Practice 22 Managing a PDB Snapshot Carousel

Practice Overview

In this practice, you will configure and test a PDB Snapshot Carousel on `PDB1`.

Specifically, you will perform the following:

- Enable PDB Snapshot Carousel on `PDB1`
- Create a PDB Snapshot and display information about it
- Study the PDB Snapshot Files and Their Location
- Create a PDB from PDB Snapshots
- Utilize Sparse Files in PDB Snapshot Carousel

Assumption

This practice assumes that you have `srv1` and its database up and running.

A. Enabling PDB Snapshot Carousel on PDB1

In this section of the practice, you will configure PDB Snapshot Carousel on `PDB1`.

1. In Oracle VirtualBox, take a snapshot for srv1.

Caution: Do not proceed with the practice before taking a snapshot for the `srv1` first.

2. Open Putty and connect to `srv1` as `oracle`

3. Run the following code to create a SQL script file named as `display_pdb_snapshots.sql`. The script displays information about the existing PDB snapshots.

```
cat >display_pdb_snapshots.sql <<EOF
set linesize 150
col CON_ID FOR 99
col CON_NAME FOR a10
col SNAPSHOT_NAME FOR a25
col SNAP_SCN FOR 9999999
col FULL_SNAPSHOT_PATH FOR A50
SELECT CON_ID, CON_NAME, SNAPSHOT_NAME,SNAPSHOT_SCN AS snap_scn,
FULL_SNAPSHOT_PATH FROM DBA_PDB_SNAPSHOTS ORDER BY SNAP_SCN;
EOF
```

4. Run the following code to create a SQL script file named as `display_max_pdb_snapshots.sql`. The script displays the value of the property `MAX_PDB_SNAPSHOTS` for the current container.

```
cat > display_max_pdb_snapshots.sql <<EOF
set linesize 150
col ID for 99
col PROPERTY_NAME for a17
col PDB_NAME for a9
col VALUE for a3
col DESCRIPTION for a43
SELECT r.CON_ID AS id, p.PDB_NAME, PROPERTY_NAME,
PROPERTY_VALUE AS value, DESCRIPTION
FROM CDB_PROPERTIES r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID AND PROPERTY_NAME='MAX_PDB_SNAPSHOTS'
ORDER BY PROPERTY_NAME;
EOF
```

5. Start SQL*Plus and login to the CDB root as `sysdba`

```
sqlplus / as sysdba
```

6. Display the value of the `MAX_PDB_SNAPSHOTS` property for `PDB1`.

The default value of the property is 8.

Note: To change this property for a PDB, the PDB must be open in read/write mode.

```
-- to display the PBD property, we should switch to it first:
ALTER SESSION SET CONTAINER=PDB1;
@display_max_pdb_snapshots
```

7. Display the default Snapshot Carousel setting for `PDB1`.

By default, the Snapshot mode is `MANUAL`. This means we can manually create a PDB snapshot.

If we want to disable the ability to create a snapshot, we could issue the following statement:

```
ALTER PLUGGABLE DATABASE SNAPSHOT MODE NONE;
```

```
col PDB_NAME for a10
col SNAPSHOT_MODE for a15
SELECT P.CON_ID, P.PDB_NAME, P.SNAPSHOT_MODE, P.SNAPSHOT_INTERVAL
FROM CDB_PDBS P ORDER BY 1;
```

8. For `PDB1`, change the setting of the PDB Snapshot mode so that a snapshot is automatically created every 24 hours.

The command displays the following error:

```
ORA-12754: Feature PDB SNAPSHOT CAROUSEL is disabled due to missing
capability.
```

```
ALTER PLUGGABLE DATABASE SNAPSHOT MODE EVERY 24 HOURS;
```

To handle the returned error, we need to enable the undocumented parameter "`_exadata_feature_on`". Enabling this parameter requires restarting the instance.

