Standby Creation

DATAGUARD:

==========

Standby:

=======

• it’s a exact copy of primary database.

• If any transctions in primary database it will automatically transfer to standby.

• Both primary and standby are in synk.

• standby is in either mount or read only.

• If primary database crashed stanby is activated as primary and all the applications will point to database.

Adv:

===

1. disaster recovery

2. Zero data loss

# How to create standby:

===================

Primary:

~~~~~~

1. db must be enable with archive log mode.

2. Force logging must be enable at database level.

(Select name,open\_mode,log\_mode,force\_logging from v$database;)

• alter database force logging;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# active duplicate standby create without backup:

=================================

1. same db name in primary and standby.

2. We can have different SID.

Primary:

~~~~~~

/prod/ram/oradata

1. create tns\_names

2. Create password copy to standby

Conn to rman

Rman target / auxiliary=sys/welcome@tns\_stby

• rman> duplicate target database for standby from active database nofilenamecheck;

Standby:

~~~~~~

1. create directories.

/prod/ramstby/oradata

2. copy env file from primary and edit.

3. copy pfile and edit

• db\_file\_name\_convert=‘/prod/ram/oradata/’,’/prod/ramstby/oradata/‘

• Log\_file\_name\_convert=‘/prod/ram/oradata/‘,’/prod/ramstby/oradata/‘

4. conn sql & startup nomount

5. Create listener

6. copy password file from primary

# How to keep primary and standby in synk:

=================================

How to enable automation:

====================

1. Archive shipping

2. Log shipping ( standby redo logfiles)

1. archive shipping:

~~~~~~~~~~~~~~

Primary:

~~~~~~

Log\_archive\_dest\_1=‘LOCATION=/prod/ram/admin/arch’

Log\_archive\_dest\_2=‘SERVICE=tns\_mouli ARCH’

[Disable :log\_archive\_dest parameters]

• tnsnames

• Password file

Standby:

~~~~~~

STANDBY\_FILE\_MANAGEMENT=AUTO

• listener

• Password file

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# to check MRP status:

=================

• select process,status from v$managed\_standby;

MRP ON:

~~~~~~~

• alter database recover managed standby database disconnect;

MRP OFF:

~~~~~~~

• alter database recover managed standby database cancel;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# To check archives applied or not:

~~~~~~~~~~~~~~~~~~~~~~~~~~

• Select sequence# ,applied from v$archived\_log;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#

LNS : it is in primary side and sends archives to standby.(log network service)

RFS :it is in standby side and receives the archives from primary (remote file service)

MRP: it is in standby side and applies archives in standby.(managed recovery process)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

=================================

# Manual standby creation using Rman hot backup:

=================================

Primary: /prod/ram/oradata ( db name ram)

~~~~~~. ( sid= abc)

1. take Rman hot backup

• backup database;

• Backup archivelog all;

• Backup current controlfile for standby;

2. transfer backups to standby.

Standby: /prod/ram/oradata (db name ram)

~~~~~~

1. create required directories.

2. Set env

Sid=ramstby

3. put db in nomount

3. copy pfile and edit

Conn rman

Rman> restore standby controlfile from ‘backup /location and /file name’;

5. Rman> alter database mount;

6. register backups in standby cf

• catalog start with ‘/backup/location/‘;

• Restore database;

• Recover database;

7. alter database open;

# if directory system is different prepare script

================================

run

{

Set newname for datafile 1 to ‘/prod/hydtst/oradata/system01.dbf’;

Set newname for datafile 2 to ‘/prod/hydtst/oradata/sysaux01.dbf’;

Set newname for datafile 3 to ‘/prod/hydtst/oradata/undotbs01.dbf’;

Set newname for datafile 4 to ‘/prod/hydtst/oradata/users01.dbf’;

restore database;

}

10. Rman>switch database to copy.

11. Rman> recover database;

Conn sql

12. alter database rename file ‘/prod/ram/oradata/redo01.log’ to ‘/prod/ramstby/oradata/redo01.log’;

13. alter database rename file ‘/prod/ram/oradata/redo02.log’ to ‘/prod/ramstby/oradata/redo02.log’;

14. alter database rename file ‘/prod/ram/oradata/redo03.log’ to ‘/prod/ramstby/oradata/redo03.log’;

15. alter database open;

16. start MRP.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Snapshot standby:(for application testing)

==============

• FRA should be enable.

• Set fra size and location.

• Enable the flashback.

• Cancel MRP

To snapshot standby

~~~~~~~~~~~~~~~~

• alter database convert to snapshot standby.

• Bounce the database.

To physical standby

~~~~~~~~~~~~~~~

• shutdown snapshots standby

• Startup in mount.

• Alter database convert to physical standby.

• Start MPR

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Switch over

=========

• both the primary and standby should be in sync before doing the activity.

on primary

——————

• select name,open\_mode,database\_role,switchover\_status from v$database;

• Disconnect the active sessions of user.

• Alter database commit to switchover to standby;

• Bounce the db.

• Check the db status again.

On standby

——————-

• select name,open\_mode,database\_role,switchover\_status from v$database;

• Alter database commit to switchover to primary;

• Check the db status again.

• bounce the db.

# enable the sync between primary and standby

—————————————————————————

Failover

======

• when the primary database is not responding.we can convert standby as primary.

• Finish the mrp process in standby.

• If once the Mrp is finished. Again it will not be converted as standby.

• Alter database recover managed standby database cancel;

• Alter database recover managed standby database finish.

• Activate standby database.

• Alter database activate standby database;

• db will go to mount state bounce the db.

• shut immediate;

• Startup;

• check the status now again.

Select name,open\_mode,database\_role from v$ database;