

셸 스크립트를 이용한 DATA GUARD 자동화

오라클과 파이썬을 활용한 데이터 전문가 양성과정
3기 오혜진

“DATA GUARD, 쉽게 자동화 구축 해보자!”

□ 개요

DATA GUARD 는 ORACLE 9i부터의 신기능으로 9i 의 목적인 e-business 의 효율을 극대화 시킬 기능 중 하나이다. 이전까지는 Primary DB에서 Standby DB로 리두 로그를 수동으로 복사 및 적용을 해야 했지만 9i 부터는 아카이브 로그뿐만 아니라 리두 로그 또한 자동적으로 적용하여 천재지변, 각종 장애로부터 보호해주는 메커니즘이다. 하나의 Primary DB 와 최대 9개의 Standby DB를 구성할 수 있으며 통신만 가능하다면 손상에 대한 복구를 자동화로 관리할 수 있다.

본문에서 다룰 내용은 이 DATA GUARD를 간편하게 생성하고 장애로부터 Primary DB 와 Standby DB를 SWITCHOVER를 하는 자동화 셸 스크립트에 대한 것이다.

□ DATA GUARD란

운영 중인 데이터베이스가 하나 이상의 Standby 데이터베이스와 연동하여 데이터베이스의 손상을 초래할 수 있는 장애와 오류로부터 데이터를 보호해주는 메커니즘이다.

□ DATA GUARD의 필요성

궁극적으로 천재지변, 각 종 H/W와 S/W의 장애로부터 기업의 중요 자원인 DB를 보호하기 위해 필요하며 수동방식이 아닌 자동화 구성으로 된 DATA GUARD는 데이터를 보호하는데 꼭 필요하다.

□ 필요한 패키지

- DataGuardAuto.zip
 - DataGuardAuto.sh
 - : Primary DB 와 Standby DB Create 및 SWITCHOVER 자동화 쉘 스크립트
 - catalog.sql, catproc.sql, pupbld.sql
 - : data dictionary 생성 스크립트
 - listener_insert.txt : listener.ora 삽입할 내용

□ 쉘 스크립트 구성

1. Primary DB Create
2. Standby DB Create
3. SWITCHOVER

□ 사전 작업

- .bash_profile
: \$HOME/.bash_profile를 백업본을 만듭니다. 기존에 ORACLE 환경변수를 지우지 않으려면 주석(#) 처리 해놓으십시오.
- listener.ora
: \$ORACLE_HOME/network/admin/listener.ora 에서 SID_LIST 내용 중에 가장 마지막 SID_NAME 이름을 확인해주세요. Standby 구성 시 필요합니다.
listener 에 사용할 IP를 확인합니다. 변경이 필요할 때 OS 상에서 수동으로 IP를 편집해야 합니다.
- tnsnames.ora
: tnsnames.ora 에 사용할 IP를 확인합니다. 변경이 필요할 때 OS 상에서 수동으로 편집합니다.
- DataGuardAuto.sh
: 셸 스크립트 동작 시 ORACLE_BASE부터 오라클 환경 변수를 .bash_profile에 삽입하는 코드가 있습니다. 오라클 경로를 확인하시고 default와 다르다면 셸 스크립트에서 수정 후 실행 바랍니다.

□ PRIMARY DATABASE 생성

- Primary DB 이름 입력합니다.

```

┌────────────────── LIST ───────────────────┐
|
| [1] PRIMARY DB
|
| [2] STANDBY DB
|
| [3] SWITCHOVER
|
└──────────────────┬──────────────────┘
| Usage : 해당 List 번호 입력 [ (ex) 1, 2, 3 ] |
└────────── EXIT : Ctrl-C ───────────┘

▶▶▶▶▶▶▶▶▶▶ INPUT [ LIST ] : 1

read c1

case $c1 in
    1)
        echo "Primary Database create start "

        echo -n "Primary Database name ? " Primary DB명
        read dname ( 영문 소문자 입력 )

        UPPER_DNAME=`echo $dname | tr '[a-z]' '[A-Z]`
        # tr 명령어로 소문자를 대문자로 변환하여 변수에 저장

