Practice

Upgrading Oracle Databases from 19c to 21c

Practice Target

In this practice, you will upgrade the Oracle CDB database 19c in `srv1` to release 21c.

In high level, you will perform the following:

- Prepare for the database upgrade in `srv1`
- Upgrade the database in `srv1` using AutoUpgrade utility
- Implement the post-upgrade actions

Assumption

The practice assumes that you have `srv1` in VirtualBox restored from its **CDB** snapshot.

Preparing for Database Upgrade in `srv1`

In this section of the practice, you will perform the preparation steps for upgrading the database in `srv1`.

1. Download Oracle database 21c for Linux x86-64. You can download it from this link or this link. The file name is `LINUX.X64_213000_db_home.zip` and its size is nearly 3.1G.

2. Copy or move the downloaded file to the sharing folder.

3. Open Putty and login to `srv1` as `root`

4. Verify that `oracle` user is a member of the `oper` group. This is a requirement by the installer, if `oper` group is there. If `oracle` is not an `oper` group member, join it to the group.

```
id oracle
usermod oracle -a -G oper
```

5. Change the current user to `oracle`, invoke SQL*Plus and login to the database as `sys`

```
su - oracle
sqlplus / as sysdba
```

6. Verify that the `REMOTE_LOGIN_PASSWORDFILE` parameter is **not** set to `SHARED` value.

If the parameter is set to `SHARED`, you need to set it to either `EXCLUSIVE` or `NONE` before starting the upgrade.

```
show parameter REMOTE_LOGIN_PASSWORDFILE
```

7. Empty Recycle Bin in all the containers.

```
PURGE DBA_RECYCLEBIN;
ALTER SESSION SET CONTAINER=PDB1;
PURGE DBA_RECYCLEBIN;
```

8. Gather statistics for the data dictionary objects and fixed tables in all the containers.

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
exec DBMS_STATS.GATHER_DICTIONARY_STATS;
exec DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
ALTER SESSION SET CONTAINER=PDB1;
exec DBMS_STATS.GATHER_DICTIONARY_STATS;
exec DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
```

9. Exit from SQL*Plus

```
exit
```

10. Create the Oracle database 21c home directory

```
mkdir -p /u01/app/oracle/product/21.0.0/db_1
```

11. Unzip the installation file into the Oracle database 21c home.

```
unzip /media/sf_staging/LINUX.X64_213000_db_home.zip -d
/u01/app/oracle/product/21.0.0/db_1
```

12. Copy the password file from the source Oracle home to the target Oracle home.

The AutoUpgrade utility does not copy the password file for you.

```
cp /u01/app/oracle/product/19.0.0/db_1/dbs/orapworadb
/u01/app/oracle/product/21.0.0/db_1/dbs/orapworadb
```

13. In Oracle VirtualBox window of `srv1`, login as `oracle` and open a terminal window.

14. Start the Oracle Universal Installer, and select a software-only installation.

Select software only option > Single Instance Database Installation > Enterprise Edition > the rest of the steps are straightforward.

Note: For the "**Prerequisite Checks**", the installer displays a warning that the clock source is set to "kvm-clock". Just click on "Ignore All" checkbox and proceed. This is the only warning that you should receive.

Note: At 68%, you will be prompted for running scripts as `root`. Click on **Yes** button.

```
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/db_1
cd /u01/app/oracle/product/21.0.0/db_1/
./runInstaller
```

15. Download the most recent version of the AutoUpgrade utility (`autoupgrade.jar`) from My Oracle Support Document 2485457.1

16. Copy the file to the staging folder.

17. Copy the file from the sharing folder to the 21c home directory.

```
mv /u01/app/oracle/product/21.0.0/db_1/rdbms/admin/autoupgrade.jar
/u01/app/oracle/product/21.0.0/db_1/rdbms/admin/autoupgrade.jar.backup
cp /media/sf_staging/autoupgrade.jar
/u01/app/oracle/product/21.0.0/db_1/rdbms/admin/autoupgrade.jar
```

18. Just to provide more free disk space in the FRA, delete the archived redo log files saved in it.

In real life scenario, do not do this action. You can take backup of the archive redo log files and then delete them.