9. Enable the undocumented parameter "`_exadata_feature_on`" and restart the instance.

```
ALTER SESSION SET CONTAINER=cdb$root;
ALTER SYSTEM SET "_exadata_feature_on"=TRUE SCOPE=SPFILE;

shutdown immediate
startup
```

10. Change the setting of the `PDB1` Snapshot mode so that a snapshot is automatically created every 24 hours.

```
ALTER SESSION SET CONTAINER=PDB1;
ALTER PLUGGABLE DATABASE SNAPSHOT MODE EVERY 24 HOURS;
```

11. Verify that the change is applied.

Observe that the view provides the `SNAPSHOT_MODE` and the `SNAPSHOT_INTERVAL`. But it does not tell at what specific time the snapshot is schedule to be taken.

```
SELECT P.CON_ID, P.PDB_NAME, P.SNAPSHOT_MODE, P.SNAPSHOT_INTERVAL
FROM CDB_PDBS P ORDER BY 1;
```

12. Retrieve information about the Scheduler job that kicks off creating a `PDB1` Snapshot.

The query provides more precise information about the Scheduler job that creates the automatic `PDB1` snapshot.

I am not sure if this is supported or not but, technically, we can modify that job to precisely control the job, including setting exactly when it should start. Just be cautious. Changing the Snapshot Mode resets or even deletes this scheduler job.

Note: Personally, if I need to precisely control the PDB Snapshots creation process, I would set the Snapshot Mode to `MANUAL` and manually create a Scheduler job to automate the snapshot creation the way that I want. I might take the code of the automatic Scheduler job to help me on creating the manual Scheduler job.

```
conn / as sysdba

set linesize 200
col OWNER for a10
col JOB_NAME for a30
col REPEAT_INTERVAL for a50

ALTER SESSION SET NLS_DATE_FORMAT='DD-MM-YY HH24:MI:SS';
SELECT OWNER, JOB_NAME, REPEAT_INTERVAL, START_DATE, NEXT_RUN_DATE, JOB_ACTION
FROM DBA_SCHEDULER_JOBS
WHERE JOB_NAME LIKE '%SNAPSHOT'
ORDER BY OWNER, JOB_NAME;
```

B. Creating a PDB Snapshot and Display Information about it

In the following steps, you will query the required data dictionary views to retrieve information about the Snapshot Carousel on `PDB1`.

13. Run the script `display_pdb_snapshots.sql` to display the existing `PDB1` snapshots in the database, if there is any.

You have not created an `PDB1` snapshot, however, the query reports that one PDB snapshot is there. It looks like that enabling the automatic PDB Snapshot Carousel creates a PDB snapshot.

```
@display_pdb_snapshots
```

14. Verify that the CDB is running in local undo mode.

```
col PROPERTY_VALUE for a20
SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES
WHERE PROPERTY_NAME='LOCAL_UNDO_ENABLED';
```

15. Issue the following commands to manually create an additional `PDB1` snapshot.

```
ALTER SESSION SET CONTAINER=PDB1;
set timing on
ALTER PLUGGABLE DATABASE SNAPSHOT;
```

C. Studying the PDB Snapshot Files and Their Location

In the following steps, you will examine the PDB snapshot files and reach to some facts about them.

16. Display list of the PDB snapshots. Observe the following about the output:

- The automatic created Snapshot names have the format `SNAP_<some ID>_<SCN ID>`, where SCN is the SCN at which the snapshot was taken. If you want to assign a snapshot name of your own at the time of creating a snapshot, use the following format of the statement:

  ```
  ALTER PLUGGABLE DATABASE SNAPSHOT <snapshot name>;
  ```

- Snapshot files are saved into a single archive `pdb` file. As you will see later in this practice, this behavior will change when sparse files are generated.

- All the snapshot files are saved in the following directory:

  ```
  $ORACLE_BASE/oradata/DB_UNIQUE_NAME
  ```

  This is the directory where the dbca creates the database data files in it. Observe that OMF is not enabled in our case. To verify that, retrieve the OMF parameter value by issuing the following command:

  ```
  show parameter DB_CREATE_FILE_DEST
  ```

```
@display_pdb_snapshots
```

In real life scenario, we might be interested in saving PDB snapshot files in a location different from the default location. Unfortunately, the command that creates PDB snapshots does not provide the option to provide the PDB snapshot files location.

The solution that I know is to set the `DB_CREATE_FILE_DEST` at the database level or to set it at the session level before issuing the command. Let's try this solution.

