Step-by-Step Lab: Creating and Managing Pluggable Databases (PDBs) within fenagoCDB

Prerequisites

- Ensure that the `fenagoCDB` CDB is created and running.
- Set the `ORACLE_HOME` and `ORACLE_SID` environment variables appropriately.

Steps

1. Set Environment Variables

```
export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
export ORACLE_SID=fenagoCDB
```

2. Connect to SQL*Plus as SYSDBA

```
sqlplus / as sysdba
```

3. Create the First PDB

a. Create the PDB using `DBCA`

```
CREATE PLUGGABLE DATABASE pdb1 ADMIN USER pdb1admin IDENTIFIED BY password
FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/fenagoCDB/pdbseed/',
'/u01/app/oracle/oradata/fenagoCDB/pdb1/');
```

b. Open the PDB

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

4. Create the Second PDB

a. Create the PDB using `DBCA`

```
CREATE PLUGGABLE DATABASE pdb2 ADMIN USER pdb2admin IDENTIFIED BY password
FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/fenagoCDB/pdbseed/',
'/u01/app/oracle/oradata/fenagoCDB/pdb2/');
```

b. Open the PDB

```
ALTER PLUGGABLE DATABASE pdb2 OPEN;
```

5. Verify the PDB Creation

```
SELECT pdb_name, status FROM CDB_PDBS;
```

6. Connect to Each PDB

a. Connect to `pdb1`

```
ALTER SESSION SET CONTAINER = pdb1;
```

b. Connect to `pdb2`

```
ALTER SESSION SET CONTAINER = pdb2;
```

7. Monitor the PDBs

a. Check PDBs Status

```
SELECT pdb_name, status FROM CDB_PDBS;
```

b. Monitor PDB Resource Usage

```
SELECT name, open_mode, restricted FROM V$PDBS;
```

8. Close the PDBs

a. Close `pdb1`

```
ALTER PLUGGABLE DATABASE pdb1 CLOSE;
```

b. Close `pdb2`

```
ALTER PLUGGABLE DATABASE pdb2 CLOSE;
```

9. Delete the PDBs (Optional)

a. Drop `pdb1`

```
DROP PLUGGABLE DATABASE pdb1 INCLUDING DATAFILES;
```

b. Drop `pdb2`

```
DROP PLUGGABLE DATABASE pdb2 INCLUDING DATAFILES;
```

Conclusion

You have successfully created, opened, monitored, and optionally deleted two Pluggable Databases (PDBs) within the `fenagoCDB` Container Database (CDB). These steps include setting environment variables, connecting to SQL*Plus, creating PDBs, and verifying their status.

This guide provides a comprehensive approach to managing PDBs in Oracle, ensuring that you can handle both typical and exceptional scenarios effectively.