# Add hosts

vi /etc/hosts

10.19.175.55 vnoradev09.hcnet.vn vnoradev09

10.19.175.56 vnoradev10.hcnet.vn vnoradev10

# Create tnsnames.ora in standby

OLAB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = vnoradev09.hcnet.vn)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = olab)

)

)

ST2OLAB=(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = vnoradev08.hcnet.vn)(PORT = 1521))(CONNECT\_DATA =(SERVER = DEDICATED)(SERVICE\_NAME = olab.hcnet.vn)))

# Add standby entry in tnsnames primary db

OLAB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = vnoradev09.hcnet.vn)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = olab)

)

)

ST2OLAB=(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = vnoradev08.hcnet.vn)(PORT = 1521))(CONNECT\_DATA =(SERVER = DEDICATED)(SERVICE\_NAME = olab.hcnet.vn)))

# Update bash\_profile on standby server

# User specific environment and startup programs

# --------------------------------------------------

# OS User: oracle

# Application: Oracle Database Software Owner

# Version: Oracle 19g Release 15

# --------------------------------------------------

export ORACLE\_SID=st2olab;

export ORACLE\_TERM=xterm;

export ORACLE\_BASE=/u01/app/oracle;

export ORACLE\_HOME=/u01/app/oracle/product/19.0.0/dbhome\_1;

export PATH=/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/oracle/bin:/bin:/OPatch;

export LD\_LIBRARY\_PATH=/lib;

export ORACLE\_UNQNAME=olab;

export JAVA\_HOME=/jdk;

export PATH=$PATH:$ORACLE\_HOME/bin:$ORACLE\_HOME/OPatch;

export CV\_ASSUME\_DISTID=OEL8; ###for oracle 19C

alias sqlplus="rlwrap sqlplus"

alias rman="rlwrap rman"

alias lsnrctl="rlwrap lsnrctl"

alias adrci="rlwrap adrci"

alias dgmgrl="rlwrap dgmgrl"

alias asmcmd="rlwrap asmcmd"

# Create listener on Standby

vi /u01/app/19.0.0/grid/network/admin/listener.ora

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC = (GLOBAL\_DBNAME = olab.hcnet.vn)(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)(SID\_NAME = st2olab) )

(SID\_DESC =(GLOBAL\_DBNAME = olab\_DGMGRL.hcnet.vn)(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)(SID\_NAME = st2olab)))

# Use tnsping to check connection

In Primary

source /home/oracle/.bash\_profile && /u01/app/oracle/product/19.0.0/dbhome\_1/bin/tnsping st2olab

In Standby

source /home/oracle/.bash\_profile && /u01/app/oracle/product/19.0.0/dbhome\_1/bin/tnsping olab

# Check force logging

In primary, use sqlplus as sysdba

SELECT force\_logging FROM v$database;

ALTER DATABASE FORCE LOGGING;

SELECT force\_logging FROM v$database;

# Create standby logfile

vi /u01/stage/ stb\_standby\_redolog.sql

ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=MANUAL;

ALTER DATABASE ADD STANDBY LOGFILE '+ASMDISK' SIZE 200M;

ALTER DATABASE ADD STANDBY LOGFILE '+ASMDISK' SIZE 200M;

ALTER DATABASE ADD STANDBY LOGFILE '+ASMDISK' SIZE 200M;

ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=AUTO;

Execute with sysdba in primary

@/u01/stage/ stb\_standby\_redolog.sql;

# Setup parameter

On primary db

alter system set LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(olab,st2olab)' scope=spfile;

alter system set LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=olab' scope=spfile;

alter system set LOG\_ARCHIVE\_DEST\_2='SERVICE="st2olab" LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=st2olab' scope=spfile;

alter system set LOG\_ARCHIVE\_DEST\_STATE\_1=enable scope=spfile;

alter system set FAL\_SERVER=st2olab scope=spfile;

alter system set FAL\_CLIENT=olab scope=spfile;

# Check password file on primary

As user grid

source /home/grid/.bash\_profile && asmcmd pwget --dbuniquename olab

# Create password file on primary

As user oracle

source /home/oracle/.bash\_profile && orapwd file='+ASMDISK' dbuniquename='olab' password=Password1 format=12 ENTRIES=2 sysdg=y <<< Password1

# Copy password file to /tmp

As user grid

source /home/grid/.bash\_profile && asmcmd pwcopy +ASMDISK/OLAB/PASSWORD/pwdolab.272.1132064825 /tmp/pwdolab

# Copy password file to standby

As user oracle

cp /tmp/pwdolab /u01/app/oracle/product/19.0.0/dbhome\_1

# Create pfile in stanby

As user oracle

vi /u01/app/oracle/product/19.0.0/dbhome\_1/dbs/pfilest2olab.ora

.db\_block\_size=8192

\*.db\_name='olab'

\*.db\_unique\_name='st2olab'

\*.fal\_server='olab'

\*.fal\_client='st2olab'

\*.log\_archive\_config='DG\_CONFIG=(olab,st2olab)'

\*.DB\_FILE\_NAME\_CONVERT=('//olab/','//st2olab/')

\*.LOG\_FILE\_NAME\_CONVERT=('+ASMDISK/olab/','+ASMDISK/st2olab/')

