**DATABASE REFRESH STEPS AND PROCEDURES**

High Level Tasks

Pre Refresh Tasks

1 > Take perf database RMAN backup or export for tables which application team want to preserve if required .

2> Create perf refresh directory, to store log files and scripts at Prod and Perf DB server.

cd ~

mkdir perf\_refresh

cd perf\_refresh

2 > save existing passwords (run script provided below) at **PERF**

set pages 0

set lines 200

set feedback off

spool password\_backup.sql

select 'alter user  '|| a.username||' identified by values '''|| b.password||''' ;' from dba\_users a, user$ b where a.username = b.name and name not in (select role from dba\_roles);

spool off

At shell prompt

sed -i '/SQL>/d' password\_backup.sql

3 > Save db links with passwords (run script provided below) at **PERF**

spool db\_links.sql

set feedback off

set heading off

set pages 0

set lines 2000

SET SERVEROUTPUT ON LONG 10000000

DECLARE

CURSOR DB\_LINKS IS

select DISTINCT A.OWNER,'create database link ' || a.DB\_LINK ||' connect to ' ||b.USERID || ' identified by values '''''|| passwordx ||''''' using ''''' ||a.host ||'''''' STMT,A.DB\_LINK DBL from dba\_db\_links a,sys.link$ b where a.DB\_LINK=b.name and a.owner not like 'PUBLIC';

begin

for rec in DB\_LINKS

loop

DBMS\_OUTPUT.PUT\_LINE('CREATE OR REPLACE PROCEDURE '||REC.OWNER||'.'||'DB\_LINK\_CREATE AS ');

DBMS\_OUTPUT.PUT\_LINE('BEGIN');

DBMS\_OUTPUT.PUT\_LINE('EXECUTE IMMEDIATE '''||REC.STMT||''';');

DBMS\_OUTPUT.PUT\_LINE('END;');

DBMS\_OUTPUT.PUT\_LINE('/');

DBMS\_OUTPUT.PUT\_LINE('EXECUTE '||REC.OWNER||'.DB\_LINK\_CREATE'||';');

DBMS\_OUTPUT.PUT\_LINE('DROP PROCEDURE '||REC.OWNER||'.DB\_LINK\_CREATE'||';');

end loop;

end;

/

select 'create public database link ' || a.DB\_LINK ||' connect to ' ||b.USERID || ' identified by values '''|| passwordx ||''' using ''' ||a.host ||''';'

   from  dba\_db\_links a,sys.link$ b where a.DB\_LINK=b.name and a.owner = 'PUBLIC'

/

spool off

At shell prompt

sed -i '/SQL>/d' db\_links.sql

sed -i '/^ /d' db\_links.sql

sed -i "s/[[:blank:]]\*$//" db\_links.sql

4 > comment cronjobs on both nodes (run script provided below) at **PERF**

crontab -l > cron\_orig.bck

sed ‘%s/^/####PERF####/g’ cron\_orig.bck > cron\_orig.mnt

crontab cron\_orig.mnt

5 > **Prepare RMAN duplicate script** Adjust datafiles based on disk group name and size (or mount point) (run script provided below) at **PROD**.

6 > Take production backup and make a note of backup completion time and SCN if possible

7 > Ask SA to mount /rman mount point to perf env

Refresh Tasks

1> set cluster database = false in PFILE/SPFILE.

2> Blackout database from OEM to avoid alerts during refresh

3> Get the list of files to be deleted at **PERF**.

Set pages 0

Set lines 200

Set feedback off

Set heading off

Spool list\_of\_Files.sh

select 'rm -rf '||name from v$datafile

union

select 'rm -rf '||name from v$controlfile

union

select 'rm -rf '||MEMBER from v$logfile

spool off

AT Shell Prompt

sed -i '/SQL>/d' list\_of\_Files.sh

3> shutdown database with srvctl stop database -d <database name>

4> stop listener

5> Disable NFS snapshots for the Perf filesystem

6> drop datafiles,temp and redolog file from prf database using Step 3 output. For ASM, invoke asmcmd and use the commands provided in Step3.

7> startup database in nomount mode ( one node only)

8> Update the db\_file\_name\_convert and log\_file\_name\_convert in SPFILE/PFILE.

9> Run rman script in nohup mode (run script provided below)

Execute the following in sqlplus of Production DB.

**A.**

set pages 0

set lines 200

set feedback off

set heading off

spool rman\_script.sh

prompt rman << ! > rman\_perf\_refresh.log

prompt connect auxiliary /

prompt run {

select 'allocate auxiliary channel a1 type disk ;' from dual;

select 'allocate auxiliary channel a2 type disk ;' from dual;

select 'allocate auxiliary channel a3 type disk ;' from dual;

select 'allocate auxiliary channel a4 type disk ;' from dual;

select 'allocate auxiliary channel a5 type disk ;' from dual;

select 'allocate auxiliary channel a6 type disk ;' from dual;

select 'set newname for datafile ' ||file\_id|| ' to '''|| file\_name|| ''';' from dba\_data\_files order by file\_id;

select 'set newname for tempfile ' ||file\_id|| ' to '''|| file\_name|| ''';' from dba\_temp\_files order by file\_id;

select 'DUPLICATE DATABASE TO $ORACLE\_SID BACKUP LOCATION '||chr(39)||'/rman/production/ORACLE\_SID'||chr(39)||' NOFILENAMECHECK ;' from dual;

prompt }

prompt !

spool off

**B.**

**Bring the rman\_script.sh from Production to the Target Database.**

**C.**

**At Perf DB server**

sed -i '/SQL>/d' rman\_script.sh

**D.**

**Change the File Name and Location with respect to the Perf Database considering the filesystem space.**

In case of ASM change the complete name to disksgroup. For example change '+DATA\_DG3/odsprdp1/datafile/brm.325.886572515' to '+DATA\_DG3'

**Change the DUPLICATE statement with correct location.**

(DUPLICATE DATABASE TO $ORACLE\_SID BACKUP LOCATION '/rman/production/ORACLE\_SID' NOFILENAMECHECK)

**E.**

**AT Perf Database**

Run the rman script at Perf DB.

8> wait for RMAN duplicate to complete

Post refresh tasks

1> set cluster database = true and start database on 2nd node

2> restore users password (run script which is created in pre refresh task )

3> Drop Prod DB links if required

4> restore DB link (run script which is created in pre refresh task )

5> run and app specific post refresh tasks

6> enable cron jobs on both nodes

crontab /tmp/cron\_orig.bck

7 > Drop and recreate temporary tablespace

DB\_FILE\_NAME\_CONVERT should be able to take care of the tempfile names. In case there is issue is creating the tempfiles, then drop the temp files and add proper tempfiles to the TEMP tablespaces.

8 > Validate OEM Monitoring.

9 > Validate Rman backup

10 > Enable NFS snapshots.