How to Open Physical Standby for Read Write Testing using Flashback

Customer: Guarantee Trust Bank plc Nigeria

Project: 19c Multitenant implementation and migration

Target: Open Physical standby database for testing.

Technology: Active Data Guard & Flashback.

Table of contents:

1- Purpose of the article:

This article is to open the Standby database in read write mode for any reporting or testing and then move it back to standby database using the flashback technology.

2- Introduction:

Using a combination of Data Guard, restore points, and Flashback Database, a physical standby database can be opened temporarily in read/write mode for development, reporting, or testing purposes, and then flashed back to a point in the past to be reverted back to a physical standby database. When the database is flashed back, Data Guard automatically synchronizes the standby database with the primary database, without the need to re-create the physical standby database from a backup copy of the primary database.

Perform the following steps to activate the physical standby database as a production database and later resynchronize it with the primary database.

3- Prerequisites for opening Standby database in read/write mode:

a. Make sure that Physical standby database in SYNC state with primary, use below queries to determine the state:

Primary as SYS user:

```
alter system archive log current;


select thread#, max(sequence#) "Last Primary Seq Generated" from v$archived_log val, v$database vdb

where val.resetlogs_change# = vdb.resetlogs_change# group by thread# order by 1;
```

Physical Standby:

```
select thread#, max(sequence#) "Last Standby Seq Applied" from v$archived_log val, v$database vdb

where val.resetlogs_change# = vdb.resetlogs_change#  and val.applied in ('YES','IN-MEMORY')

group by thread# order by 1;
```

The output will fetch two rows, one row for each instance.

If the output of the second query from physical standby is similar to the first statement which is executed at primary, that means that Physical standby database is in SYNC state with primary.

Note: if Standby database has huge gap refer to document**: create Physical Standby Database using duplicate section 8**. In case the gap is note huge and can be recovered, refer to document: **Rolling Forward a Physical Standby Using Recover from Service**

b. Shutdown both nodes from the cluster using below command as oracle user:
   srvctl stop database -d HOBANKDR
c. Start one node at mount stage using below command as oracle user:
   srvctl start instance -i HOBANKDR1 -d HOBANKDR
d. and start replication as sys user:
   alter database recover managed standby database disconnect from session;

4- Steps to open Physical standby database in read/write mode using flashback.

In Standby database

a. Make sure to give enough space for to FRA.
b. Cancel Redo Apply and create a guaranteed restore point:

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

CREATE RESTORE POINT Standby_flashback_testing GUARANTEE FLASHBACK DATABASE;

To confirm the details of restore point and its scn and time stamp run:

select NAME,SCN,TIME from v$restore_point;

In Primary Database

a. On the primary database, switch logs so the SCN of the restore point will be archived on the physical standby database. When using standby redo log files, this step is essential to ensure the database can be properly flashed back to the restore point.
   ALTER SYSTEM ARCHIVE LOG CURRENT;
b. Defer log archive destinations pointing to the standby that will be activated.
   ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_3=DEFER SID='*';
   Note:
   Be careful choose the correct parameter of LOG_ARCHIVE_DEST_STATE_n to point to correct destination.

In Standby database

a. Activate the physical standby database:

   ALTER DATABASE ACTIVATE STANDBY DATABASE;

Once its done you can check the controlfile status will be changed from **Standby** to **Current**

select CONTROLFILE_TYPE from v$database;

b. Then open the database.
   ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
   ALTER DATABASE OPEN;

Once the standby database has been activated, you can run reporting tools or perform other testing and activities for days or even weeks, independent of the primary database

5- Switch standby database back to Physical standby database

In standby database

Revert the active standby database back to Physical standby database:

a. Mount the database:
   STARTUP MOUNT FORCE;
b. Flashback the database to restore point:
   FLASHBACK DATABASE TO RESTORE POINT Standby_flashback_testing ;

You can confirm the same by checking the controlfile status. It will be now **backup** controlfile

   select controlfile_type from v$database;

c. Convert to Standby database:
   ALTER DATABASE CONVERT TO PHYSICAL STANDBY;

   Stop the database as oracle: srvctl stop database -d HOBANKDR
   Start the database as oracle: srvctl start database -d HOBANKDR –o mount

d. Start media recovery process:
   ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;

   SQL> select controlfile_type from v$database; → STANDBY

In Primary database

Re-enable archiving to the physical standby database:

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_3=ENABLE SID='*';
```

In standby database

Drop restore point:

DROP RESTORE POINT Standby_flashback_testing ;