Practice 8: Creating a PDB by Relocating a PDB

Practice Overview

In this practice you will relocate `PDB1` from `CDB1` (in `srv1`) to `CDB2` (in `srv2`).

The major advantage of creating a PDB with the relocation method over all other methods is that it provides high availability. It allows you to relocated a PDB from one CDB to another with minimum downtime. You will test this advantage in this section of the practice.

Practice Assumptions

- You have the appliance `srv1` and its CDB database up and running.
- You have the appliance `srv2` up and running.

Creating a PDB by relocating a PDB

In the following sections, you will create a new PDB by relocating a remote PDB. Relocating a PDB cannot be done within the same CDB. Therefore, you will create a CDB in `srv2` and relocate `PDB1` from `CDB1` to it.

A. Creating CDB2 in srv2

1. In VirtualBox, login to `srv2` and as `oracle` user.
2. Shutdown the non-CDB database ORADB.

```
export ORACLE_SID=ORADB
sqlplus / as sysdba
shutdown immediate
```

3. Run the dbca utility.
4. Follow the screens of the dbca to create a CDB named `CDB2`. Response to the screens as follows:

Window	Response
Database Operation	• **Create Database**
Creation Mode	• **Advanced Configuration**
Deployment Type	• **General Purpose or Transaction processing**
Database Identification	• **Global Database Name:** `CDB2` • **Sid:** `CDB2` • **Mark "Create as Database Container"** • **Mark "Use Local Undo tablespaces for PDBs"** • **Create an empty Container database**
Storage Option	• **Keep the default selection "Use template file"**
Fast Recovery Option	• **Mark** "Specify the Fast Recovery Area" • Leave the options with their default settings • **Mark** "Enable Archiving"
Network Configuration	• Make sure the **LISTENER** is selected
Data Vault Option	• Make sure the check boxes are unmarked
Configuration Options	• Keep everything with their default settings.
Management Options	• Unmark the "Configure Enterprise Manager (EM) database express"
Use Credentials	• Select "User the same adminstrative password for all accounts" • Set the password (it has been set to "oracle" in my demonstrations)
Creation Option	• Make sure "Create database" is selected.
Summary	• click on **Finish**

B. Preparing for the relocation process

5. In `srv2`, make sure that `CDB1` is configured in `tnsnames.ora` file. If not, configure it as follows.

```
-- to test the configuration:
sqlplus system/oracle@CDB1

-- to add a naming descriptor of CDB1 in tnsnames.ora file:
vi $TNS_ADMIN/tnsnames.ora
CDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDB1.localdomain)
    )
  )
```

6. In `srv2`, create a database link in `CDB2` that points to `CDB1`.

```
export ORACLE_SID=CDB2
sqlplus / as sysdba
CREATE DATABASE LINK cdb1_lnk CONNECT TO system IDENTIFIED BY oracle USING
'CDB1';

SELECT SYSDATE FROM DUAL@CDB1_LNK;
```

7. In `srv1`, grant `SYSOPER` privilege to `SYSTEM` user.

`SYSTEM` user was used in the connection credential of the database link. Without this privilege, it will not be able to create a pluggable database.

```
sqlplus / as sysdba
GRANT SYSOPER TO system CONTAINER=ALL;
```

C. Relocating PDB1 from CDB1 to CDB2

In this section you will relocate PDB1 from CDB1 to CDB2 and test the availability option in this process.

8. Open a new Putty session to srv1. In this new session, connect to PDB1 as SYSTEM user.

You will use this terminal window for testing purposes. This terminal windows will be referred to in the rest of the practice as "testing window".

```
sqlplus system/oracle@pdb1
```

9. Create a new table and insert a row in it for testing purpose, as follows:

```
CREATE TABLE SYSTEM.TEST ( RID  NUMBER(2), NOTES VARCHAR2(80)) ;
INSERT  INTO SYSTEM.TEST VALUES (1,'BEFORE RELOCATING PDB1');
COMMIT;
```

10. In the testing window, verify the session is connected to srv1 and its container is CDB1:

```
SELECT SYS_CONTEXT('USERENV','HOST') CON_ID FROM DUAL;
SELECT INSTANCE_NAME FROM V$INSTANCE;
```

11. In srv2, re-connect to the CDB2 root.

```
export ORACLE_SID=CDB2
sqlplus / as sysdba
```

12. In srv2, define the OMF in the session level.

Because OMF is configured in the source database, Oracle database needs to know the location of the datafiles in the destination system. And because OMF is not configured in the target database level, you define it in the session level.

```
ALTER SESSION SET DB_CREATE_FILE_DEST='/u01/app/oracle/oradata';
```

13. Relocate PDB1 from CDB1 to CDB2. Do not wait for the command to finish. After you execute this command, go to the next step.

```
CREATE PLUGGABLE DATABASE pdb1 FROM pdb1@CDB1_LNK
RELOCATE AVAILABILITY MAX;
```

14. Insert the following row in the testing window:

```
INSERT INTO SYSTEM.TEST VALUES (2,'DURING RELOCATING PDB1');
COMMIT;
```

15. In srv1, check out the status of pdb1.

```
SELECT PDB_NAME, STATUS FROM CDB_PDBS WHERE PDB_NAME='PDB1';
```

16. Wait till the "CREATE PLUGGABLE DATABASE" command is finished.

17. Verify that the testing window is still active and connecting to `srv1`:

```
SELECT SYS_CONTEXT('USERENV','HOST') FROM dual;
```

18. In `srv2`, open `pdb1` in read/write mode

```
ALTER PLUGGABLE DATABASE pdb1 OPEN;
```

19. Check if the testing window is still active.

```
SELECT SYSDATE FROM DUAL;
```

20. In `srv2`, check the contents of the testing table.

The data inserted before and during the relocation is there.

```
ALTER SESSION SET CONTAINER=PDB1;
SELECT * FROM SYSTEM.TEST;
```

21. In the testing window in `srv1`, make a new connection to `pdb1`.

The connection should be successful, even though it has been made in `srv1`.

```
sqlplus system/oracle@pdb1
```

22. Check the name of the instance the testing window is connected to.

Even though the connection was made through the listener in `srv1`, the client session is actually made to `srv2`.

```
SELECT SYS_CONTEXT('USERENV','INSTANCE_NAME') FROM DUAL;
```

23. Check out the status of `PDB1` in `srv1` and in `srv2`.

```
SELECT PDB_NAME, STATUS FROM CDB_PDBS WHERE PDB_NAME='PDB1';
```

24. In `srv1`, check the services in the listener.

Observe that the listener is forwarding the connection it receives for the service `pdb1` to the machine `srv2`.

```
lsnrctl services
```

Note

Do not drop `CDB2` at this stage. You will use it in the next practice.

Summary

Relocating a PDB allows you to move a PDB from one CDB to another CDB with minimum down time. When the `AVAILABILITY MAX` clause is used, the relocate operation configures the listener to relocate connections to the new location.