
Active Data Guard 12c NF

Author	이대혁
Creation Date	2018-04-05
Last Updated	2019-07-12
Version	1.2
Copyright(C) 2018 GoodusData Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2018-04-05	이대혁	문서 최초 작성(ver1.0)
2	2019-03-31	이대혁	내용 추가 및 일부 수정(ver1.1)
3	2019-07-12	이대혁	테스트 내용 일부 추가 작성 완료 (ver1.2)

Contents

1. Oracle DataGuard 소개	3
2. 12c New Feature	3
2.1 SYSDG	3
2.1.1 SYSDG 를 사용하기 위한 암호 파일 인증 생성	4
2.1.2 orapwd Utility 를 사용하여 Password File 생성	4
2.1.3 SYSDBA, SYSOPER, SYSBACKUP, SYSDG, SYSKM 권한 부여	4
2.1.4 인증 파일을 이용하여 Connect	5
2.2 Fal Sync	5
2.2.1 Far Sync 서버를 사용할 경우 오버헤드 감소	6
2.2.2 Far Sync 서버를 사용할 경우 라이선스	6
2.3 Fast Sync	6
2.3.1 Data Protection Modes	7
2.3.2 Data Protection Modes 에 필요한 Redo Transport 속성	8
2.3.3 12c Fast Sync 설정	8
2.4 Real-Time Cascade	9
2.5 12c Database ActiveDataGuard Writable Temp Tables	9
2.6 Simpler Rol Transitions	9
2.7 Rolling Upgrade	10
2.8 12c Moving Online Data FILES	10
3 테스트 구성	11
3.1 RAC to RAC Active Data Guard 구성	11
3.1.1 Oracle 12.2.0.1 RAC S/W 2SET 설치	11
3.1.2 Oracle 12cR2 Parameter 변경 사항	11
3.1.3 Primary DB & Standby DB Information	13
3.1.4 Tnsnames.ora 설정	14
3.1.5 Static Listener 생성	14
3.1.6 Password File 배포	15
3.1.7 Primary force_logging / supplemental log 설정	16
3.1.8 Archive Log 설정	16
3.1.9 Add Standby Logfile 추가	17
3.1.10 Init Parameter 추가	18
3.1.11 경로 생성	20
3.1.12 RMAN Duplication 수행	20
3.1.13 MRP 기동 및 Apply 확인	21
3.2 12c Far Sync 구성	22
3.2.1 Primary 에서 Controlfile 을 생성	22
3.2.2 Parameter 생성	23
3.2.3 구성 확인	25

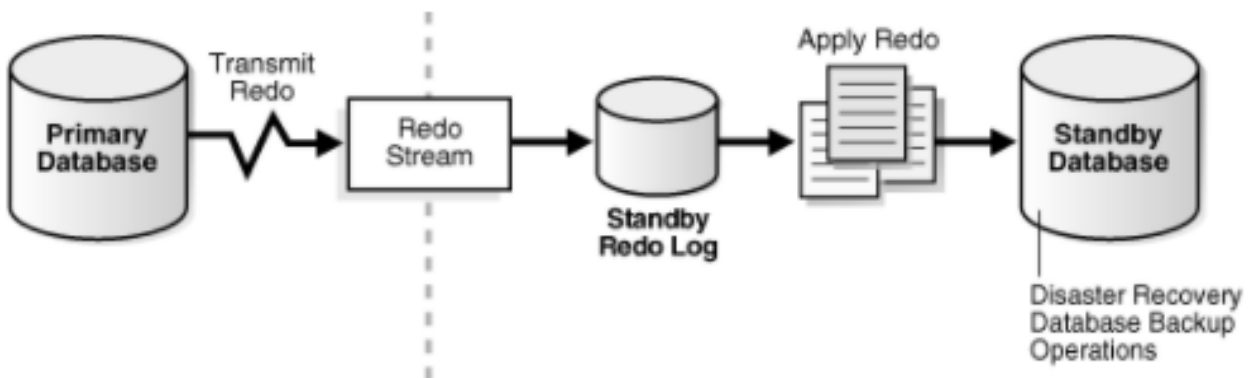
1. Oracle DataGuard 소개

■ DataGuard란?

Oracle Data Guard는 고가용성, 데이터 보호 및 재난 재해 복구를 보장하는 기능입니다. Data Guard는 하나 이상의 Standby Database 를 생성, 유지, 관리, 모니터링하면서 운영 서버를 복제하여 관리 가능합니다. 의도치 않은 운영 서버 중단으로 인한 production Database 를 사용 불가능하게 되면 모든 Standby Database 를 Active Role 로 전환하여 장애 시간을 최소화 할 수 있습니다.

이런 Oracle Data Guard 는 전통적인 백업, 복구 및 Cluster Technology 와 함께 사용하여 수준 높은 데이터 보호 및 가용성을 제공합니다. 이러한 데이터 복제 서비스는 Oracle Streams 및 Oracle Golden Gate 와 같은 다른 Oracle 기능에서도 Source Database 에서 하나 이상의 원격 대상으로 효율적이고 안정적인 전송을 위해 사용됩니다.

<일반적인 Data Guard 구성의 예>



- 참조- <https://docs.oracle.com/cloud/latest/db121/SBYDB/concepts.htm#SBYDB4704>

위 그림은 기본 데이터베이스와 물리적 대기 데이터베이스로 구성된 Data Guard 구성이며 Primary Database 에서 Redo 가 전송되어 Standby Database 에 적용되기까지의 과정을 보여줍니다.

11g 까지의 (Active) DataGuard 에 대한 소개 및 구성은 '굿어스 기술노트 21호 Data Guard' , '굿어스 기술노트 52호 11gR2_RAC ActiveDataGuard_v1' 를 참조하시길 바랍니다.

2. 12c New Feature

2.1 SYSDBG

- Oracle 12c New feature 로 Data Guard 에 대한 권한이 세분화 되었습니다.
- SYSDBG 는 데이터 가드 작업을 수행할 때 필요한 권한을 가지고 있습니다.
- 이 권한은 Data Guard Broker 또는 DGMGRL Command Line Interface 를 사용할 수 있는 권한입니다.
- SYSDBG 암호를 사용하여 데이터베이스에 연결하려면 암호 파일을 만들어야합니다.

