643 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start]
XAGTask retcode = 0
2022-04-18 11:52:43.643 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [start] XAG
HealthCheck after start returned 0
2022-04-18 11:52:43.643 :    AGFW:558036736: {1:30281:59063} Command: start for resource:
xag.<INSTANCE_NAME>.goldengate 1 1 completed with status: SUCCESS
2022-04-18 11:52:43.643 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [check]
Executing action script: /u01/oracle/XAG_MA/bin/aggoldengatescaas[check]
2022-04-18 11:52:43.644 :    AGFW:549631744: {1:30281:59063} Agent sending reply for:
RESOURCE_START[xag.<INSTANCE_NAME>.goldengate 1 1] ID 4098:4125749
2022-04-18 11:52:43.795 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [check] GG
agent running command 'check' on xag.<INSTANCE_NAME>.goldengate
2022-04-18 11:52:45.548 :CLSDYNAM:558036736: [xag.<INSTANCE_NAME>.goldengate]{1:30281:59063} [check]
XAGTask retcode = 0
2022-04-18 11:52:45.548 :    AGFW:549631744: {1:30281:59063} xag.<INSTANCE_NAME>.goldengate 1 1 state
changed from: STARTING to: ONLINE
```

A.4 - GoldenGate deployment log files

Location: <Goldengate_deployment_directory>/<instance_name>/var/log
Example location: /mnt/dbfs/goldengate/deployments/<instance_name>/var/log

File names: `adminsrvr.log, recvsrvr.log, pmsrvr.log, distsrvr.log`
Contains the output of start, stop, and status checks of the Oracle GoldenGate deployment processes (Administration Server, Distribution Server, Receiver Server, and Performance Metrics Server).

Example:

```
2022-04-18T11:52:42.645-0400 INFO | Setting deploymentName to '<instance_name>'. (main)
2022-04-18T11:52:42.665-0400 INFO | Read SharedContext from store for length 19 of file
'/mnt/dbfs/goldengate/deployments/<instance_name>/var/lib/conf/adminsrvr-resources.dat'. (main)
2022-04-18T11:52:42.723-0400 INFO | XAG Integration enabled (main)
2022-04-18T11:52:42.723-0400 INFO | Configuring security. (main)
2022-04-18T11:52:42.723-0400 INFO | Configuring user authorization secure store path as
'/mnt/dbfs/goldengate/deployments/<instance_name>/var/lib/credential/secureStore/'. (main)
2022-04-18T11:52:42.731-0400 INFO | Configuring user authorization as ENABLED. (main)
2022-04-18T11:52:42.749-0400 INFO | Set network configuration. (main)
2022-04-18T11:52:42.749-0400 INFO | Asynchronous operations are enabled with default synchronous wait time
of 30 seconds (main)
2022-04-18T11:52:42.749-0400 INFO | HttpServer configuration complete. (main)
2022-04-18T11:52:42.805-0400 INFO | SIGHUP handler installed. (main)
2022-04-18T11:52:42.813-0400 INFO | SIGINT handler installed. (main)
2022-04-18T11:52:42.815-0400 INFO | SIGTERM handler installed. (main)
2022-04-18T11:52:42.817-0400 WARN | Security is configured as 'disabled'. (main)
2022-04-18T11:52:42.818-0400 INFO | Starting service listener... (main)
2022-04-18T11:52:42.819-0400 INFO | Mapped 'ALL' interface to address 'ANY:9101' with default IPV4/IPV6
options identified by 'exadb-node1.us.oracle.com'. (main)
2022-04-18T11:52:42.821-0400 INFO | Captured 1 interface host names: 'exadb-node1.us.oracle.com' (main)
2022-04-18T11:52:42.824-0400 INFO | The Network ipACL specification is empty. Accepting ANY address on ALL
interfaces. (main)
2022-04-18T11:52:42.826-0400 INFO | Server started at 2022-04-18T11:52:42.827-05:00 (2022-04-
18T15:52:42.827Z GMT) (main)
```

A.5 - GoldenGate report files

Location: <Goldengate_deployment_directory>/<instance_name>/var/lib/report
Example location: /mnt/dbfs/goldengate/deployments/<instance_name>/var/lib/report

The GoldenGate report files contain important information, warning messages, and errors for all GoldenGate processes, including the Manager processes. If any of the GoldenGate processes fail to start or abend when running, the process report file will contain important information that can be used to determine the cause of the failure.

Example errors from an Extract report file:

```
2022-04-23 13:01:50  ERROR   OGG-00446  Unable to lock file "
/mnt/acfs_gg/deployments/<instance_name>/var/lib/checkpt/EXT_1A.cpe" (error 95, Operation not supported).
2022-04-23 13:01:50  ERROR   OGG-01668  PROCESS ABENDING.
```

Appendix B: Example Configuration Problems

Below are some configuration problems that can be encountered with GoldenGate in a RAC environment and how to diagnose and resolve them.

B.1 - Incorrect parameter settings in the `mount-dbfs.conf` file

When XAG fails to mount DBFS, the failure will be reported either on the command line (if you are running the manual `agctl` command) or in the XAG log file:

```
$ agctl start goldengate <instance_name> --node exadb-node1
CRS-2672: Attempting to start 'dbfs_mount' on 'exadb-node1'
CRS-2674: Start of 'dbfs_mount' on 'exadb-node1' failed
CRS-2679: Attempting to clean 'dbfs_mount' on 'exadb-node1'
CRS-2681: Clean of 'dbfs_mount' on 'exadb-node1' succeeded
CRS-4000: Command Start failed, or completed with errors.
```

The XAG log file (`agctl_goldengate_grid.trc`) has the advantage that it shows timestamps that can be used when looking at other log or trace files:

```
2022-04-19 15:32:16: executing cmd: /u01/app/19.0.0.0/grid/bin/crsctl start resource
xag.<INSTANCE_NAME>.goldengate -f -n exadb-node1
2022-04-19 15:32:19: Command output:
>  CRS-2672: Attempting to start 'dbfs_mount' on 'exadb-node1'
>  CRS-2674: Start of 'dbfs_mount' on 'exadb-node1' failed
>  CRS-2679: Attempting to clean 'dbfs_mount' on 'exadb-node1'
>  CRS-2681: Clean of 'dbfs_mount' on 'exadb-node1' succeeded
>  CRS-4000: Command Start failed, or completed with errors.
>End Command output
2022-04-19 15:32:19: start resource failed rc=1
```

Next, check the CRS trace file (`crsd_scriptagent_oracle.trc`), which shows why DBFS failed to mount. Below are some example errors caused by incorrect parameter settings in the `mount-dbfs.conf` file.

- Incorrect DBNAME

```
2022-04-19 15:32:16.679 :    AGFW:1190405888: {1:30281:17383} dbfs_mount 1 1 state changed from: UNKNOWN
to: STARTING
2022-04-19 15:32:16.680 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383} [start] Executing action script:
/u01/oracle/scripts/mount-dbfs.sh[start]
2022-04-19 15:32:16.732 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383} [start] mount-dbfs.sh mounting
DBFS at /mnt/dbfs from database ggdg
2022-04-19 15:32:17.883 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383} [start] ORACLE_SID is
2022-04-19 15:32:17.883 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383} [start] No running ORACLE_SID
available on this host, exiting
2022-04-19 15:32:17.883 :    AGFW:1192507136: {1:30281:17383} Command: start for resource: dbfs_mount 1 1
completed with invalid status: 2
```

- Incorrect MOUNT_POINT

```
2022-04-19 16:45:14.534 :    AGFW:1734321920: {1:30281:17604} dbfs_mount 1 1 state changed from: UNKNOWN
to: STARTING
2022-04-19 16:45:14.535 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] Executing action script:
/u01/oracle/scripts/mount-dbfs.sh[start]
2022-04-19 16:45:14.586 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] mount-dbfs.sh mounting
DBFS at /mnt/dbfs from database ggdgs
```

```
2022-04-19 16:45:15.638 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] ORACLE_SID is ggdg1
2022-04-19 16:45:15.738 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] spawning dbfs_client
command using SID ggdg1
2022-04-19 16:45:20.745 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] fuse: bad mount point
`/mnt/dbfs': No such file or directory
2022-04-19 16:45:21.747 :CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604} [start] Start - OFFLINE
2022-04-19 16:45:21.747 :    AGFW:1736423168: {1:30281:17604} Command: start for resource: dbfs_mount 1 1
completed with status: FAIL
```

- Incorrect DBFS_USER or DBFS_PASSWD

```
2022-04-19 16:47:47.855 :    AGFW:1384478464: {1:30281:17671} dbfs_mount 1 1 state changed from: UNKNOWN
to: STARTING
2022-04-19 16:47:47.856 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] Executing action script:
/u01/oracle/scripts/mount-dbfs.sh[start]
2022-04-19 16:47:47.908 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] mount-dbfs.sh mounting
DBFS at /mnt/dbfs from database ggdgs
2022-04-19 16:47:48.959 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] ORACLE_SID is ggdg1
2022-04-19 16:47:49.010 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] spawning dbfs_client
command using SID ggdg1
2022-04-19 16:47:55.118 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] Fail to connect to
database server. Error: ORA-01017: invalid username/password; logon denied
2022-04-19 16:47:55.118 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start]
2022-04-19 16:47:56.219 :CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671} [start] Start - OFFLINE
2022-04-19 16:47:56.220 :    AGFW:1386579712: {1:30281:17671} Command: start for resource: dbfs_mount 1 1
completed with status: FAIL
```

- Incorrect ORACLE_HOME