```

- .bash_profile content 확인 및 삽입
sed 명령어 사용 시 “ ” (큰따옴표) 으로 사용하면 문자가 아닌 변수로 인식할 수 있게 됩니다.

```
# 기존 환경 변수를 화면에 출력
grep 'export ORACLE' $HOME/.bash_profile
grep 'export PATH' $HOME/.bash_profile
# 아래 내용을 .bash_profile에 추가한다고 화면 출력
echo " [ Insert the following contents in .bash_profile ]"
echo "ORACLE_BASE=/u01/app/oracle"
echo "ORACLE_HOME=W$ORACLE_BASE/product/11.2.0/db_1"
echo "ORACLE_SID=$UPPER_DNAME"
echo "PATH=W$ORACLE_HOME/bin:W$ORACLE_HOME/jdk/bin:W$PATH"
echo "alias $dname='export ORACLE_SID='$UPPER_DNAME''"
echo "unset LANG"

echo -n " INSERT OK ? y/n "
read yn

if [ $yn = 'y' -o $yn = 'Y' ]
then

##### remove 꺾칠 수 있는 환경변수 삭제
sed -i '/^export ORACLE/d' $HOME/.bash_profile
sed -i '/^export PATH=W$ORACLE/d' $HOME/.bash_profile
sed -i '/^unset/d' $HOME/.bash_profile
sed -i '/^alias $dname/d' $HOME/.bash_profile

##### insert
echo "export ORACLE_BASE=/u01/app/oracle" >> $HOME/.bash_profile
echo "export
ORACLE_HOME=W$ORACLE_BASE/product/11.2.0/dbhome_1" >>
$HOME/.bash_profile
echo "export ORACLE_SID=$UPPER_DNAME" >> $HOME/.bash_profile
echo "export
PATH=W$ORACLE_HOME/bin:W$ORACLE_HOME/jdk/bin:W$PATH" >>
$HOME/.bash_profile
echo "unset LANG" >> $HOME/.bash_profile
echo "alias $dname='export ORACLE_SID='$UPPER_DNAME''" >>
$HOME/.bash_profile
```

```
##### application
        source $HOME/.bash_profile
        echo " [ .bash_profile Insert Ok. ]"
    else
        echo " [ Not inserted. ]"
    fi
```

○ Primary DB Directory 생성

```
# 기존 디렉토리명과 같으면 삭제
rm -rf /u01/app/oracle/oradata/$UPPER_DNAME
mkdir -p /u01/app/oracle/oradata/$UPPER_DNAME
cd /u01/app/oracle/oradata/$UPPER_DNAME
mkdir disk1 disk2 disk3 disk4 disk5
```

○ Primary DB init.ora 생성

pfile 옵션은 특정 파라미터 파일을 지정해서 instance를 구성하려고 pfile 옵션을 사용하는 것이다.

※ 셸 스크립트 변경 시 유의 바랍니다. line 수가 맞지 않으면 명령어 동작시 잘못된 파일을 생성할 수 있습니다.

```
cd $ORACLE_HOME/dbs
rm -rf $ORACLE_HOME/dbs/init$UPPER_DNAME.ora
touch $ORACLE_HOME/dbs/init$UPPER_DNAME.ora
```

아래 주석처리된 내용으로 init.ora 생성

```
#####db_name = $UPPER_DNAME
#####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/$UPPER_DNAME/disk1/ctrl1.ctl ,
#####/u01/app/oracle/oradata/$UPPER_DNAME/disk2/ctrl2.ctl ,
#####/u01/app/oracle/oradata/$UPPER_DNAME/disk3/ctrl3.ctl )

# 위 주석 부분의 sed 명령어로 변수 값으로 치환
sed -i "106,115s/W$UPPER_DNAME/$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh
# sed 명령어로 치환된 init.ora 내용을 init.ora에 추가
sed -n '106,115p' $HOME/DataGuardAuto.sh >>
$ORACLE_HOME/dbs/init$UPPER_DNAME.ora
# init.ora 파일에 주석 제거
sed -i "s/#####/g" $ORACLE_HOME/dbs/init$UPPER_DNAME.ora
# 셸 스크립트에 변수값을 변수명으로 다시 변경
sed -i "106,115s/$UPPER_DNAME/W$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh

ls -al $ORACLE_HOME/dbs/init$UPPER_DNAME.ora
# 파일이 생성 되었는지 확인
if [ -e $ORACLE_HOME/dbs/init$UPPER_DNAME.ora ]; then
    echo " [ Primary DB init.ora create complete ]"
else
    echo " [ Primary DB init.ora create fail ]"
fi
```