- SYSDBG 권한

- STARTUP
- SHUTDOWN
- ALTER DATABASE
- ALTER SESSION
- ALTER SYSTEM
- CREATE RESTORE POINT (GUARANTEED 복원 지점 포함)
- CREATE SESSION
- DROP RESTORE POINT (GUARANTEED 복원 지점 포함)
- FLASHBACK DATABASE
- SELECT ANY DICTIONARY
- SELECT

- x\$ table (Fixed Tables)
- v\$ 및 gv\$ views (Dynamic Performance Views)
- APPQOSSYS.WLM_CLASSIFIER_PLAN
- DELETE
- APPQOSSYS.WLM_CLASSIFIER_PLAN
- EXECUTE
- SYS.DBMS_DRS

2.1.1 SYSDG 를 사용하기 위한 암호 파일 인증 생성

Oracle Database Instance 및 Oracle ASM (Oracle Automatic Storage Management) Instance 에 대한 인증 암호 파일을 사용할 수 있습니다. Oracle 데이터베이스의 암호 파일을 데이터베이스 암호 파일이라고하며 Oracle ASM의 암호 파일을 Oracle ASM 암호 파일이라고합니다.

■ 12c SYSDG 권한 확인

```
SQL> select * from dba_sys_privs where GRANTEE='SYSDG';
```

GRANT PRIVILEGE	ADM	COM	INH
SYSDG ALTER SESSION	NO	NO	NO
SYSDG ALTER SYSTEM	NO	NO	NO
SYSDG ALTER DATABASE	NO	NO	NO
SYSDG SELECT ANY DICTIONARY	NO	NO	NO

2.1.2 orapwd Utility 를 사용하여 Password File 생성

orapwd Command 구문은 다음과 같습니다.

■ Syntax

```
orapwd FILE=filename [ENTRIES=numusers] [FORCE={y|n}] [ASM={y|n}]
[DBUNIQUENAME=dbname] [FORMAT={12|legacy}] [SYSBACKUP={y|n}] [SYSDG={y|n}]
[SYSKM={y|n}] [DELETE={y|n}] [INPUT_FILE=input-fname]
```

ex)

```
$ orapwd FILE=filename ENTRIES=max_users FORMAT=12
```

- Format 인수 값이 12 인 경우 암호 파일은 Oracle Database 12c 형식으로 생성 됩니다. 이 형식은 SYSBACKUP, SYSDG, SYSKM 관리 권한을 지원합니다.
 - ➔ SYSDBA, SYSDG 및 SYSKM 은 FORMAT=12 인수로 작성된 경우에만 Password File 에서 지원됩니다. FORMAT 은 12 가 기본값 입니다.
- 기본적으로 Password File 암호는 대소문자를 구분합니다.
- REMOTE_LOGIN_PASSWORDFILE 파라미터 exclusive (Default) 설정 확인
REMOTE_LOGIN_PASSWORDFILE 정적 파라미터로 변경이 필요할 시 재시작하지 않으면 변경할 수 없습니다.
- SYSDG 인수는 선택 사항이며 암호를 묻는 메시지를 Y 로 설정하여 지정하여 줍니다. 암호는 작성된 파일에 저장됩니다.

2.1.3 SYSDBA, SYSOPER, SYSBACKUP, SYSDG, SYSKM 권한 부여

```
SQL> GRANT SYSDG to mydba;
```

2.1.4 인증 파일을 이용하여 Connect

암호 파일 인증을 사용하면 관리 사용자를 SQL * Plus CONNECT 명령 을 사용하여 로컬 또는 원격 데이터베이스에 연결하고 인증 할 수 있습니다. 기본적으로 암호는 대소 문자를 구분합니다.

예를 들어 사용자 mydba 에게 SYSDBG 권한 이 부여 된 경우 다음과 같이 연결할 수 있습니다.

```
CONNECT mydba AS SYSDBG
```

2.2 Far Sync

Active Data Guard 는 물리적 standby DB 를 읽기 전용으로 열어서 사용하는 동시에, redo 전송 및 스탠바이 적용 프로세스가 모두 액티브 상태가 되도록 할 수 있습니다.

액티브 상태의 standby DB 에서 실행되는 쿼리는 업데이트된 결과를 반환 합니다.

Active Data Guard Far Sync 는 Oracle DB 12c 의 새로운 기능이며 거리에 관계없이 동기화된 standby DB 를 두고 성능에 영향을 받지 않으면서도 최소한의 비용과 복잡성을 가지고 데이터 손실을 0%로 데이터를 보호할 수 있습니다.

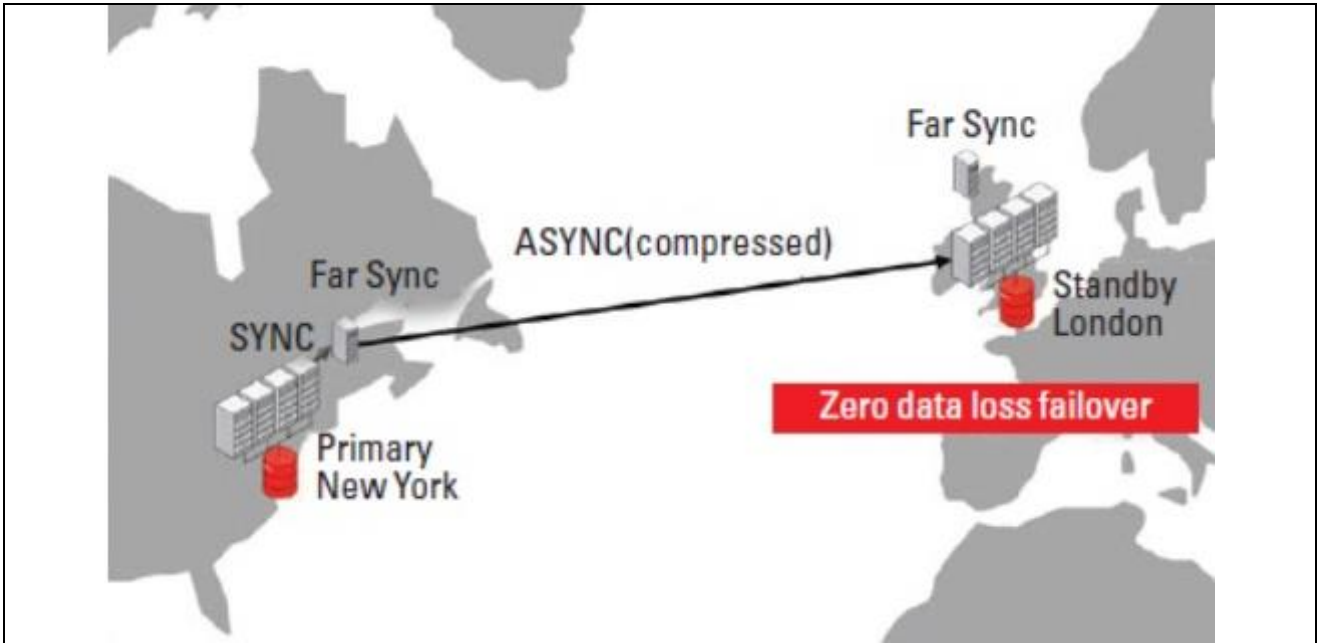
Far Sync 는 대기 상태에서 지리적으로 멀리 떨어져 대기 상태가되고 성능상의 이유로 MaxAvailability 보호 모드를 선택할 수 없는 경우 매우 좋은 솔루션입니다. 이 솔루션을 사용하면 Far Sync 인스턴스에 장애가 발생할 경우 Primary Database (Active) 에 대한 대체 아카이브 대상을 선택할 수도 있습니다. 이 경우 기본 데이터베이스는 비동기적으로 Standby Database 에 직접 데이터를 보냅니다.

