Practice 9: Creating a Proxy PDB

Practice Overview

In this practice you will create a proxy PDB. You will then perform basic testing on it.

Practice Assumptions

- You have the CDB1 (in srv1) and CDB2 (in srv2) databases up and running.

Creating a Proxy PDB

In the following steps, you will create a proxy PDB called `PDB2PXY` in the machine `srv2`. It points to `PDB2` in `srv1` machine.

1. Connect to the root of `CDB2` database as `SYSDBA`.

2. Verify that the database link to `CDB1` is valid and working. If it is not there, create one.

```
export ORACLE_SID=CDB2
sqlplus / as sysdba
SELECT SYSDATE FROM DUAL@CDB1_LNK;
-- if it isn’t there, create the database link as follows:
CREATE DATABASE LINK cdb1_lnk CONNECT TO system IDENTIFIED BY oracle USING
'CDB1';
```

3. In `srv2`, define the OMF at the session level.

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

4. Create the proxy PDB and then open it.

```
CREATE PLUGGABLE DATABASE pdb2pxy AS PROXY FROM pdb2@cdb1_lnk;
ALTER PLUGGABLE DATABASE pdb2pxy OPEN;
```

5. Verify that the new PDB is a proxy PDB.

```
SELECT PDB_NAME, CON_ID, IS_PROXY_PDB FROM CDB_PDBS WHERE PDB_NAME='PDB2PXY';
```

6. In `srv2`, switch the current container to the new proxy PDB.

```
ALTER SESSION SET CONTAINER= pdb2pxy;
```

7. Display the current container name.

```
SHOW CON_NAME
```

   The command should return the following errors:

```
ORA-01017: invalid username/password; logon denied
ORA-02063: preceding line from PROXYPDB$DBLINK
```

   Those errors are returned when you connect to the root using OS authentication and then switch to a proxy PDB. If you connect to the root using the service login authentication, you should be alright. It is not an expected behavior. Let’s try the workaround.

8. Connect to the root using its service name and try re-executing the same statement.

   The proxy PDB is seen by the users who are connected to it as `PDB2`.

```
conn sys/oracle@//srv2:1521/CDB2 as sysdba
ALTER SESSION SET CONTAINER= pdb2pxy;
SHOW CON_NAME
```

9. Retrieve list of the datafiles of the proxy PDB.

Observe that the proxy PDB has copies of the SYSTEM, SYSAUX and UNDO tablespace datafiles. The other tablespaces will remain in the referenced PDB.

Do not use the `CDB_DATA_FILES` view to obtain the required list. It reports the datafiles in the referenced PDB.

The code below assumes that the CON_ID of PDB2 is 4. Change it accordingly, if this is not true in your case.

```
conn sys/oracle@//srv2:1521/CDB2 as sysdba

SELECT T.NAME TABLESPACE_NAME, D.NAME
FROM V$TABLESPACE T, V$DATAFILE D
WHERE T.TS# = D.TS# AND T.CON_ID=4 AND D.CON_ID=4;
```

Testing the Proxy PDB functionality

In the following steps, you will test the proxy PDB that you created in the previous section. You will create a user in the PDB, insert data into a testing table from the referenced PDB, insert data into the same table from the proxy PDB, and examine the result.

10. In `srv1`, login as SYSDBA to `PDB2` and create a testing user in it. Grant create session and create table privileges to the user.

```
sqlplus / as sysdba
ALTER SESSION SET CONTAINER=PDB2;

CREATE USER tuser IDENTIFIED BY oracle
 DEFAULT TABLESPACE users
 QUOTA UNLIMITED ON users;

GRANT CREATE SESSION, CREATE TABLE TO tuser;
```

11. Connect to `PDB2` using the newly created user, create a test table, and insert a row in it.

```
conn tuser/oracle@//srv1:1521/pdb2.localdomain

CREATE TABLE tb1 (id NUMBER, notes varchar2(20));
INSERT INTO tb1 VALUES (1,'from PDB2');
COMMIT;
```

12. Connect to `PDB2PXY` and insert a row into the testing table.

Note: connecting to proxy PDB using the local user `tuser` will return the following error:

```
ORA-01017: invalid username/password; logon denied
```

This is not the expected behavior and I believe it is a bug. The workaround is to log as `SYSDBA` using password authentication and then switch the current container to `PDB2PXY`, as shown in the following block code.

```
CONN sys/oracle@//srv2:1521/CDB2 as sysdba
ALTER SESSION SET CONTAINER= pdb2pxy;

INSERT INTO tuser.tb1 VALUES (2,'from pdb2pxy');
COMMIT;
```

13. Query the testing table in both PDBs:

```
SHOW CON_NAME
SELECT * FROM TUSER.TB1;

conn tuser/oracle@//srv1:1521/pdb2.localdomain
SELECT * FROM tb1;
```

Note

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

Summary

- When users connected to a proxy PDB make data manipulation, the changes will actually affect the referenced PDB.
- Physically, the proxy PDB has copies of the datafiles of the `SYSTEM`, `SYSAUX`, and `UNDO` tablespaces. They will automatically get synchronized with the referenced PDB.
- When you want to switch the current container to a proxy PDB, first log in to the root using username/password credential rather than using the OS authentication.