

Oracle GoldenGate Microservices Architecture on Oracle Cloud Infrastructure

Configuration Guide

ORACLE WHITE PAPER | APRIL 2019





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Oracle GoldenGate lifecycle management is independent of Oracle Database on Oracle Cloud Infrastructure and must be handled by clients.

Revision History

The following revisions have been made to this white paper since its initial publication:

Date	Revision
April 18, 2019	Initial publication

You can find the most recent versions of the Oracle Cloud Infrastructure white papers at <https://cloud.oracle.com/iaas/technical-resources>.



Table of Contents

Overview	4
Oracle Software	4
Component Overview	4
Configuration Overview	6
Configure Oracle GoldenGate on Oracle Cloud Infrastructure	7
Set Up the Oracle Cloud Infrastructure DB System	7
Configure a VCN	7
Install an Oracle Client	8
Set Up a File System	9
Configure the Databases	19
Configure Oracle GoldenGate	20
Create a TNS Alias	31
Configure HA for Oracle GoldenGate	32
Configure Reverse Proxy and HA	36
Configure Data Replication	38
Conclusion	41



Overview

Oracle GoldenGate is a comprehensive software package for real-time data integration and replication in heterogeneous IT environments. The product set enables high-availability solutions, real-time data integration, transactional change data capture, data replication, transformations, and verification between operational and analytical enterprise systems.

Oracle GoldenGate supports two architectures, the Classic Architecture and the Microservices Architecture. The Microservices Architecture provides REST-enabled services as part of the Oracle GoldenGate environment. The REST-enabled services provide remote configuration, administration, and monitoring through HTML5 web pages, command line, and APIs.

This white paper describes best practices for configuring Oracle GoldenGate Microservices Architecture on Oracle Cloud Infrastructure virtual machine (VM), bare metal, or Exadata databases. Configuring Oracle GoldenGate on on-premise databases is not covered in this white paper. For details, see [Oracle GoldenGate Performance Best Practices](#) and [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Oracle Software

Use Oracle GoldenGate Release 18.1 or later to take advantage of increased functionality and enhanced performance features. You can [download the latest release of Oracle GoldenGate](#). Patches and updates are available from [My Oracle Support](#). Confirm the latest recommended patches from [Latest GoldenGate/Database \(OGG/RDBMS\) Patch recommendations \(Doc ID 2193391.1\)](#).

We recommend using Oracle GoldenGate with Oracle Database Cloud Enterprise Edition for enhanced performance. To use the integrated Extract and Replicat GoldenGate features, you must have at least Oracle Database 11g Release 2 (11.2.0.4). For details, see [Oracle GoldenGate Performance Best Practices](#).

Component Overview

The following components are involved in configuring Oracle GoldenGate on Oracle Cloud Infrastructure.

Oracle Database Cloud Service

The Oracle Database Cloud Service provided on Oracle Cloud Infrastructure enables users to access databases on bare metal, VM, and Exadata DB systems.



Oracle GoldenGate

Oracle GoldenGate is a data-replication solution that enables the exchange and manipulation of data at the transaction level among multiple, heterogeneous platforms across the enterprise. It moves committed transactions with transaction integrity and minimal overhead on your existing infrastructure. It can be installed in various topologies, such as unidirectional, bi-directional, peer-to-peer, broadcast, and consolidation cascading. For details, see the [Oracle GoldenGate website](#).

Oracle Grid Infrastructure Bundled Agents

Oracle Grid Infrastructure Bundled Agents (XAG) are Oracle Grid Infrastructure components that provide a high-availability (HA) framework to application resources and resource types managed through the agent management interface, Agent Control (AGCTL). The framework provides a complete, ready-to-use application HA solution that contains predefined Oracle Grid Infrastructure resource configurations and agents to integrate applications for complete application HA. The agent command-line utility, AGCTL, is used to start and stop Oracle GoldenGate and can also be used to relocate Oracle GoldenGate between the nodes in the cluster.

When you use Oracle GoldenGate Microservices Architecture, you must use XAG version 9 or later. You can download the [latest agent software](#).

Oracle Database File System

The Oracle Database File System (DBFS) creates a file system interface to files stored in a 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 HA and disaster-recovery (DR) capabilities provided by the Oracle Database.

Oracle Automatic Storage Management Cluster File System

The Oracle Automatic Storage Management Cluster File System (ACFS) is a general-purpose industry-standard POSIX, X/OPEN, and Windows compliant cluster file system that supports multiple OS and server platforms, including Oracle Exadata Database Machine, Oracle Database Appliance, and traditional server environments. Oracle ACFS uses Oracle Automatic Storage Management (ASM) files and inherits ASM features, including striping, mirroring, and rebalancing.



Configuration Overview

This section provides an overview of the steps that you need to follow to configure Oracle GoldenGate on Oracle Cloud Infrastructure. The rest of the paper provides the details of these steps.

1. **Set Up the Oracle Cloud Infrastructure DB System:** To configure Oracle GoldenGate on Oracle Cloud Infrastructure, you need a VM, bare metal, or Exadata DB system on Oracle Cloud Infrastructure. To learn about these database systems, see [Bare Metal and Virtual Machine DB Systems](#) and [Exadata DB Systems](#).
2. **Configure a VCN:** A virtual cloud network (VCN) is a software-defined network that enables you to securely connect to your resources, such as databases, on Oracle Cloud Infrastructure. You must configure VCN components such as security lists and firewalls for Oracle GoldenGate to function properly. To learn more about VCNs and security lists, including instructions for creating them, see the [Oracle Cloud Infrastructure Networking documentation](#).
3. **Install an Oracle Client:** Install an Oracle Client to provide dedicated access to Oracle Client libraries for Oracle GoldenGate. Oracle GoldenGate uses client libraries so that it doesn't depend on Oracle Database libraries.
4. **Set Up a File System:** You must set up either Oracle DBFS or Oracle ACFS for configuring HA on Oracle Cloud Infrastructure with Oracle GoldenGate.
5. **Configure Databases:** Use best practices to configure the source and target databases in an Oracle GoldenGate replicated environment.
6. **Configure Oracle GoldenGate:** Use best practices to install and configure Oracle GoldenGate components on Oracle Cloud Infrastructure.
7. **Create a TNS Alias:** You create a TNS alias to simplify database connectivity of the Oracle GoldenGate processes when switching between Oracle RAC nodes.
8. **Configure HA for Oracle GoldenGate:** You configure Oracle GoldenGate for HA on Oracle Cloud Infrastructure.
9. **Configure Reverse Proxy and HA:** Configure reverse proxy and HA by using Nginx.
10. **Configure Data Replication:** Create and configure Oracle GoldenGate Extract, Replicat, and Path processes need for data replication.

Configure Oracle GoldenGate on Oracle Cloud Infrastructure

This section provides the detailed steps for configuring Oracle GoldenGate on Oracle Cloud Infrastructure.

Set Up the Oracle Cloud Infrastructure DB System

To get started, you need a VM, bare metal, or Exadata DB system on Oracle Cloud Infrastructure for Oracle GoldenGate deployment. You can deploy Oracle GoldenGate with an existing DB system or launch a new DB system, according to your business needs. For instructions on launching and managing a DB system on Oracle Cloud Infrastructure, see [Managing Bare Metal and Virtual Machine DB Systems](#) or [Managing Exadata DB Systems](#).

Configure a VCN

This section provides instructions for configuring a VCN to use Oracle GoldenGate on Oracle Cloud Infrastructure. For

Security List

Open ingress port 443 to connect the Oracle GoldenGate service using Nginx as a reverse proxy. For more information, see [Updating the Security List for the DB System](#).

