Author : ACHBANI Mohamed
Date : 16-01-2016

Oracle - How to Create Logical Standby

This note covers the steps for creating a configuration "Logical Standby" (Data Guard).
You will need a database and a Linux environment in this example.
In my case, I'll create a database named DB1 and a DB2 database (Logical Standby Database).

1- Preparation of the base Primary data (server name: primary)

If you do not already have one, create and start a listener for your database example:

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

LISTENER_DB1 =
 (DESCRIPTION_LIST =
        (DESCRIPTION =
                (ADDRESS_LIST =
                (ADDRESS = (PROTOCOL = TCP)(HOST = primary)(PORT = 1521))
                ) ))

# lsnrctl start LISTENER_DB1

En outre, ajouter des références pour ce listener dans votre fichier de réseau:

        $ORACLE_HOME/networking/admin/tnsnames.ora

LISTENER_DB1 =
        (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = primary)(PORT = 1521))
        )

DB1 =
(DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = primary)(PORT = 1521))
        (CONNECT_DATA =
                (SERVER = DEDICATED)
                (SERVICE_NAME = DB1.momo.com)
        )
)
```

Author : ACHBANI Mohamed
Date : 16-01-2016

Create init.ora file for your database

```
$ORACLE_HOME/dbs/initDB1.ora

*.db_name=DB1
*.db_domain=momo.com
*.db_block_size=8192
*.service_names='DB1.momo.com'
*.control_files=("/u01/app/oracle/oradata/DB1/control01.ctl")
*.memory_target=256M
*.processes=100
*.background_dump_dest=/u01/app/oracle/admin/DB1/bdump
*.user_dump_dest=/u01/app/oracle/admin/DB1/udump
*.core_dump_dest=/u01/app/oracle/admin/DB1/cdump
*.undo_management=AUTO
*.undo_tablespace=UNDOTBS1
*.undo_retention=3600
*.local_listener=LISTENER_DB1
```

Create the password file

```
# orapwd file=$ORACLE_HOME/dbs/orapwDB1 password={password}
```

Add the entry in the /etc/oratab

```
# echo "DB1:/u01/app/oracle/product/11.2.0:N" >> /etc/oratab
```

Create the database

```
# export ORACLE_SID=DB1
#. oraenv
```

```
# create_DB1.sql:

        shutdown immediate;
        startup nomount;
        CREATE DATABASE
        CONTROLFILE REUSE
        MAXINSTANCES 32
        MAXLOGHISTORY 10000
        MAXLOGMEMBERS 5
        MAXLOGFILES 64
        DATAFILE '/u01/app/oracle/oradata/DB1/system01.dbf' SIZE 400M REUSE      AUTOEXTEND
        ON NEXT 100M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
        SYSAUX DATAFILE '/u01/app/oracle/oradata/DB1/sysaux01.dbf' SIZE 400M     REUSE
        AUTOEXTEND ON NEXT 100M MAXSIZE 5G
        DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
        '/u01/app/oracle/oradata/DB1/temp01.dbf' SIZE 100M REUSE
        UNDO TABLESPACE "UNDOTBS1" DATAFILE
        '/u01/app/oracle/oradata/DB1/undotbs1_01.dbf' SIZE 200M REUSE
        CHARACTER SET UTF8
        LOGFILE
        GROUP 1 ('/u01/app/oracle/oradata/DB1/redo_1a.dbf',
        '/u01/app/oracle/oradata/DB1/redo_1b.dbf') SIZE 100M REUSE,
        GROUP 2 ('/u01/app/oracle/oradata/DB1/redo_2a.dbf',
        '/u01/app/oracle/oradata/DB1/redo_2b.dbf') SIZE 100M REUSE;
        exit;
```

```
# sqlplus "/ as sysdba" @create_DB1.sql

ORA-01507: database not mounted
ORACLE instance shut down.
ORACLE instance started.
```

Create the Dictionary

```
# create_dd.sql:

@$ORACLE_HOME/rdbms/admin/catalog.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql
@$ORACLE_HOME/rdbms/admin/catclust.sql
@$ORACLE_HOME/rdbms/admin/utlrp.sql
exit;
```

```
# sqlplus "/ as sysdba" @create_dd.sql
```

Run the following command to force the basis for any "logger"

```
        SQL> alter database force logging;
```

Create the Standby Redo Log Groups

```
SQL> alter database add standby logfile group 3
('/u01/app/oracle/oradata/DB1/redo_3a.dbf','/u01/app/oracle/oradata/DB1/redo_3b.dbf') size 100M;
SQL> alter database add standby logfile group 4
('/u01/app/oracle/oradata/DB1/redo_4a.dbf','/u01/app/oracle/oradata/DB1/redo_4b.dbf') size 100M;

SQL> alter database add standby logfile group 5
('/u01/app/oracle/oradata/DB1/redo_5a.dbf','/u01/app/oracle/oradata/DB1/redo_5b.dbf') size 100M;
```

Create the spfile file and restart the database

```
SQL> create spfile from pfile;
SQL> shutdown immediate;
        Database closed.
        Database dismounted.
        ORACLE instance shut down.
SQL> startup;
```

Edit the initialization parameters

