❁ 59. 데이터 가드 구현하기

```
회사에서 제일 중요한 데이터가 있는 메인 서버가 있는데 이 메인 서버가 있는 서버실에
불이 나게 되면 rman으로 열심히 백업을 많이 해놨어도 복구를 할 수 없는 상황이 된다.

DATA GUARD
- 운영 중에 데이터 베이스가 하나 이상의 standy db와 연동하여 운영 중인 데이터 베이스에서 발생할
수 있는 오류와 손상으로부터 데이터를 보호해주는 메커니즘
- 서버실 화재, 지진, 홍수, 테러로 인한 건물 붕괴로 인해서 서버가 손상되었을 때를 대비해서
원격에다가 별도의 서버를 항상 준비를 시켜주는 메커니즘

primary db ------------------------ standy db
현재 운영중인 서버                      복구만 하고 있는 서버
archive log file -------------------> 복구를 하고 있는 상태
```

```
※ Primary DB 구성 방법
1. .bash_profile을 구성한다. (오라클로 접속하기 위한 환경 구성 파일)
(PROD)
vi .bash_profile
--------------------
esc + ':%s/psh/PROD/g'

-> psh를 모두 PROD로 바꾼다.
--------------------
source .bash_profile
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export JAVA_HOME=/usr/java/jdk1.6.0_18
export PATH=$JAVA_HOME/bin:$PATH
export PS1='[`echo $ORACLE_SID`:\W]$ '

export ORACLE_BASE=/u01/app/oracle
export ORACLE_SID=psh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
unset LANG

alias alert='cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace'
alias net='cd $ORACLE_HOME/network/admin'
alias ss='sqlplus sys/oracle_1234  as sysdba'
alias scott='sqlplus scott/tiger'
alias dbs='cd $ORACLE_HOME/dbs'
alias oradata='cd /u01/app/oracle/oradata/psh/'
alias r='rman target sys/oracle_1234 nocatalog'

export LANG=ko_KR.UTF-8
export GRID_HOME=/u01/app/oracle/product/11.2.0/grid

NLS_LANG=american_america.we8iso8859p15
NLS_DATE_FORMAT='RRRR/MM/DD:HH24:MI:SS'
export NLS_LANG
export NLS_DATE_FORMAT
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:%s/psh/PROD/g
```

2. database 를 생성할 디렉토리를 생성한다.
(PROD)

```
mkdir -p /u01/app/oracle/oradata/PROD

oradata

mkdir disk1 disk2 disk3 disk4 disk5

3. parameter file 을 만든다.
cd $ORACLE_HOME/dbs
-> initPROD.ora 또는 spfilePROD.ora 파일이 있다면 rm으로 지워야 한다.

vi $ORACLE_HOME/dbs/initPROD.ora
-----------------------------------
db_name = PROD
compatible=11.2.0.1.0

sga_target = 256M
```

```
undo_management = AUTO
undo_tablespace = UNDOTBS

processes = 100

remote_login_passwordfile = EXCLUSIVE

control_files = (/u01/app/oracle/oradata/PROD/disk1/ctrl1.ctl,
                 /u01/app/oracle/oradata/PROD/disk2/ctrl2.ctl,
                 /u01/app/oracle/oradata/PROD/disk3/ctrl3.ctl)
-----------------------------------
4. instance 를  nomount 로 올린다.
sqlplus / as sysdba

(PROD sys)
startup  nomount  pfile=$ORACLE_HOME/dbs/initPROD.ora
@i
```

```
PROD(SYS) > startup  nomount  pfile=$ORACLE_HOME/dbs/initPROD.ora
ORACLE instance started.

Total System Global Area  267825152 bytes
Fixed Size                  1335924 bytes
Variable Size              92278156 bytes
Database Buffers          167772160 bytes
Redo Buffers                6438912 bytes
PROD(SYS) > @i

INSTANCE_NAME    STATUS
---------------- ------------
PROD             STARTED
```

```
5. create database 스크립트를 수행한다.
- database 를 생성한다는것은 어떠한 파일들을 생성하면 db 가 생성되는 것인가 ?
1. data file
2. control file
3. redo log file
4. parameter file
5. archive log file
6. password file

- database 를 수동으로 생성하는 스크립트를 수행하면 위의 파일중 3가지가 생성된다.

1. data file
2. control file
3. redo log file

create database PROD
user sys identified by oracle
user system identified by oracle
datafile '/u01/app/oracle/oradata/PROD/disk1/system01.dbf'
size 100M autoextend on maxsize unlimited extent management local
sysaux
datafile '/u01/app/oracle/oradata/PROD/disk2/sysaux01.dbf'
size 50M autoextend on maxsize unlimited
default temporary tablespace temp
tempfile '/u01/app/oracle/oradata/PROD/disk3/temp01.dbf'
size 50M autoextend on maxsize unlimited
undo tablespace undotbs
datafile '/u01/app/oracle/oradata/PROD/disk4/undotbs01.dbf'
size 50M autoextend on maxsize unlimited
```

```
logfile
group 1 ('/u01/app/oracle/oradata/PROD/disk4/redoG1M1.rdo',
        '/u01/app/oracle/oradata/PROD/disk5/redoG1M2.rdo') size 100M,
group 2 ('/u01/app/oracle/oradata/PROD/disk4/redoG2M1.rdo',
        '/u01/app/oracle/oradata/PROD/disk5/redoG2M2.rdo') size 100M,
group 3 ('/u01/app/oracle/oradata/PROD/disk4/redoG3M1.rdo',
        '/u01/app/oracle/oradata/PROD/disk5/redoG3M2.rdo') size 100M,
group 4 ('/u01/app/oracle/oradata/PROD/disk4/redoG4M1.rdo',
        '/u01/app/oracle/oradata/PROD/disk5/redoG4M2.rdo') size 100M,
group 5 ('/u01/app/oracle/oradata/PROD/disk4/redoG5M1.rdo',
         '/u01/app/oracle/oradata/PROD/disk5/redoG5M2.rdo') size 100M;

@i

-> db가 만들어지면 open 상태로 올라온다.
```

```
PROD(SYS) >
PROD(SYS) > @i

INSTANCE_NAME    STATUS
---------------- ------------
PROD             OPEN
```

```
6. data dictionary 를 생성하는 스크립트를 수행한다.
(PROD sys)
@$ORACLE_HOME/rdbms/admin/catalog.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql

connect system/oracle

(PROD system)
@$ORACLE_HOME/sqlplus/admin/pupbld.sql

7. PROD DB 에 운영 DATA 생성 (scott 계정 생성하고  demobld 스크립트를 수행한다.)
(PROD sys)
create user scott identified by tiger;

grant dba to scott;

connect scott/tiger

(PROD scott)
@demobld

select count(*) from emp;

(PROD)
cd $ORACLE_HOME/sqlplus/admin
vi glogin.sql
------------------------------
define_editor='vi'
-> 이미 있으면 추가 안해도 된다.
------------------------------
```

```
--   This script is automatically run
--
define _editor='vi'
set sqlprompt "_connect_identifier'('_user')' > "
~
~
~
~
~
~
```