Far Sync 는 매우 가벼운 프로세스입니다.

하나의 물리적 서버가 여러 개의 Far Sync 인스턴스를 지원할 수 있으며, 각각은 원격 대상에 대한 데이터 손실을 제로 수준으로 방지합니다. 그림 6 은 Primary Database (Active) 가 단일 실제 시스템에서 Far Sync Hub 로 작동하는 단일 실제 시스템에서 다중 Far Sync 인스턴스로 제공되는 예 입니다. 기본 및 Far Sync 허브는 클라우드에 원격으로 배포 된 Standby Database 가 있는 구내에 있습니다. 이러한 구성 (기본 및 대기 데이터베이스 호스트 및 Far Sync 호스트)의 모든 시스템은 My Oracle Support Note 413484.1 에 설명 된 Data Guard 구성의 호환성 요구 사항을 충족해야 합니다.

Far Sync 는 데이터 손실 방지 기능을 구현하려는 사용자를 위해 재해 복구 사이트 위치를보다 유연하게 제공합니다. 이미 Data Guard 전송을 배포한 사용자라도 Primary Database 에 대한 성능 영향을 줄이기 위해 현재 Standby Database 위치 보다 Primary 에 가까운 Far Sync Instance 를 구성하면 도움이 됩니다.

- Data Guard 은 비동기식(데이터 손실률 거의 0)과 동기식(데이터 손실률 0)중에 선택할 수 있는 기능을 제공합니다.
- 비동기식 구성은 적용하기 쉽고, Primary 성능에 영향이 없고, primary와 standby DB의 거리에 제한도 없습니다.
- 동기식 전송은, 성능에 어느 정도 영향이 있고 이에 따라 거리에 제한도 생기게 됩니다. 성능에 영향을 받는 이유는 Primary DB가 Standby Acknowledges를 받기 전까지는 다음 Transaction을 진행하지 않기 때문입니다.
- Acknowledgement를 기다리는데 걸리는 시간은 Primary와 Standby DB 사이의 거리에 비례하며 직접적으로 어플리케이션 반응 속도와 처리량에 영향을 끼치게 됩니다. Fast Sync와 Active Data Guard Far Sync는 Oracle DB 12c의 새로운 2개의 기능이며, 이 한계점에 대해 다루고 있습니다.
- Oracle Database 12c에서는 이런 원거리 Data Guard에서도 제로 데이터 손실을 보장하기 위해서 Data Guard Far Sync라는 새로운 기능을 추가하였으며 Data Guard Far Sync는 Primary와 Standby 사이에서 변경되는 데이터를 중계해주는 중계소와 같은 역할을 하며 Controlfile, Redolog File, Archivelog 로만 구성되는 굉장히 가벼운 Instance 입니다.
- Primary에서부터 Oracle Data Guard Far Sync 인스턴스까지의 짧은 거리는 동기식으로 구성되며 Oracle Data Guard Far Sync Instance부터 Standby 까지의 먼 거리는 비동기식으로 구성하게 됩니다.
- Oracle Data Guard Far Sync 인스턴스는 가까운 Oracle Data Guard 원격 목적지로써 Primary Database로부터 Redo를 받아서 Oracle Data Guard를 구성하는 Standby 멤버들에게 그 Redo를 전달하는 역할을 하게 됩니다.
- Far Sync Instance는 standby와 동일하게 control file을 관리하며, redo를 받아 Standby Redo Logs (SRLs)로 전달, 그 SRL들을 Local Archived Redo Logs에 보관합니다. 그러나 Far sync 인스턴스는 User Data Files를 가지고 있지 않고, 접근 가능하도록 OPEN되지 않으며, Redo Apply를 실행할 수 없고, 절대로 Primary로써의 역할을 하지 못하고 어떤 종류의 standby DB로도 바뀔 수 없습니다.



<그림> 12c Active Data Guard Far Sync – Zero Data Loss Protection at any Distance

2.2.1 Far Sync 서버를 사용할 경우 오버헤드 감소

- Standby Database (예 : 네트워크 또는 대기 데이터베이스 중단 이후)가 수신 한 아카이브 로그의 갭을 해결합니다.
- 여러 Standby Database 가 있는 구성의 전송 오버 헤드를 다시 실행합니다. 기본 서버는 Far Sync 인스턴스로 한 번 배송되며 Far Sync 는 여러 대상으로의 배송을 담당합니다.
- Far Sync Instance 는 오프 호스트 네트워크 압축을 수행하여 Primary Database 성능에 영향을 미치지 않고 WAN 대역폭을 보존합니다.
- Far Sync Instance 는 로그 파일 전송을 위한 최소한의 디스크 공간만 필요하고 발생할 수 있는 GAP 간격을 해결하기 위해 Archiving 된 Redo Log를 유지해야 합니다.
- Far Sync Instance 는 최소한의 SGA 만을 필요로 하며 하나 이상의 CPU를 소비 합니다. (Redo 전송 시 압축 기능을 수행하는 경우는 제외)