○ Password file 생성

```
cd $ORACLE_HOME/dbs
# 기존 패스워드 파일을 삭제
rm -rf orapw$UPPER_DNAME
# 패스워드 파일 생성
orapwd file=orapw$UPPER_DNAME password=oracle
pwd
ls -al orapw$UPPER_DNAME
# 패스워드 파일 생성 확인
if [ -e $ORACLE_HOME/dbs/init$UPPER_DNAME ]; then

    echo " [ Primary DB Password file create complete ]"
else
    echo " [ Primary DB Password file create fail ]"
```

- Primary DB 생성
pfile을 이용해서 instance 구성합니다.

```
# pfile로 nomount 상태로 startup
sqlplus -s / as sysdba <<EOF
startup nomount pfile=$ORACLE_HOME/dbs/init$UPPER_DNAME.ora
EOF

# mount.txt에 mount 상태를 저장
sqlplus -s / as sysdba <<EOF
spool $HOME/mount.txt
select status from vW$instance;
spool off
EOF

STATUS_M=`grep STARTED $HOME/mount.txt | cut -d ' ' -f 1`
# 변수값이 STARTED 인지 확인
if [ $STATUS_M = 'STARTED' ]
then
    echo " [ $dname startup nomount ]"
else
    echo " [ $dname startup fail ]"
fi

# DB 생성시 User sys 와 system의 기본 패스워드를 'oracle' 로
# 지정 변경을 원한다면 쉘 스크립트 수정 요망
echo " User 'sys', 'system' Password is 'oracle'(default) " >>

# DB 생성 ( user, data file, redo log )
sqlplus -s / as sysdba <<EOF
create database $UPPER_DNAME
user sys identified by oracle
user system identified by oracle
datafile '/u01/app/oracle/oradata/$UPPER_DNAME/disk1/system01.dbf'
size 100M autoextend on maxsize unlimited extent management local
sysaux
datafile '/u01/app/oracle/oradata/$UPPER_DNAME/disk2/sysaux01.dbf'
size 50M autoextend on maxsize unlimited
```



```

default temporary tablespace temp
tempfile '/u01/app/oracle/oradata/$UPPER_DNAME/disk3/temp01.dbf'
size 50M autoextend on maxsize unlimited
undo tablespace undotbs
datafile '/u01/app/oracle/oradata/$UPPER_DNAME/disk4/undotbs01.dbf'
size 50M autoextend on maxsize unlimited
logfile
group 1 ('/u01/app/oracle/oradata/$UPPER_DNAME/disk4/redoG1M1.rdo',
'/u01/app/oracle/oradata/$UPPER_DNAME/disk5/redoG1M2.rdo') size
100M,
group 2 ('/u01/app/oracle/oradata/$UPPER_DNAME/disk4/redoG2M1.rdo',
'/u01/app/oracle/oradata/$UPPER_DNAME/disk5/redoG2M2.rdo') size
100M,
group 3 ('/u01/app/oracle/oradata/$UPPER_DNAME/disk4/redoG3M1.rdo',
'/u01/app/oracle/oradata/$UPPER_DNAME/disk5/redoG3M2.rdo') size
100M,
group 4 ('/u01/app/oracle/oradata/$UPPER_DNAME/disk4/redoG4M1.rdo',
'/u01/app/oracle/oradata/$UPPER_DNAME/disk5/redoG4M2.rdo') size
100M,
group 5 ('/u01/app/oracle/oradata/$UPPER_DNAME/disk4/redoG5M1.rdo',
'/u01/app/oracle/oradata/$UPPER_DNAME/disk5/redoG5M2.rdo') size
100M;
EOF
# nomount 상태에서 확인 가능한 redo log file, control_file 확인
sqlplus -s / as sysdba <<EOF
select member from vW$logfile;
show parameter control_files;
EOF

```

◦ Data Dictionary create script

```

sqlplus -s / as sysdba <<EOF
@$ORACLE_HOME/rdbms/admin/catalog.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql
EOF
# system user로 스크립트를 실행
sqlplus -s system/oracle <<EOF
@$ORACLE_HOME/sqlplus/admin/pupbld.sql
EOF

