Using FLASHBACK DATABASE for [destructive] D.R. Testing

{This has been posted to my blog}

Testing your Disaster Recovery strategy with an Oracle Standby Database can be at different "levels" for the database :
1. Graceful Switchover to the D.R. site and reversing roles between the two databases, but only querying* data at the D.R. site
2. Shutdown of the Production site and Failover to the D.R. site and only *querying* data at the D.R. site
3. Shutdown of the Production site and Failover to the D.R. site with *destructive* testing at the D.R. site followed by restore (or flashback) of the D.R. site database to "throwaway" all changes
3. Either Switchover or Failover with role reversal and *destructive* testing at the D.R. site, validation that data changes flow back to the Production site and, finally, restore (or flashback) of the database at both sites.

Restoring a large database at one or both sites can take time.
You may have taken a Snapshot of the database(s) and just restore the snapshot.
Or you may FLASHBACK the database(s).

{for details on how I created this Standby database configuration in 19c, see my previous posts here and here}

I will try to use FLASHBACK DATABASE here.

I start with the Primary running at the Production site :

```
oracle19c>sqlplus hemant/hemant@orclpdb1

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:22:26 2020
Version 19.3.0.0.0

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

Last Successful login time: Thu Mar 26 2020 23:22:02 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> drop table my_transactions purge;

Table dropped.

SQL> create table my_transactions (txn_id number, txn_data varchar2(50));

Table created.

SQL> insert into my_transactions values (1,'First at ProductionDC:Primary');

1 row created.

SQL> commit;

Commit complete.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
oracle19c>
```

I then verify the state of both databases (the "oracle19c" prompt is at the Production site, the "STDBYDB" prompt is at the D.R. site)

```
oracle19c>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:23:48 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON       CURRENT_SCN
------- ----------- ---------------- ------------------ -----------
CURRENT NOT ALLOWED PRIMARY          NO                     4796230

SQL>



STDBYDB>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:25:02 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select controlfile type, open resetlogs, database role, flashback on, standby became primary scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
STANDBY REQUIRED    PHYSICAL STANDBY NO
                         0     4796205


SQL>
```

So, currently, the Standby is slightly behind (SCN#4796205) the Primary (SCN#4796230). Note that FLASHBACK is *not* enabled in the databases.

I first create my RESTORE POINT on the Standby and then on the Primary.

```
{at the current Standby at the D.R. site}
SQL> alter database recover managed standby database cancel;

Database altered.

SQL> show parameter db_recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest                string      /opt/oracle/FRA/STDBYDB
db_recovery_file_dest_size           big integer 10G
SQL> create restore point dr_before_switch guarantee flashback database;

Restore point created.

SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

NAME
--------------------------------------------------------------------------------
RESTORE_POINT_TIME
---------------------------------------------------------------------------
DATABASE_INCARNATION#        SCN GUA
-------------------- ---------- ---
DR_BEFORE_SWITCH

                     2    4796590 YES


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

Database altered.

SQL>


{at the current Primary at the Production site}
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON       CURRENT_SCN
------- ----------- ---------------- ------------------ -----------
CURRENT NOT ALLOWED PRIMARY          NO                     4796230

SQL> alter system switch logfile;

System altered.

SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON       CURRENT_SCN
------- ----------- ---------------- ------------------ -----------
CURRENT NOT ALLOWED PRIMARY          NO                     4796968

SQL> create restore point production_before_switch guarantee flashback database;

Restore point created.

SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

NAME
--------------------------------------------------------------------------------
RESTORE_POINT_TIME
---------------------------------------------------------------------------
DATABASE_INCARNATION#        SCN GUA
-------------------- ---------- ---
PRODUCTION_BEFORE_SWITCH

                     2    4797182 YES


SQL>
```

At each site, I have created a Restore Point (with Guarantee Flashback Database). I have ensured that the Restore Point for the current Standby Database at the D.R. site is at a *lower* SCN (4796590) than that for the current Primary (4797182) (at the Production site). To further ensure this, I did a log swich and verified the CURRENT_SCN at the Primary before creating the Restore Point.

(Note that both sites have a DB_RECOVERY_FILE_DEST configured for the GUARANTEEd Restore Point).

(*a small note : I have to disable Recovery at the Standby database before I can create a Restore Point and then re-enable Recovery after that. A Restore Point cannot be created when a database is in Recovery mode)*.