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

- **Source:** Source CIDR block
- **IP Protocol:** TCP
- **Source Port Range:** All
- **Destination Port Range:** 443
- **Allows:** TCP traffic for ports: 443 HTTPS

Source: 0.0.0.0/0	IP Protocol: TCP	Source Port Range: All	Destination Port Range: 443	Allows: TCP traffic for ports: 443 HTTPS
-------------------	------------------	------------------------	-----------------------------	--

Firewall

Update the firewall on the host DB system to open port 443 for ingress activity.

```
[ssh into DB System as opc user]
sudo iptables-save > /tmp/iptables.orig
sudo iptables -I INPUT 8 -p tcp -m state --state NEW -m tcp --dport 443 -j
ACCEPT -m comment --comment "for Nginx"
sudo service iptables save
sudo service iptables restart
sudo iptables --list -n
```

Install an Oracle Client

Use these instructions to install an Oracle Client to provide dedicated access to Oracle Client libraries for Oracle GoldenGate.

The client home requires about 1.7 GB space in the /u01 mount point. Ensure that enough free space exists in the mount point when you are performing any lifecycle management operations, such as patching or creating additional homes, on these environments using the available cloud tooling.

1. [Download the Oracle Client software](#) that is the same version as Oracle Database. (Click the **See All** link to access the client file).
2. After downloading the file, upload it to each database host.
3. Create an installation directory. Following is an example for the Oracle Database 18c client:

```
[ssh into DB System as opc user]
sudo mkdir -p /u01/app/oracle/product/18.0.0.0/client
sudo chown -R oracle:oinstall /u01/app/oracle/product/18.0.0.0/client
```

4. Edit the response file:

```
[ssh into DB System as opc user]
sudo su - oracle
cd <installation media>
vi ./response/client_install.rsp
```


Following is an example of the parameters in response file:

```
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=<client home>
ORACLE_BASE=/u01/app/oracle
oracle.install.client.installType=Custom
oracle.install.client.customComponents="oracle.sqlplus:18.0.0.0.0","oracle.rdbms.oci:18.0.0.0.0"," oracle.rdbms.util:18.0.0.0.0"
```

5. Install the Oracle Client:

```
[ssh into DB System as opc user]
sudo su - oracle
cd <installation media>
./runInstaller -silent -nowait -responseFile <path of response file>
```

Ignore any "insufficient swap space" warnings while installing the client software.

6. Copy the `tnsnames.ora` file from the database home to the client home:

```
[ssh into DB System as opc user]
sudo su - oracle
cp <database home>/network/admin/tnsnames.ora <client home>/network/admin
```

A TNS alias will be added in the `tnsnames.ora` file in the client home and be used by Oracle GoldenGate.

Note: The cloud tooling provided by Oracle doesn't support lifecycle management operations on the Oracle Client home. It is the customer's responsibility to patch this client home.

Set Up a File System

You must set up either an Oracle ACFS or an Oracle DBFS file system for HA. If Oracle Data Guard is already configured on your DB system, we recommend using DBFS.

Set Up ACFS

1. Create the file system by using ASMCMD.

Modify the file system size according to the determined size requirements. If DATA disk group doesn't have enough space, consider using RECO disk group. For details, see the [disk requirements](#) in the Oracle GoldenGate documentation.

```
[ssh into DB System as opc user]
sudo su - grid
asmcmd volcreate -G DATA -s 10G acfs_gg
```

2. Confirm the volume device:

```
[ssh into DB System as opc user]
sudo su - grid
asmcmd volinfo -G DATA acfs_gg
```

Following is an example of the ACFS volume device output:

```
Diskgroup Name: DATA
Volume Name: ACFS_GG
Volume Device: /dev/asm/acfs_gg-453
State: ENABLED
Size (MB): 10240
Resize Unit (MB): 512
Redundancy: UNPROT
Stripe Columns: 8
Stripe Width (K): 1024
Usage: ACFS
Mountpath:
```

3. Create the ACFS file system:

```
/sbin/mkfs -t acfs <volume device>
```

4. Create a directory for ACFS:

```
[ssh into DB System as opc user]
su -
mkdir /mnt/acfs_gg
```

5. Confirm the ORACLE_HOME and PATH environment variables for the grid user:

```
[ssh into DB System as opc user]
sudo su - grid
cat .bashrc
```

Following is an example for Grid Infrastructure 12.2:

```
ORACLE_HOME=/u01/app/12.2.0.1/grid; export ORACLE_HOME
PATH=$PATH:/u01/app/12.2.0.1/grid/bin; export PATH
LD_LIBRARY_PATH=/u01/app/12.2.0.1/grid/lib; export LD_LIBRARY_PATH
ORACLE_SID=+ASM1; export ORACLE_SID
```

6. Create the Cluster Ready Services (CRS) resource for the new ACFS file system, if it's not already created:

```
[ssh into DB System as opc user]
su -
export ORACLE_HOME=<ORACLE_HOME environmental variable of grid user>
export PATH=<PATH environmental variable of grid user>
export LD_LIBRARY_PATH=<LD_LIBRARY_PATH environmental variable of grid user>
```

```
export ORACLE_SID=<ORACLE_SID environmental variable of grid user>
srvctl add filesystem -device <device name> -volume acfs_gg -diskgroup
DATA -path /mnt/acfs_gg -user oracle -autostart NEVER
```

7. To verify the currently configured ACFS file systems, use the following command to view the file system details:

```
[ssh into DB System as opc user]
sudo su - grid
srvctl config filesystem
```

8. Check the status of the ACFS resource and mount it:

```
[ssh into DB System as opc user]
sudo su - grid
srvctl status filesystem -volume acfs_gg -diskgroup DATA
srvctl start filesystem -volume acfs_gg -diskgroup DATA
```

The CRS resource that is created is named using the format `ora.<diskgroup name>.<volume name>.acfs`. In the preceding file system example, the CRS resource is called `ora.datacl.acfs_gg.acfs`.

9. After the file system is mounted, create the directory for storing the Oracle GoldenGate deployment and the Service Manager:

```
[ssh into DB System as opc user]
sudo su - oracle
cd /mnt/acfs_gg
mkdir deployments
```

Set Up DBFS

Oracle DBFS provides a file-system interface to files that are stored in an Oracle database. Files stored in an Oracle database are usually stored as SecureFiles LOBs, and pathnames, directories, and other file system information are stored in database tables. A set of PL/SQL procedures implement the file-system access primitives like open and create. On Linux, the `DBFS_client` interfaces with the FUSE kernel module to implement a file-system mount point that provides transparent access to the files stored in the database.

For more information about DBFS, see the DBFS [documentation](#).

