19c Physical Standby Switchover Best Practices using SQL*Plus

Customer: Guarantee Trust Bank plc Nigeria

Project: 19c Multitenant implementation and migration

Target: Switchover to Physical standby database.

Technology: Active Data Guard & SQL*Pus.

Table of contents:

1- Purpose of the article:

This Document explain about switchover steps for 19c.

2- Prerequisites

Setup/configuration verification

- Verify the alert logfiles and make sure there are no erorrs
- Make sure primary and physical standby configuration are good and there are no errors in redo transport and redo apply.
- Check If Physical standby is in SYNC state using below:

```
You can also optionally, use the below queries to check the redo transport and apply
status


On primary

To check the remote redo transport status and if there are any
errors,  V$ARCHIVE_DEST.ERROR will show the details

SQL> col DEST_NAME for a20
SQL> col DESTINATION for a25
SQL> col ERROR for a15
SQL> col ALTERNATE for a20
SQL> set lines 1000
SQL> select DEST_NAME,DESTINATION,ERROR,ALTERNATE,TYPE,status,VALID_TYPE,VALID_ROLE
from V$ARCHIVE_DEST where STATUS <>'INACTIVE';

To check the last archivelog created at the primary:

SQL> select   thread#, max(sequence#) "Last Primary Seq Generated"
              from     gv$archived_log val, gv$database vdb
              where    val.resetlogs_change# = vdb.resetlogs_change#
              group by thread# order by 1;

On Standby:

Using the below query, check the last received Archivelog from primary database (if
database is RAC, then result will be displayed for each thread)


Query output is: last archive log sequence Applied by standby

SQL> select    thread#, max(sequence#) "Last Standby Seq Applied"
              from     gv$archived_log val, gv$database vdb
              where    val.resetlogs_change# = vdb.resetlogs_change#
              and      val.applied in ('YES','IN-MEMORY')
              group by thread# order by 1;



If Physical standby database is in SYNC, the output of above will be similar to the
one above it
```

- Verify Initialization Parameters

Before Switchover, set below parameters at Main site: at SPFILE level for all SIDs:

SwichOver to Onsite ADG:

FAL_SERVER=REP;

FAL_CLIENT=PRIMARY;

DB_FILE_NAME_CONVERT='HOBANKREP','HOBANK'

LOG_FILE_NAME_CONVERT='HOBANKREP','HOBANK'

PDB_FILE_NAME_CONVERT='HOBANKREP','HOBANK'

SwichOver to DR ADG:

FAL_SERVER=DR;

FAL_CLIENT=PRIMARY;

DB_FILE_NAME_CONVERT='HOBANKDR','HOBANK'

LOG_FILE_NAME_CONVERT='HOBANKDR','HOBANK'

PDB_FILE_NAME_CONVERT='HOBANKDR','HOBANK'

- Check the MRP process status (it should be started running and applying the logs)
  `select * from gv$dataguard_process;`

- Restart managed recovery process and start it with no delay option:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE NODELAY;
```

For any reason, If standby database recovery (MRP) started with delay OR if the standby always maintained with lag then switchover will consume time to apply the logs to be sync.
Before switchover, try to maintain minimal archive log apply lag, which will reduce the total switchover time window.

3- Switchover:

Verify the switchover

```
SQL> ALTER DATABASE SWITCHOVER TO <standby db_name> VERIFY;
```

For Onsite ADG: HOBANKREP

For DR ADG: HOBANKDR

In case of error, fix an issue and then rerun switchover verify command.

Switchover steps

If switchover verify is successful, then execute the command to switchover the database.

Stop one of RAC at main, and in case we are switching over to DR site , shutdown one node at DR.

- Execute in the current primary

```
SQL> ALTER DATABASE SWITCHOVER TO <standby db_name>;
```

For Onsite ADG: HOBANKREP

For DR ADG: HOBANKDR

- Open Primary database

```
SQL> ALTER DATABASE OPEN;
```

- Old primary (current/new standby) should be mounted

```
SQL> STARTUP MOUNT;
```

- Start redo apply in new standby

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
```