# Create dump location in standby

As user oracle

mkdir -p /u01/app/oracle/admin/st2olab/adump

# Startup nomount in standby

As user oracle

cd /u01/app/oracle/product/19.0.0/dbhome\_1/dbs

startup nomount pfile=pfilest2olab.ora

# Reload listener on both primary and standby

As user grid

source /home/grid/.bash\_profile && lsnrctl reload

# Create duplicate script in primary

As user oracle

vi /home/oracle/bin/ rman\_duplicate\_stb.sh

#!/bin/bash

#

# Date: 2020-11-27

#

# phat.trant@homecredit.vn

#

rman target sys/Password1@olab auxiliary sys/Password1@st2olab << EOF

run {

allocate channel prmy1 type disk;

allocate auxiliary channel stby type disk;

duplicate target database for standby from active database dorecover NOFILENAMECHECK

spfile

parameter\_value\_convert 'olab','st2olab'

set db\_name='olab'

set db\_unique\_name='st2olab'

set db\_file\_name\_convert='+ASMDISK/olab/','+ASMDISK/st2olab/'

set log\_file\_name\_convert='+ASMDISK/olab/','+ASMDISK/st2olab/'

set control\_files='+ASMDISK','+ASMDISK'

set fal\_client='st2olab'

set fal\_server='olab'

set db\_recovery\_file\_dest='+ASMDISK'

set standby\_file\_management='AUTO'

set log\_archive\_config='dg\_config=(olab,st2olab)'

set LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=olab'

set LOG\_ARCHIVE\_DEST\_2='SERVICE=st2olab LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=st2olab'

;

}

EXIT;

# Execute script

As user oracle in primary

/home/oracle/bin/ rman\_duplicate\_stb.sh

# Error (ignore if OK)

RMAN-00571: ===========================================================

RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============

RMAN-00571: ===========================================================

RMAN-03002: failure of Duplicate Db command at 03/22/2023 16:27:12

RMAN-05501: aborting duplication of target database

RMAN-06136: Oracle error from auxiliary database: ORA-00200: control file could not be created

ORA-00202: control file: '+ASMDISK'

ORA-17502: ksfdcre:4 Failed to create file +ASMDISK

ORA-15001: diskgroup "ASMDISK" does not exist or is not mounted

ORA-01034: ORACLE not available

ORA-27121: unable to determine size of shared memory segment

Linux-x86\_64 Error: 13: Permission denied

Additional information: 8210

Additional information: 5

Solution:

As user root

Chmod 06751 $ORACLE\_HOME/bin/oracle

# Create stb\_mrp script in standby

As user oracle

vi /home/oracle/bin/stb\_mrp.sh

sqlplus / as sysdba << EOF

show parameter standby\_archive\_dest;

show parameter DB\_RECOVERY\_FILE\_DEST;

alter system set STANDBY\_ARCHIVE\_DEST='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST OPTIONAL';

alter system set "\_query\_on\_physical"=false scope=spfile;

alter database recover managed standby database disconnect from session;

alter system set LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=st2olab' scope=both sid='\*';

alter system set LOG\_ARCHIVE\_DEST\_2='SERVICE=olab LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=olab' scope=both sid='\*';

shu abort;

startup mount;

EXIT;

EOF

# Execute start MRP script in standby

As user oracle

/home/oracle/bin/stb\_mrp.sh

# Enable dataguard broker in primary

As user oracle

alter system set dg\_broker\_config\_file1='+ASMDISK/OLAB/dr1olab.dat' sid='\*';

alter system set dg\_broker\_config\_file2='+ASMDISK/OLAB/dr2olab.dat' sid='\*';

alter system set dg\_broker\_start=true;

# Enable dataguard broker in standby

As user oracle

alter system set dg\_broker\_config\_file1='+ASMDISK/ST2OLAB/dr1st2olab.dat' sid='\*';

alter system set dg\_broker\_config\_file2='+ASMDISK/ST2OLAB/dr2st2olab.dat' sid='\*';

alter system set log\_archive\_max\_processes=4 scope=spfile;

alter system set log\_archive\_min\_succeed\_dest=1 scope=spfile;

alter system set archive\_lag\_target=0 scope=spfile;

alter system set DATA\_GUARD\_SYNC\_LATENCY=0 scope=spfile;

alter system set dg\_broker\_start=true;

# Reset log\_archive\_dest\_2 in standby

alter system set LOG\_ARCHIVE\_DEST\_2='';

# Create script and execute for data guard broker

As user oracle

vi /home/oracle/bin/dgmgrl\_config.sh

dgmgrl / << EOF

create configuration 'dgOLAB' as primary database is 'OLAB' connect identifier is OLAB;

add database 'ST2OLAB' as connect identifier is ST2OLAB maintained as physical ;

enable configuration ;

EXIT;

EOF

dgmgrl sys/<pass> << EOF

edit database 'ST2OLAB' set property 'LogArchiveTrace'=0;

edit database 'ST2OLAB' set property 'LogArchiveFormat'='%t\_%s\_%r.dbf';

edit database 'ST2OLAB' set property StandbyArchiveLocation='USE\_DB\_RECOVERY\_FILE\_DEST';

edit database 'ST2OLAB' set state='APPLY-ON';

EOF

/home/oracle/bin/dgmgrl\_config.sh

# Set parameter in standby

alter system set LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=st2olab' scope=spfile;

alter system set LOG\_ARCHIVE\_DEST\_2='SERVICE=olab LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=olab' scope=spfile;

alter system set LOG\_ARCHIVE\_DEST\_STATE\_2=enable;

# Backup parameter file in standby

create pfile='/tmp/initst2.ora' from spfile;

# Add database on standby server

As user oracle

source /home/oracle/.bash\_profile && srvctl add database -db st2olab -oraclehome $ORACLE\_HOME -role PHYSICAL\_STANDBY -startoption MOUNT -dbname olab

# Start mount database on standby

As user oracle

source /home/oracle/.bash\_profile && srvctl start database -d st2olab -o MOUNT