Upgrading Oracle GoldenGate to 19c

Practice Overview

In this practice you will upgrade Oracle GoldenGate from 12c in the course practice environment to the release 19c.

Upgrade Reference

- **Documentation:** Oracle Fusion Middleware Upgrading Oracle GoldenGate, 19c (19.1.0), E98075-02, July 2019. Do not consider this practice as reference for upgrade.

Pre-requisites

- Download Oracle GoldenGate 19.1.0.0.4 for Oracle on Linux x86-64 from **edelivery.oracle.com**

  The steps are as follows:

  - On the browser go to the site **edelivery.oracle.com**
  - Sign in using an Oracle account.
  - In the search box, type “Oracle GoldenGate” then press [Enter]
  - In the top returned search result, you should see the link “**DLP: Oracle GoldenGate 19.1.0.0 (Oracle GoldenGate)**”. Click on the “**Add to Cart**” button the is close to it
  - Click on “**Check Out**” link in the top right corner.
  - The browser should display a checkbox labelled as “**Oracle GoldenGate 19.1.0.0.0**”.
  - Click on the **drop list** under the column “**Platforms/Languages**”. Select the option “**Linux x86-64**”
  - Click on “**Continue**” button
  - You should see a text box containing “**Oracle Standard Terms and Restrictions**”
  - Scroll down till the end of the window.
  - **Mark** the checkbox “**I reviewed and accept the Oracle License Agreement**”
  - Click on “**Continue**” button
  - A list of Oracle GoldenGate products will be displayed.
  - Select only the following product:

    V983658-01.zip Oracle GoldenGate 19.1.0.0.4 for Oracle on Linux x86-64

  - Click on “**Download**” button

Preparing the Environment

In this section of the practice, you will modify specific settings on the practice VirtualBox appliances. These changes are not required for the upgrade but they make the upgrade procedure easier for us.

Specifically, you will perform the following:

- Enable "Shared Clipboard" setting
- Create and configure a staging directory
- Automate database startup and shutdown
- Take snapshots for the virtual appliances

A. Enabling "Shared Clipboard" setting

1. Make sure the `ggsrv1` and `ggsrv1` appliances are turned off. If not, shut them down.

2. In the VirtualBox window, for **each appliance**, perform the following:

Click on "**Settings**" button, "**General**", "**Advanced**" tab, and set the "**Shared Clipboard**" to "**Bidirectional**", click on **OK** button.

B. Creating and configuring a staging directory

In the following steps you will create a staging directory in the hosting PC. This directory will be accessed by the VirtualBox appliances `ggsrv1` and `ggsrv2`. It will be used to exchange files between the VM appliances and the hosting PC.

3. In your hosting machine, under the disk drive letter that has the most free disk space, create a staging directory.

The code examples in the practice document assumes that the staging directory is as follows. Whenever you see in the practice steps a reference to this directory, replace it with the one that you created in your PC.

```
D:\staging\Linux
```

4. In the **VirtualBox Manager**, for **each appliance**, open the "**Settings**", click on "**Shared Folders**" link in the right-hand pane. Add shared folder by pressing "**plus**" icon. Then select path to `D:\staging\Linux`, and mark the "**Auto-mount**" box. Change the "**Folder Name**" to "**extdisk**"

5. Perform the following steps to add `oracle` user to the `vboxsf` group. This group has privilege to access the shared folder.

a. Start the appliance `ggsrv1`

b. Start Putty and login as `root` to `ggsrv1`

c. Open a terminal window and execute the following command to make sure the shared folder is seen by the appliance:

```
ls -ld /media/sf_extdisk/
```

d. Add `oracle` to `vboxsf` group.

```
usermod -a -G vboxsf oracle
```

e. Repeat the same steps for `ggsrv2`

C. Automating database startup and shutdown

In the following steps you will configure the appliances so that their databases are automatically started when you start the appliances and automatically shut down when the appliances are shut down.

