Oracle 19c-Step by Step Manual Data Guard Switchover

Switchover Operation Procedure:

After configuring data guard, databases would be either primary and standby database role, and we can altered these roles without loss of data or without resetting logs called Switchover.

A database can be in one of two mutually exclusive modes (primary or standby). These roles can be altered at runtime without loss of data or resetting of redo logs. This process is known as a Switchover and can be performed using the following statements.

Data guard Configuration Details:-

Environment Details	Primary DB	Standby DB
DB SID	PROD	PROD_DR
DB Role	PRIMARY	PHYSICAL STANDBY
Server IP	192.168.0.109	192.168.0.125
DB Version	19.3.0.0.0	19.3.0.0.0
OS	Redhat 7.9	Redhat 7.9

Step 1:- Check database role and database name

Primary database:-

SQL> select name,open_mode,database_role from v$database;


```
SQL> select name,open_mode,database_role from v$database;

NAME      OPEN_MODE            DATABASE_ROLE
--------- -------------------- ----------------
PROD      READ WRITE           PRIMARY
```

Standby database:-

SQL> select name,open_mode,database_role from v$database;


```
SQL> select name,open_mode,database_role from v$database;

NAME      OPEN_MODE            DATABASE_ROLE
--------- -------------------- ----------------
PROD      MOUNTED              PHYSICAL STANDBY
```

Precheck for Switchover:- (PRIMARY SIDE)

Before performing switchover, kindly verify the state of data guard on both the instances by following SQL queries:

Step 2:-

SQL> ALTER SESSION SET nls_date_format='DD-MON-YYYY HH24:MI:SS';

```
SQL> ALTER SESSION SET nls_date_format='DD-MON-YYYY HH24:MI:SS';

Session altered.
```

SQL> SELECT sequence#, first_time, next_time, applied FROM v$archived_log ORDER BY sequence#;

```
oracle@prod:~
SQL>  SELECT sequence#, first_time, next_time, applied FROM v$archived_log ORDER BY sequence#

 SEQUENCE# FIRST_TIME           NEXT_TIME            APPLIED
---------- -------------------- -------------------- ---------
         8 16-JUL-2025 20:05:00 16-JUL-2025 23:00:40 YES
         9 16-JUL-2025 23:00:40 16-JUL-2025 23:04:04 YES
        10 16-JUL-2025 23:04:04 16-JUL-2025 23:10:25 YES
        11 16-JUL-2025 23:10:25 16-JUL-2025 23:15:21 YES
        12 16-JUL-2025 23:15:21 16-JUL-2025 23:17:58 YES
        13 16-JUL-2025 23:17:58 17-JUL-2025 20:53:28 YES
        14 17-JUL-2025 20:53:28 17-JUL-2025 21:47:07 YES
        14 17-JUL-2025 20:53:28 17-JUL-2025 21:47:07 YES
        15 17-JUL-2025 21:47:07 17-JUL-2025 22:01:12 YES
        15 17-JUL-2025 21:47:07 17-JUL-2025 22:01:12 YES
        16 17-JUL-2025 22:01:12 17-JUL-2025 22:07:55 YES
        16 17-JUL-2025 22:01:12 17-JUL-2025 22:07:55 YES
        17 17-JUL-2025 22:07:55 17-JUL-2025 22:08:00 YES
        17 17-JUL-2025 22:07:55 17-JUL-2025 22:08:00 YES
        18 17-JUL-2025 22:08:00 17-JUL-2025 22:14:39 YES
        18 17-JUL-2025 22:08:00 17-JUL-2025 22:14:39 YES
        19 17-JUL-2025 22:14:39 17-JUL-2025 22:54:23 YES
        19 17-JUL-2025 22:14:39 17-JUL-2025 22:54:23 YES
        20 17-JUL-2025 22:54:23 17-JUL-2025 22:54:36 YES
        20 17-JUL-2025 22:54:23 17-JUL-2025 22:54:36 YES
```

Step 3:-

SQL> select switchover_status from v$database;

```
SQL>  select switchover_status from v$database;

SWITCHOVER_STATUS
-------------------
TO STANDBY
```

Note: This command will give you appropriate message about the data guard current status.

After getting confirmation on data guard smooth activity, we can instantiate switchover activity by issuing following sort of commands:

The **switchover_status** column of **v$database** can have the following values:

Not Allowed:-Either this is a standby database and the primary database has not been switched first, or this is a primary database and there are no standby databases.
Session Active:- Indicates that there are active SQL sessions attached to the primary or standby database that need to be disconnected before the switchover operation is permitted.

Switchover Pending:- This is a standby database and the primary database switchover request has been received but not processed.

Switchover Latent:- The switchover was in pending mode, but did not complete and went back to the primary database.

To Primary:- This is a standby database, with no active sessions, that is allowed to switch over to a primary database.
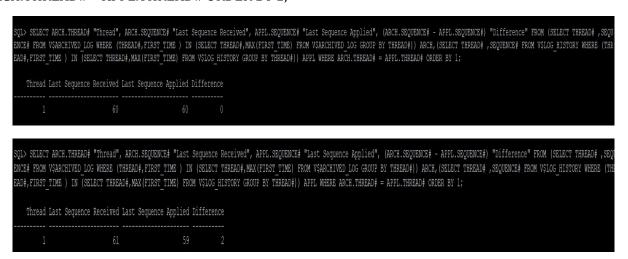
To Standby:- This is a primary database, with no active sessions, that is allowed to switch over to a standby database.