1. If you're using a VM or bare metal DB system, install the following RPMs on all nodes of the Oracle RAC cluster. If you're using an Exadata DB system, you don't need to install RPMs. To set up the yum repository on a VM or bare metal DB system, see [OS Updates](#).

```
#yum install fuse-libs
#yum install fuse-devel
```

2. Create a tablespace for DBFS. If you are using a VM or bare metal DB system, use the +DATA disk group.

```
SQL> Alter session set container=<pdb_name>;
SQL> create bigfile tablespace dbfstfs datafile '+DATA1' size 32g
autoextend on next 8g maxsize 300g EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO ;
SQL> create user dbfs_user identified by <password> default tablespace
dbfstfs quota unlimited on dbfstfs;
SQL> grant create session, create table, create view, create procedure,
dbfs_role to dbfs_user;
```

3. As the user root, create the DBFS mount point on all nodes of the Oracle RAC cluster and create DBFS in the tablespace:

```
# cd /mnt
# mkdir dbfs
# chown oracle:oinstall dbfs/
```

4. Create a TNS_ADMIN directory for use by the DBFS mount script on all nodes:

```
<alias name> =
  (DESCRIPTION =
    (SDU=2097152)
    (ADDRESS = (PROTOCOL = TCP)(HOST =<hostname>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <pdb service name>)
    )
  )
```

5. Change to the oracle user and create the DBFS file system:

```
#su - oracle
$cd $ORACLE_HOME/rdbms/admin
$ sqlplus dbfs_user/password@pdb
SQL> start dbfs_create_filesystem dbfstfs ogg
```

6. Verify that the LOB segment is using NOCACHE LOGGING and verify the output:

```
SQL>alter session set container=pdb;
SQL> SELECT table_name, segment_name, logging, cache FROM user_lobs WHERE
tablespace_name='DBFSTS';
```

The output should look something like this:

TABLE_NAME	SEGMENT_NAME	LOGGING	CACHE
SFSS\$_FST_11	LOB_SFSS\$_FST_11	YES	NO

If the LOB segment is not using NOCACHE LOGGING, alter it as follows:

```
SQL> ALTER TABLE DBFS.<TABLE_NAME> MODIFY LOB (FILEDATA) (NOCACHE LOGGING);
```

7. Create a symbolic (soft) link on all the nodes.

For Oracle Database release 18.x, set the library path on all the nodes by using the following commands. For other database versions, substitute the correct RDBMS ORACLE_HOME and the correct library, libnncz12.a or libnncz11.a. Run the following commands as the root user:

```
#export ORACLE_HOME=<client_home>
#ln -s $ORACLE_HOME/lib/libnncz18.a /usr/local/lib/libnncz18.a
#ln -s $ORACLE_HOME/lib/libclntsh.so /usr/local/lib/libclntsh.so
#ln -s /usr/lib64/libfuse.so.2 /usr/local/lib/libfuse.so.2
#echo /usr/local/lib >> /etc/ld.so.conf.d/usr_local_lib.conf
```

8. Verify that the mount point works on both nodes. Ensure that passwd.txt contains the password for dbfs_user.

For a VM or bare metal DB system, change the permission of /usr/bin/fusermount. For details, see [Oracle Support Document 1908868.1](#).

```
# chmod +x /usr/bin/fusermount
```

For Oracle Database version 12.2 or later, uncomment the user_allow_other parameter in the /etc/fuse.conf file. For details, see [Oracle Support Document 2388733.1](#).

Mount the DBFS by using the following command:

```
nohup $ORACLE_HOME/bin/dbfs_client dbfs_user@<alias name> -o allow_other,direct_io,failover,nolock /mnt/dbfs < ~/passwd.txt &
```

After confirming that it works, stop the dbfs_client process to register and start it again as a cluster resource in the next step.

9. Register DBFS as a cluster resource.

- A. Download the mount-dbfs.zip file that is attached to [Oracle Support Document 1054431.1](#) and place it on one database server in a temporary location (like /tmp). Then, unzip the file, which contains the mount-dbfs.sh and mount-dbfs.conf files.

B. Edit the variable settings in `mount-dbfs.conf` for your environment. Edit or confirm the settings for the following variables in the script file. Use the comments in the script to help you confirm the values for these variables.

- DBNAME: DB unique name
- MOUNT_POINT
- DBFS_USER
- ORACLE_HOME: The oracle_client home directory
- LOGGER_FACILITY: Used by syslog to log the messages and output from this script
- MOUNT_OPTIONS
- DBFS_PASSWD: Used only if WALLET=false
- DBFS_PWDFILE_BASE: Used only if WALLET=false
- WALLET: Must be true or false
- TNS_ADMIN: Used only if WALLET=true
- DBFS_LOCAL_TNSALIAS: <client_home>/network/admin/tnsnames.ora alias name for connecting to the database
- IS_PDB: Set to true if using pluggable databases (PDBs)
- PDB: PDB name, if applicable
- PDB_SERVICE: Name of the service that you created

You must create a service resource for the PDB:

```
$ srvctl add service -d <db unique name> -pdb <pdb name> -service  
pdb_service -preferred "<instance name1>,<instance_name2>"  
$ srvctl start service -d <db unique name> -pdb <pdb name>
```

If you create and use Oracle Wallet to store the password of `dbfs_user`, create a `sqlnet.ora` file that includes your wallet location in the `<client_home>/network/admin` directory. For information about creating a wallet to store the `dbfs_user` password, see [Oracle Support Document 1054431.1](#).

Sample mount-dbfs.sh Script

You can use the following script as an example for registering DBFS with Oracle Clusterware. Change the parameters as needed and copy in `/tmp/mount-dbfs.conf`.

```
### This file provides configuration for mount-dbfs.sh.  
### Note 1054431.1 provides information about the setup required to use this  
script
```

```

#####
### Everyone must set these values
#####
### Database name for the DBFS repository as used in "srvctl status database -d
$DBNAME"
### If using PDB/, this should be set to the CDB name
DBNAME=<db unique name>

### Mount point where DBFS should be mounted
MOUNT_POINT=/mnt/dbfs

### Username of the DBFS repository owner in database $DBNAME
DBFS_USER=dbfs_user

### RDBMS ORACLE_HOME directory path → Here this should be client home
ORACLE_HOME=/u01/app/oracle/product/18.0.0.0/client

### GRID HOME directory path
##GRID_HOME=/u01/app/18.1.0.0/grid

### Syslog facility name (default user)
### Changed default from local3 to user for Solaris default support on 17-FEB-
2012
### This will allow us to log messages to the syslog
### (/var/log/messages on Linux, /var/adm/messages on Solaris)
LOGGER_FACILITY=user

### mount options for dbfs_client; these are used for both wallet and non-
wallet mounting
#MOUNT_OPTIONS=allow_other,direct_io
### if tracing is required, maybe consider parameter like the example below
### following example is commented out, only uncomment if directed by Oracle
Support
### for 11.2.0.3 and earlier, fix_control=32 added per bug 13340960 to allow
async statfs response. Use trace only for troubleshooting purpose.
#MOUNT_OPTIONS=allow_other,direct_io,fix_control=32,trace_level=1,trace_file=/t
mp/dbfs_client_trace.$$log,trace_size=100
### for 11.2.0.4 and later, remove fix control. Trace file only required while
troubleshooting.
MOUNT_OPTIONS=allow_other,direct_io,failover,nolock,spool_max=64,trace_level=1,
trace_file=/tmp/dbfs_client_trace.$$log,trace_size=100
### PERL_ALARM_TIMEOUT is number of seconds to wait for response from status
command.
### After this, if no response, the script will run clean.
### NOTE: If this is longer than the clusterware check interval, bad things may
happen.
### Adjust the CHECK_INTERVAL to ensure it is at least 2x as long as
PERL_ALARM_TIMEOUT.
### Example:
### $ crsctl status res dbfs_mount -p|grep ^CHECK
### CHECK_INTERVAL=30
### $ crsctl modify res dbfs_mount -attr "CHECK_INTERVAL=32"
### $ crsctl status res dbfs_mount -p|grep ^CHECK
### CHECK_INTERVAL=32
PERL_ALARM_TIMEOUT=14