※ Standby DB 구성 방법
1. standby DB 생성시 필요한 환경을 Primary DB 에 설정해준다.
- Data Guard 를 구성하기 위해서 Primaryh DB 쪽에 필요한 환경설정 3가지
1-1. Password file 인증 방법 이어야한다.
(PROD sys)
show parameter remote_login_passwordfile

```
PROD(SYS) > show  parameter  remote_login_passwordfile

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
remote_login_passwordfile            string      EXCLUSIVE
```

1-2. Archive log mode여야한다
archive log list

```
PROD(SYS) > archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            /u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch
Oldest online log sequence     3
Current log sequence           7
```

shutdown immediate
startup mount
alter database archivelog;
alter database open;
archive log list

```
PROD(SYS) > archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch
Oldest online log sequence     3
Next log sequence to archive   7
Current log sequence           7
```

1-3. Force logging이 활성화 되어 있어야 한다. (무조건 log 정보가 생기게 하겠다.)
alter database force logging;

alter database add supplemental log data (primary key, unique index) columns;
-> primary db에서 수행한 dml 문장들이 저장이 된다.

alter system archive log current;
@logsw

select name from v$archived_log;

select FORCE_LOGGING from v$database;

```
PROD(SYS) > select name from v$archived_log;

NAME
--------------------------------------------------------------------------------
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch1_7_1163599696.dbf
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch1_8_1163599696.dbf
```

```
PROD(SYS) > select FORCE_LOGGING from v$database;

FOR
---
YES
```

2. Primary DB 의 파라미터 파일을 수정한다.
(PROD)
cd $ORACLE_HOME/dbs
vi initPROD.ora
---
db_unique_name=PROD
-> standby db 쪽의 db 이름도 PROD일것이므로 primary db 쪽에 db_unique_name 을 PROD 라고 세팅 해야한다.

standby_file_management=auto
-> primary db 쪽에서 테이블 스페이스를 생성하면 standby db 쪽에서도 똑같은 테이블 스페이스가 자동으로 만들어지게 하는 파라미터

db_file_name_convert='/home/oracle/SBDB','/u01/app/oracle/oradata/PROD'
-> Primary db쪽에는 data file 이 /u01/app/oracle/oradata/PROD 에 위치해 있고
Standby db 쪽에는 data file 이 /home/oracle/SBDB 에 위치해 있다는 것을 알려준다.

log_file_name_convert='/home/oracle/SBDB','/u01/app/oracle/oradata/PROD'
-> Primary db쪽에는 redo log file 이 /u01/app/oracle/oradata/PROD 에 위치해 있고
Standby db 쪽에는 redo log file 이 /home/oracle/SBDB 에 위치해 있다는 것을 알려준다.

log_archive_dest_1='location=/home/oracle/PROD/arch valid_for=(all_logfiles,all_roles)'
-> primary db 쪽에서 생성될 archive log file의 위치
---
cd
mkdir PROD
cd PROD
mkdir arch

cd arch
pwd

```
[PROD:PROD]$ cd arch
[PROD:arch]$ pwd
/home/oracle/PROD/arch
```

dbs
vi initPROD.ora
---
log_archive_dest_2='service=SBDB LGWR SYNC AFFIRM valid_for=(online_logfiles,primary_roles)'
-> primary db 쪽에 archive log file의 위치

#fal_server=SBDB
#fal_client=PROD
-> standby db 쪽에 생성할 아카이브 로그파일의 위치

-> 나중에 primary db 가 standby db 가 될수있기 때문에 미리 적어놓은 파라미터

```
db_name = PROD
compatible=11.2.0.1.0

sga_target = 256M

undo_management = AUTO
undo_tablespace = UNDOTBS

processes = 100

remote_login_passwordfile = EXCLUSIVE

control_files = (/u01/app/oracle/oradata/PROD/disk1/ctrl1.ctl,
                 /u01/app/oracle/oradata/PROD/disk2/ctrl2.ctl,
                 /u01/app/oracle/oradata/PROD/disk3/ctrl3.ctl)

db_unique_name=PROD
standby_file_management=auto
db_file_name_convert='/home/oracle/SBDB','/u01/app/oracle/oradata/PROD'
log_file_name_convert='/home/oracle/SBDB','/u01/app/oracle/oradata/PROD'
log_archive_dest_1='location=/home/oracle/PROD/arch valid_for=(all_logfiles,all_role)'
log_archive_dest_2='service=SBDB LGWR SYNC AFFIRM valid_for=(online_logfiles,primary_role)'

#fal_server=SBDB
#fal_client=PROD
```