```

□ STANDBY DATABASE 생성

- Standby DB 이름, Primary DB 로 사용할 DB 이름 입력 그리고 사전 확인 작업한 마지막 SID_NAME을 입력.

```
2)
echo -n "Standby Database name ? " Standby DB명
read dname                      ( 영문 소문자 입력 )

echo -n "Primary Database name ? " Primary DB명
read pdname                      ( 영문 소문자 입력 )

# tr 명령어로 소문자를 대문자로 변환하여 변수에 저장
UPPER_DNAME=`echo $dname | tr 'a-z' 'A-Z'`
echo $UPPER_DNAME
UPPER_PDNAME=`echo $pdname | tr 'a-z' 'A-Z'`
echo $UPPER_PDNAME
# Standby DB alias를 .bash_profile에 추가

echo "Insert Standby DB alias"
echo "alias $dname='export ORACLE_SID='$UPPER_DNAME''" >>
$HOME/.bash_profile
source $HOME/.bash_profile

echo -n "Listener last SID_NAME ? " 리스너의 마지막 SID명
read lsname
```

- Standby DB 생성 전, Primary DB 사전 작업

```
# 작업할 DB의 ORACLE_SID를 export
export ORACLE_SID=$UPPER_PDNAME
# Primary DB를 Archive log mode 변경, Force logging 활성화
sqlplus -s / as sysdba <<EOF
select instance_name, status from vW$instance;
shutdown immediate
startup mount
alter database archivelog;
alter database open;
```

```

archive log list
alter database force logging;
alter database add supplemental log data (primary key , unique index)
columns;
alter system switch logfile;
select name from v$archived_log;
EOF

```

○ Standby DB를 위한 init.ora(Primary) 에 추가

```

#####db_unique_name=$UPPER_PDNAME
#####standby_file_management=auto
#####db_file_name_convert='/home/oracle/$UPPER_DNAME','/u01/app/o
racle/oradata/$UPPER_PDNAME'
#####log_file_name_convert='/home/oracle/$UPPER_DNAME','/u01/app/
oracle/oradata/$UPPER_PDNAME'
#####log_archive_dest_1='location=/home/oracle/$UPPER_PDNAME/arch
valid_for=(all_logfiles, all_roles)'
#####log_archive_dest_2='service=$UPPER_DNAME LGWR SYNC
AFFIRM valid_for=(online_logfiles, primary_role)'
#####standby_archive_dest=/home/oracle/$UPPER_PDNAME/arch
#####fal_server=$UPPER_DNAME
#####fal_client=$UPPER_PDNAME
# 위 주석 부분의 sed 명령어로 변수 값으로 치환
sed -i "319,327s/W$UPPER_PDNAME/$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh
sed -i "319,327s/W$UPPER_DNAME/$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh
# sed 명령어로 치환된 init.ora 내용을 init.ora에 추가
sed -n '319,327p' $HOME/DataGuardAuto.sh >>
$ORACLE_HOME/dbs/init$UPPER_DNAME.ora
# init.ora 파일에 주석 제거
sed -i "s/#####/g" $ORACLE_HOME/dbs/init$UPPER_DNAME.ora
# 쉘 스크립트에 변수값을 변수명으로 다시 변경
sed -i "319,327s/$UPPER_PDNAME/W$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh
sed -i "319,327s/$UPPER_DNAME/W$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh

```

- Standby DB Directory 생성

```
# 기존 디렉토리명과 같으면 삭제
rm -rf /home/oracle/$UPPER_DNAME
mkdir -p /home/oracle/$UPPER_DNAME
```

- Primary DB datafile, temp file를 Standby DB로 복사

```
# 작업할 DB의 ORACLE_SID를 export
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
shutdown immediate
EOF

echo " [ datafile, temp file .. Copy START!! ]"
# Primary DB에서 Standby DB 로 복사
cp -rp /u01/app/oracle/oradata/$UPPER_PDNAME/*
/home/oracle/$UPPER_DNAME

echo " [ datafile, temp file .. Copy END!! ]"
```

- Archive log file, Flashback database log 저장할 디렉토리 생성

```
# 기존 디렉토리명과 같으면 삭제
rm -rf /home/oracle/$UPPER_PDNAME/arch
rm -rf /home/oracle/$UPPER_PDNAME/flash
# Primary DB arch, flash 디렉토리 생성
mkdir -p /home/oracle/$UPPER_PDNAME/arch
mkdir -p /home/oracle/$UPPER_PDNAME/flash

# 기존 디렉토리명과 같으면 삭제
rm -rf /home/oracle/$UPPER_DNAME/arch
rm -rf /home/oracle/$UPPER_DNAME/flash
# Primary DB arch, flash 디렉토리 생성
mkdir -p /home/oracle/$UPPER_DNAME/arch
mkdir -p /home/oracle/$UPPER_DNAME/flash
```

- Standby DB Controlfile 생성

Standby DB용 Controlfile 은 Primary에서 생성하고 이를 위해 Primary DB를 mount 상태로 올립니다.