```

```

#####
### If using password-based authentication, set these
#####
### This is the plain text password for the DBFS_USER user
DBFS_PASSWD=xxxxxxx

### The file used to temporarily store the DBFS_PASSWD so dbfs_client can read
it
### This file is removed immediately after it is read by dbfs_client
### The actual filename used will have the PID appended to the name for
uniqueness
### This variable should be a full pathname including a directory and the first
part of a filename.
#`DBFS_PWDFILE_BASE=/home/oracle/passwd.txt

#####
### If using wallet-based authentication, modify these
#####
### WALLET should be true if using a wallet, otherwise, false
WALLET=false

### TNS_ADMIN is the directory containing tnsnames.ora and sqlnet.ora used by
DBFS
TNS_ADMIN=/u01/app/oracle/product/18.0.0.0/client/network/admin

### TNS alias used for mounting with wallets
DBFS_LOCAL_TNSALIAS=dest

#####
### If using PDBs, modify these
#####
### Configure the following if this is a Pluggable Database (PDB)
### IS_PDB should be set to "true" or "false"
### PDB should be a TNS alias defined per instructions in Note 1054431.1.
### PDB_SERVICE should be a cluster database service defined per
### instructions in Note 1054431.1.
IS_PDB=true
PDB=pdb
PDB_SERVICE=dbfspdb

```

- C. After editing the files as needed, copy `mount-dbfs.sh` (renaming it if necessary) to the correct directory (`GI_HOME/crs/script` or your own directory) on the database nodes and set the proper permissions on it, as the root user. Repeat these steps for `mount-dbfs.conf`, copying it to `/etc/oracle`.
- D. Change the permission/ownership of the `mount-dbfs.sh` files on all the nodes. Run the following commands as the root user:

```

# dos2unix /tmp/mount-dbfs.sh
# dos2unix /tmp/mount-dbfs.conf
# cp -f /tmp/mount-dbfs.sh /home/oracle
# chown oracle: oinstall /home/oracle/mount-dbfs.sh
# chmod 750 /home/oracle/mount-dbfs.sh
# cp -f /tmp/mount-dbfs.conf /etc/oracle

```



```
# chown oracle:oinstall /etc/oracle/mount-dbfs.conf
# chmod 640 /etc/oracle/mount-dbfs.conf
```

E. Make the following changes in the `mount-dbfs.sh` file:

```
1> #SRVCTL=$ORACLE_HOME/bin/srvctl --> Comment this variable value

2> ### set the ORACLE_SID dynamically based on OCR info, if it is
running --> Comment out below line from this section

# if RAC, set SID with following command
# if single instance, set ORACLE_SID based on below command instead
## if [ -z "$ORACLE_SID" ]; then
##     export ORACLE_SID=$(($SRVCTL config db -d $DBNAME | $GREP
'instance' | $AWK '{print $3}')
## fi
# logit info "ORACLE_SID is $ORACLE_SID"
### further checks required if we are using a PDB
### if the local instance isn't running (because ORACLE_SID is null)
don't bother
## if [ -n "$ORACLE_SID" -a "$IS_PDB" = 'true' ]; then
##     logit info "IS_PDB = true"
##     HOSTNAME=`$HN -s`
### by checking to see if the service is running on the local
instance,
### we can determine whether we should continue or not
##     PDB_SERVICE_STATE=$(($SRVCTL status service -d $CDB -s
$PDB_SERVICE | \
##         $GREP "$ORACLE_SID" )
### if the output from the last command is > 0, service is running
locally
##     if [ -n "$PDB_SERVICE_STATE" ]; then
##         logit info "PDB service is online"
##     else
##         logit error "PDB service is offline. Exiting"
##         exit 2
##     fi
## fi

3 > ### if there's no SID defined locally or it isn't running, stop -
-> Comment out following line from this section

## if [ -z "$ORACLE_SID" -a "$WALLET" = 'false' ]; then
##     logit error "No running ORACLE_SID available on this host,
exiting"
##     exit 2
## fi
```

When registering the resource with Oracle Clusterware, create it as a cluster resource rather than a local resource. If it's a cluster resource, the file system can be mounted only on a single node at a time. This prevents accidentally mounting DBFS from concurrent nodes, which creates the potential of concurrent file writes, which causes file corruption problems.

F. Add the script to register the DBFS resource as follows:

```
##### start script add-dbfs-resource.sh
#!/bin/bash
add-dbfs-resource.sh
ACTION_SCRIPT=/home/oracle/mount-dbfs.sh
RESNAME=dbfs_mount
DBNAME=<db_unique_name>
DBNAMEL=`echo $DBNAME | tr A-Z a-z`
ORACLE_HOME=/u01/app/18.0.0.0/grid
PATH=$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(ora.$DBNAMEL.db)pullup(ora.$DBNAMEL.db)', \
STOP_DEPENDENCIES='hard(ora.$DBNAMEL.db)', \
SCRIPT_TIMEOUT=300"
##### end script add-dbfs-resource.sh
```

G. Run the `add-dbfs-resource.sh` script to register DBFS with the cluster. Run the script as the oracle user on one database server:

```
$ sh ./add-dbfs-resource.sh
```

H. Start the DBFS resource:

```
$ <GRID_HOME>/bin/crsctl start resource dbfs_mount
```

Note: File locking is handled by DBFS. For details, see Step 1: Set Up a File System on Oracle RAC in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#) and [Oracle Support Document 1054431.1](#).

Configure the Databases

This section describes the configuration best practices for the source and target databases used in an Oracle GoldenGate replicated environment. Follow the configuration steps recommended in the [Oracle GoldenGate Performance Best Practices](#) white paper while performing the following steps:

- Set `alter system set enable_goldengate_replication=true`.
- Run the Oracle GoldenGate source database in ARCHIVELOG mode.
- If an Extract is running, enable FORCE LOGGING mode.
- If an Extract is running, enable minimal supplemental logging. Also, add schema-level or table-level logging for all replicated objects.
- Configure the streams pool in the System Global Area (SGA).
- Configure redo log reading performance parameters for Extract.
- Install the Oracle Streams Performance Advisor (UTL_SPADV) PL/SQL package to monitor integrated Extract and integrated Replicat performance.

Note: Redo log generation increases if you enable the FORCE LOGGING mode and supplemental logging. Doing this also increases Fast Recovery Area consumption and Archive log backup times.

You also need to create a database user for Oracle GoldenGate. For instructions, see [Establishing Oracle GoldenGate Credentials](#).

Extra Steps for Oracle Database 11.2

1. Get Patch 20448066 from [My Oracle Support](#) and apply it. For more information, see [Oracle Support Document 2348973.1](#).

As of February 2019, the patch is available for Exadata 11.2.0.4.180116 / 11.2.0.4.180116 and 11.2.0.4.180417 / 11.2.0.4.180417. If the patch level is higher, create a service request to Oracle Support. For instructions, see [Contacting Support](#).

2. Disable the database vault. This step applies only to Enterprise Edition High Performance and Enterprise Edition Extreme Performance of Oracle Database Cloud Service.

For more information, see [Oracle Support Document 1513366.1](#) and [Oracle Support Document 803948.1](#).

3. Run the script in the Oracle GoldenGate home directory.

For instructions on running the post script, see [Oracle Support Document 2304095.1](#).

Configure Oracle GoldenGate

Use the instructions in this section to install and configure Oracle GoldenGate.

Install Oracle GoldenGate

1. [Download the Oracle GoldenGate software](#) and upload it to the system.
2. Install the Oracle GoldenGate software locally on all nodes in the Oracle RAC configuration that will be part of the Oracle GoldenGate configuration. Ensure that the installation directory is the same on all nodes.

Following is an example response file for a silent installation. Use the OS oracle user as the installation user and create the installation directory (for example,

/u01/app/oracle/product/ogg181).

Run the following installation command:

```
[ssh into DB System as opc user]
sudo su - oracle
cd <directory installation media is
uploaded>/fbo_ggs_Linux_x64_services_shiphome/Disk1
./runInstaller -silent -nowait -responseFile <path of response file>
```

An example response file for Oracle Database 12c follows. The path for the response file in the installation media is <directory installation media is uploaded>/fbo_ggs_Linux_x64_services_shiphome/Disk1/response/oggcore.rsp.