```
SQL> alter system set log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/DB1/arch/
VALID_FOR=
(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=DB1' scope=spfile;
SQL> alter system set log_archive_dest_2='SERVICE=DB2 LGWR ASYNC VALID_FOR=
(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=DB2' scope=spfile;
SQL> alter system set log_archive_dest_state_1=enable scope=spfile;
SQL> alter system set log_archive_dest_state_2=enable scope=spfile;
SQL> alter system set log_archive_format='DB1_%t_%s_%r.arc' scope=spfile;
SQL> alter system set log_archive_max_processes=4 scope=both;
SQL> alter system set fal_server=DB2 scope=spfile;
SQL> alter system set fal_client=DB1 scope=spfile;
SQL> alter system set db_file_name_convert='/DB2/','/DB1/' scope=spfile;
SQL> alter system set log_file_name_convert='/DB2/','/DB1/' scope=spfile;
SQL> alter system set standby_file_management=auto scope=spfile;
```

Putting the database in archivelog mode

```
SQL> shutdown immediate;
                Database closed.
                Database dismounted.
                ORACLE instance shut down.
SQL> startup mount;
                ORACLE instance started.
SQL> alter database archivelog;
SQL> alter database open;
```

Create an RMAN backup that will be used later to create us "standby" database

```
# rman target=/
RMAN> backup full database format '/u01/app/oracle/oradata/DB1/backups/%d_%U.bckp'
plus archivelog format '/u01/app/oracle/oradata/DB1/backups/%d_%U.bckp';
```

Create a standby controlfile backup via RMAN

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/u01/app/oracle/oradata/DB1/backups/%U';
RMAN> BACKUP CURRENT CONTROLFILE FOR STANDBY;
```

Copy all backups to the standby server via scp command

```
# cd /u01/app/oracle/oradata/DB1/backups
# scp * oracle@standby:/u01/app/oracle/oradata/DB1/backups/
```

Before you create the standby database, change the primary database by adding an entry in the tnsnames.ora file

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

DB2 =
        (DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = standby)(PORT = 1521))
        (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = DB2.momo.com) ) )
```

2. Preparation of the standby database (server name: standby)

If you do not already have one, create and start a listener for your sample database

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

LISTENER_DB2 =
(DESCRIPTION_LIST =
        (DESCRIPTION =
        (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = standby)(PORT = 1521))
        ) ) )
```

```
# lsnrctl start LISTENER_DB2
```

Add references to the two instances in the tnsnames.ora file

```
$ORACLE_HOME/networking/admin/tnsnames.ora

LISTENER_DB2 =
        (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = standby)(PORT = 1521)) )

DB1 =
(DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = primary)(PORT = 1521))
        (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = DB1.momo.com)))

DB2 =
(DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = standby)(PORT = 1521))
        (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = DB2.momo.com)))
```

Create the initialization file of parameters by copying the primary database

```
SQL> create pfile='/tmp/initDB2.ora' from spfile;
        File created.
```

```
# scp /tmp/initDB2.ora oracle@standby:/u01/app/oracle/product/11.2.0/dbs/

$ORACLE_HOME/dbs/initDB2.ora
```

```
*.background_dump_dest='/u01/app/oracle/admin/DB2/bdump'
*.cluster_database=FALSE
*.control_files='/u01/app/oracle/oradata/DB2/control01.ctl'
*.core_dump_dest='/u01/app/oracle/admin/DB2/cdump'
*.db_block_size=8192
*.db_domain='momo.com'
*.db_file_name_convert='/DB1/','/DB2/'
*.db_name='DB1'
*.db_unique_name='DB2'
*.fal_client='DB2'
*.fal_server='DB1'
*.local_listener='LISTENER_DB2'
*.log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/DB2/arch/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=DB2'
*.log_archive_dest_2='SERVICE=DB1 LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=DB1'
*.log_archive_dest_state_1='ENABLE'
*.log_archive_dest_state_2='ENABLE'
*.log_archive_format='DB2_%t_%s_%r.arc'
*.log_archive_max_processes=4
*.log_file_name_convert='/DB1/','/DB2/'
*.pga_aggregate_target=64M
*.processes=100
*.service_names='DB2.momo.com'
*.memory_target=256M
*.standby_file_management='AUTO'
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.undo_retention=3600
*.user_dump_dest='/u01/app/oracle/admin/DB2/udump'
```

Create a password file to the standby

```
SQL> startup mount
```

```
# orapwd file=$ORACLE_HOME/dbs/orapwDB2 password={password}
```

Add the entry in the /etc/oratab

```
# echo "DB2:/u01/app/oracle/product/11.2.0:N" >> /etc/oratab
```

Start the standby instance in nomount mode

```
SQL> startup nomount;
```

Duplicate database using RMAN

create_standby.rman :

```
DUPLICATE TARGET DATABASE FOR STANDBY NOFILENAMECHECK;
```

```
# rman target=sys/{password}@DB1 auxiliary=/ @create_standby.rman
```

Create the spfile file and start the database in mount mode

```
SQL> shutdown immediate;
SQL> create spfile from pfile;
SQL> startup mount;
```