```
rman target /
DELETE ARCHIVELOG ALL;
```

Upgrading the Database in `srv1` Using AutoUpgrade Utility

In this section of the practice, you will use AutoUpgrade utility to upgrade the CDB database in `srv1` from release 19c to 21c.

19. Create a directory to save the AutoUpgrade log files in it.

```
mkdir /home/oracle/autoupgrade/
```

20. Run the following command to create the upgrade configuration file:

```
cat > upgrade21c.cfg << EOF
global.autoupg_log_dir=/home/oracle/autoupgrade/
19c21c.source_home=/u01/app/oracle/product/19.0.0/db_1
19c21c.target_home=/u01/app/oracle/product/21.0.0/db_1
19c21c.sid=oradb
EOF
```

21. Run the AutoUpgrade utility in analyze mode.

After the job is finished, it generates two files. One in HTML format and the other one in text format.

```
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/db_1
export PATH=$PATH:$ORACLE_HOME/jdk/bin
$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config
upgrade21c.cfg -mode ANALYZE
```

22. We can monitor, manage and control the jobs from the autoupgrade console using the following commands:

`lsj` – to list the jobs

`status` – to show the job status

`tasks` – shows the tasks executing

23. In the VirtualBox of `srv1`, open the generated HTML file with FireFox browser. Click on the "**Prechecks Report**" link.

The file lists all precheck Errors, warnings and recommendations.

```
firefox /home/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html &
```

In the following steps, you will address the issues reported by the report:

24. In the Putty session, set the value `DB_RECOVERY_FILE_DEST_SIZE` to 15G.

```
source ~/.bash_profile
sqlplus / as sysdba
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=15G SCOPE=BOTH;
```

25. In `PDB1`, disable concurrent statistics gathering.

```
ALTER SESSION SET CONTAINER=PDB1;
exec DBMS_STATS.SET_GLOBAL_PREFS('CONCURRENT','OFF');
```

26. In `PDB1`, compile the invalid objects.

```
@$ORACLE_HOME/rdbms/admin/utlrp.sql
```

27. Exit from SQL*Plus

```
exit
```

28. Run the AutoUpgrade utility in Analyze mode again.

```
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/db_1
$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config
upgrade21c.cfg -mode ANALYZE
```

29. In the VirtualBox of `srv1`, open the generated HTML file with FireFox. Click on the "**Checks Report**" link.

Now the database is ready for the upgrade process.

```
firefox /home/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html &
```

Note: At this stage, in real life scenario, you should take a back up of the source database.

The AutoUpgrade utility does not touch the existing Listener. But we want the Listener to run from the 21c Oracle home. In the following steps, you will delete the existing running Listener and create a new one in the 21c home.

Note: At this stage, we expect that the upgrade process will start. We should stop application users from connecting to the database.

30. Copy the network configuration files from the existing `TNS_ADMIN` directory to their location in the 21c home directory.

Note: AutoUpgrade by default does not copy the network configuration files. However, it can be instructed to do so using the configuration file parameter `manage_network_files`. For more information, refer to the Oracle Database Upgrade Guide documenatin.

```
cp /u01/app/oracle/product/19.0.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/21.0.0/db_1/network/admin/tnsnames.ora
cp /u01/app/oracle/product/19.0.0/db_1/network/admin/sqlnet.ora
/u01/app/oracle/product/21.0.0/db_1/network/admin/sqlnet.ora
```

31. In the VirtualBox of `srv1`, run the `netca` from the 19c database and follow the assistance windows to delete the Listener.

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
/u01/app/oracle/product/19.0.0/db_1/bin/netca
```

32. Run the netca from the 21c home and create a Listener from it.

```
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/db_1
/u01/app/oracle/product/21.0.0/db_1/bin/netca
```

33. Deploy the upgrade. It normally takes longer than an hour to finish.

AutoUpgrade in deploy mode executes all the upgrade stages from the pre-upgrade analysis to post-upgrade checks.