작업할 DB의 ORACLE_SID를 export

```
export ORACLE_SID=$UPPER_PDNAME
```

```
sqlplus -s / as sysdba <<EOF
```

```
startup mount
```

```
alter database create standby controlfile
```

```
as
```

```
'$HOME/physical.ctl' reuse;
```

```
EOF
```

생성된 controlfile 확인

```
if [ -e $HOME/physical.ctl ]; then
```

```
    echo " [ Completed to create controlfile for Standby
```

```
DB ]"
```

```
else
```

```
    echo " [ Failed to create controlfile for Standby DB ]"
```

```
fi
```

- Controlfile Standby DB로 복사

Standby DB disk1, disk2, disk3 디렉토리에 ctrl1, ctrl2, ctrl3 복사

```
cp -rf $HOME/physical.ctl /home/oracle/$UPPER_DNAME/disk1/ctrl1.ctl
```

```
cp -rf $HOME/physical.ctl /home/oracle/$UPPER_DNAME/disk2/ctrl2.ctl
```

```
cp -rf $HOME/physical.ctl /home/oracle/$UPPER_DNAME/disk3/ctrl3.ctl
```

- listener.ora 수정

생성된 Primary DB 와 Standby DB 의 정보를 listener.ora 에 추가합니다. 이때 \$HOME/listener_insert.txt 파일의 내용으로 추가됩니다.

listener_insert.txt에 변수명을 변수값으로 치환 후 저장

```
sed -i "s/W$UPPER_DNAME/$UPPER_DNAME/g" $HOME/listener_insert.txt
```

```
sed -i "s/W$UPPER_PDNAME/$UPPER_PDNAME/g"
```

```
$HOME/listener_insert.txt
```

listener.ora 의 마지막 SID_NAME 아래쪽에 listener_insert.txt 내용 삽입 후 저장

```
sed -i "(/$lsname)/r $HOME/listener_insert.txt"
```

```
$ORACLE_HOME/network/admin/listener.ora
```

listener_insert.txt에 변수값을 변수명으로 치환 후 저장

```
sed -i "s/$UPPER_DNAME/W$UPPER_DNAME/g" $HOME/listener_insert.txt
```

```
sed -i "s/$UPPER_PDNAME/W$UPPER_PDNAME/g"
```

```
$HOME/listener_insert.txt
```

```
lsnrctl stop
```

```
lsnrctl start
```

```
lsnrctl status > $HOME/lsnrctl_status.txt
```

Primary DB 와 Standby DB가 올라왔는지 확인

```
LS_P=`grep service lsnrctl_status.txt | grep $UPPER_PDNAME | cut -d '"' -f 2`
```

```
LS_S=`grep service lsnrctl_status.txt | grep $UPPER_DNAME | cut -d '"' -f 2`
```

```
PN=$UPPER_PDNAME
```

```
SN=$UPPER_DNAME
```

```
if [ $LS_P = $PN -a $LS_S = $SN ]; then
```

```
    echo " listener service OK "
```

```
else
```

```
    echo " listener service fail.. "
```

```
fi
```

- tnsnames.ora 수정

Primary DB와 Standby DB의 정보를 추가합니다.

```
# tnsnames.ora 에 추가할 내용
# IP 변경 시 셸 스크립트에서 수정 요망
#####$UPPER_PDNAME =
##### (DESCRIPTION =
##### (ADDRESS = (PROTOCOL = TCP)(HOST =
192.168.56.104)(PORT = 1521))
##### (CONNECT_DATA =
##### (SERVER = DEDICATED)
##### (SERVICE_NAME = $UPPER_PDNAME)
##### )
##### )
#####$UPPER_DNAME =
##### (DESCRIPTION =
##### (ADDRESS = (PROTOCOL = TCP)(HOST =
192.168.56.104)(PORT = 1521))
##### (CONNECT_DATA =
##### (SERVER = DEDICATED)
##### (SERVICE_NAME = $UPPER_DNAME)
##### )
##### )

# 위 주석 부분의 sed 명령어로 변수 값으로 치환
sed -i "472,487s/W$UPPER_PDNAME/$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh
sed -i "472,487s/W$UPPER_DNAME/$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh

# 치환된 내용을 tnsnames.ora에 추가
sed -n '472,487p' $HOME/DataGuardAuto.sh >>
$ORACLE_HOME/network/admin/tnsnames.ora

# tnsnames.ora 파일에 주석 제거
sed -i "s/#####/g" $ORACLE_HOME/network/admin/tnsnames.ora

# 셸 스크립트에 변수로 치환된 자리를 변수명으로 다시 변경
sed -i "472,487s/$UPPER_PDNAME/W$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh
sed -i "472,487s/$UPPER_DNAME/W$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh
```