17. Create a directory in the sharing folder and create a `PDB1` snapshot in it.

```
set timing on
host mkdir -p /media/sf_staging/pdb1/snapshots
ALTER SESSION SET CONTAINER=PDB1;
ALTER SESSION SET DB_CREATE_FILE_DEST='/media/sf_staging/pdb1/snapshots';
ALTER PLUGGABLE DATABASE SNAPSHOT;
```

18. Display list of the PDB snapshots.

Observe the newly created snapshot is created in the sharing folder.

```
@display_pdb_snapshots
```

19. Check out the size of the generated PDB snapshot files.

Observe that the PDB snapshot files have the same full data size of `PDB1`.

```
host ls -alsh $ORACLE_BASE/oradata/ORADB/pdb1/snap_*
host ls -alsh /media/sf_staging/pdb1/snapshots
```

20. Query the view `DBA_PDB_SNAPSHOTFILE`.

The view is supposed to retrieve the PDB Snapshot files but this view retrieves snapshot files that are saved as sparse files. You will make the database save PDB snapshot files as sparse files later in this practice.

```
SELECT SNAPSHOT_FILENAME FROM DBA_PDB_SNAPSHOTFILE ;
```

D. Creating a PDB from PDB Snapshots

In the following steps, you will create a PDB from `PDB1` snapshots.

21. Create a directory to save the new PDB datafiles in it.

```
host mkdir /u01/app/oracle/oradata/ORADB/pdb1_copy
```

22. Issue the following statement to create a PDB from a PDB snapshot. Substitute the `<SNAP_Name>` with a name of a snapshot from the `PDB1` snapshots.

```
conn / as sysdba
CREATE PLUGGABLE DATABASE pdb1_copy FROM pdb1 USING SNAPSHOT <SNAP_Name> ;
```

The statement returns the following error:

```
ORA-65016: FILE_NAME_CONVERT must be specified
```

The statement fails because OMF is disabled and it does not know where to create the new PDB data files. However, unfortunately, the returned error is misleading. Setting the `FILE_NAME_CONVERT` will not resolve the error. Because OMF is disabled at the database level, the issue can be resolved by setting the `DB_CREATE_FILE_DEST` at the session level.

23. Issue the following statement to create a PDB from a PDB snapshot.

```
ALTER SESSION SET DB_CREATE_FILE_DEST='/u01/app/oracle/oradata/ORADB/pdb1_copy';
CREATE PLUGGABLE DATABASE pdb1_copy FROM pdb1 USING SNAPSHOT <SNAP_NAME> ;
```


24. Open the new PDB and retrieve some data for testing purpose.

```
ALTER PLUGGABLE DATABASE pdb1_copy OPEN;
ALTER SESSION SET CONTAINER = pdb1_copy;
SELECT COUNT(*) FROM HR.EMPLOYEES;
```

25. As a cleanup, drop the pluggable database `pdb1_copy`

```
conn / as sysdba
ALTER PLUGGABLE DATABASE pdb1_copy CLOSE;
DROP PLUGGABLE DATABASE pdb1_copy INCLUDING DATAFILES;
```

26. Drop the PDB snapshots.

```
ALTER SESSION SET CONTAINER = pdb1;
ALTER PLUGGABLE DATABASE pdb1 SET MAX_PDB_SNAPSHOTS=0;

-- verify:
@display_pdb_snapshots

-- set MAX_PDB_SNAPSHOTS back to some value:
ALTER PLUGGABLE DATABASE pdb1 SET MAX_PDB_SNAPSHOTS=4;
```

E. Utilizing Sparse Files in PDB Snapshot Carousel

In the following steps, you will make the database save the PDB Snapshot Carousel files as sparse files to save disk space taken up by snapshot files.

Note: Refer to the concepts lecture to learn about the recommendations on using this feature.

27. Enable the database initialization parameter `CLONEDB` then restart the database instance.

Caution: Be cautious about setting this parameter in production systems. This parameter does not only affect the PDB Snapshot files. It affects the datafiles as well.

```
conn / as sysdba
ALTER SYSTEM SET CLONEDB=TRUE SCOPE=SPFILE;
shutdown immediate
startup
```

28. Create a snapshot for `PDB1`.

Observe that the snapshot was created in a few seconds.

```
set timing on
ALTER SESSION SET CONTAINER=PDB1;
ALTER PLUGGABLE DATABASE SNAPSHOT;
```

29. Display information about the created snapshot.

Observe that the snapshot files are saved in a directory this time and not in an archive file.

