Practice 15

Creating Policy-managed Oracle RAC Database

Practice Overview

In this practice, you will create a policy-managed RAC database. Specifically, you will perform the following:

- Create server pools
- Create a policy-managed CDB RAC database
- Create dynamic database services associated with server pools
- Test the node failover in the policy-managed RAC database

Practice Assumptions

- The practice assumes that you have the virtual machines `srv1` and `srv2` up and running.

Creating Server Pools

In this section of the practice, you will create two server pools.

1. Open a Putty session to `srv1`, and login as `grid`.

2. List the available servers and the server pools they belong to.

```
crsctl status server -f | egrep -w 'NAME|ACTIVE_POOLS'
```

3. Create two sever pools with using the code below.

   In the Clusterware, `srvctl` is the command you use to create server pools for databases. `crsctl` is the command you use to create server pools for other applications.

   Notice that `spool1` has higher importance than `spool2`.

```
srvctl add srvpool -serverpool spool1 -importance 1 -min 1 -max 1
srvctl add srvpool -serverpool spool2 -importance 0 -min 1 -max 1

# verify:
srvctl config srvpool
```

4. List the servers again and make sure they are assigned to the new server pools.

   Observe how the server pool names change in this output. It has `ora.` appended to it. This is automatically done by the Clusterware, every time you create server pools using the srvctl command.

   When you later use the crsctl commands to manage the database server pools, you should include the “ora.” characters when you pass the server pool names to the crsctl command. You will also observe the “ora.” characters in the command output.

   But when you use the srvctl commands to manage the database server pools, you do not include the “ora.” characters in the server pool names.

```
crsctl status server -f | egrep -w 'NAME|ACTIVE_POOLS'
```

5. List the server pools and display the servers in each pool:

   The defined server policy guarantees that each server pool has at least one server. However, it does not guarantee that `srv1` will be assigned to `spool1`. Either of the servers could be assigned to either of the server pools.

```
crsctl stat serverpool
```

Creating a Policy-managed CDB RAC Database

In this section of the practice, you will create a new Oracle Policy-managed CDB RAC database.

6. In the VirtualBox window of `srv1`, login as `oracle`.

7. Start the `dbca` utility and use it to create a new policy-managed CDB RAC database. Response to the utility windows as follows:

Utility Window	Response
Database Operations	Select **Create Database**
Creation Mode	Select **Advanced Mode**
Deployment Type	Database Type: **Oracle Real Application Cluster (RAC) Database** Configuration Type: **Policy Managed** Select the **General Purpose or Transaction Processing** template
Server Pool	Select the radio button that reads: "**Use Existing Server pool for this database**". Select the two server pools **spool1** and **spool2**.
Database Identification	Global Database Name: `rac.localdomain` **Mark "Create as Container Database**" Select **User Local Undo tablespace for PDBs** Number of PDBs: **1** PDB name: **pdb1**
Storage Options	In the Database files storage type: select **Automatic Storage Management (ASM)** as the Storage Type. Enter **+DATA/{DB_UNIQUE_NAME}** in the Database File Locations field (can be selected by clicking on the **Browse** button). Select **Oracle-Managed Files**
Fast Recovery Option	Select **Specify Fast Recovery Area** and enter `+FRA` in the Fast Recovery Area field. Storage Type: **Automatic Storage Management (ASM)** Fast Recovery Area: **+FRA** Fast Recovery Area Size: **10240** **Unmark Enable Archiving**
Oracle Vault Option	click on **Next** button

Configuration Options	Under Memory tab: SGA Size: **979** PGA Size: **327** Under Sizing tab: Processes: **500**
Management Options	**Unmark** "Run Cluster Verification Utility..." **Unmark** "Configure Enterprise Manager (EM) Database Express”
User Credentials	Select "Use the same Administrative password." Enter `oracle` as the password.
Creation Options	Select **Create Database** and click **Next**
Pre-requisites Checks	Select **Ignore All**
Summary	Click on **Finish** button
Progress Page	Click on **Close** button

8. In the Putty session, switch user to `oracle`.

9. Using the `srvctl` oracle utility, check the database configuration and its status.

Observe that the instance names have been set to `rac_1` and `rac_2`.