3. Standby DB 를 위한 Directory 를 생성
(PROD)
mkdir -p /home/oracle/SBDB

4. 모든 datafile, temp file을 standby DB로 전달
(PROD sys)
shutdown immediate

(PROD)
cp -rp /u01/app/oracle/oradata/PROD/* /home/oracle/SBDB

cd /home/oracle/SBSB/
ls

```
[PROD:~]$ cd SBDB/
[PROD:SBDB]$ ls
disk1  disk2  disk3  disk4  disk5
```

5. Primary DB 에서 standby DB 를 위한 controlfile 을 생성한 후, 전달한다.
(PROD)
mkdir -p /home/oracle/PROD/arch -> archive log file의 위치
mkdir -p /home/oracle/PROD/flash -> flashback database log 위치

6. standby db 쪽에 archive log file 과 flasback database log 가 저장될 위치를 생성하시오.
mkdir -p /home/oracle/SBDB/arch
mkdir -p /home/oracle/SBDB/flash

=> 이 때까지 primary db에서 standy db로 모든 data file들, tempfile, redo log file을 넘겼다.

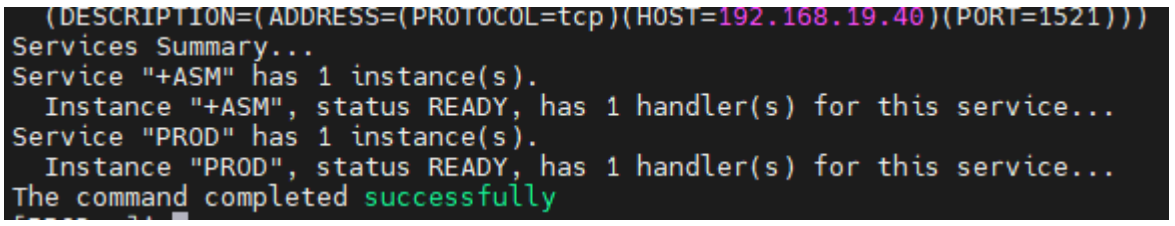
7. PROD DB 를 mount 로 올리고 standby 용 controlfile 을 생성한다.

(PROD sys)
startup monut

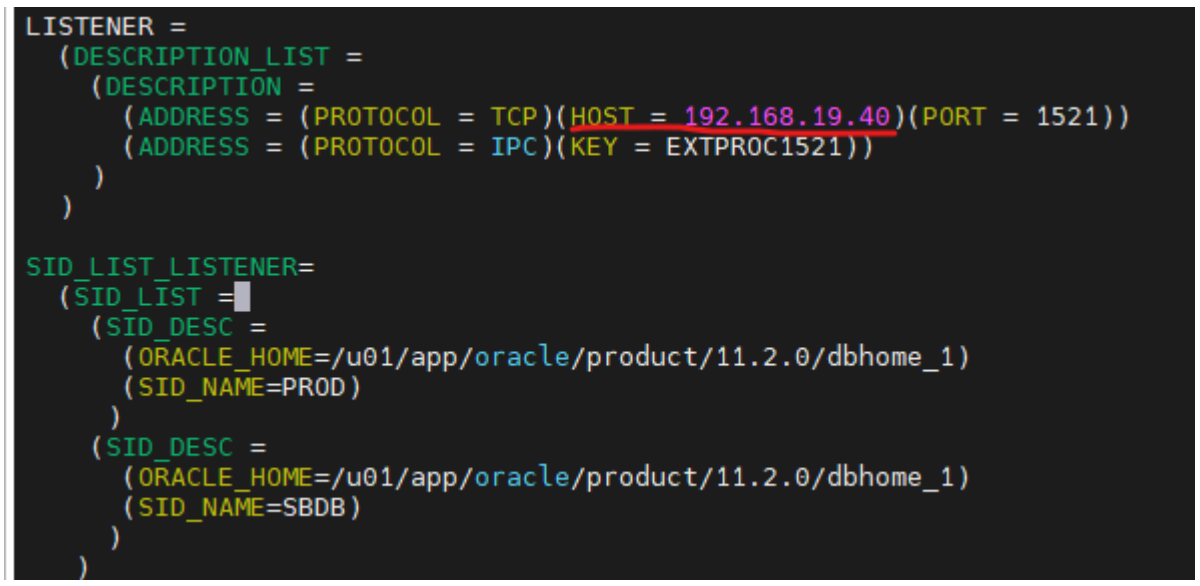
alter database create standby controlfile as '$HOME/physical.ctl' reuse;