- Primary DB, Standby DB password file 생성

```
# 생성하는 password file명과 동일하면 백업
cd $ORACLE_HOME/dbs
cp orapw$UPPER_PDNAME orapw$UPPER_PDNAME.bak
cp orapw$UPPER_DNAME orapw$UPPER_DNAME.bak
# 기존에 존재했다면 삭제 후 생성
# 대소문자 구분 얹히는 옵션 사용
rm -rf $ORACLE_HOME/dbs/orapw$UPPER_PDNAME
rm -rf $ORACLE_HOME/dbs/orapw$UPPER_DNAME
orapwd file=orapw$UPPER_PDNAME password=oracle ignorecase=Y
orapwd file=orapw$UPPER_DNAME password=oracle ignorecase=Y

        if [ -e $ORACLE_HOME/dbs/orapw$UPPER_PDNAME -a -e
$ORACLE_HOME/dbs/orapw$UPPER_DNAME ]; then
                echo " [ Primary DB, Standby DB password file create
complete ]"
        else
                echo " [ Primary DB, Standby DB password file create
fail ]"
        fi
```

- SQL*PLUS 접속 확인

```
# 작업할 DB의 ORACLE_SID를 export
export ORACLE_SID=$UPPER_PDNAME
echo $ORACLE_SID
sqlplus -s sys/oracle@$pdname as sysdba <<EOF
EOF
sqlplus -s sys/ORACLE@$pdname as sysdba <<EOF
EOF

# 작업할 DB의 ORACLE_SID를 export
export ORACLE_SID=$UPPER_DNAME
echo $ORACLE_SID
sqlplus -s sys/oracle@$dname as sysdba <<EOF
EOF
sqlplus -s sys/ORACLE@$dname as sysdba <<EOF
EOF
```


○ Standby DB init.ora 생성

init.ora 생성 시 내용

```
#####compatible=11.2.0.1.0
#####control_files = (/home/oracle/$UPPER_DNAME/disk1/ctrl1.ctl ,
#####                  /home/oracle/$UPPER_DNAME/disk2/ctrl2.ctl ,
#####                  /home/oracle/$UPPER_DNAME/disk3/ctrl3.ctl )
#####db_block_size=8192
#####db_name=$UPPER_PDNAME
#####service_names=$UPPER_DNAME
#####global_names=true
#####job_queue_processes=10
#####open_cursors=500
#####processes=100
#####remote_login_passwordfile='EXCLUSIVE'
#####sga_max_size=120M
#####sga_target=120M
#####undo_management='AUTO'
#####undo_tablespace='UNDOTBS'
#####db_recovery_file_dest_size=4G
#####db_recovery_file_dest=/home/oracle/$UPPER_DNAME/flash
#####db_unique_name=$UPPER_DNAME
#####standby_file_management=auto
#####db_file_name_convert='/u01/app/oracle/oradata/$UPPER_PDNAME'
#####',/home/oracle/$UPPER_DNAME'
#####log_file_name_convert='/u01/app/oracle/oradata/$UPPER_PDNAME'
#####',/home/oracle/$UPPER_DNAME'
#####log_archive_dest_1='location=/home/oracle/$UPPER_DNAME/arch
#####valid_for=(all_logfiles, all_roles)'
#####log_archive_dest_2='service=$UPPER_PDNAME LGWR SYNC
#####AFFIRM valid_for=(online_logfiles, primary_role)'
#####standby_archive_dest=/home/oracle/$UPPER_DNAME/arch
#####recovery_parallelism=4
#####fal_server=$UPPER_PDNAME
#####fal_client=$UPPER_DNAME
```