Observe also that the output of the second command displays the "Server pools" where the database instances are distributed. There is no guarantee to have `rac_1` running in `srv1` or `rac_2` running in `srv2`. Instances are assigned to server pools, not servers.

```
srvctl status database -db rac
srvctl config database -db rac
```

10. Display how the servers are distributed on the server pools.

```
srvctl status srvpool -detail

# the grid would use the following command instead:
crsctl stat serverpool
```

11. In the command line prompt, set `ORACLE_SID` to the running local instance and try connecting to the local instance using the OS authentication.

```
# local instance can be obtained from either of the following commands:
ps -ef | grep ora_pmon
srvctl status database -db rac

export ORACLE_SID=<local instance>
sqlplus / as sysdba
```

12. Mark out the ORACLE_SID setting in the bash_profile file of oracle user in srv1 and srv2. With the current configuration, the instance running in a server is not fixed.

```
vi .bash_profile

# in srv1 (add hash symbol in the beginning of the line):
# ORACLE_SID=rac_1; export ORACLE_SID

# in srv2 (do the same):
# ORACLE_SID=rac_2; export ORACLE_SID
```

13. Connect to pdb1 container using Easy Connect method.

```
sqlplus system/oracle@//srv1/pdb1.localdomain
```

Creating Dynamic Database Services

In this section of the practice, you will create two new dynamic database services. One will operate in the server pool `spool1` and the other will operate in the server pool `spool2`.

14. Create two uniform services for `pdb1` named `service1` and `service2`. They operate from `spool1` and `spool2` respectively. Start the services and verify they are running.

Note: the load balancing, TAF, and FAN settings that you learnt earlier in this course are all applicable with the server pool-based services.

```
srvctl add service -db rac -pdb pdb1 -service service1 -serverpool spool1 -
cardinality uniform

srvctl add service -db rac -pdb pdb1 -service service2 -serverpool spool2 -
cardinality uniform

srvctl start service -db rac -service service1
srvctl start service -db rac -service service2

srvctl config service -db rac -service service1
srvctl config service -db rac -service service2
```

15. In each server, check out which service is running.

```
lsnrctl services | grep -e service1 -e service2
```

16. Invoke SQL*Plus and connect to each service using Easy Connect method and check on which instance the session is connected in each connection.

```
sqlplus system/oracle@//srv1/service1.localdomain
SELECT INSTANCE_NAME FROM V$INSTANCE;

conn system/oracle@//srv2/service2.localdomain
SELECT INSTANCE_NAME FROM V$INSTANCE;
```

Testing Node Failover

In this section of the practice, you will simulate crashing the server running in `spool1` and examine the consequences.

17. Exit from the Putty session connected to `srv1`.

18. Invoke Putty and connect to the server contained in `spool2` as `grid`. It could be `srv1` or `srv2`.

19. Reboot the server contained in `spool1`. This reboot will be considered as a node failure by the clusterware. Do not wait for the reboot to finish, go to next step straight away.

20. In the Putty session, run the following command to display how the servers are being migrated among the server pools.

The Clusterware migrated the server from `spool2` to `spool1` because the `IMPORTANCE` attribute of the server pool `spool1` is higher than it in `spool2`. The result of this failover is having `service2` unavailable because there is no server left in its server pool.

```
crsctl stat serverpool
```

21. Wait for the reboot to complete.

22. Execute the command again in the Putty session and observe that the rebooted server is assigned to the server pool `spool2`.

23. Set the `IMPORTANCE` to zero for the server pool `spool1` so that is has now the same `IMPORTANCE` value as its value in `spool2`.

```
srvctl modify srvpool -serverpool spool1 -importance 0
```

Note: Do not shutdown the appliances. Keep them up for the next practice lecture.

Summary

Policy-managed RAC database is associated with server pools. Servers are automatically added and removed from the server pool according to the server pool policy and the service needs.