8. Primary DB 쪽에서 생성한 standby 용 controlfile 을 Standby 쪽으로 넘겨준다.

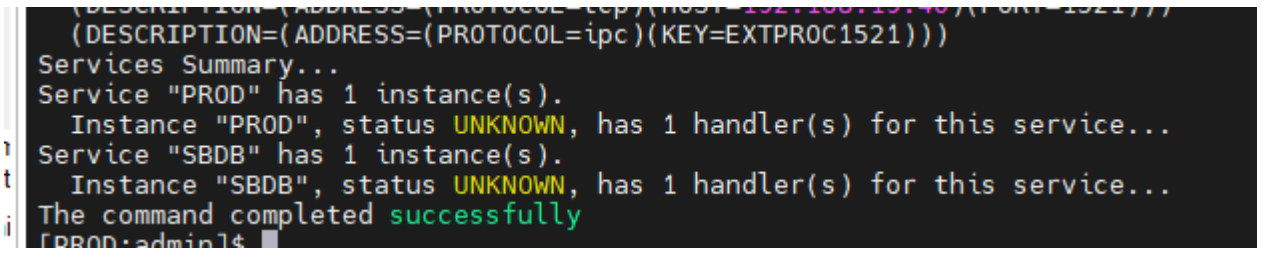
```
(PROD)
cp $HOME/physical.ctl  /home/oracle/SBDB/disk1/ctrl1.ctl
cp $HOME/physical.ctl  /home/oracle/SBDB/disk2/ctrl2.ctl
cp $HOME/physical.ctl  /home/oracle/SBDB/disk3/ctrl3.ctl

9. 리스너가 인식하고 있는 서비스가 무엇인지 확인한다. 동적 서비스 등록을 정적 서비스 등록으로
변경한다.
lsnrctl  status
```




```
cd $ORACLE_HOME/network/admin

mv listener.ora listener.bak

vi listener.ora
--------------------------------------
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER=
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME=PROD)
     )
    (SID_DESC =
      (ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME=SBDB)
     )
  )
----------------------------------------
```



```
lsnrctl stop
lsnrctl start
```




```
10. tnsnames.ora 파일의 아래의 내용을 추가한다.
cd  $ORACLE_HOME/network/admin
vi  tnsnames.ora
-----------------------------------------
PROD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PROD)
    )
  )

SBDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SBDB)
    )
  )
-----------------------------------------

11. PROD 와 SBDB 를 위한 패스워드 파일 생성
(PROD)
dbs
orapwd  file=orapwSBDB  password=oracle  ignorecase=Y
orapwd  file=orapwPROD  password=oracle  ignorecase=Y

-> 패스워드파일을 생성하는 이유는 원격에서 리스너 통해서 sys 유져로 접속하려고 생성
ignorecase=Y 는 패스워드 oracle 을 대문자로 쓰든 소문자로 쓰든 상관없이 접속하게 하겠다.

12. 위에서 구성한 nework 설정이 정상인지 확인하시오.
sqlplus  sys/oracle@prod as sysdba
sqlplus  sys/ORACLE@prod as sysdba
sqlplus  sys/oracle@sbdb as sysdba
sqlplus  sys/ORACLE@sbdb as sysdba

13. .bash_profile 을 열어서 prod 와 sbdb 로 접근하기 위한 alias 를 생성한다.
(PROD)
vi  .bash_profile
-------------------------------------
alias  prod='export ORACLE_SID=PROD'
alias  sbdb='export ORACLE_SID=SBDB'
-------------------------------------
source .bash_profile
```

```
sbdb
prod

(PROD sys)
shutdown immediate
startup mount

14. SBDB  용 파라미터 파일 구성
(PROD)
dbs
vi initSBDB.ora
-----------------------------------
compatible=11.2.0.1.0
control_files = (/home/oracle/SBDB/disk1/ctrl1.ctl ,
                 /home/oracle/SBDB/disk2/ctrl2.ctl ,
                 /home/oracle/SBDB/disk3/ctrl3.ctl )
db_block_size=8192
db_name=PROD
service_names=SBDB
global_names=true
job_queue_processes=10
open_cursors=500
processes=100
remote_login_passwordfile='EXCLUSIVE'
sga_max_size=120M
sga_target=120M
undo_management='AUTO'
undo_tablespace='UNDOTBS'
db_recovery_file_dest_size=4G
db_recovery_file_dest=/home/oracle/SBDB/flash
db_unique_name=SBDB
standby_file_management=auto
db_file_name_convert='/u01/app/oracle/oradata/PROD','/home/oracle/SBDB'
log_file_name_convert='/u01/app/oracle/oradata/PROD','/home/oracle/SBDB'
log_archive_dest_1='location=/home/oracle/SBDB/arch valid_for=(all_logfiles, all_roles)'
log_archive_dest_2='service=PROD LGWR SYNC AFFIRM valid_for=(online_logfiles, primary_role)'
#standby_archive_dest=/home/oracle/SBDB/arch
#recovery_parallelism=4
fal_server=PROD
fal_client=SBDB
-----------------------------------
※ standby DB 쪽에서 반드시 활성화 되어져야하는 파라미터
  1. fal_server=PROD  -> 아카이브 로그파일을 보내줘야하는 DB
  2. fal_client=SBDB  -> 아카이브 로그파일을 받아야하는 DB

15. standby 용 redo log file 을 prod 와 sbdb 에서 각각 생성
- PROD
(PROD sys)
alter  database  add   standby  logfile
'/u01/app/oracle/oradata/PROD/disk1/standby01.log'  size  10m;

alter  database  add   standby  logfile
'/u01/app/oracle/oradata/PROD/disk1/standby02.log'  size  10m;

(PROD)
sbdb
- SBDB
(SBDB sys)
```

```
startup

alter  database  add   standby  logfile
'/home/oracle/SBDB/disk1/standby01.log'  size  10m;

alter  database  add   standby  logfile
'/home/oracle/SBDB/disk1/standby02.log'  size  10m;

16. PROD 와 SBDB의 PMON이 잘 떠있는지 확인한다.
(SBDB)
ps -ef | grep pmon
```

```
[SBDB:~]$   ps -ef | grep pmon
oracle    5338     1  0 09:45 ?        00:00:04 asm_pmon_+ASM
oracle   12236     1  0 15:59 ?        00:00:00 ora_pmon_PROD
oracle   12318     1  0 16:06 ?        00:00:00 ora_pmon_SBDB
oracle   12375  5062  0 16:07 pts/1    00:00:00 grep pmon
```

```
※※※ 여기서 부터가 중요  (sbdb 를 복구모드로 변환하는 작업) ※※※
17. standby 디비인 SBDB 를 복구모드로 변환한다.
(SBDB sys)
shutdown  immediate
startup mount

alter  database  flashback  on;

-> MRP(Media Recovery Process)가 떠 있는지 확인하기 위해 아래의 작업수행
select process, status  from  v$managed_standby;

-> MRP가 아직 안 뜬다.
```

```
SBDB(SYS) > select process, status  from  v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
```

```
- 복구모드로 변환하는 명령어를 수행한다.
recover  managed  standby  database  disconnect;


select process, status  from  v$managed_standby;
-> MRP가 뜬다.
```

```
SBDB(SYS) > recover  managed  standby  database  disconnect;
Media recovery complete.
SBDB(SYS) > select process, status  from  v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
MRP0      WAIT_FOR_LOG
```

```
18. prod 와 sbdb 쪽에서 각각 아래의 명령어를 수행해서 datafile 의 개수가 똑같은지 확인하시오.
(PROD sys)
select  name from v$datafile;
```

```
(SBDB sys)
select  name from v$datafile;
```

```
PROD(SYS) >   select  name from v$datafile;

NAME
--------------------------------------------------
/u01/app/oracle/oradata/PROD/disk1/system01.dbf
/u01/app/oracle/oradata/PROD/disk2/sysaux01.dbf
/u01/app/oracle/oradata/PROD/disk4/undotbs01.dbf
```

```
SBDB(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------
/home/oracle/SBDB/disk1/system01.dbf
/home/oracle/SBDB/disk2/sysaux01.dbf
/home/oracle/SBDB/disk4/undotbs01.dbf
```

```
19. prod 쪽 db 를 올리고 테이블스페이스를 생성해서 standby db 로 넘어가서 반영되는지 확인한다.
(PROD sys)
alter database open;

create tablespace ts9000 datafile '/u01/app/oracle/oradata/PROD/disk2/ts9000.dbf' size 10m;

20. 아래의 작업을 양쪽에서 각각 수행한다.
(PROD sys)
select  name from v$datafile;
@logsw
@ckpt

(SBDB sys)
select  name from v$datafile;

=> primary db는 open 된 상태이고 standy db는 mount상태다. standby db는 mount에서 계속 복구만
한다.
```

```
PROD(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------
/u01/app/oracle/oradata/PROD/disk1/system01.dbf
/u01/app/oracle/oradata/PROD/disk2/sysaux01.dbf
/u01/app/oracle/oradata/PROD/disk4/undotbs01.dbf
/u01/app/oracle/oradata/PROD/disk2/ts9000.dbf
```

```
SBDB(SYS) > select name from v$datafile;

NAME
--------------------------------------------------
/home/oracle/SBDB/disk1/system01.dbf
/home/oracle/SBDB/disk2/sysaux01.dbf
/home/oracle/SBDB/disk4/undotbs01.dbf
/home/oracle/SBDB/disk2/ts9000.dbf
```

```
(PROD sys)
select process, status from v$managed_standby;
```

```
PROD(SYS) > select process, status from v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CLOSING
ARCH      CLOSING
ARCH      CONNECTED
ARCH      CLOSING
LGWR      WRITING
```

```
(SBDB sys)
select process, status from v$managed_standby;
```

```
SBDB(SYS) > select process, status from v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CLOSING
MRP0      WAIT_FOR_LOG
RFS       IDLE
RFS       IDLE
RFS       IDLE
```