Use the commands lsj, tasks, and status to monitor the upgrade process.

To look into what commands the utility is executing, you can look into the log files while the utility is processing.

```
$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config
upgrade21c.cfg -mode DEPLOY
```

When the upgrade is successfully finished, it should report a message containing the following:

```
Jobs finished                    [1]
```

It should also provide a name for a guaranteed restore point that should be later dropped.

34. In the VirtualBox window of the vm, open the generated report in FireFox browser.

The STATUS of all the stages must be SUCCESS.

```
firefox /home/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html &
```

35. Login to the database as SYS and submit the following queries.

```
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/db_1
cd $ORACLE_HOME/bin
./sqlplus / as sysdba
SELECT VERSION_FULL FROM V$INSTANCE;
SELECT NAME, OPEN_MODE FROM V$DATABASE;

exit
```

36. Make sure all the database services are registered in the new Listener.

```
lsnrctl service
```

37. Login as SOE to PDB1 via the Listener just to verify that tnsnames.ora file is seen by the Listener.

```
sqlplus soe/ABcd##1234@pdb1

exit
```

Implementing Post-upgrade Actions

In this section of the practice, you will implement the actions that are normally implemented after upgrading Oracle databases in production systems.

38. As instructed by the AutoUpgrade output message, drop the GRP. It is not needed anymore.

Replace the GRP name in the code with the one reported by the AutoUpgrade utility.

```
sqlplus / as sysdba
SELECT NAME FROM V$RESTORE_POINT;
Drop restore point AUTOUPGRADE_****_ORADB1916000 ;
exit
```

39. Fix the `ORACLE_HOME` variable value in the `.bash_profile` file for `oracle` user.

```
vi ~/.bash_profile

ORACLE_HOME=$ORACLE_BASE/product/21.0.0/db_1; export ORACLE_HOME
```

40. Exit from the current `oracle` shell and change the current user to `oracle` again.

```
exit
su - oracle
```

41. Verify that SQL*Plus is executed by default from the 21c Oracle home.

```
which sqlplus
```

42. Check the value of the `COMPATIBLE` parameter.

Observer that the AutoUpgrade does not update this parameter to the latest release number.

```
sqlplus / as sysdba
show parameter COMPATIBLE
```

43. Change the parameter value to the 21c release number.

In real life scenario, normally we upgrade the `COMPATIBLE` parameter only after we make sure that the upgrade was successful.

```
ALTER SYSTEM SET COMPATIBLE='21.0.0' SCOPE=SPFILE;
SHUTDOWN IMMEDIATE;
STARTUP;
show parameter COMPATIBLE
```

44. As in our case we do not need Oracle 19c home anymore, we can deinstall the 19c Oracle home.

This action is not mandatory. We assume that this home is not needed anymore and that is why we deinstall it.

```
cd /u01/app/oracle/product/19.0.0/db_1/deinstall/
./deinstall
```

45. Verify that the 19c home is removed from `/etc/oratab`

```
cat /etc/oratab
```

46. Delete the AutoUpgrade logs directory.

```
cd
rm -rf autoupgrade
```

Note: In real life scenario, we would implement the following actions after the upgrade as well:

- Take a full backup of the database along with archive logs.
- Apply the most recent RU on the new release

47. Shutdown `srv1`

48. (optional) Now you have a system with Oracle database 21c installed in it. However, the rest of the course is still based on Oracle 19c. If you are interested in keeping this Oracle database 21c vm, in Oracle VitualBox, export this vm so that you can import it later when you want it.

49. Restore `srv1` from the **CDB** snapshot so that it reverts back to the Oracle 19c database.

Note: AutoUpgrade utility has a lot of advanced options to implement the complex upgrade scenarios. In this practice, we just demonstrated using the utility to implement upgrading a single-instance standalone Oracle database.

Summary

- AutoUpgrade makes it so easy to upgrade Oracle database. It helps on executing the pre-upgrade checkups, the upgrade itself, and the post-upgrade actions.

- We cannot solely rely on the AutoUpgrade utility. Some actions must still be taken by the DBA.