Practice

Performing Recovery Part II

Practice Target

In this practice you will perform further recovery scenarios on `oradb` database.

Practice Overview

In this practice, you will perform the recovery procedures for the following scenarios:

- Recovery of datafiles loss by switching to image copies
- Database point-in-time recovery (DBPITR)

Assumptions

- This practice assumes that you have `srv1` up and running from the **non-CDB** snapshot.

Recovery Scenario 4:

Recovery of datafiles loss by switching to image copies

Scenario assumptions:

- The database is running in `ARCHIVELOG` mode
- One or more datafiles are lost
- **Image copies** backup of the database are available in the FRA

Preparing for the scenario

1. Start Putty and login to `srv1` as `oracle`

2. Invoke RMAN with connecting to `oradb` as target and take a full backup of the database as image copies.

It is expected that this backup type takes longer time that taking backup as backupsets.

```
rman target /

BACKUP AS COPY DATABASE TAG 'DB_COPY';

-- just to simulate that some user activity happened after backup:
ALTER SYSTEM SWITCH LOGFILE;
```

Simulating the Crash

You will use a manual method to simulate a tablespace loss.

3. Retrieve the datafile full name of the `users` tablespace. Take a note of the datafile number.

```
SELECT FILE#, NAME FROM V$DATAFILE WHERE TS# = (SELECT TS# FROM V$TABLESPACE WHERE
NAME='USERS');
```

4. Delete the `USERS` tablespace datafile.

```
host 'rm /u01/app/oracle/oradata/ORADB/datafile/*_users*.dbf';
```

5. Verify that the database instance is still up and running.

```
SELECT * FROM V$VERSION;
```

Recovery Actions

6. Validate the database.

Observe that the command reports that the datafile of the users tablespace is lost.

```
VALIDATE DATABASE;
```

7. Issue the following commands on the tablespace to recover it by switching the tablespace data file to the backup image copy.

This method is much faster than restoring from backupset because it does not involve restoring the datafile.

```
ALTER DATABASE DATAFILE 4 OFFLINE;
SWITCH DATAFILE 4 TO COPY;
RECOVER DATAFILE 4;
ALTER DATABASE DATAFILE 4 ONLINE;
```

8. Validate the database. It should succeed.

```
VALIDATE DATABASE;
```

Clean up

9. Shutdown `srv1` and restore it to the snapshot "**oradb non-CDB database**".

Recovery Scenario 5:

Performing database point-in-time recovery (DBPITR) when the recovery point is known

Scenario Target

In this scenario, some data has been mistakenly deleted from the database and after discussing the incident with the management, you decided to rewind the database to the point before deleting the data.

In this scenario, you know roughly at what time the incident happened and therefore you know that if you rewind the database to a point-in-time before that time, the lost data will definitely be recovered.

Note: Although the DBPITR works just fine for the purpose, to rewind a database to a short-period of time in the past, database flashback is the best technology choice for this purpose. However, for long-period point-in-time recovery, DBPITR is the best to fit this purpose.

Scenario assumptions:

- The database is running in `ARCHIVELOG` mode

Preparing for the scenario

10. Open a Putty session to `srv1` as `oracle`, invoke RMAN and take a full backup of the database as backupset.

```
rman target /
BACKUP DATABASE TAG 'DB_FULL';
```

11. Exit from RMAN.

```
exit
```

Simulating the data loss

12. Simulate data loss by performing the following steps:

a. Start SQL*Plus and connect as to `ORADB` as sysdba

```
sqlplus / as sysdba
```

b. Take a note of the current time.

```
SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY HH24:MI:SS') FROM DUAL;
```

c. Issue the following command:

```
ALTER SYSTEM SWITCH LOGFILE;
```

d. Drop the `HR.EMPLOYEES` table.

This action is the destructive unintentional action that you want to recover from.

```
DROP TABLE HR.EMPLOYEES CASCADE CONSTRAINTS;
```

e. Switch the redo logfile

```
ALTER SYSTEM SWITCH LOGFILE;
```

Recovery Actions

13. Exit from SQL*Plus then invoke RMAN and connect to `ORADB` as target

```
exit
rman target /
```

14. Shutdown the database and start it up in `MOUNT` state.

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
```

15. Perform the DBPITR to rewind the database to the time noted earlier. Replace the time in the following code with noted time.

The time format must match the value of `$NLS_DATE_FORMAT`. To view its value from RMAN, issue the following command:

```
host 'echo $NLS_DATE_FORMAT';
```

```
ALTER SESSION SET NLS_DATE_FORMAT='DD-MON-YYYY HH24:MI:SS';

SELECT SYSDATE FROM DUAL ;

RUN { SET UNTIL TIME '<DD-MON-YYYY HH24:MI:SS>';
      RESTORE DATABASE;
      RECOVER DATABASE;}
```

16. Open the database using `RESETLOGS` option.

```
ALTER DATABASE OPEN RESETLOGS;
```

17. Check if the data has been recovered.

```
SELECT COUNT(*) FROM HR.EMPLOYEES;
```

Note

If you have been provided a wrong recovery point that is later than the incident time, you would not see the lost data. In this case, you need to rewind the database to some point-in-time further backward in the past. But because you opened the database in `RESETLOGS` option, it is now running in an incarnation version that is different from its incarnation version at that point.

In that case, you have to issue the command "`RESET DATABASE TO INCARNATION <DBINC>`", just after mounting the database. DBINC is the previous database incarnation and can be obtained from the command "`LIST INCARNATION OF DATABASE`".

Clean up

18. Shutdown `srv1` and restore it to the snapshot "**oradb non-CDB database**".

Summary

In this practice, you performed the full recovery procedure to the following scenarios:

- Recovery of datafiles loss by switching to image copies
- Database point-in-time recovery (DBPITR)