Redo Apply

```
SQL> alter database recover managed standby database disconnect from session;
```

To test :

- On the primary database

```
SQL> alter system switch logfile;
SQL> alter system archive log current;
```

```
SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /u01/app/oracle/oradata/DB1/arch/
Oldest online log sequence     10
Next log sequence to archive   11
Current log sequence           11
```

- On the standby database

```
SQL> SELECT SEQUENCE#,APPLIED FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
    SEQUENCE# APPLIED
    ---------- -----
    6                    YES
    7                    YES
    8                    YES
    9                    YES
    10                   YES
```

3. Convert a physical standby to a logical standby

Stop the process of recovery on the standby

```
SQL> alter database recover managed standby database cancel;
```

Edit the archive destinations swapping roles on the primary

```
SQL> alter system set log_archive_dest_3='LOCATION=/u01/app/oracle/oradata/DB1/arch2/ valid_for=
(standby_logfiles,standby_role) db_unique_name=DB1' scope=both;
```

```
SQL> alter system set log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/DB1/arch/ valid_for=
(online_logfiles,all_roles) db_unique_name=DB1' scope=BOTH;
```

```
SQL> alter system set log_archive_dest_state_3=enable scope=both;
```

Compiling the dictionary LogMiner on the primary

```
SQL> EXECUTE DBMS_LOGSTDBY.BUILD;
```

Convert the physical standby to a logical standby

```
SQL> alter database recover to logical standby DB2;
```

Recreate password file on standby

```
# cd $ORACLE_HOME/dbs
# rm orapwDB2
# orapwd file=$ORACLE_HOME/dbs/orapwDB2 password={password}
```

Edit the archive destinations swapping roles on standby

```
SQL> alter system set log_archive_dest_3='LOCATION=/u01/app/oracle/oradata/DB2/arch2/ valid_for=
(standby_logfiles,standby_role) db_unique_name=DB2' scope=both;

SQL> alter system set log_archive_dest_state_3=enable scope=both;
SQL> alter system set log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/DB2/arch/ valid_for=
(online_logfiles,all_roles) db_unique_name=DB2' scope=both;
```

Create the standby redo log files

```
SQL> alter database add standby logfile group 3
('/u01/app/oracle/oradata/DB2/redo_3a.dbf','/u01/app/oracle/oradata/DB2/redo_3b.dbf') size 100M;
SQL> alter database add standby logfile group 4
('/u01/app/oracle/oradata/DB2/redo_4a.dbf','/u01/app/oracle/oradata/DB2/redo_4b.dbf') size 100M;
SQL> alter database add standby logfile group 5
('/u01/app/oracle/oradata/DB2/redo_5a.dbf','/u01/app/oracle/oradata/DB2/redo_5b.dbf') size 100M;
```

Start logical standby

```
SQL> shutdown immdiate;
SQL> startup mount;
SQL> alter database open resetlogs;
SQL> alter database start logical standby apply immediate;
```

Once converted, the file conversion settings do not work as they do for physical standby.
You can create a handler that performs the equivalent replacement file path on the standby

create_df_skip_handler_momo.sql :

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
SQL> CREATE OR REPLACE PROCEDURE sys.handle_tbs_ddl (
        old_stmt  IN  VARCHAR2,
        stmt_typ  IN  VARCHAR2,
        schema    IN  VARCHAR2,
        name      IN  VARCHAR2,
        xidusn    IN  NUMBER,
        xidslt    IN  NUMBER,
        xidsqn    IN  NUMBER,
        action    OUT NUMBER,
        new_stmt  OUT VARCHAR2
      ) AS
      BEGIN
            new_stmt := replace(old_stmt, '\DB1\', '\DB2\');
            action := DBMS_LOGSTDBY.SKIP_ACTION_REPLACE;
      EXCEPTION  WHEN OTHERS THEN
            action := DBMS_LOGSTDBY.SKIP_ACTION_ERROR;
            new_stmt := NULL;
      END handle_tbs_ddl;
      /

SQL> EXECUTE DBMS_LOGSTDBY.SKIP (stmt => 'TABLESPACE', proc_name =>
'SYS.HANDLE_TBS_DDL');
SQL> SHOW ERRORS;
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

```
# sqlplus "/ as sysdba" @create_df_skip_handler_momo.sql
```

To test:

- On the primary

```
SQL> create tablespace users datafile '/u01/app/oracle/oradata/DB1/users01.dbf' size 100M segment
space management auto;
```

- On standby

```
SQL> SELECT EVENT_TIME, STATUS, EVENT FROM DBA_LOGSTDBY_EVENTS ORDER BY
EVENT_TIMESTAMP, COMMIT_SCN;

SQL> SELECT APPLIED_SCN, LATEST_SCN, MINING_SCN, RESTART_SCN FROM
V$LOGSTDBY_PROGRESS;

SQL> SELECT APPLIED_TIME, LATEST_TIME, MINING_TIME, RESTART_TIME FROM
V$LOGSTDBY_PROGRESS;

SQL> SELECT SESSION_ID, STATE FROM V$LOGSTDBY_STATE;
```