⇒ WAIT_FOR_LOG : archive 된 redo 정보를 standby db에 적용

⇒ RFS (Remote File Server) : Primary db로부터 redo 정보를 수신하고 redo를 standby db의 redo log와 archive redo log에 직접 기록하는 프로세서

⇒ LNS (Logwrite Network Server) : redo를 standby db의 RFS에게 보낸다.

```
문제 1. prod 쪽에서 테이블 스페이스를 다음과 같이 생성하고 sbdb쪽에 관련 테이블 스페이가 똑같이
만들어지는지 확인하시오.
(PROD sys)
create tablespace ts801 datafile '/u01/app/oracle/oradata/PROD/disk2/ts801.dbf' size 10m;

@logsw
@ckpt
select name from v$datafile;

(SBDB sys)
select name from v$datafile;
```

```
PROD(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------------------
/u01/app/oracle/oradata/PROD/disk1/system01.dbf
/u01/app/oracle/oradata/PROD/disk2/sysaux01.dbf
/u01/app/oracle/oradata/PROD/disk4/undotbs01.dbf
/u01/app/oracle/oradata/PROD/disk2/ts9000.dbf
/u01/app/oracle/oradata/PROD/disk2/ts801.dbf
```

```
SBDB(SYS) > select name from v$datafile;

NAME
--------------------------------------------
/home/oracle/SBDB/disk1/system01.dbf
/home/oracle/SBDB/disk2/sysaux01.dbf
/home/oracle/SBDB/disk4/undotbs01.dbf
/home/oracle/SBDB/disk2/ts9000.dbf
/home/oracle/SBDB/disk2/ts801.dbf
```

문제 (이수자 평가)
영문 이니셜 db를 생성하고 영문 이니셜 db를 primary db로 생성하고 sbdb2라는 db를 standby db로
해서 데이터 가드를 구현하시오.
PRIMARY DB 구성
(PROD)
vi .bash_profile