Edit the `INSTALL_OPTION` and `SOFTWARE_LOCATION` parameters appropriately.

```
#####
## Copyright(c) Oracle Corporation 2018. All rights reserved.    ##
##                                                                ##
## Specify values for the variables listed below to customize    ##
## your installation.                                           ##
##                                                                ##
## Each variable is associated with a comment. The comment      ##
## can help to populate the variables with the appropriate      ##
## values.                                                       ##
##                                                                ##
## IMPORTANT NOTE: This file should be secured to have read    ##
## permission only by the oracle user or an administrator who  ##
## own this installation to protect any sensitive input values. ##
##                                                                ##
#####
#-----
-----
```

```

# Do not change the following system generated value.
#-----
-----
oracle.install.responseFileVersion=
/oracle/install/rspfmt_ogginstall_response_schema_v12_1_2

#####
#####
##
##
## Oracle GoldenGate installation option and details
##
##
##
#####
#####

#-----
-----
# Specify the installation option.
# Specify ORA18c for installing Oracle GoldenGate for Oracle Database 18c
or
#       ORA12c for installing Oracle GoldenGate for Oracle Database 12c
or
#       ORA11g for installing Oracle GoldenGate for Oracle Database 11g
#-----
-----
INSTALL_OPTION=<version>

#-----
-----
# Specify a location to install Oracle GoldenGate
#-----
-----
SOFTWARE_LOCATION=<installation location>

#####
#####
##
##
## Specify details to Create inventory for Oracle installs
##
## Required only for the first Oracle product install on a system.
##
##
##
#####
#####

```

```
#-----  
-----  
# Specify the location which holds the install inventory files.  
# This is an optional parameter if installing on  
# Windows based Operating System.  
#-----  
-----  
INVENTORY_LOCATION=/u01/app/oraInventory  
  
#-----  
-----  
# Unix group to be set for the inventory directory.  
# This parameter is not applicable if installing on  
# Windows based Operating System.  
#-----  
-----  
UNIX_GROUP_NAME=oinstall
```

Configure Oracle GoldenGate

After you install the Oracle GoldenGate software, create a deployment by using the Oracle GoldenGate Configuration Assistant (OGGCA).

Following are recommendations for creating the Oracle GoldenGate deployment:

- To create the new Service Manager, specify the following values:
 - Service Manager Deployment Home location for ACFS:
/mnt/acfs_gg/deployments/ggsm01
 - Service Manager Deployment Home location for DBFS: /mnt/dbfs/gg/oggsn
 - Specify localhost as the Listening hostname/address field. Using localhost allows the deployment to be started on all of the Oracle RAC nodes without the need for a virtual IP address (VIP).
 - Select to integrate with XAG.
- Specify the Deployment Home directory on the shared DBFS or ACFS file system:
 - ACFS: /mnt/acfs_gg/deployments/gg01
 - DBFS: /mnt/dbfs/gg/deployments

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

- ADMINISTRATOR_PASSWORD
- SERVICEMANAGER_DEPLOYMENT_HOME

- OGG_SOFTWARE_HOME
- OGG_DEPLOYMENT_HOME
- ENV_ORACLE_HOME (Specify the Oracle Client home directory that was installed earlier.)
- ENV_LD_LIBRARY_PATH (under the Oracle Client home directory)
- ENV_TNS_ADMIN (under the Oracle Client home directory)

Example response file

```
#####
## Copyright(c) Oracle Corporation 2016, 2018. All rights reserved.      ##
##                                                                    ##
## Specify values for the variables listed below to customize your      ##
## installation.                                                         ##
##                                                                    ##
## Each variable is associated with a comment. The comments can help to ##
## populate the variables with the appropriate values.                  ##
##                                                                    ##
## IMPORTANT NOTE: This file should be secured to have read permission  ##
## by the Oracle user or an administrator who owns this configuration to ##
## protect any sensitive input values.                                   ##
##                                                                    ##
#####

#-----
# Do not change the following system generated value.
#-----
oracle.install.responseFileVersion=/oracle/install/rspfmt_oggca_response_schema_
v18_1_0

#####
##                                                                    ##
## Oracle GoldenGate deployment configuration options and details        ##
##                                                                    ##
#####

#####
##                                                                    ##
## Instructions to fill out this response file                          ##
## -----                                                                    ##
## Fill out section A, B, and C for general deployment information      ##
## Additionally:                                                         ##
## Fill out sections D, E, F, G, H, I, and J for adding a deployment    ##
## Fill out section K for removing a deployment                        ##
##                                                                    ##
#####
```

```

#####
#
#           SECTION A - GENERAL           #
#
#####

#-----
# Specify the configuration option.
# Specify:
# - ADD      : for adding a new GoldenGate deployment.
# - MODIFY  : for modifying an existing GoldenGate deployment.
# - REMOVE  : for removing an existing GoldenGate deployment.
#-----
CONFIGURATION_OPTION=ADD

#-----
# Specify the name for the new or existing deployment.
#-----
DEPLOYMENT_NAME=ogg_deployment

#####
#
#           SECTION B - ADMINISTRATOR ACCOUNT           #
#
#####

#-----
# Specify the administrator account username.
#-----
ADMINISTRATOR_USER=oggadmin

#-----
# Specify the administrator account password.
#-----
ADMINISTRATOR_PASSWORD=<password for oggadmin>

#####
#
#           SECTION C - SERVICE MANAGER           #
#
#####

#-----
# Specify the location for the Service Manager deployment.
# This is only needed if the Service Manager deployment doesn't exist already.
#-----
SERVICEMANAGER_DEPLOYMENT_HOME=<location for service manager>

```



```
-----
# Specify the host for the Service Manager.
#-----
HOST_SERVICEMANAGER=localhost

#-----
# Specify the port for the Service Manager.
#-----
PORT_SERVICEMANAGER=9100

#-----
# Specify if SSL / TLS is or will be enabled for the deployment.
# Specify true if SSL / TLS is or will be enabled, false otherwise.
#-----
SECURITY_ENABLED=false

#-----
# This option is only needed when CONFIGURATION_OPTION is ADD
#
# Specify if a new Service Manager should be created.
# Specify true if a new Service Manager should be created, false otherwise.
#-----
CREATE_NEW_SERVICEMANAGER=true

#-----
# This option is only needed when CONFIGURATION_OPTION is ADD
# This option does not apply to Windows platform
#
# Specify if Service Manager should be registered as a service/daemon. This
option is mutually exclusive with the 'INTEGRATE_SERVICEMANAGER_WITH_XAG'
option.
# Specify true if Service Manager should be registered as a service, false
otherwise.
#-----
REGISTER_SERVICEMANAGER_AS_A_SERVICE=false

#-----
# This option is only needed when CONFIGURATION_OPTION is ADD
#
# Specify if Service Manager should be integrated with XAG. This option is
mutually exclusive with the 'REGISTER_SERVICEMANAGER_AS_A_SERVICE' option.
# Specify true if Service Manager should be integrated with XAG, false
otherwise.
#-----
INTEGRATE_SERVICEMANAGER_WITH_XAG=true

#-----
# This option is only needed when CONFIGURATION_OPTION is ADD
#
# If using an existing Service Manager, specify if it is integrated with XAG.
```

```

# Specify true if the existing Service Manager is integrated with XAG, false
otherwise.
#-----
EXISTING_SERVICEMANAGER_IS_XAG_ENABLED=false

#####
#
#           SECTION D - SOFTWARE HOME
#
#
#####

#-----
# Specify the existing OGG software home location.
#-----
OGG_SOFTWARE_HOME=<GoldenGate home directory>

#####
#
#           SECTION E - DEPLOYMENT DIRECTORIES
#
#
#####

#-----
# Specify the location of the new or existing OGG deployment.
#-----
OGG_DEPLOYMENT_HOME=<GoldenGate deployment home>

#-----
# Specify the location for OGG_ETC_HOME.
#-----
OGG_ETC_HOME=

#-----
# Specify the location for OGG_CONF_HOME.
#-----
OGG_CONF_HOME=

#-----
# Specify the location for OGG_SSL_HOME.
#-----
OGG_SSL_HOME=

#-----
# Specify the location for OGG_VAR_HOME.
#-----
OGG_VAR_HOME=

#-----
# Specify the location for OGG_DATA_HOME.
#-----
OGG_DATA_HOME=