위 주석 부분의 sed 명령어로 변수 값으로 치환

```
ssed -i "561,588s/W$UPPER_DNAME/$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh
```

```
sed -i "561,588s/W$UPPER_PDNAME/$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh
# 치환된 내용을 init.ora에 추가
sed -n '561,588p' $HOME/DataGuardAuto.sh >>
$ORACLE_HOME/dbs/init$UPPER_DNAME.ora

# init.ora 파일에 주석 제거
sed -i "s/#####/g" $ORACLE_HOME/dbs/init$UPPER_DNAME.ora

# 셸 스크립트에 변수로 치환된 자리를 변수명으로 다시 변경
sed -i "561,588s/$UPPER_DNAME/W$UPPER_DNAME/g"
$HOME/DataGuardAuto.sh
        sed -i "561,588s/$UPPER_PDNAME/W$UPPER_PDNAME/g"
$HOME/DataGuardAuto.sh

ls -al $ORACLE_HOME/dbs/init$UPPER_DNAME.ora

# 파일이 생성 되었는지 확인
if [ -e $ORACLE_HOME/dbs/init$UPPER_DNAME.ora ]; then
    echo " [ Standby DB init.ora create complete ]"
else
    echo " [ Standby DB init.ora create fail]"
fi
```

- Standby DB redo log file 생성
Standby DB용 redo log file을 생성하기 위해 Primary DB에도 동일한 이름으로 redo log file을 생성해줍니다.

```
# 작업할 DB의 ORACLE_SID를 export
# 기존 redo log file 삭제 후, 생성
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
alter database drop standby logfile
'/u01/app/oracle/oradata/$UPPER_PDNAME/disk1/standby01.log';
alter database drop standby logfile
'/u01/app/oracle/oradata/$UPPER_PDNAME/disk1/standby02.log';
alter database add standby logfile
'/u01/app/oracle/oradata/$UPPER_PDNAME/disk1/standby01.log' size
10m;
alter database add standby logfile
'/u01/app/oracle/oradata/$UPPER_PDNAME/disk1/standby02.log' size
10m;
EOF

# 작업할 DB의 ORACLE_SID를 export
# pfile로 Standby DB nomount startup
# 기존 redo log file 삭제 후, 생성
export ORACLE_SID=$UPPER_DNAME
sqlplus -s / as sysdba <<EOF
startup nomount pfile=$ORACLE_HOME/dbs/init$UPPER_DNAME.ora
alter database mount;
select status from vW$instance;
alter database drop standby logfile
'/home/oracle/$UPPER_DNAME/disk1/standby01.log';
alter database drop standby logfile
'/home/oracle/$UPPER_DNAME/disk1/standby02.log';
alter database add standby logfile
'/home/oracle/$UPPER_DNAME/disk1/standby01.log' size 10m;
alter database add standby logfile
'/home/oracle/$UPPER_DNAME/disk1/standby02.log' size 10m;
EOF
```

◦ Standby DB Recover Mode

Recover Mode를 확인 하기 위해서 recover_mode.txt 파일이 생성된다.

복구 모드 설정을 위해 mount 상태로 startup

```
rm -rf /home/oracle/recover_mode.txt
export ORACLE_SID=$UPPER_DNAME
sqlplus -s / as sysdba <<EOF
spool /home/oracle/recover_mode.txt
shutdown immediate
startup mount
alter database flashback on;
recover managed standby database disconnect;
select process, status from vW$managed_standby;
spool off
EOF
```

복구 모드 설정 후 MRP0 프로세스 체크

```
MRP_P=`grep MRP0 /home/oracle/recover_mode.txt | cut -d ' ' -f 1`

echo " [ Standby DB recover mode, $MRP_P process check !! ]"
```

Primary DB에서 LGWR 가 떠있는지 체크

```
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
spool /home/oracle/recover_mode.txt
select process, status from vW$managed_standby;
spool off
EOF
```

복구 모드 설정 후 LGWR 프로세스 체크

```
LGWR_P=`grep LGWR /home/oracle/recover_mode.txt | cut -d ' ' -f 1`

echo " [ Primary DB, $LGWR_P process check !! ]"
```

○ Primary DB Open

```
# 작업할 DB의 ORACLE_SID를 export
# Primary DB open
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
alter database open;
EOF

echo " Finishing create the $dname DB Standby 복구 모드 완료. 테스트
필수"
```

□ SWITCHOVER

자연재해와 장애로 인해 운영 중인 DB를 손상을 일으켜 Standby DB를 손상된 Primary DB를 대신해 Primary DB로 쓰고 Primary DB를 Standby DB로 바꾸는 기능이다.