Note: this procedure is applicable in our case because the Oracle Restart has not been configured. If the Oracle Restart was configured, you would have followed different procedure.

6. Make sure you login as `root` to `ggsrv1`

7. Edit the `oratab` file

```
vi /etc/oratab
```

8. Change the last field for the database line to **`Y`**

```
DB1:/u01/app/oracle/product/12.1.0/db_1:Y
```

9. Create the file `/etc/init.d/dbora` and add the following code in it:

```
#! /bin/sh
# description: Oracle auto start-stop script.
ORA_HOME=/u01/app/oracle/product/12.1.0/db_1
ORA_OWNER=oracle

case "$1" in
'start')
    # Start the Oracle databases:
    # Remove "&" if you don't want startup as a background process.
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart $ORA_HOME" &
    touch /var/lock/subsys/dbora
    ;;

'stop')
    # Stop the Oracle databases:
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut $ORA_HOME" &
    rm -f /var/lock/subsys/dbora
    ;;
esac
```

10. Change the group of the `dbora` file to `dba`, and set its permissions to 750

```
chgrp dba /etc/init.d/dbora
chmod 750 /etc/init.d/dbora
```

11. Create symbolic links to the `dbora` script in the appropriate run-level script directories

```
ln -s /etc/init.d/dbora /etc/rc.d/rc0.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
```

12. Restart `ggsrv1` and wait for a few minutes to allow the database to automatically starts up.

13. Start Putty and login as `oracle` to `srv1` and verify that the database was automatically started.

```
sqlplus / as sysdba
```

14. Repeat the same steps on `ggsrv2`

D. Taking Snapshots for the VirtualBox Appliances

15. Shutdown `ggsrv1` and `ggsrv2`

16. In Oracle VirtualBox manager, take a snapshot for each appliance.

Upgrading Oracle GoldenGate to 19c

In this section of the practice, you will implement the procedure to upgrade Oracle GoldenGate in `ggsrv1` and `ggsrv2` to the release 19c. Meanwhile, you will observe the changes on the trail files as a result of the upgrade.

In high level, the procedure will be as follows:

- Stop the GoldenGate processes in the proper way
- Check the version of the current trail files
- Take backup of the GoldenGate homes in the machines
- Extract and run the installation files in the configuration machines
- Run the 19c installer and upgrade the GoldenGate homes
- Start the processes with the required options (some of the options are mandatory)
- Check the version of the new trail files
- Test the replication configuration

Note: While a lot of administrators prefer to delete the current processes and create them again after the upgrade, in this practice you will perform the upgrade without deleting currently running processes.

Note: The upgrade procedure demonstrated in this section is applicable to the classic GoldenGate configuration. There is a slight change in the procedure for the integrated processes.

E. Pre-upgrade Procedure

In this sub-section of the practice, you will perform the actions that should be taken before upgrading the GoldenGate homes.

17. Make sure no DML or DDL is issued by the users.

Note: This step can be avoided.

Note: To make sure no change could be made by users, some administrators prefer to restart the database in restricted mode.

18. Start the current configuration processes and make sure they are running fine.

a. In `ggsrv1` and `ggsrv2`, issue `ggsci` and start the Manager process.

```
cd $GG_HOME
ggsci
ggsci> start mgr
```

b. In `ggsrv1` and `ggsrv2`, start the other processes and make sure they are all running without any lag or error.

```
ggsci> info all
ggsci> start *
ggsci> info all
```

19. In `ggsrv1` (the source), exit from `ggsci` and display the Extract trail files.

```
ls $GG_HOME/dirdat/*
```

20. Issue the `logdump` utility and open the latest trail file generated by the Extract. Replace the filename in the following code with the latest filename in your environment.

```
logdump
open /u01/app/oracle/product/ogg/dirdat/es000000001
```

21. In `logdump` command prompt, issue the `env` command and observe the version of the trail file.

The trail file version is 12.2.