I now put in another transaction at the Primary (Production site database) and then Switchover to to the D.R. site.

```
SQL> connect hemant/hemant@orclpdb1
Connected.
SQL> insert into my_transactions values (2,'Second, after R.P. at ProductionDC:Primary');

1 row created.

SQL> commit;

Commit complete.

SQL> connect / as sysdba
Connected.
SQL> alter database switchover to stdbydb;

Database altered.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
oracle19c>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:41:57 2020
Version 19.3.0.0.0

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

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, current_scn
  2  from v$databasse
  3
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON       CURRENT_SCN
------- ----------- ---------------- ------------------ -----------
STANDBY ALLOWED     PHYSICAL STANDBY RESTORE POINT ONLY     4899284

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

Database altered.

SQL>
```

So, now the database at the Production site is a Standby database.

I now connect to the database at the D.R. site that is now a Primary

```
STDBYDB>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:45:02 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
CURRENT NOT ALLOWED PRIMARY          RESTORE POINT ONLY
                   4899284           0


SQL> shutdown ;
ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL>
SQL> alter pluggable database orclpdb1 open;

Pluggable database altered.

SQL> connect hemant/hemant@STDBYPDB1
Connected.
SQL> select * from my_transactions order by 1;

    TXN_ID TXN_DATA
---------- --------------------------------------------------
         1 First at ProductionDC:Primary
         2 Second, after R.P. at ProductionDC:Primary

SQL>
SQL> insert into my_transactions values (3,'Destructive change at DRDC');

1 row created.

SQL> commit;

Commit complete.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
STDBYDB>
```

{Note that "STDBYDPDB1" is my tnsnames entry for the PDB which still has the name "orclpdb1" at the D.R. site.}

I have created a "destructive" change with the third row which should not be in production. However, I will switch back to the Production data centre and verify that the row has replicated back.

```
{at the D.R. site}
STDBYDB>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:50:29 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> alter database switchover to orclcdb;

Database altered.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
STDBYDB>



{at the Production site}
oracle19c>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:52:21 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> shutdown immediate
ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
SQL> alter pluggable database orclpdb1 open;
alter pluggable database orclpdb1 open
*
ERROR at line 1:
ORA-65019: pluggable database ORCLPDB1 already open


SQL>
SQL> connect hemant/hemant@orclpdb1
Connected.
SQL> select * from my_transactions order by 1;

    TXN_ID TXN_DATA
---------- --------------------------------------------------
         1 First at ProductionDC:Primary
         2 Second, after R.P. at ProductionDC:Primary
         3 Destructive change at DRDC

SQL>
```

So, I have been able to
1. SWITCHOVER from the Production site to the D.R. site
2. Create a new row when the database is Primary at the D.R. site
3. SWITCHOVER back to the Production site
4. Verify that the destructive row is now at the Production site.

I now need to reset both databases to the state they were in before I began the test.

```
{at the Production site}
oracle19c>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 26 23:56:16 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
CURRENT NOT ALLOWED PRIMARY          RESTORE POINT ONLY
                   5000964          0


SQL>
SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

NAME
--------------------------------------------------------------------------------
RESTORE_POINT_TIME
---------------------------------------------------------------------------
DATABASE_INCARNATION#        SCN GUA
-------------------- ---------- ---
PRODUCTION_BEFORE_SWITCH

                     2    4797182 YES


SQL>
SQL> FLASHBACK DATABASE TO RESTORE POINT PRODUCTION_BEFORE_SWITCH;

Flashback complete.

SQL> alter database open resetlogs ;

Database altered.

SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
CURRENT NOT ALLOWED PRIMARY          RESTORE POINT ONLY
                   5000964     4798237


SQL>
```

```
SQL> connect hemant/hemant@orclpdb1
Connected.
SQL> select * from my_transactions order by 1;

    TXN_ID TXN_DATA
---------- --------------------------------------------------
         1 First at ProductionDC:Primary

SQL>
```

So, now, the database at the Production site has reverted to the Restore Point and all changes after the Restore Point have been discarded.

This includes TXN_ID=2 which I had added to demonstrate propagation of a change from the Production site to the D.R. site ---- in your testing, you must ensure that you do not make any changes after the Restore Point is created. Typically, you'd create your Production Restore Point with the applications disconnecte, database shutdown and re-mounted just before switchover. Remember, this is for **D.R. testing** when you do have control over applications and database shutdown and startup.

What about the database at the D.R. site ? Can I flashback it and resume it's role as a Standby ?
Remember that the Restore Point I created on the D.R. site was at a *lower* SCN than that for the Production site.

```
STDBYDB>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 27 00:08:25 2020
Version 19.3.0.0.0

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

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
STANDBY ALLOWED     PHYSICAL STANDBY RESTORE POINT ONLY
                         0     5000964


SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

NAME
--------------------------------------------------------------------------------
RESTORE_POINT_TIME
---------------------------------------------------------------------------
DATABASE_INCARNATION#        SCN GUA
-------------------- ---------- ---
DR_BEFORE_SWITCH

                     2    4796590 YES

PRODUCTION_BEFORE_SWITCH_PRIMARY

                     2    4797182 NO


SQL>  FLASHBACK DATABASE TO RESTORE POINT DR_BEFORE_SWITCH;

Flashback complete.
```

```
SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
STANDBY ALLOWED     PHYSICAL STANDBY RESTORE POINT ONLY
                         0     4796590


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

Database altered.

SQL>
```

Now the database at the Production site has resumed as a Primary database, at SCN#4798237 and the database at the D.R. site has resumed as a Standby database at SCN#4796590 (lower than the Primary).