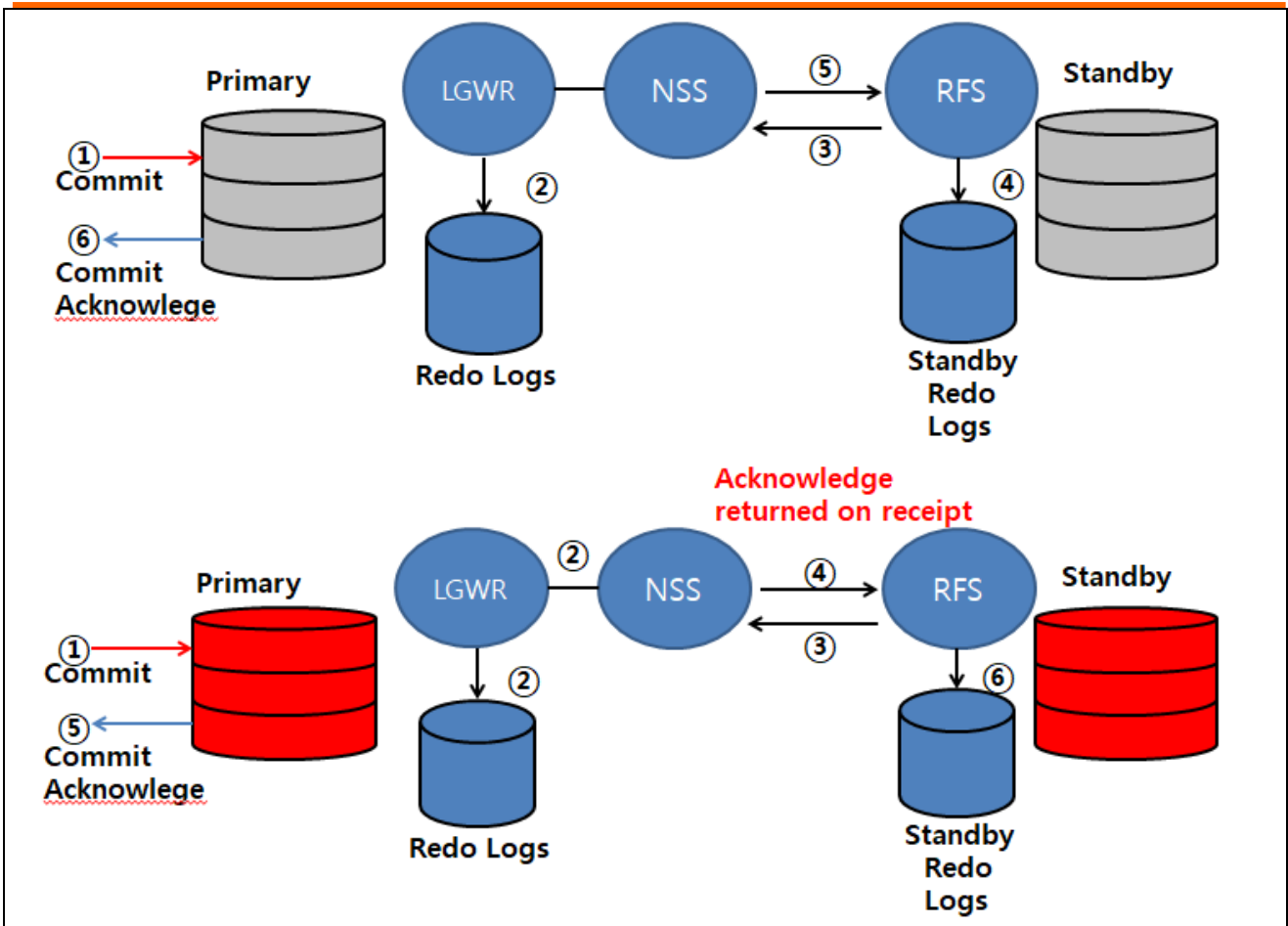
2.2.2 Far Sync 서버를 사용할 경우 라이선스

- Far Sync Instance 에는 사용자 데이터 파일, 미디어 복구 및 Oracle Database 라이선스가 필요하지 않습니다.
- Far Sync HA 구성 시 인스턴스 중단은 Primary / Standby Database 의 가용성에 영향을 주지 않습니다.
- Far Sync HA의 효과적인 대안은 Far Sync를 Oracle RAC에 배치하는 것 입니다. (이 경우 Oracle RAC 라이선스가 필요)
- Far Sync는 최대 30 개의 원격 대상을 지원 가능 (remote 29개) 합니다.

2.3 Fast Sync

12c에서는 Fast Sync 기능을 통해 zero 데이터 손실을 위한 동기적 구성에서 성능을 향상시킬 수 있는 손쉬운 방법을 제공합니다. Fast Sync에서는 standby DB가 redo 로그 파일로 디스크 I/O가 일어나기를 대기할 필요 없이 메모리상에 redo를 받자마자 primary로 ACK을 할 수 있도록 해줍니다.

이렇게 함으로써 전체 round-trip time을 줄일 수 있게 되어 동기적 전송에 대한 primary DB 성능에 미치는 영향을 보다 최소화할 수 있습니다.



<그림> 12c Data Guard Fast Sync

2.3.1 Data Protection Modes

- Data Guard 구성에 사용할 수 있는 Protection Mode 는 세 가지가 존재합니다.

- **Maximum Protection**

해당 모드는 Primary Database Fail 과 관련해 No Data Loss 를 보장 합니다. 이를 지원하기 위해 각 트랜잭션에 대해 Recover 할 수 있는 Redo Data 가 트랜잭션 Commit 전에 반드시 Local Online Redo 로그와 최소한 하나 이상의 Standby Redo Log 에 쓰여져야 합니다. Data loss 가 일어나지 않았음을 확인하기 위해 Primary Database 는 오류로 인해 최소한 하나 이상의 Remote Standby Redo Log 에 Redo Stream 을 쓰지 못한 경우 Shutdown 됩니다.

- **Maximum Availability**

Maximum Protection Mode 와 마찬가지로 Redo Data 는 최소 하나 이상의 Remote Standby Redo Log 와 Local Online Redo 에 쓰여져야 Commit 합니다. 다른 점으로는 Remote Standby Redo Log 에 대한 Redo Stream 에 쓰기를 실패하더라도 Primary Database 를 Shutdown 하지 않습니다. 대신에 Maximum Performance 로 동작을 하는데 문제가 문제가 해소되고 Redo Log Gap 이 해소되면은 자동적으로 Maximum Availability 모드로 복귀하게 됩니다.

- **Maximum Performance**

이는 Primary Database 의 Performance 에 주는 영향을 최소화 한 Protection Mode 입니다. 이 경우 commit 에 대해 Online Redo Log 에 Redo Log 를 씌으로써 달성하며 최소한 하나 이상의 Standby Database 에 대해 Redo Data Stream 이 쓰여집니다.

하지만 이 데이터는 Redo Data 를 만들어낸 트랜잭션의 커밋에 대해 비동기적(Async)으로 쓰여집니다. 해당 모드는 Primary 와 Standby Database 간에 Network 상황이 좋지 않을 때 Primary Database 의 Performance 침해를 최소화 할 수 있습니다.

2.3.2 Data Protection Modes 에 필요한 Redo Transport 속성

- **AFFIRM** : Redo Transport 대상에서 수신된 Redo Data를 Standby Redo Log에 기록한 후 해당 데이터를 승인하도록 지정합니다.
- **NOAFFIRM** : Redo Transport 대상에서 수신된 리두 데이터를 Standby Redo Log에 기록하기 전에 해당 데이터를 승인하도록 지정합니다.
- AFFIRM이나 NOAFFIRM 속성을 모두 지정하지 않는 경우 SYNC 속성을 지정하면 **기본값(Default) 이 AFFIRM**이 되고 **ASYNC 속성을 지정하면 기본값이 NOAFFIRM** 이 됩니다.
- SYNC 에서는 11~31 터미널 구성이 지원이 가능하지만 ASYNC 에서는 1~10 터미널 구성 지원이 가능합니다.