아래의 그림처럼 변경
---
source .bash_profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export JAVA_HOME=/usr/java/jdk1.6.0_18
export PATH=$JAVA_HOME/bin:$PATH
export PS1='[`echo $ORACLE_SID`:\W]$ '

export ORACLE_BASE=/u01/app/oracle
export ORACLE_SID=psh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
unset LANG

alias alert='cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace'
alias net='cd $ORACLE_HOME/network/admin'
alias ss='sqlplus sys/oracle_1234  as sysdba'
alias scott='sqlplus scott/tiger'
alias dbs='cd $ORACLE_HOME/dbs'
alias oradata='cd /u01/app/oracle/oradata/psh/'
alias r='rman target sys/oracle_1234 nocatalog'
alias  prod='export ORACLE_SID=PROD'
alias  sbdb='export ORACLE_SID=SBDB'
```

(psh)
mkdir -p /u01/app/oracle/oradata/psh

oradata
mkdir disk1 disk2 disk3 disk4 disk5

cd $ORACLE_HOME/dbs

vi $ORACLE_HOME/dbs/initpsh.ora
---
db_name = psh
compatible=11.2.0.1.0

sga_target = 256M

undo_management = AUTO
undo_tablespace = UNDOTBS

processes = 100

remote_login_passwordfile = EXCLUSIVE

control_files = (/u01/app/oracle/oradata/psh/disk1/ctrl1.ctl,
                 /u01/app/oracle/oradata/psh/disk2/ctrl2.ctl,

```
                /u01/app/oracle/oradata/psh/disk3/ctrl3.ctl)
-----------------------------------

sqlplus / as sysdba

(psh sys)
startup nomount pfile=$ORACLE_HOME/dbs/initpsh.ora
@i
```

```
psh(SYS) > startup nomount pfile=$ORACLE_HOME/dbs/initpsh.ora
ORACLE instance started.

Total System Global Area  267825152 bytes
Fixed Size                  1335924 bytes
Variable Size              92278156 bytes
Database Buffers          167772160 bytes
Redo Buffers                6438912 bytes
psh(SYS) > @i

INSTANCE_NAME    STATUS
---------------- ------------
psh              STARTED
```

```
create database psh
user sys identified by oracle
user system identified by oracle
datafile '/u01/app/oracle/oradata/psh/disk1/system01.dbf'
size 100M autoextend on maxsize unlimited extent management local
sysaux
datafile '/u01/app/oracle/oradata/psh/disk2/sysaux01.dbf'
size 50M autoextend on maxsize unlimited
default temporary tablespace temp
tempfile '/u01/app/oracle/oradata/psh/disk3/temp01.dbf'
size 50M autoextend on maxsize unlimited
undo tablespace undotbs
datafile '/u01/app/oracle/oradata/psh/disk4/undotbs01.dbf'
size 50M autoextend on maxsize unlimited
logfile
group 1 ('/u01/app/oracle/oradata/psh/disk4/redoG1M1.rdo',
        '/u01/app/oracle/oradata/psh/disk5/redoG1M2.rdo') size 100M,
group 2 ('/u01/app/oracle/oradata/psh/disk4/redoG2M1.rdo',
        '/u01/app/oracle/oradata/psh/disk5/redoG2M2.rdo') size 100M,
group 3 ('/u01/app/oracle/oradata/psh/disk4/redoG3M1.rdo',
        '/u01/app/oracle/oradata/psh/disk5/redoG3M2.rdo') size 100M,
group 4 ('/u01/app/oracle/oradata/psh/disk4/redoG4M1.rdo',
        '/u01/app/oracle/oradata/psh/disk5/redoG4M2.rdo') size 100M,
group 5 ('/u01/app/oracle/oradata/psh/disk4/redoG5M1.rdo',
         '/u01/app/oracle/oradata/psh/disk5/redoG5M2.rdo') size 100M;

@i
```

```
(psh sys)
@$ORACLE_HOME/rdbms/admin/catalog.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql

connect system/oracle

(psh system)
@$ORACLE_HOME/sqlplus/admin/pupbld.sql
```

```
(psh sys)
create user scott identified by tiger;
grant dba to scott;
connect scott/tiger

(psh scott)
@demobld

select count(*) from emp;

(psh)
cd $ORACLE_HOME/sqlplus/admin
vi glogin.sql
-----------------------------
define_editor='vi'
-> 이미 있으면 추가 안해도 된다.
-----------------------------

STANDBY DB 구성
(psh sys)
show  parameter  remote_login_passwordfile
```

```
psh(SYS) > show  parameter  remote_login_passwordfile

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
remote_login_passwordfile            string      EXCLUSIVE
```

```
archive log list
```

```
psh(SYS) > archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            /u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch
Oldest online log sequence     3
Current log sequence           7
```

```
shutdown immediate
startup mount
alter database archivelog;
alter database open;
archive log list
```