Note: If you check the version of the trail files in the target system, it will be the same version.

```
env
```

22. Exit from `logdump` utility and invoke `ggsci`

23. Issue the `SEND EXTRACT` command with the `LOGEND` option until it shows there is no more redo data to capture.

```
SEND EXTRACT esrv1 LOGEND
```

24. Stop the Extract and Data Pump processes

```
STOP EXTRACT esrv1
STOP EXTRACT psrv1
```

25. In `ggsrv2`, issue the `SEND REPLICAT` command with the `STATUS` option until it shows a status of "`At EOF`" to indicate that it finished processing all of the data in the trail.

```
SEND REPLICAT rsrv2 STATUS
```

26. Stop the Replicat process

```
STOP REPLICAT rsrv2
```

27. In `ggsrv1` and `ggsrv2`, stop the Manager process.

```
STOP mgr
```

28. In `ggsrv2`, exit from `ggsci` then verify no gg `server` process is running.

```
ps -ef | grep server
```

29. In `ggsrv1`, exit from `ggsci` then take backup of the current gg home and save it in the staging directory.

```
cd $GG_HOME
cd ..

# check the total size of gg home
du -sh $GG_HOME

# make sure there is enough free disk space to backup the gg home
df -h

# backup gg home
tar -zcvf ogg12csrv1.tgz ogg
mv ogg12csrv1.tgz /media/sf_extdisk
```

30. In `ggsrv2`, take backup of the current gg home and save it in the staging directory.

```
cd $GG_HOME
cd ..
tar -zcvf ogg12csrv2.tgz ogg
mv ogg12csrv2.tgz /media/sf_extdisk
```

F. Upgrading Oracle GoldenGate 12c to 19c

In this sub-section of the practice, you will upgrade the existing GoldenGate homes from release 12c to 19c.

31. Copy the installation file `V983658-01.zip` to the staging directory.

32. In `ggsrv1`, create a directory under `oracle` home. This directory will be used for staging the installation files.

```
cd
mkdir gg_install
```

33. Decompress the installation file into the installation directory.

```
unzip /media/sf_extdisk/V983658-01.zip -d ~/gg_install
```

34. In the VirtualBox window of the machine, login as `oracle` then open a terminal window.

35. Issue the Oracle GG 19c installer.

```
cd /home/oracle/gg_install/fbo_ggs_Linux_x64_shiphome/Disk1
./runInstaller
```

36. Respond to the installer windows as follows:

Window Name	Action
Installation Option	Select the following option: - **Oracle GoldenGate for Oracle Database 12c**
Installation Details	**Software location:** enter the gg home directory `/u01/app/oracle/product/ogg` **Deselect** the Start Manager option On the confirmation message, click on **Yes** button.
Summary	Click on **Install** button

37. Close the terminal window.

38. In the `ggsrv1` Putty window, invoke SQL*Plus and login to the database as sysdba. Then run the script `ulg.sql`

Note: Do not proceed with the procedure if the script returns any error.

```
sqlplus / as sysdba
@ $GG_HOME/ulg.sql
```

39. Delete the installation directory.

```
rm -r ~/gg_install
```

40. Repeat the same upgrade steps in `ggsrv2`

By reaching to this point, GG homes are upgraded to 19c. But the processes are not running yet.

G. Post-Upgrade Procedure

In this sub-section of the practice, you will perform the actions that should be implemented after upgrading GG homes to the release 19c.

41. In `ggsrv1` and `ggsrv2`, invoke `ggsci` then start the Manager

```
cd $GG_HOME
ggsci
start mgr
```

42. In `ggsrv1`, issue the following command so that the Extract process creates a new trail file.

The `INFO` output of the command informs to modify the Extract Pump process so that it takes its input from the new file.