Maximum Availability	Maximum Performance	Maximum Protection
AFFIRM	NOAFFIRM	AFFIRM
SYNC	ASYNC	SYNC
DB_UNIQUE_NAME	DB_UNIQUE_NAME	DB_UNIQUE_NAME

2.3.3 12c Fast Sync 설정

- 12cR1 (12.1.0.2) 이전 버전까지는 Maximum Availability 에서는 Sync / AFFIRM 만 지원되었습니다.
- 12c 이후부터는 **SYNC / NOAFFIRM** 으로 전송이 가능합니다.

```
SQL> alter system set log_archive_dest_2='service=STANDBY sync noaffirm
2 valid_for=(online_logfiles,primary_role) db_unique_name=STANDBY' scope=both;
```

System altered.

```
SQL> select protection_mode from v$database;
```

```
PROTECTION_MODE
-----
MAXIMUM PERFORMANCE
```

```
SQL> alter database set standby database to maximize availability;
```

Database altered.

```
SQL> select protection_mode from v$database;
```

```
PROTECTION_MODE
-----
MAXIMUM AVAILABILITY
```

→ Parameter File 을 Pfile 사용할 경우, log_archive_dest_2 를 SYNC NOAFFIRM 으로 기재, OPEN 후 MAXIMIZE AVAILABILITY 로 변경하면 됩니다.

2.4 Real-Time Cascade

12c 에선 Primary -> Local Standby -> Remote Standby Redo(Multiple Standby Database환경) 전송 시 Real-Time Cascade 로 Remote Standby 전송가능하며 Primary Log Switch 로 인한 대기 지연이 없으며 성능 오버헤드가 적습니다.

2.5 12c Database ActiveDataGuard Writable Temp Tables

Primary DB 에서 SYNC 모드 설정 시 발생하는 성능 저하가 감소하였으며 일시적으로 Read/Write 가 필요한 경우 ADG Standby 에서 Temporary Table 사용 가능합니다.

```
SQL> alter system set temp_undo_enabled=true;
```

System altered.

➔ System Level 로 TEMP_UNDO_ENABLED 파라미터를 TRUE 사용

```
SQL> show parameter temp_undo_enabled
```

```
NAME TYPE VALUE
```

```
-----  
temp_undo_enabled boolean TRUE
```

2.6 Simpler Rol Transitions

12c 에선 RAC 환경에서 Physical Standby Database 로 Switch-Over 를 위해서 한 노드를 제외한 전체 노드를 Shutdown 할 필요가 없습니다.

```
SQL> alter database switchover to india verify;
```

```
alter database switchover to india verify
```

```
*
```

```
ERROR at line 1:
```

```
ORA-16475: succeeded with warnings, check alert log for more details
```

```
SQL>
```

```
-- Black Box
```

```
alter database switchover to india verify
```

```
Wed Aug 06 22:49:50 2014
```

```
SWITCHOVER VERIFY WARNING: no standby redo logfiles configured. Standby redo logfiles  
are recommended configuration for physical standby database.
```

```
SWITCHOVER VERIFY: Send VERIFY request to switchover target india
```

```
SWITCHOVER VERIFY COMPLETE
```

```
ORA-16475 signalled during: alter database switchover to india verify...
```

```
SQL> alter database switchover to india verify;
```

```
Database altered.
```

```
SQL>
```

Simple Switchover with “SQL> alter database switchover to india verify;”

Simple **Failover** with “SQL> alter database failover to india **verify**;”

Verify 옵션을 사용해 Switchover 를 준비

```
SQL> ALTER DATABASE SWITCHOVER TO STBY VERIFY;
```

- Switchover 가 준비되면 Database Altered 메시지가 발생
- Switchover 가 가능하지 않으면, 상황에 맞는 error 메시지를 발생
- Initial switchover 명령이 완료되지 않으면, FORCE 옵션 사용 가능

2.7 Rolling Upgrade

Oracle 12c부터 DBMS_ROLLING Package 를 사용하여 S/W Upgrade 가 가능합니다.

이 해당 패키지(DBMS_ROLLING) Package 는 Database 관리 작업에도 사용 가능합니다.

- Adding Partitioning to nonpartitioned tables
- Changing BasicFiles Lobs to SecureFiles LOBs
- Changing XML-CLOB to Binary XML
- Altering a table to be OLTP-Compressed

2.8 12c Moving Online Data Files

```
SQL> select database_role from v$database;

DATABASE_ROLE
-----
PHYSICAL STANDBY
SQL> select file_id,file_name from dba_data_files where file_id=3;
  FILE_ID FILE_NAME
-----
          3 /u01/app/oracle/oradata/ORCL/datafile/sysaux_sd_01.dbf
SQL> alter database move datafile 3 to '/u01/app/oracle/oradata/ORCL/sysaux_nl_01.dbf';

Database altered.

SQL> select file_id,file_name from dba_data_files where file_id=3;
  FILE_ID FILE_NAME
-----
          3 /u01/app/oracle/oradata/ORCL/sysaux_nl_01.dbf
SQL>
```

- SQL> alter database move datafile ~
- Active Data Guard 에서 Physical Standby Recover 중일 경우에도 MRP 를 종료할 필요가 없습니다.
- STANDBY_FILE_MANAGEMENT 를 MANUAL 로 변경 할 필요가 없습니다.
- Physical Standby Database 는 반드시 Read-Only Open 상태여야 합니다.

3 테스트 구성

3.1 RAC to RAC Active Data Guard 구성

3.1.1 Oracle 12.2.0.1 RAC S/W 2SET 설치

Guard 구성을 위한 특별한 설치 Option 은 없으며 RAC(Primary) to RAC(Standby) 구성으로 진행하였습니다.

3.1.2 Oracle 12cR2 Parameter 변경 사항

Oracle 12cR2 에서는 DataGuard 관련 파라미터 중 다음 파라미터가 Deprecated 되었습니다.

- 참조 : <https://docs.oracle.com/database/121/UPGRD/deprecated.htm#UPGRD60057>

LOG_ARCHIVE_START

```
SQL> startup pfile='/u01/oracle/product/122/db/dbs/initSTANDBY1.ora'  
ORA-32006: LOG_ARCHIVE_START initialization parameter has been deprecated  
ORACLE instance started.  
  
Total System Global Area 1174405120 bytes  
Fixed Size 8619984 bytes  
Variable Size 553650224 bytes  
Database Buffers 603979776 bytes  
Redo Buffers 8155136 bytes  
Database mounted.  
Database opened.  
SQL>
```

- 12c Deprecated And Desupported Parameters

8.1.5 Deprecated and Desupported Parameters

In Oracle Database 12c, some database parameters are deprecated and desupported, or removed.