```

```

#####
#
#           SECTION F - ENVIRONMENT VARIABLES           #
#
#####

#-----
# Specify the value for the ORACLE_HOME environment variable.
#-----
ENV_ORACLE_HOME=<Oracle Client home>

#-----
# Specify the value for the LD_LIBRARY_PATH environment variable.
#-----
ENV_LD_LIBRARY_PATH==<Oracle Client home>/lib

#-----
# Specify the value for the TNS_ADMIN environment variable.
#-----
ENV_TNS_ADMIN=<Oracle Client home>/network/admin

#-----
# Specify the value for the ORACLE_SID environment variable.
#-----
ENV_ORACLE_SID=

#-----
# This option is only needed when Sharding will be enabled.
# Specify the value for the STREAMS_POOL_SIZE environment variable.
#-----
ENV_STREAMS_POOL_SIZE=

#-----
# Specify any additional environment variables.
#-----
ENV_USER_VARS=

#####
#
#           SECTION G - SECURITY           #
#           This section is only needed if Security will be enabled           #
#
#####

#-----
# Specify the Cipher Suites to be used for SSL.
#-----

```

```

CIPHER_SUITES=TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_
WITH_3DES_EDE_CBC_SHA,
SSL_RSA_WITH_RC4_128_MD5,SSL_RSA_WITH_RC4_128_SHA

#-----
# If SSL / TLS will be enabled, specify the server Wallet to be copied.
#-----
SERVER_WALLET=

#-----
# If SSL / TLS will be enabled, specify the Oracle GoldenGate server certificate
# to be imported.
#-----
SERVER_CERTIFICATE=

#-----
# If the Oracle GoldenGate server certificate is encrypted, specify the key file
#-----
SERVER_CERTIFICATE_KEY_FILE=

#-----
# If the Oracle GoldenGate server certificate is encrypted, specify the key file
# password
#-----
SERVER_CERTIFICATE_KEY_FILE_PWD=

#-----
# If SSL / TLS will be enabled, specify the client Wallet to be copied.
(Optional)
#-----
CLIENT_WALLET=

#-----
# If SSL / TLS will be enabled, specify the Oracle GoldenGate client certificate
# to be imported. (Optional)
#-----
CLIENT_CERTIFICATE=

#-----
# If the Oracle GoldenGate client certificate is encrypted, specify the key file
#-----
CLIENT_CERTIFICATE_KEY_FILE=

#-----
# If the Oracle GoldenGate server certificate is encrypted, specify the key file
# password
#-----
CLIENT_CERTIFICATE_KEY_FILE_PWD=

#####

```

```

#                                                                 #
#           SECTION H - SHARDING                               #
#                                                                 #
#####

#-----
# Specify if Sharding will be enabled for this deployment.
# Specify true if Sharding will be enabled for this deployment, false otherwise.
#-----
SHARDING_ENABLED=false

#-----
# This option is only needed when Sharding will be enabled.
# Specify the Sharding user.
#-----
SHARDING_USER=

#####

#                                                                 #
#           SECTION I - SERVICES                               #
#                                                                 #
#####

#-----
# Specify the port for Administration Server.
#-----
PORT_ADMINSRVR=9101

#-----
# Specify the port for Distribution Server.
#-----
PORT_DISTSRVR=9102

#-----
# If security is disabled, specify if this non-secure deployment will be used
# to send trail data to a secure deployment.
#-----
NON_SECURE_DISTSRVR_CONNECTS_TO_SECURE_RCVRSRVR=false

#-----
# Specify the port for Receiver Server.
#-----
PORT_RCVRSRVR=9103

#-----
# Specify if Performance Metrics server will be enabled.
# Specify true if Performance Metrics server will be enabled, false otherwise.
#-----
METRICS_SERVER_ENABLED=true

```

```

#-----
# Specify if Performance Metrics server is a critical service.
# Specify true if Performance Metrics server is a critical service, false
otherwise.
# This is optional and only takes effect when Performance Metrics server will be
enabled.
# Also, this option should only be set when the Service Manager is integrated
with XAG.
# The default value is false.
#-----
METRICS_SERVER_IS_CRITICAL=true

#-----
# This option is only needed when Performance Metrics server will be enabled.
# Specify the port for Performance Metrics server (TCP).
#-----
PORT_PMSRVR=9104

#-----
# This option is only needed when Performance Metrics server will be enabled.
# Specify the port for Performance Metrics server (UDP).
#-----
UDP_PORT_PMSRVR=9105

#-----
# This option is only needed when Performance Metrics server will be enabled.
# Specify the DataStore type for Performance Metrics server.
# Valid values are: BDB, LMDB
#-----
PMSRVR_DATASTORE_TYPE=BDB

#-----
# Specify the DataStore home location for Performance Metrics server.
# This is optional and only takes effect when Performance Metrics server will be
enabled.
#-----
PMSRVR_DATASTORE_HOME=

#####
#
#           SECTION J - REPLICATION OPTIONS
#
#####

#-----
# Specify the value for the GoldenGate schema.
#-----
OGG_SCHEMA=c##ggadmin

```

```
#####
#                                                                 #
#           SECTION K - REMOVE DEPLOYMENT OPTIONS                 #
#                                                                 #
#####

#-----
# Specify if the deployment files should be removed from disk.
# Specify true if the deployment files should be removed, false otherwise.
#-----
REMOVE_DEPLOYMENT_FROM_DISK=
```

Run the following commands to perform a silent configuration:

```
[ssh into DB System as opc user]
sudo su - oracle
export OGG_HOME=/u01/app/oracle/product/ogg181
export TNS_ADMIN=$ORACLE_HOME/network/admin
export PATH=$OGG_HOME/bin:$ORACLE_HOME/bin:$PATH
export OGG_BASE_DIR=/mnt/acfs_gg
export OGG_DEPLOYMENT=$OGG_BASE_DIR/deployments/gg01
export JAVA_HOME=$OGG_HOME/jdk
oggca.sh -silent -responseFile <path of response file>
```

For more information, see "Step 4: Create an Oracle GoldenGate Deployment" in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Create a TNS Alias

To the `tnsnames.ora` file in all DB system nodes, add a TNS alias to connect to local non-CDB or CDB instances. The `tnsnames.ora` file is located in the `<client_home>/network/admin/` directory (`client_home` is the installation location of the Oracle Client that was installed earlier).

```
<Alias name> =
  (DESCRIPTION =
    (SDU=2097152)
    (ADDRESS = (PROTOCOL = TCP)(HOST =<hostname>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME =<db_unique_name>.<domain>)
    )
  )
```

If the Oracle Database release is 12c or later, you also need to create a TNS alias for the PDB. Following is an example of the TNS alias to connect to a local PDB instance:

```
<Alias name> =
  (DESCRIPTION =
    (SDU=2097152)
    (ADDRESS = (PROTOCOL = TCP)(HOST =<hostname>)(PORT = 1521))
    (CONNECT_DATA =
```

```
(SERVER = DEDICATED)
(SERVICE_NAME = oggpdbserv.<domain>)
)
```

For more information, see "Step 5: Create Oracle Net Alias for Oracle GoldenGate Database Connections" in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Configure HA for Oracle GoldenGate

To configure HA, follow the steps in "Step 6: Oracle Clusterware Configuration" in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#) and add Oracle Cloud Infrastructure specific topics as described in this section.