Recovery Needed:- This is a standby database that has not received the switchover request.

On both side check sync status and difference before switchover:

Step 4:

SQL> SELECT ARCH.THREAD# "Thread", ARCH.SEQUENCE# "Last Sequence Received", APPL.SEQUENCE# "Last Sequence Applied", (ARCH.SEQUENCE# - APPL.SEQUENCE#) "Difference" FROM (SELECT THREAD# ,SEQUENCE# FROM V$ARCHIVED_LOG WHERE (THREAD#,FIRST_TIME) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$ARCHIVED_LOG GROUP BY THREAD#)) ARCH,(SELECT THREAD# ,SEQUENCE# FROM V$LOG_HISTORY WHERE (THREAD#,FIRST_TIME) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$LOG_HISTORY GROUP BY THREAD#)) APPL WHERE ARCH.THREAD# = APPL.THREAD# ORDER BY 1;

```
SQL> SELECT ARCH.THREAD# "Thread", ARCH.SEQUENCE# "Last Sequence Received", APPL.SEQUENCE# "Last Sequence Applied", (ARCH.SEQUENCE# - APPL.SEQUENCE#) "Difference" FROM (SELECT THREAD# ,SEQU
ENCE# FROM V$ARCHIVED_LOG WHERE (THREAD#,FIRST_TIME ) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$ARCHIVED_LOG GROUP BY THREAD#)) ARCH,(SELECT THREAD# ,SEQUENCE# FROM V$LOG_HISTORY WHERE (THR
EAD#,FIRST_TIME ) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$LOG_HISTORY GROUP BY THREAD#)) APPL WHERE ARCH.THREAD# = APPL.THREAD# ORDER BY 1;

    Thread Last Sequence Received Last Sequence Applied Difference
---------- ---------------------- --------------------- ----------
         1                     60                    60          0
```

```
SQL> SELECT ARCH.THREAD# "Thread", ARCH.SEQUENCE# "Last Sequence Received", APPL.SEQUENCE# "Last Sequence Applied", (ARCH.SEQUENCE# - APPL.SEQUENCE#) "Difference" FROM (SELECT THREAD# ,SEQU
ENCE# FROM V$ARCHIVED_LOG WHERE (THREAD#,FIRST_TIME ) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$ARCHIVED_LOG GROUP BY THREAD#)) ARCH,(SELECT THREAD# ,SEQUENCE# FROM V$LOG_HISTORY WHERE (THR
EAD#,FIRST_TIME ) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM V$LOG_HISTORY GROUP BY THREAD#)) APPL WHERE ARCH.THREAD# = APPL.THREAD# ORDER BY 1;

    Thread Last Sequence Received Last Sequence Applied Difference
---------- ---------------------- --------------------- ----------
         1                     61                    59          2
```

On Primary database:-

Step 5:

SQL> alter database commit to switchover to standby;

```
SQL> alter database commit to switchover to standby;

Database altered.
```

Step 6:-

shutdown immediate

startup nomount

alter database mount standby database

```
Version 19.3.0.0.0
[oracle@prod ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Jul 20 16:09:42 2025
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.
```

```
SQL> startup nomount;

ORACLE instance started.

Total System Global Area 1124072640 bytes
Fixed Size                   8895680 bytes
Variable Size              805306368 bytes
Database Buffers           301989888 bytes
Redo Buffers                 7880704 bytes
```

SQL> alter database mount standby database;

```
SQL>
SQL> alter database mount standby database;

Database altered.
```

SQL> alter database recover managed standby database disconnect from session;

```
SQL> alter database recover managed standby database disconnect from session;

Database altered.
```

Step 7:-

SQL> select name,open_mode,database_role from v$database;

```
SQL> select name,open_mode,database_role from v$database;

NAME      OPEN_MODE            DATABASE_ROLE
--------- -------------------- ----------------
PROD      MOUNTED              PHYSICAL STANDBY
```

On Standby database:

Step 8:-alter database commit to switchover to primary;

```
SQL> alter database commit to switchover to PRIMARY;

Database altered.
```

Step 9:-

SQL> select name,open_mode,database_role from v$database;

```
SQL> select name,open_mode,database_role from v$database;

NAME       OPEN_MODE             DATABASE_ROLE
---------  --------------------  ----------------
PROD       MOUNTED               PRIMARY
```

Switch over activity have completed successfully, our old primary database has become standby and old standby database has become primary database.

Note:-To test above switchover activity, generate multiple archive logs on primary database and verify those archive logs being transferred on standby database.

After switchover activity,

Environment Details	Primary DB	Standby DB
DB SID	PROD_DR	PROD
DB Role	PHYSICAL STANDBY	PPRIMARY
Server IP	192.168.0.125	192.168.0.109
DB Version	19.3.0.0.0	19.3.0.0.0
OS	Redhat 7.9	Redhat 7.9

THANKYOU