```
psh(SYS) > archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch
Oldest online log sequence     3
Next log sequence to archive   7
Current log sequence           7
```

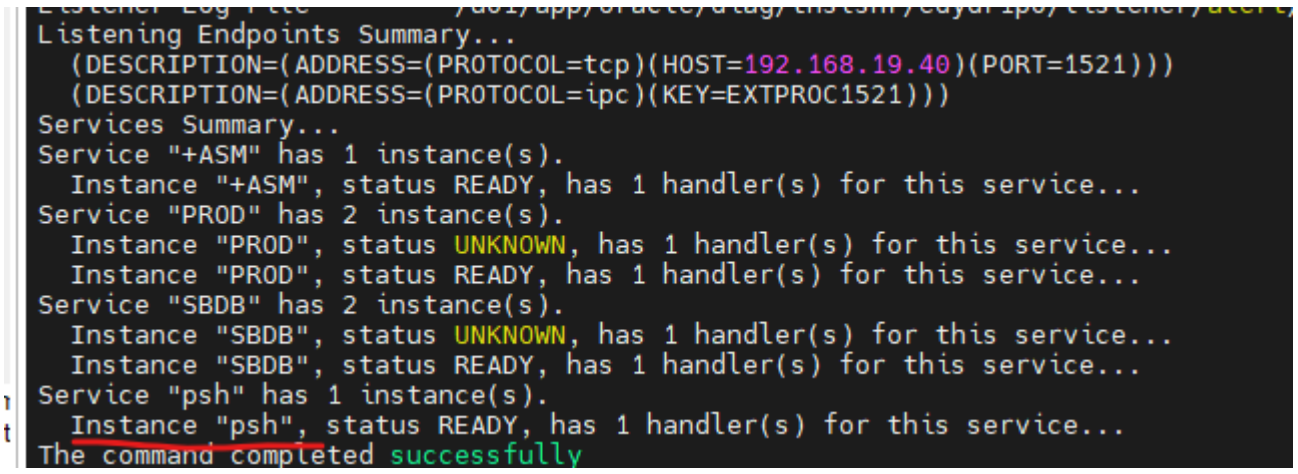
```
alter database force logging;
alter database add supplemental log data (primary key, unique index) columns;
alter system archive log current;
@logsw

select name from v$archived_log;
select FORCE_LOGGING from v$database;
```

```
psh(SYS) > select name from v$archived_log;

NAME
--------------------------------------------------------------------------------
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch1_7_1163608977.dbf
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/arch1_8_1163608977.dbf
```

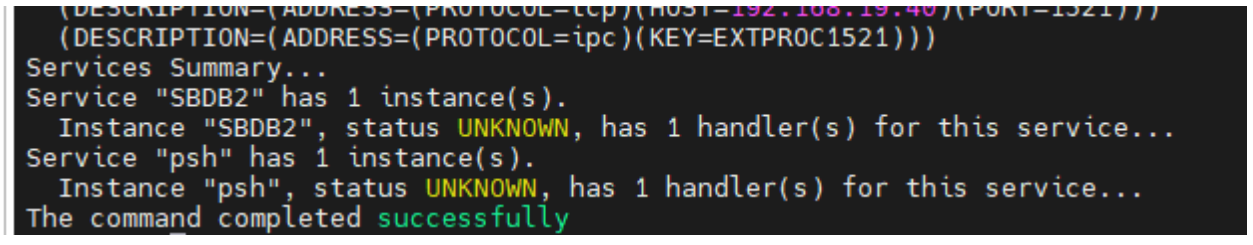
```
psh(SYS) > select FORCE_LOGGING from v$database;

FOR
---
YES
```

```
(psh)
dbs
vi initpsh.ora
-------------------------
db_unique_name=psh
standby_file_management=auto
db_file_name_convert='/home/oracle/SBDB2','/u01/app/oracle/oradata/psh'
log_file_name_convert='/home/oracle/SBDB2','/u01/app/oracle/oradata/psh'
log_archive_dest_1='location=/home/oracle/psh/arch valid_for=(all_logfiles,all_roles)'
log_archive_dest_2='service=SBDB2 LGWR SYNC AFFIRM valid_for=(online_logfiles,primary_roles)'
#fal_server=SBDB
#fal_client=PROD
-------------------------
(psh)
mkdir psh
cd psh
mkdir arch

mkdir -p /home/oracle/SBDB2

(psh sys)
shutdown immediate

(psh)
cp -rp /u01/app/oracle/oradata/psh/* /home/oracle/SBDB2

cd /home/oracle/SBDB2
ls

(psh)
mkdir -p /home/oracle/psh/arch
mkdir -p /home/oracle/psh/flash

mkdir -p /home/oracle/SBDB2/arch
mkdir -p /home/oracle/SBDB2/flash

(psh sys)
startup mount

alter database create standby controlfile as '$HOME/physical.ctl' reuse;

(psh)
cp $HOME/physical.ctl  /home/oracle/SBDB2/disk1/ctrl1.ctl
cp $HOME/physical.ctl  /home/oracle/SBDB2/disk2/ctrl2.ctl
cp $HOME/physical.ctl  /home/oracle/SBDB2/disk3/ctrl3.ctl
```

```
lsnrctl status
```



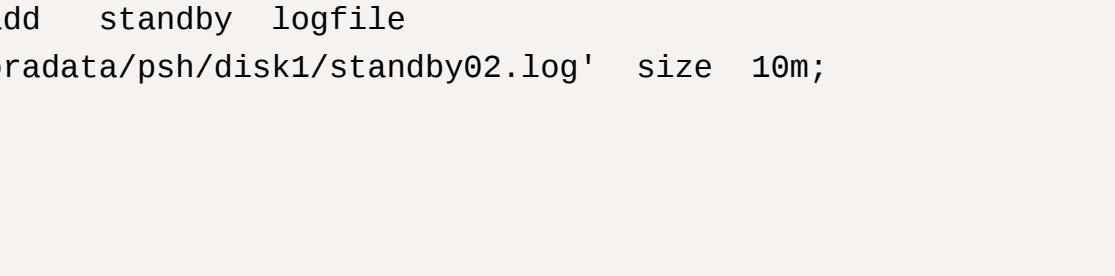

```
cd $ORACLE_HOME/network/admin
mv listener.ora listener.bak

vi listener.ora
---------------------------------------
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER=
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME=psh)
     )
    (SID_DESC =
      (ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME=SBDB2)
     )
   )
-----------------------------------------
lsnrctl stop
lsnrctl start
```