Install the Oracle Grid Infrastructure Standalone Agent

[Download the Oracle Grid Infrastructure Standalone Agent software](#). Oracle GoldenGate Microservices Architecture requires Oracle Grid Infrastructure Agent (XAG) version 9 or later.

The XAG software must be installed in a directory that is outside of the Oracle Grid Infrastructure home directory to avoid conflicts with software patching. You can install XAG separately on each node by using the following command. Setting the ORACLE_BASE variable avoids errors against the `agctl config` command.

```
[ssh into DB System as opc user]
sudo mkdir /u01/app/xag
sudo chown grid:oinstall /u01/app/xag
sudo su - grid
cd <xag media directory>
export ORACLE_BASE=/u01/app
./xagsetup.sh --install --directory /u01/app/xag
```

Update the PATH Variable

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

Add the following line to `/home/oracle/.bashrc` file:

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

For more information, see step 2 in the "Step 6: Oracle Clusterware Configuration" section of [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Create an Application Virtual IP Address

You need to add a floating IP address on the VCN as an application virtual IP address (VIP) for Oracle GoldenGate. A floating IP lets you access Oracle GoldenGate on each cluster node by using the same IP address. In the following example, the Oracle Cloud Infrastructure API calls are used for creating the floating IP. Each API request will be signed with an SSH private key and Oracle will use the public key to verify the authenticity of these requests. As a prerequisite, the API signing keys must be created and uploaded to the specific user who is making these requests. The [steps for setting up the API authentication](#) are provided in the documentation. This task must be performed from the client machine, not the database server.

1. Define UNIX Shell functions on the machine that will call the API.

Specifically, create the `oci-curl` function to call the request for managing the floating IP on the DB system easily. For more information, see [Request Signatures](#).

You can create a script to create the `oci-curl` function by copying the script at https://docs.cloud.oracle.com/iaas/Content/Resources/Assets/signing_sample_bash.txt and saving it as `oci-curl.sh`.

```
curl -X GET
https://docs.cloud.oracle.com/iaas/Content/Resources/Assets/signing_sample
_bash.txt >> oci-curl.sh
```

Before creating the function, edit following values:

- `tenancyId`: Tenancy OCID (Oracle-assigned unique ID called an Oracle Cloud Identifier) that can be confirmed on the web console
- `authUserId`: User OCID that can be confirmed on the web console
- `keyFingerprint`: After you add the API key on the user, you can confirm the fingerprint on the web console
- `privateKeyPath`: File path of your API private key on the client

For example:

```
$cat oci-curl.sh
.
.
    local tenancyId="ocidl.tenancy.oc1..exampleuniqueID";
    local authUserId="ocidl.user.oc1..exampleuniqueID";
    local
keyFingerprint="6c:3e:67:5c:2e:a2:dd:ea:7e:14:f8:07:f8:f7:53:03";
    local privateKeyPath="/Users/ygoto/keys/ygoto_api_key.pem";
```

Define the `oci-curl` function by sourcing the `oci-curl.sh` command:

```
$ . oci-curl.sh
or
$ source oci-curl.sh
```

Use the `typeset -F` command to ensure that the `oci-curl` function is defined in the current shell:

```
# typeset -F |grep oci-curl
declare -f      oci-curl
```

2. Query the VNIC ID of the client network.

The floating IP is attached to the client network of the system. You can find the VNIC ID by querying the instance metadata of the DB system, as shown in the following example:

```
# curl -s "http://169.254.169.254/opc/v1/vnics/0/vnicId"
ocidl.vnic.ocl.uk-london-
1.abwgiljtopawtklo3tgiqv7cekhwokpalsrww2bgv6w4mzchfbf22kqrr5vq
```

3. Create a floating IP.

- A. Create a JSON request body to create a private IP (see the [API documentation](#)). Specify an IP address that is not used anywhere in the subnet as the floating IP. Following is an example JSON request body:

```
{ "displayName": "<your own display name>", "hostnameLabel": "<your own display name >", "ipAddress": "<IP address for floating IP>", "vnicId": "<vnicId you confirmed in preceding step>" }
```

For example:

```
$cat createVip.json
{ "displayName": "ggappvip2", "hostnameLabel": "ggappvip2", "ipAddress": "10.0.1.32",
  "vnicId": "ocidl.vnic.ocl.iad.exampleuniqueID" }
```

- B. Run the following `oci-curl` command to create the floating IP:

```
oci-curl <iaas host> post ./createVip.json "/20160918/privateIps" -k
```

To confirm the IaaS host, see Core Services API in [API Reference and Endpoints](#).

For example:

```
$ oci-curl iaas.us-ashburn-1.oraclecloud.com post ./createVip.json
"/20160918/privateIps" -k
.
.
```

```
{
  "availabilityDomain" : "KDFA:US-ASHBURN-AD-2",
  "compartmentId" : "ocidl.compartment.oc1..exampleuniqueID",
  "displayName" : "ggappvip2",
  "hostnameLabel" : "ggappvip2",
  "id" : "ocidl.privateip.oc1.iad.exampleuniqueID",
  "ipAddress" : "10.0.1.32",
  "isPrimary" : false,
  "subnetId" : "ocidl.subnet.oc1.iad.exampleuniqueID",
  "timeCreated" : "2017-09-19T02:57:01.869Z",
  "vnicId" : "ocidl.vnic.oc1.iad.exampleuniqueID"
}
```

4. Confirm the list of VIPs.

To confirm the floating IP that you created, you can confirm the list of VIPs by using the following command:

```
oci-curl <iaas host> get "/20160918/privateIps?vnicId=<vnicid>" -k |grep ipAdd
```

For example:

```
$ oci-curl iaas.us-ashburn-1.oraclecloud.com get
"/20160918/privateIps?vnicId=ocidl.vnic.oc1.iad.exampleuniqueID" -k |grep
ipAdd
"ipAddress" : "10.0.1.32",
"ipAddress" : "10.0.1.27",
"ipAddress" : "10.0.1.23",
"ipAddress" : "10.0.1.25"
```

Create an Application Service

Create an application service that will be integrated with the Oracle GoldenGate service, as follows:

```
[ssh into DB system as opc user]
sudo su - oracle
srvctl add service -d $ORACLE_UNQNAME -service oggserv -role PRIMARY -preferred
<preferred instance name> -available <available instance name>
srvctl start service -d $ORACLE_UNQNAME -service oggserv
```

For Oracle Database 12c or later, create a service for the PDB.

```
[ssh into DB system as opc user]
sudo su - oracle
srvctl add service -d $ORACLE_UNQNAME -service oggpdbserv -role PRIMARY -
preferred <preferred instance name> -available <available instance name> -pdb
<pdb name>
srvctl start service -d $ORACLE_UNQNAME -service oggpdbserv
```

For more information, see step 4 in the "Step 6: Oracle Clusterware Configuration" section of [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Configure Oracle Grid Infrastructure Agent

To implement high availability of the Oracle GoldenGate service, integrate Oracle GoldenGate with Oracle Grid Infrastructure Agent (XAG) as follows:

```
[ssh into DB system as opc user]
sudo su -
cd /u01/app/xag/bin
./agctl add goldengate ggscal --gg_home <GoldenGate Home directory> --
service_manager --config_home <GoldenGate Service Manager Deployment Home
directory>/etc/conf --var_home <GoldenGate Service Manager Deployment Home
directory>/var --port 9100 --adminuser oggadmin --user oracle --group oinstall -
-network 1 --ip <floating ip address> --filesystems ora.data.acfs_gg.acfs --
db_services ora.<ORACLE_UNQNAME>.<service name>.svc --use_local_services

(You will be required to enter password of admin user)

./agctl start goldengate ggscal --node <hostname of RAC node1>
```

For more information, see step 5 in the "Step 6: Oracle Clusterware Configuration" section of [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#).