To obtain a current list of deprecated parameters, run the following query in SQL*Plus:

```
SQL> SELECT name from v$parameter
        WHERE isdeprecated = 'TRUE' ORDER BY name;
```

The query returns a list of the deprecated parameters as of the date and time that you run it. The following example shows the query results that appear:

```
active_instance_count
background_dump_dest
buffer_pool_keep
buffer_pool_recycle
commit_write
cursor_space_for_time
fast_start_io_target
global_context_pool_size
instance_groups
lock_name_space
log_archive_start
max_enabled_roles
parallel_automatic_tuning
parallel_io_cap_enabled
parallel_server
parallel_server_instances
plsql_v2_compatibility
remote_os_authent
resource_manager_cpu_allocation
sec_case_sensitive_logon
serial_reuse
sql_trace
standby_archive_dest
user_dump_dest
```

3.1.3 Primary DB & Standby DB Information

NAME	DBNAME	Instance Name	hostname	Version	Protection Mode
Primary DB	PRIMARY	PRIMARY1	primary1	12.2.0.1	MAXIMUM AVAILABILITY (ASYNC NOAFFIRM)
Primary DB	PRIMARY	PRIMARY2	primary2	12.2.0.1	MAXIMUM AVAILABILITY (ASYNC NOAFFIRM)
Standby DB	STANDBY	STANDBY1	stdby1	12.2.0.1	MAXIMUM AVAILABILITY (ASYNC NOAFFIRM)
Standby DB	STANDBY	STANDBY2	stdby2	12.2.0.1	MAXIMUM AVAILABILITY (ASYNC NOAFFIRM)

[primary ifconfig]

```

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

#Public
12.12.12.6 primary1.localdomain primary1
12.12.12.7 primary2.localdomain primary2

#Private
10.10.10.127 primary1-priv.localdomain primary1-priv
10.10.10.128 primary2-priv.localdomain primary2-priv

#Vip
12.12.12.8 primary1-vip.localdomain primary1-vip
12.12.12.9 primary2-vip.localdomain primary2-vip

#Grid Scan(dns)
12.12.12.10 primary-scan.localdomain primary-scan

```

[standby ifconfig]

```

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

#Public
12.12.12.11 stdby1.localdomain stdby1
12.12.12.12 stdby2.localdomain stdby2

```

```

#Private
10.10.10.125      stdb1-priv.localdomain  stdb1-priv
10.10.10.126      stdb2-priv.localdomain  stdb2-priv

#Vip
12.12.12.13      stdb1-vip.localdomain  stdb1-vip
12.12.12.14      stdb2-vip.localdomain  stdb2-vip

#Grid Scan(dns)
12.12.12.15      stdb-scan.localdomain  stdb-scan

```

3.1.4 Tnsnames.ora 설정

[Primary1 / Primary2 tnsnames.ora]

```

STANDBY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 12.12.12.15)(PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = STANDBY)
    )
  )
)

```

[Standby1 / Standby2 tnsnames.ora]

```

PRIMARY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 12.12.12.10)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PRIMARY)
    )
  )
)

```

3.1.5 Static Listener 생성

- Standby 한 쪽 Node 에서 oracle 유저로 Static Listener 기동
- Primary 와 Standby 의 Role 변경 시 반대로도 Static Listener 가 필요
- \$ORACLE_HOME/db/network/listener.ora

```

LISTENER_STANDBY =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = 12.12.12.15)(PORT = 1522))
      )
    )
  )
)

SID_LIST_LISTENER_STANDBY =
  (SID_DESC =
    (GLOBAL_DBNAME = STANDBY)
    (ORACLE_HOME = /u01/oracle/product/122/db)
  )
)

```

```
(SID_NAME = STANDBY1)
```

```
)
```

```
/u01/oracle/product/122/db/network/admin $lsnrctl status LISTENER_STANDBY
```

```
LSNRCTL for Linux: Version 12.2.0.1.0 - Production on 24-APR-2018 17:15:55
```

```
Copyright (c) 1991, 2016, Oracle. All rights reserved.
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=12.12.12.15)(PORT=1522)))  
STATUS of the LISTENER
```

```
-----  
Alias                LISTENER_STANDBY  
Version              TNSLSNR for Linux: Version 12.2.0.1.0 - Production  
Start Date           13-APR-2018 17:16:40  
Uptime               10 days 23 hr. 59 min. 15 sec  
Trace Level          off  
Security             ON: Local OS Authentication  
SNMP                 OFF  
Listener Parameter File /u01/oracle/product/122/db/network/admin/listener.ora  
Listener Log File    /u01/oracle/diag/tnslsnr/stdby1/listener_standby/alert/log.xml
```

```
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=12.12.12.15)(PORT=1522)))
```

```
Services Summary...
```

```
Service "STANDBY" has 1 instance(s).
```

```
Instance "STANDBY1", status UNKNOWN, has 1 handler(s) for this service...
```

```
The command completed successfully
```

3.1.6 Password File 배포

- 모든 Node 에 orapw (Password File 배포) 파일을 배포합니다.

```
ASMCMD> pwget --dbuniquefilename primary
```

```
+VOL3/PRIMARY/PASSWORD/pwdprimary.256.971456723
```

```
ASMCMD>
```

```
ASMCMD> pwcopy +VOL1/ASM/PASSWORD/pwdasm.256.971435451 /tmp/orapwPRIMARY1
```

```
[root@primary1 ~]# cd /tmp
```

```
[root@primary1 tmp]# cp orapwPRIMARY1 /u01/oracle/product/122/db/dbs
```

```
[root@primary1 tmp]# su - oracle
```

```
/u01/oracle $cd $ORACLE_HOME/dbs
```

```
/u01/oracle/product/122/db/dbs $ls -l
```

```
total 16
```

```
-rw-rw---- 1 oracle dba 1544 Apr  8 12:46 hc_PRIMARY1.dat
```

```
-rw-r--r-- 1 oracle dba 3079 May 15 2015 init.ora
```

```
-rw-r--r-- 1 oracle dba 1714 Mar 27 18:35 initPRIMARY1.ora
```

```
-rw-rw---- 1 oracle dba 3072 Apr  9 14:22 orapwPRIMARY1
```

```
[root@primary1 dbs]# scp orapwPRIMARY1
```

```
root@12.12.12.7:/u01/oracle/product/122/db/dbs/orapwPRIMARY2
```

```
[root@primary1 dbs]# scp orapwPRIMARY1
```

```
root@12.12.12.11:/u01/oracle/product/122/db/dbs/orapworapwSTANDBY1
```

```
[root@primary1 dbs]# scp orapwPRIMARY1
```

```
root@12.12.12.12:/u01/oracle/product/122/db/dbs/orapworapwSTANDBY2
```

```
[root@primary1 dbs]# scp orapwPRIMARY1
root@172.40.40.130:/u01/oracle/product/122/db/dbs/orapwFALSYNC

chown -R oracle:dba orapw<SID>
chmod 660 orapw<SID>
```