Note: In our case, this is required because we upgraded the Replicate process home to 19c as well. If we were in a situation where the Replicat process home is not included in the upgrade, we do not have to use the `ETROLLOVER` option. The 19c Extract is able to create 12c but you have to use the following parameter. This parameter causes Extract to write a version of the trail that is compatible with the older version of Replicat.

```
{EXTTRAIL | RMTTRAIL} file_name FORMAT RELEASE major.minor
```

```
ALTER EXTRACT esrv1, ETROLLOVER
```

43. Start the Extract process

```
start esrv1
```

44. Issue the `info` command on the Extract

```
info esrv1
```

45. Exit from `ggsci`, display the Extract trail files. Observe the new file created by the Extract and its sequence number.

```
ls $GG_HOME/dirdat/*
```

46. Issue the `logdump` utility and open the new trail file generated by the Extract. Replace the file name in the code with the file name generated in your environment.

```
logdump
open /u01/app/oracle/product/ogg/dirdat/es000000002
```

47. Issue the `env` command to verify that the new file is of the new release.

The new file is of release 19.1

```
env
```

48. Exit form `logdump` command prompt.

```
exit
```

49. Issue `ggsci`, then issue the following command. Substitute the `<seq_no>` with the new sequence number of the new trail file. This is mandatory to make the process take its input from the new file.

```
ggsci
ggsci> ALTER EXTRACT psrv1, EXTSEQNO <seq_no>, EXTRBA 0
```

50. In `ggsrv2`, list the Replicat trail files, just to observe the **latest** sequence number of the trail files **before** the upgraded Replicat process is started.

```
ls $GG_HOME/dirdat
```

51. In `ggsrv1`, start the Pump Extract. Use `info` command to verify that the process is reading from the new file.

```
start psrv1
info psrv1
```

52. In `ggsrv2`, list the Replicat trail files and observe new trail file(s) are generated. The least sequence number of the generated files should be the latest sequence number of the trail file before the upgrade plus one.

```
ls $GG_HOME/dirdat
```

53. In `ggsrv2`, use `logdump` utility to verify that the new trail file is of release 19c.

```
logdump
open /u01/app/oracle/product/ogg/dirdat/rt000000003
```

54. In `ggsrv2`, invoke ggsci then issue the following command to make the Replicat take its input file from the new file. Substitute the `<seq_no>` with the sequence number of the new version trail files.

```
ALTER REPLICAT rsrv2, EXTSEQNO <seq_no>, EXTRBA 0
```

55. Start the Replicat process and make sure it runs fine.

```
START REPLICAT rsrv2
info all
info rsrv2
```

56. In each machine, exit from the `ggsci` and examine the latest contents of the `ggserr.log` file. Check if there is any error or warning resulted from the upgrade.

```
tail ggserr.log
```

H. Test the Replication Configuration

In this sub-section of the practice, you will verify that the replication configuration works fine after the upgrade.

57. In `ggsrv1`, invoke SQL*Plus and connect to the database as `hr`

```
sqlplus hr/oracle
```

58. Issue the following SQL statement.

```
INSERT INTO JOBS (JOB_ID,JOB_TITLE,MIN_SALARY,MAX_SALARY)
VALUES('TEST_ID','TEST_JOB',0,0);
COMMIT;
```

59. In `ggsrv1`, invoke SQL*Plus and connect to the database as `hrtrg`. Verify that the update statement is replicated to the target database.

```
sqlplus hrtrg/oracle
SELECT * FROM JOBS WHERE JOB_ID='TEST_ID';
```

Note: Technically speaking, you can allow the DML activities to resume after you start the Replicat. However, it is better to resume them after you finish the whole procedure and verify that configuration is working fine.

Note: If the trigger-based DDL support is configured, more steps should be included in the procedure. For more information, refer to the reference documentation mentioned in the beginning of this practice document.

Clean up

60. If the procedure finished successfully, delete the VirtualBox snapshots created in the beginning of the document.

61. Delete the backup files created in the staging folder.

Summary

In high level, upgrading GoldenGate to 19c includes stopping the processes, upgrading the software homes, then starting the processes with specific options.