Configure Reverse Proxy and HA

Configure reverse proxy on both the database server and HA by using Nginx. Reverse proxy allows a single point of contact for various microservices associated with an Oracle GoldenGate Microservices Architecture deployment. Reverse proxy is optional, but we recommend it to ensure easy access to microservices and provide enhanced security.

Install Nginx

1. Log in to the host that has Oracle GoldenGate running and set the repository location to install Nginx by using yum:

```
[ssh into DB system as opc user]
sudo su -
cat <<EOF > /etc/yum.repos.d/nginx.repo
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/rhel/6/\$basearch/
gpgcheck=0
enabled=1
EOF
```

2. Install Nginx:

```
[ssh into DB system as opc user]
sudo su -
yum install -y nginx
```

Configure Nginx-Based Reverse Proxy

1. Create an Nginx configuration file for Oracle GoldenGate:

```
[ssh into DB system as opc user]
sudo su - oracle
export JAVA_HOME=<GoldenGate home>/jdk
<GoldenGate home>/lib/utl/reverseproxy/ReverseProxySettings -u oggadmin -P
<password of oggadmin> -o ogg.conf http://localhost:9100
```

2. Copy the Nginx configuration file. Ensure that no other files exist in the target directory.

```
[ssh into DB system as opc user]
sudo cp /home/oracle/ogg.conf /etc/nginx/conf.d/
```

3. Create the certificate:

```
[ssh into DB system as opc user]
sudo sh /etc/ssl/certs/make-dummy-cert /etc/nginx/ogg.pem
```

4. Test the Nginx configuration:

```
[ssh into DB system as opc user]
sudo nginx -t
```

5. If there are no errors, disable check config to integrate Nginx with Clusterware in a later step:

```
[ssh into DB system as opc user]
sudo chkconfig nginx off
```

6. Relocate the XAG service to Oracle RAC node 2 to set up Nginx on that node:

```
[ssh into DB system as opc user]
sudo su -
cd /u01/app/xag/bin
./agctl relocate goldengate ggscal
```

Repeat "Install Nginx" and "Configure Nginx-Based Reverse Proxy" on Oracle RAC node 2. For more information, see [Reverse Proxy Support](#) in the Oracle GoldenGate documentation.

Integrate Nginx with Clusterware and XAG

Integrate Nginx with Clusterware and XAG to implement integrated high availability with the Oracle GoldenGate service.

1. Confirm the home directory of Oracle Grid Infrastructure:

```
[ssh into DB system as opc user]
sudo su - grid
echo $ORACLE_HOME
```

2. Add Nginx as a Clusterware resource:

```
[ssh into DB system as opc user]
sudo su -
cd <Grid Infrastructure home directory>/bin
./crsctl add resource nginx -type generic_application -attr
"EXECUTABLE_NAMES=named,START_PROGRAM='/etc/rc.d/init.d/nginx
start',STOP_PROGRAM='/etc/rc.d/init.d/nginx
stop',CHECK_PROGRAMS='/etc/rc.d/init.d/nginx status'"
./crsctl setperm resource nginx -u user:oracle:r-x
```

3. Add Nginx to the XAG GoldenGate service:

```
[ssh into DB system as opc user]
sudo su -
cd /u01/app/xag/bin/
./agctl modify goldengate ggscal --filesystems ora.data.acfs_gg.acfs,nginx
--db_services ora.<ORACLE_UNQNAME>.<service name>.svc --use_local_services
./agctl stop goldengate ggscal
./agctl start goldengate ggscal --node <hostname of RAC node1>
```

You can access GoldenGate Service Manager through <https://<public IP address>>.

Configure Data Replication

Create the Oracle GoldenGate Extract, Replicat, and Path processes to configure data replication. See the following resources for details:

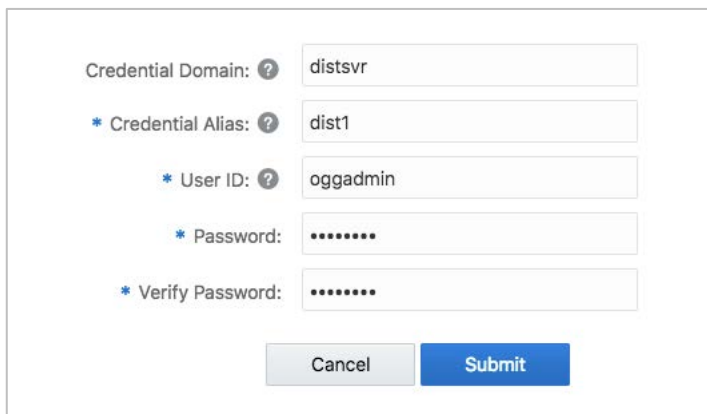
- **General configuration:** See the following topics in the Oracle GoldenGate documentation:
 - [Working with Data Replications](#)
 - [Working with Paths](#)
 - [Working with Trails](#)

- **Additional setup for HA:** See the following sections in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#):
 - Step 7: Configure Oracle GoldenGate Processes to Use the Shared File System
 - Step 8: Configure Autostart of Extract and Replicat Processes

Configure the Oracle GoldenGate Path Process with Nginx Reverse Proxy

This section shows an example of an Oracle GoldenGate Path process that transfers data to a target system using the Nginx reverse proxy.

1. Create a credential for basic authentication on the distribution server. Log in to the Administration Server and click **Configuration** in right-side menu. Add the credential with the following values:



The screenshot shows a configuration dialog box for creating a credential. It contains the following fields and values:

Credential Domain: ?	distsvr
* Credential Alias: ?	dist1
* User ID: ?	oggadmin
* Password:	*****
* Verify Password:	*****

At the bottom of the dialog are two buttons: "Cancel" and "Submit".

2. Create the Path process, entering Nginx specific settings. Log in to Distribution Server and click **[+]**. Enter values as follows, and then click **Create and Run**.

Note: Specify the floating IP on the target side.



* Path Name:

Description:

* Source:

Generated Source URI:

Reverse proxy enabled?

* Target:

Use Basic Authentication:

Generated Target URI:

Sequence Length:

Trail Size (MB):

Configure Trail Format:

Begin:

Source Sequence Number:

Source RBA Offset:

Critical:

Auto Restart:

Auto Restart Options

Retries:

Delay:

Rule-set Configuration

Filtering

► More Options



Conclusion

This white paper describes in detail the steps and best practices required to configure Oracle GoldenGate on Oracle Cloud Infrastructure. Specifically, consider the following best practices:



- Use Oracle GoldenGate release 18.1 or later with Oracle Enterprise Edition for enhanced performance.
- Open 443 port in the VCN security list and firewall on each database host for Nginx.
- Create a floating IP address by `oci-curl` for XAG setup.
- Use ACFS as the default option for setting up a file system. Use DBFS only if Oracle Data Guard is configured with the DB system.
- Follow the instructions in [Oracle GoldenGate Microservices Architecture with Oracle RAC Configuration Best Practices](#) to set up Oracle GoldenGate, a shared file system (ACFS or DBFS), and XAG.
- Use Nginx as a reverse proxy for secure deployment and better user experience.
- Integrate Nginx with Clusterware and XAG for HA.



Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0419

Oracle GoldenGate Microservices Architecture on Oracle Cloud Infrastructure
April 2019
Author: Oracle Corporation