3.1.7 Primary force_logging / supplemental log 설정

- DDL 문의 NOLOGGING 옵션 시에도 Redo Data 의 기록을 강제 하기 위한 설정

```
SQL> select force_logging from v$database;

FOR
---
NO

SQL> alter database force logging;

Database altered.

SQL> select force_logging from v$database;

FORCE_
-----
YES

SQL> alter database add supplemental log data;

SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;

SUPPLEME FORCE_LOGGING
-----
YES      YES
```

3.1.8 Archive Log 설정

```
SQL> create pfile from spfile;
→ Pfile 로 전환 기존 spfile 변경

vi initPRIMARY1.ora
vi initPRIMARY2.ora
```



```
PRIMARY1.LOG_ARCHIVE_FORMAT=PRIMARY1_%t_%s_%r.arc
PRIMARY2.LOG_ARCHIVE_FORMAT=PRIMARY2_%t_%s_%r.arc
PRIMARY1.LOG_ARCHIVE_DEST_1='LOCATION=+VOL2 VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=PRIMARY'
PRIMARY2.LOG_ARCHIVE_DEST_1='LOCATION=+VOL2 VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=PRIMARY'
```

```
SQL> startup mount
```

```
ORACLE instance started.
```

```
Total System Global Area 1174405120 bytes
```

```
Fixed Size 8619984 bytes
```

```
Variable Size 553650224 bytes
```

```
Database Buffers 603979776 bytes
```

```
Redo Buffers 8155136 bytes
```

```
Database mounted.
```

```
SQL> alter database archivelog;
```

```
Database altered.
```

```
SQL> alter database open;
```

```
Database altered.
```

```
SQL> archive log list
```

```
Database log mode Archive Mode
```

```
Automatic archival Enabled
```

```
Archive destination +VOL2
```

```
Oldest online log sequence 46
```

```
Next log sequence to archive 47
```

```
Current log sequence 47
```

→ 양 노드 각각 수행, RAC 환경에서 모든 클러스터 노드의 아카이브 생성 경로가 공유되어야 합니다.

3.1.9 Add Standby Logfile 추가

- Standby Redo Logfile 은 Primary DB Thread#별 Online Redo Logfile 보다 1개 더 많이 구성

```
SQL> set line 200
```

```
SQL> set pages 9999
```

```
SQL> set long 50000
```

```
SQL> col file_name for a50
```

```
SQL> col member for a50
```

```
SQL> select a.group#, a.thread#, a.bytes/1024/1024 MB, b.member
2   from v$log a, v$logfile b
3   where a.group# = b.group#;
```

GROUP#	THREAD#	MB	MEMBER
1	1	200	+VOL3/PRIMARY/ONLINELOG/group_1.258.971456765
2	1	200	+VOL3/PRIMARY/ONLINELOG/group_2.259.971456765
3	2	200	+VOL3/PRIMARY/ONLINELOG/group_3.266.971458279
4	2	200	+VOL3/PRIMARY/ONLINELOG/group_4.267.971458281

```
SQL> alter database add standby logfile thread 1 group 5 ('+VOL3/PRIMARY/ONLINELOG/group_5') size 500m;
SQL> alter database add standby logfile thread 1 group 6 ('+VOL3/PRIMARY/ONLINELOG/group_6') size 500m;
SQL> alter database add standby logfile thread 1 group 7 ('+VOL3/PRIMARY/ONLINELOG/group_7') size 500m;
SQL> alter database add standby logfile thread 2 group 8 ('+VOL3/PRIMARY/ONLINELOG/group_8') size 500m;
SQL> alter database add standby logfile thread 2 group 9 ('+VOL3/PRIMARY/ONLINELOG/group_9') size 500m;
SQL> alter database add standby logfile thread 2 group 10 ('+VOL3/PRIMARY/ONLINELOG/group_10') size
500m;
```

3.1.10 Init Parameter 추가

■ Primary

```
*.DB_UNIQUE_NAME=PRIMARY
*.log_archive_config='dg_config=(PRIMARY,STANDBY)'
PRIMARY1.LOG_ARCHIVE_FORMAT=PRIMARY1_%t_%s_%r.arc
PRIMARY2.LOG_ARCHIVE_FORMAT=PRIMARY2_%t_%s_%r.arc
PRIMARY1.LOG_ARCHIVE_DEST_1='LOCATION=+VOL2VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=PRIMARY'
PRIMARY2.LOG_ARCHIVE_DEST_1='LOCATION=+VOL2 VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=PRIMARY'
PRIMARY1.LOG_ARCHIVE_DEST_2='SERVICE=STANDBY LGWR
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=STANDBY'
PRIMARY2.LOG_ARCHIVE_DEST_2='SERVICE=STANDBY LGWR
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=STANDBY'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.log_archive_dest_state_3='enable'
PRIMARY1.fal_server=STANDBY
PRIMARY2.fal_server=STANDBY
```

```

PRIMARY1.fal_client=PRIMARY
PRIMARY2.fal_client=PRIMARY
*.DB_FILE_NAME_CONVERT='VOLUME3','+VOL3'
*.LOG_FILE_NAME_CONVERT='VOLUME3','+VOL3'
*.log_archive_start=TRUE
*.LOG_ARCHIVE_MAX_PROCESSES=8
*.STANDBY_FILE_MANAGEMENT=AUTO

```

■ Standby

```

*.control_files='+VOLUME3/STANDBY/CONTROLFILE/current.257.971456761' ### Duplicatiion 전 주석, 경로 변경
*.db_create_file_dest='+VOLUME3' ### 경로 변경

*.db_name='PRIMARY'
*.DB_UNIQUE_NAME='STANDBY'
*.log_archive_config='dg_config=(STANDBY,PRIMARY)'
STANDBY1.LOG_ARCHIVE_FORMAT=PRIMARY1_%t_%s_%r.arc
STANDBY2.LOG_ARCHIVE_FORMAT=PRIMARY2_%t_%s_%r.arc
STANDBY1.LOG_ARCHIVE_DEST_1='LOCATION=+VOLUME2 VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=STANDBY'
STANDBY2.LOG_ARCHIVE_DEST_1='LOCATION=+VOLUME2 VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=STANDBY'
STANDBY1.LOG_ARCHIVE_DEST_2='SERVICE=PRIMARY LGWR
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=PRIMARY'
STANDBY2.LOG_ARCHIVE_DEST_2='SERVICE=PRIMARY LGWR
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=PRIMARY'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.log_archive_dest_state_3='enable'
STANDBY1.fal_server=PRIMARY
STANDBY2.fal_server=PRIMARY
STANDBY1.fal_client=STANDBY
STANDBY2.fal_client=STANDBY
*.DB_FILE_NAME_CONVERT='+VOL3','VOLUME3'
*.LOG_FILE_NAME_CONVERT='+VOL3','VOLUME3'
*.log_archive_start=TRUE
*.LOG_ARCHIVE_MAX_PROCESSES=8
*.STANDBY_FILE_MANAGEMENT=AUTO

