Practice 12: Managing PDB Lockdown Profiles

Practice Overview

In this practice you will use the PDB lockdown profile to:

- Disable a database option in a PDB
- Force a policy on using the `ALTER SYSTEM` statement in a PDB

Practice Assumptions

- `CDB1` (in `srv1`) database is up and running.

Managing PDB Lockdown Profiles

A. Using PDB Lockdown Profile to disable a database option

In the following steps, you will use the PDB lockdown profile to disable the partitioning database option.

1. Login to the root as `sysdba` and create a lockdown profile.

The `PRIVATE_DBAAS`, `PUBLIC_DBAAS` and `SAAS` lockdown profiles are empty profiles created automatically when the database is created.

Lockdown profiles can only be created in the CDB root.

```
sqlplus / as sysdba

CREATE LOCKDOWN PROFILE LD_PROFILE1;

col profile_name format a15
SELECT PROFILE_NAME FROM DBA_LOCKDOWN_PROFILES;
```

2. Disable the partitioning option in the created profile.

```
ALTER LOCKDOWN PROFILE LD_PROFILE1 DISABLE OPTION = ('PARTITIONING');

col rule format a15
col clause format a10
SELECT PROFILE_NAME, RULE, CLAUSE, STATUS
FROM DBA_LOCKDOWN_PROFILES
WHERE PROFILE_NAME='LD_PROFILE1';
```

3. Apply the lockdown profile to `PDB2`.

```
ALTER SESSION SET CONTAINER=PDB2;

ALTER SYSTEM SET PDB_LOCKDOWN=LD_PROFILE1;
```

4. Connect to `PDB2` and try creating a partitioned table.

As a result of applying the lockdown profile, you should receive the following error:

`ORA-00439: feature not enabled: Partitioning`

```
CONN SYSTEM/oracle@PDB2

CREATE TABLE PTABLE(RID NUMBER ) PARTITION BY HASH (RID)  PARTITIONS 2;
```

B. Using PDB Lockdown Profile to disable specific SQL Statement

In the following steps, you will use the PDB lockdown profile to disable a specific SQL statement in PDB2.

5. Verify that you can change the value of OPTIMIZER_DYNAMIC_SAMPLING parameter in PDB2.

```
CONN / as sysdba

ALTER SESSION SET CONTAINER=PDB2;
ALTER SYSTEM SET OPTIMIZER_DYNAMIC_SAMPLING=3;
```

6. Modify the lockdown profile so that it allows the users to issue ALTER SYSTEM statement only for killing the session. The other ALTER SYSTEM clauses are **not** allowed.

```
CONN / as sysdba

ALTER LOCKDOWN PROFILE LD_PROFILE1 DISABLE STATEMENT= ('ALTER SYSTEM') CLAUSE ALL
EXCEPT = ('KILL SESSION');

col profile_name format a15
col rule format a15
col clause format a12

SELECT PROFILE_NAME, RULE, CLAUSE, STATUS FROM DBA_LOCKDOWN_PROFILES
WHERE PROFILE_NAME='LD_PROFILE1';
```

7. Login to PDB2 and try to use the ALTER SYSTEM statement to change the parameter value again.

   You should receive an error because the lockdown profile disallows using the ALTER SYSTEM statement for anything other than killing the sessions.

```
ALTER SESSION SET CONTAINER=PDB2;
ALTER SYSTEM SET OPTIMIZER_DYNAMIC_SAMPLING=4;
```

8. Cleanup steps

```
CONN / AS sysdba
DROP LOCKDOWN PROFILE LD_PROFILE1;

ALTER SESSION SET CONTAINER=PDB2;
SHOW PARAMETER PDB_LOCKDOWN
ALTER SYSTEM SET PDB_LOCKDOWN='';
SHOW PARAMETER PDB_LOCKDOWN
```

Summary

Lockdown profile provides a layer of security that allows you to control the available options and statements to execute in the PDBs.

In this practice, you learnt how to use the lockdown profiles to disable a database option and to force specific usage of the `ALTER SYSTEM` statement in a PDB.