```
2022-04-19 16:50:38.952 :    AGFW:567502592: {1:30281:17739} dbfs_mount 1 1 state changed from: UNKNOWN
to: STARTING
2022-04-19 16:50:38.953 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start] Executing action script:
/u01/oracle/scripts/mount-dbfs.sh[start]
2022-04-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start] mount-dbfs.sh mounting
DBFS at /mnt/dbfs from database ggdgs
2022-04-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start]
/u01/oracle/scripts/mount-dbfs.sh: line 136: /u01/app/oracle/product/19.0.0.0/rdbms/bin/srvctl: No such
file or directory
2022-04-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start]
/u01/oracle/scripts/mount-dbfs.sh: line 139: /u01/app/oracle/product/19.0.0.0/rdbms/bin/srvctl: No such
file or directory
2022-04-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start] ORACLE_SID is
2022-04-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739} [start] No running ORACLE_SID
available on this host, exiting
2022-04-19 16:50:39.004 :    AGFW:569603840: {1:30281:17739} Command: start for resource: dbfs_mount 1 1
completed with invalid status: 2
```

To resolve these configuration issues, set the correct parameter values in `mount-dbfs.conf`.

B.2 - Problems with file locking on DBFS

If using Oracle Database 12c Release 2 (12.2) and the `nolock` DBFS mount option is not used, there can be problems with GoldenGate processes trying to lock checkpoint or trail files. The same problem will be encountered if using Oracle Database 11g Release 2 (11.2.0.4) or 12c Release 1 (12.1) with a patch for bug 22646150 applied. This patch changes how DBFS handles file locking

to match Oracle Database 12c Release 2 (12.2). To add the `nolock` DBFS mount option, a patch for bug 27056711 must be applied to the database. If the patch for bug 22646150 has not been applied to the database, the patch for bug 27056711 and the `nolock` mount option is not required.

Below is an example of diagnosing a GoldenGate Microservices Architecture locking problem.

When starting a deployment with XAG, one or more processes may not start due to detecting a locking conflict on one or more files. This will often occur after a RAC node failover where the deployment did not get a chance to shut down cleanly.

When one of the deployment server processes fails to start (Administration Server, Performance Metrics Server, Distribution Server, Receiver Server, or Service Manager), check the log file for the particular server located in the deployment `var/log` directory.

For example, the log file `/mnt/dbfs/goldengate/deployments/<INSTANCE_NAME>/var/log/pmsrvr.log` shows the following error on startup:

```
2022-04-11T12:41:57.619-0700 ERROR| SecureStore failed on open after retrying due to extended file lock.
(main)
2022-04-11T12:41:57.619-0700 ERROR| SecureStore failed to close (28771). (main)
2022-04-11T12:41:57.619-0700 INFO | Set network configuration. (main)
2022-04-11T12:41:57.619-0700 INFO | Asynchronous operations are enabled with default synchronous wait time
of 30 seconds (main)
2022-04-11T12:41:57.619-0700 INFO | HttpServer configuration complete. (main)
2022-04-11T12:42:07.674-0700 ERROR| Unable to lock process file, Error is [1454] - OGG-01454 (main)
2022-04-11T12:42:07.675-0700 ERROR| Another Instance of PM Server is Already Running (main)
```

An Extract process will report start-up failures in the ER-events.log logfile located in the deployment log file directory.

For example, /mnt/dbfs/goldengate/deployments/<instance_name>/var/log/ER-events.log shows the following error:

```
2022-04-11T00:14:56.845-0700  ERROR   OGG-01454  Oracle GoldenGate Capture for Oracle, EXT1.prm:  Unable
to lock file "/mnt/dbfs/goldengate/deployments/<instance_name>/var/run/EXT1.pce" (error 11, Resource
temporarily unavailable).  Lock currently held by process id (PID) 237495.
2022-04-11T00:14:56.861-0700  ERROR   OGG-01668  Oracle GoldenGate Capture for Oracle, EXT1.prm:  PROCESS
ABENDING.
```

Next, check to ensure the process failing to start up is not running on any of the RAC nodes.

Example:

```
$ ps -ef|grep EXT1|grep -v grep
```

Once it has been determined that the process is not running, the deployment must be shutdown cleanly, the file system unmounted, and the correct DBFS patch applied.

Example:

```
$ agctl stop goldengate <INSTANCE_NAME>
$ crsctl stop resource dbfs_mount
```

Check the DBFS mount options:

```
$ ps -ef|grep dbfs_client
oracle   204017      1  0 14:37 ?        00:00:00
/u01/app/oracle/product/19.1.0.0/dbhome_1/bin/dbfs_client dbfs@dbfs.local -o
allow_other,failover,direct_io /mnt/dbfs
```

It is clear the nolock mount option was not used, which leads to the locking errors.

Use the guidelines above on page 36 to determine if a DBFS patch is required. After which, add the `nolock` mount option to the `mount-dbfs.conf` file on all RAC nodes that are part of the deployment.

Example:

```
MOUNT_OPTIONS=allow_other,direct_io,failover,nolock
```

Finally, restart the deployment:

```
$ agctl start goldengate <INSTANCE_NAME>
```