SWITCHOVER를 시킬 Primary DB 와 Standby DB 입력

```
3)
echo -n " Primary DB name ? " Primary DB명
read pdname ( 영문 소문자 입력 )
echo -n " Standby DB name ? " Standby DB명
read sdname ( 영문 소문자 입력 )
```

tr 명령어로 소문자를 대문자로 변환하여 변수에 저장

```
UPPER_PDNAME=`echo $pdname | tr 'a-z' 'A-Z'`
UPPER_SDNAME=`echo $sdname | tr 'a-z' 'A-Z'`
```

작업할 DB의 ORACLE_SID를 export

Primary db 를 Standby DB로 쓰겠다고 설정.

```
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
select open_mode, database_role from vW$database;
alter database commit to switchover to physical standby with session
shutdown wait;
shutdown immediate
EOF

echo " [Primary DB ( $pdname ) -> Standby DB ]"
```

```

# 작업할 DB의 ORACLE_SID를 export
# Standby DB를 Primary db 로 쓰겠다고 설정한다.
export ORACLE_SID=$UPPER_SDNAME
sqlplus -s / as sysdba <<EOF
select open_mode, database_role from vW$database;
alter database commit to switchover to primary with session shutdown
wait;
shutdown immediate
EOF

echo " [Standby DB ( $sdname ) -> Primary DB ]"

```

○ init.ora 파일 수정

```

# 작업할 DB의 ORACLE_SID를 export
# sed 명령어로 Primary DB 에 있는 fal_server, fal_client 옵션의
주석을 제거
export ORACLE_SID=$UPPER_PDNAME
sed -i 's/#fal_server/fal_server/'
$ORACLE_HOME/dbs/init$UPPER_PDNAME.ora
sed -i 's/#fal_client/fal_client/'
$ORACLE_HOME/dbs/init$UPPER_PDNAME.ora

# 작업할 DB의 ORACLE_SID를 export
# sed 명령어로 Standby DB 에 있는 fal_server, fal_client 옵션의
주석 처리
export ORACLE_SID=$UPPER_SDNAME
sed -i 's/fal_server/#fal_server/'
$ORACLE_HOME/dbs/init$UPPER_SDNAME.ora
sed -i 's/fal_client/#fal_client/'
$ORACLE_HOME/dbs/init$UPPER_SDNAME.ora

echo " [ Primary DB , Standby DB init file modified ]"

```

- Primary DB, Standby DB startup

```

# 작업할 DB의 ORACLE_SID를 export
# SWITCHOVER 한 Standby DB가 된 DB를 mount
# PHYSICAL STANDBY 확인
export ORACLE_SID=$UPPER_PDNAME
sqlplus -s / as sysdba <<EOF
startup mount
recover managed standby database disconnect parallel 4;
select open_mode, database_role from v/$database;
EOF

        echo " [Primary DB ( $pdname ) -> Standby DB * PHYSICAL
STANDBY * ]"

# 작업할 DB의 ORACLE_SID를 export
# SWITCHOVER 한 Primary DB가 된 DB를 mount
# PRIMARY 확인
export ORACLE_SID=$UPPER_SDNAME
sqlplus -s / as sysdba <<EOF
startup
select open_mode, database_role from vW$database;
EOF

        echo " [Standby DB ( $sdname ) -> Primary DB * PRIMARY *
]"

echo " [ $pdname DB $sdname DB SWITCHOVER OK ]"

;;

esac

```

- 쉘 스크립트 실행 후 남기는 파일

파일명	경로	설명
status.log	\$HOME/	실행 결과에 대한 로그, 순차적으로 어떤 작업이 일어났는지 알 수 있다.
recover_mode.txt	\$HOME/	Primary DB, Standby DB 의 복구 모드 시 프로세스 체크가 담겨 있다.
mount.txt	\$HOME/	Primary DB 생성 시 mount 상태 저장.

- 참 고 문 헌 -

- 오라클 9i - 김태근

글쓴이소개 -----



- 컴퓨터보안학과 졸업
- 오라클과 파이썬을 활용한 데이터 전문가 양성 과정 이수
- 보안 솔루션 경력 3년 5개월, 일을 하다 접하게 된 여러 DB종류로 인해 SQL 입문을 하여 공부를 하다 보니 관심과 흥미가 있어 DB인의 길로 들어섰다.
- PL/SQL, UNIX 계열 서버 및 셸 스크립트 작성에 강하고 튜닝에 관심이 높아 맹렬하게 파고드는 중.
- 타자 소리가 좋지만 수동보다 자동화가 편하다. 그래서 배울 때 가장 애를 먹었던 DATAGUARD를 자동화 하고 싶었다.
- 어려울수록 내 것이 될 때 쾌감이 있는 법! 그 매력적인 분야에 내가 오래도록 있길 바라고 해낼 때마다 여전히 기분 좋기를...