In addition to that, the parent directory that hosts the PDB snapshot files looks different from the one used to save the snapshot archive file before enabling the sparse files option. It is under the following directory structure this time:

```
$ORACLE_HOME/dbs/snapshots/pdb_<some-pdb-id>/<snap-id>/
```

```
@display_pdb_snapshots
```

30. Display the contents of the snapshot directory.

Observe the following from the output:

- The directory does not actually host the snapshot files themselves (except the xml file). It has soft links to the snapshot files.
- The snapshot datafiles are saved in the following directory:

  `$ORACLE_BASE/oradata/ORADB/pdb1/<snap-id>/`
- The snapshot archive files are saved in a single file in the FRA directory

```
host ls -lsh $ORACLE_HOME/dbs/snapshots/<pdb-id>/<snap-id>/
```

Note: An alternative way to retrieve the disk consumption space of files is to use the following command:

```
host du -ka $ORACLE_HOME/dbs/snapshots/<pdb-id>/<snap-id>/
```

31. Display the contents of the directory that hosts the snapshot datafiles.

From the command output, compare between the file sizes in the first column and in the sixth column. They are the same. The first PDB-managed PDB snapshot is full.

```
host ls -lsh $ORACLE_BASE/oradata/ORADB/pdb1/<snap-id>/
```

32. Take another PDB snapshot.

```
ALTER PLUGGABLE DATABASE SNAPSHOT;
@display_pdb_snapshots
```

33. Display the contents of the directory that hosts the snapshot datafiles. Substitute the `<snap-id>` with its id returned from the preceding command output.

From the command output, compare between the file sizes in the first column and in the sixth column. The space actually taken from the disk (sizes in the first column) is a fraction of the total file size.

```
host ls -lsh $ORACLE_BASE/oradata/ORADB/pdb1/<snap-id>/
```

34. Create a PDB from the second snapshot.

In theory this command should succeed. But in our testing case, the command returns an internal error. To best of my knowledge, at the time of this writing, Oracle development is working on a fix to this issue and I expect a patch will be issued to fix it.

```
conn / as sysdba
ALTER SESSION SET DB_CREATE_FILE_DEST='/u01/app/oracle/oradata/ORADB/pdb1_copy';
CREATE PLUGGABLE DATABASE pdb1_copy FROM pdb1 USING SNAPSHOT <SNAP_NAME>;
```


Update on the Returned Internal Error

Oracle support reported that the returned internal error is due to the following bug:

```
BUG 28968211 - SR20.1 HCACHE_RECKEEPABTCIM - TRC - KPDBOPENARCHIVE ...
```

It suggested to apply the Patch 28968211 which is available for 19.8 DB RU on top of database home. I applied the patch but when I tried the step again, I was hit by the following error:

```
ORA-65227: unable to open pdb archive file …
```

At the time of this writing, I am still in contact with Oracle support on resolving the second issue. Will keep this updated.

Suggested Extra Practice

Optionally, if you want to perform extra practice on PDB Snapshot Carousel topic, consider performing the following:

- Create a refreshable PDB clone from PDB1 and then try to configure PDB Snapshot Carousel on it. You should not be able to configure it because the default value of `MAX_PDB_SNAPSHOTS` in the refreshable PDB clones is zero and we cannot change this property for them.

Cleanup

35. Shutdown `srv1` and restore it to the snapshot taken in the beginning of the practice.

36. Delete the snapshot taken in the beginning of the practice.

Summary

In this practice, we reached to the following conclusions:

- We can easily configure a PDB Snapshot Carousel for a PDB but unfortunately it requires to enable the undocumented parameter "`_exadata_feature_on`". Enabling this parameter requires restarting the database instance.
- For non-sparse PDB Snapshot Carousel, we can set its file location by defining the OMF (at the system or session levels).
- We can create full PDB from PDB Snapshot Carousel.
- We need to set the initialization parameter `CLONEDB` to `TRUE` to enable using sparse files in PDB snapshots.
- We might need to apply a patch on the database to create a full PDB from sparse-files-based PDB snapshots.
- We have limited control on defining the sparse-files-based PDB snapshot file locations.