If you noticed the second entry in v$restore_point at the D.R. site -- Restore Point name "PRODUCTION_BEFORE_SWITCH_PRIMARY" -- this is a 19c enhancement where a Restore Point created on the Primary automatically propagates to the Standby, with the suffix "_PRIMARY" (to indicate that it came from a database in PRIMARY role) attached to the Restore Point name.

Can I really really be sure that I have reverted both databases to their intended roles ?

I can verify this again :

```
{at the Production site}
SQL> connect hemant/hemant@orclpdb1
Connected.
SQL> insert into my_transactions values (1001,'After DR Testing, back to normal life');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from my_transactions order by 1;

    TXN_ID TXN_DATA
---------- --------------------------------------------------
         1 First at ProductionDC:Primary
      1001 After DR Testing, back to normal life

SQL>



{at the D.R site}
SQL> alter database recover managed standby database cancel;

Database altered.

SQL> alter database open read only;

Database altered.

SQL> alter pluggable database orclpdb1 open;

Pluggable database altered.

SQL> connect hemant/hemant@stdbypdb1
Connected.
SQL> select * from my_transactions order by 1;

    TXN_ID TXN_DATA
---------- --------------------------------------------------
         1 First at ProductionDC:Primary
      1001 After DR Testing, back to normal life

SQL>
```

```
SQL> connect / as sysdba
Connected.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
SQL> alter database recover managed standby database disconnect from session;

Database altered.

SQL> select controlfile_type, open_resetlogs, database_role, flashback_on, standby_became_primary_scn,
current_scn
  2  from v$database
  3  /

CONTROL OPEN_RESETL DATABASE_ROLE    FLASHBACK_ON
------- ----------- ---------------- ------------------
STANDBY_BECAME_PRIMARY_SCN CURRENT_SCN
-------------------------- -----------
STANDBY REQUIRED    PHYSICAL STANDBY RESTORE POINT ONLY
                         0     4802358


SQL>
```

To verify the behaviour, I added a new row (TXN_ID=1001) in the Primary database at the Production site and then did an OPEN READ ONLY of the Standby database at the D.R. site to check the table.
Note : So as to not require an Active Data Guard licence, I stopped Recovery on the Standby before I did an OPEN READ ONLY.

Of course, after the verification, I resumed the Standby database in Recovery mode.

This whole exercise also did NOT need the databases to be "permanently" in FLASHBACK ON mode. I used the Guaranteed Restore Point feature with the Recovery File Dest to generate the minimal flashback logs. At the end of the exercise, I can DROP the Restore Points.

```
{at the Production site}
oracle19c>sqlplus

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 27 00:37:47 2020
Version 19.3.0.0.0

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

Enter user-name: / as sysdba

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1207955552 bytes
Fixed Size                  9134176 bytes
Variable Size             436207616 bytes
Database Buffers          754974720 bytes
Redo Buffers                7639040 bytes
Database mounted.
SQL> drop restore point PRODUCTION_BEFORE_SWITCH;

Restore point dropped.

SQL> alter database open;

Database altered.

SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

no rows selected

SQL>


{at the D.R. site}
STDBYDB>sqlplus '/ as sysdba'

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 27 00:40:47 2020
Version 19.3.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> alter database recover managed standby database cancel;

Database altered.

SQL> select name from v$restore_point;

NAME
--------------------------------------------------------------------------------
DR_BEFORE_SWITCH
PRODUCTION_BEFORE_SWITCH_PRIMARY

SQL>
SQL> drop restore point PRODUCTION_BEFORE_SWITCH_PRIMARY;

Restore point dropped.

SQL> drop restore point DR_BEFORE_SWITCH;

Restore point dropped.

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

```
Database altered.

SQL>
SQL> select name, restore_point_time, database_incarnation#, scn, guarantee_flashback_database
  2  from v$restore_point
  3  /

no rows selected

SQL>
```

The only "catch" is that I had to bring up the Production site (Primary) database in MOUNT mode before I could drop the Restore Point. So, you need to factor this into you D.R. testing.

This article also posted at https://hemantoracledba.blogspot.com/2020/03/using-flashback-database-for_27.html