```
cd  $ORACLE_HOME/network/admin
vi  tnsnames.ora
-----------------------------------------
psh =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
```

```
      (SERVICE_NAME = psh)
    )
  )

SBDB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.19.40)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SBDB2)
    )
  )
-----------------------------------------
(psh)
dbs
orapwd  file=orapwSBDB2  password=oracle  ignorecase=Y
orapwd  file=orapwpsh  password=oracle  ignorecase=Y

sqlplus  sys/oracle@psh as sysdba
sqlplus  sys/ORACLE@psh as sysdba
sqlplus  sys/oracle@sbdb2 as sysdba
sqlplus  sys/ORACLE@sbdb2 as sysdba

(psh)
vi  .bash_profile
-------------------------------------
alias  psh='export ORACLE_SID=psh'
alias  sbdb2='export ORACLE_SID=SBDB2'
-------------------------------------
source .bash_profile

sbdb2
psh

(psh sys)
shutdown immediate
startup mount

(psh)
dbs
vi initSBDB2.ora
-----------------------------------
compatible=11.2.0.1.0
control_files = (/home/oracle/SBDB2/disk1/ctrl1.ctl ,
                 /home/oracle/SBDB2/disk2/ctrl2.ctl ,
                 /home/oracle/SBDB2/disk3/ctrl3.ctl )
db_block_size=8192
db_name=psh
service_names=SBDB2
global_names=true
job_queue_processes=10
open_cursors=500
processes=100
remote_login_passwordfile='EXCLUSIVE'
sga_max_size=120M
sga_target=120M
undo_management='AUTO'
undo_tablespace='UNDOTBS'
```

```
db_recovery_file_dest_size=4G
db_recovery_file_dest=/home/oracle/SBDB2/flash
db_unique_name=SBDB2
standby_file_management=auto
db_file_name_convert='/u01/app/oracle/oradata/psh','/home/oracle/SBDB2'
log_file_name_convert='/u01/app/oracle/oradata/psh','/home/oracle/SBDB2'
log_archive_dest_1='location=/home/oracle/SBDB2/arch valid_for=(all_logfiles, all_roles)'
log_archive_dest_2='service=psh LGWR SYNC AFFIRM valid_for=(online_logfiles, primary_role)'
#standby_archive_dest=/home/oracle/SBDB2/arch
#recovery_parallelism=4
fal_server=psh
fal_client=SBDB2
-----------------------------------
-psh
(psh sys)
alter  database  add   standby  logfile
'/u01/app/oracle/oradata/psh/disk1/standby01.log'  size  10m;

alter  database  add   standby  logfile
'/u01/app/oracle/oradata/psh/disk1/standby02.log'  size  10m;

(psh)
sbdb2

-SBDB2
(SBDB2 sys)
startup

alter  database  add   standby  logfile
'/home/oracle/SBDB2/disk1/standby01.log'  size  10m;

alter  database  add   standby  logfile
'/home/oracle/SBDB2/disk1/standby02.log'  size  10m;

(psh)
ps -ef | grep pmon
```




```
(SBDB2 sys)
shutdown  immediate
startup mount

alter database flashback on;

select process,status from v$managed_standby;
```

```
SBDB2(SYS) > select process, status  from  v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
```

```
recover  managed  standby  database  disconnect;

select process, status  from  v$managed_standby;
```

```
SBDB2(SYS) > select process, status  from  v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
MRP0      WAIT_FOR_LOG
```

```
(psh sys)
select name from v$datafile;

(SBDB2 sys)
select name from v$datafile;
```

```
psh(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/psh/disk1/system01.dbf
/u01/app/oracle/oradata/psh/disk2/sysaux01.dbf
/u01/app/oracle/oradata/psh/disk4/undotbs01.dbf
```

```
SBDB2(SYS) > select name from v$datafile;

NAME
--------------------------------------------------------------------------------
/home/oracle/SBDB2/disk1/system01.dbf
/home/oracle/SBDB2/disk2/sysaux01.dbf
/home/oracle/SBDB2/disk4/undotbs01.dbf
```

```
(psh sys)
alter database open;


create tablespace ts9000 datafile '/u01/app/oracle/oradata/psh/disk2/ts9000.dbf' size 10m;

select  name from v$datafile;
@logsw
@ckpt

(SBDB2 sys)
select  name from v$datafile;
```

```
psh(SYS) > psh(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/psh/disk1/system01.dbf
/u01/app/oracle/oradata/psh/disk2/sysaux01.dbf
/u01/app/oracle/oradata/psh/disk4/undotbs01.dbf
/u01/app/oracle/oradata/psh/disk2/ts9000.dbf
```

```
SBDB2(SYS) > select  name from v$datafile;

NAME
--------------------------------------------------
/home/oracle/SBDB2/disk1/system01.dbf
/home/oracle/SBDB2/disk2/sysaux01.dbf
/home/oracle/SBDB2/disk4/undotbs01.dbf
/home/oracle/SBDB2/disk2/ts9000.dbf
```

```
(psh sys)
select process, status from v$managed_standby;
```

```
psh(SYS) > select process, status from v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CLOSING
ARCH      CLOSING
ARCH      CONNECTED
ARCH      CLOSING
LGWR      WRITING
```

```
(SBDB2 sys)
select process, status from v$managed_standby;
```

```
SBDB2(SYS) > select process, status from v$managed_standby;

PROCESS   STATUS
--------- ------------
ARCH      CLOSING
ARCH      CONNECTED
ARCH      CONNECTED
ARCH      CONNECTED
MRP0      WAIT_FOR_LOG
RFS       IDLE
RFS       IDLE
RFS       IDLE

8 rows selected.
```