```

3.1.11 경로 생성

```
ASMCMD> pwd
```

```
+VOLUME3/primary/onlineolog
```

```
+VOLUME3/STANDBY
```

```
Standby 1 Node : mkdir -p /u01/oracle/admin/STANDBY/adump
```

```
Standby 2 Node : mkdir -p /u01/oracle/admin/STANDBY/adump
```

3.1.12 RMAN Duplication 수행

■ STANDBY 에서 수행 (STANDBY1 / STANDBY2)

```
SQL> startup nomount (1,2 Node 모두)
```

```
SQL> exit
```

```
$ rman target sys/oracle@PRIMARY(티엔에스네임) auxiliary sys/oracle@STANDBY(스탠드바이 티엔에스네임)
```

```
duplicate target database for standby from active database;
```

```
Starting restore at 2018-04-13:14:32:15
```

```
using channel ORA_AUX_DISK_1
```

```
channel ORA_AUX_DISK_1: starting datafile backup set restore
```

```
channel ORA_AUX_DISK_1: using network backup set from service PRIMARY
```

```
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_AUX_DISK_1: restoring datafile 00001 to +VOLUME3
```

```
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:08
```

```
channel ORA_AUX_DISK_1: starting datafile backup set restore
```

```
channel ORA_AUX_DISK_1: using network backup set from service PRIMARY
```

```
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_AUX_DISK_1: restoring datafile 00002 to +VOLUME3
```

```
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:15
```

```
channel ORA_AUX_DISK_1: starting datafile backup set restore
```

```
channel ORA_AUX_DISK_1: using network backup set from service PRIMARY
```

```
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_AUX_DISK_1: restoring datafile 00003 to +VOLUME3
```

```
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:02
```

```
channel ORA_AUX_DISK_1: starting datafile backup set restore
```

```
channel ORA_AUX_DISK_1: using network backup set from service PRIMARY
```

```
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_AUX_DISK_1: restoring datafile 00004 to +VOLUME3
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: using network backup set from service PRIMARY
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00005 to +VOLUME3
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:02
Finished restore at 2018-04-13:14:32:43
```

sql statement: alter system archive log current

contents of Memory Script:

```
{
  switch clone datafile all;
}
```

executing Memory Script

datafile 1 switched to datafile copy

```
input datafile copy RECID=6 STAMP=973347595 file
name=+VOLUME3/STANDBY/DATAFILE/system.263.973347565
```

datafile 2 switched to datafile copy

```
input datafile copy RECID=7 STAMP=973347595 file
name=+VOLUME3/STANDBY/DATAFILE/sysaux.262.973347573
```

datafile 3 switched to datafile copy

```
input datafile copy RECID=8 STAMP=973347595 file
name=+VOLUME3/STANDBY/DATAFILE/undotbs1.261.973347589
```

datafile 4 switched to datafile copy

```
input datafile copy RECID=9 STAMP=973347595 file
name=+VOLUME3/STANDBY/DATAFILE/undotbs2.260.973347591
```

datafile 5 switched to datafile copy

```
input datafile copy RECID=10 STAMP=973347595 file
name=+VOLUME3/STANDBY/DATAFILE/users.259.973347591
```

Finished Duplicate Db at 2018-04-13:14:32:54

→ 정상 수행 화면

3.1.13 MRP 기동 및 Apply 확인

- Duplication 이 완료 되면 \$ORACLE_HOME/dbs/initSTANDBY1.ora /

*.control_files='+VOLUME3/STANDBY/CONTROLFILE/current.269.973353245' -> 로그 확인 후 경로 추가

■ MRP 기동

```
○
SQL> startup mount

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
Database altered.
→ Start Redo Apply

SQL> alter database recover managed standby database cancel;
Database altered.
→ Stop

SQL> alter database open read only;
Database altered.
→ Active DataGaurd 가 아니면 OPEN X

SQL> alter database recover managed standby database using current logfile disconnect;
→ Real-Time Redo Apply

SQL> select open_mode from v$database;

OPEN_MODE
-----
READ ONLY WITH APPLY
```

3.2 12c Far Sync 구성

- 위에서 언급한대로 Far Sync Instance 는 원거리 Data Guard 에서도 제로 데이터 손실을 보장 받기 위해 중계소 같은 역할을 하며, Controlfile, Redolog File, Archive Log 로만 구성되는 굉장히 가벼운 Instance 로 절대로 Primary 로써의 역할을 수행하지 못하고 어떤 종류의 Standby DB 로도 바뀔 수 없습니다.

3.2.1 Primary 에서 Controlfile 을 생성

```
SQL> ALTER DATABASE CREATE FAR SYNC INSTANCE CONTROLFILE AS '/arch/BEIJING/control01.ctl';
```

→ Primary DB 는 꼭 OPEN 상태일 필요는 없으며 최소 Mount 상태여야 합니다.

3.2.2 Parameter 생성

- 최소한 Far Sync Instance 의 DB_UNIQUE_NAME 과 Control_files 의 경로는 반드시 수정되어야 합니다.
- Primary Database 는 서울, FarSync Database 는 베이징, Standby Database 는 런던 예시 입니다.
- 테스트는 MAXIMUM PERFORMANCE - LGWR 만 설정하여 테스트를 진행하였습니다.

Primary Database SEOUL

```
*.sga_target=1185m
*.undo_tablespace='UNDOTBS1'

### PRIMARY ###
*.DB_UNIQUE_NAME=SEOUL
*.FAL_SERVER=LONDON
*.FAL_CLIENT=SEOUL
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(SEOUL,BEIJING,LONDON) '
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SEOUL '
*.LOG_ARCHIVE_DEST_2='SERVICE=BEIJING LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=BEIJING '
*.LOG_ARCHIVE_DEST_3='SERVICE=LONDON LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=LONDON '
*.LOG_ARCHIVE_DEST_STATE_2=ENABLE
*.LOG_ARCHIVE_DEST_STATE_3=ENABLE
*.DB_FILE_NAME_CONVERT='/data/LONDON','/data/SEOUL '
*.LOG_FILE_NAME_CONVERT='/data/LONDON','/data/SEOUL '

```

```
### PRIMARY ###
*.DB_UNIQUE_NAME=SEOUL
*.FAL_SERVER=LONDON
*.FAL_CLIENT=SEOUL
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(SEOUL,BEIJING,LONDON) '
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SEOUL '
*.LOG_ARCHIVE_DEST_2='SERVICE=BEIJING LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=BEIJING '
*.LOG_ARCHIVE_DEST_3='SERVICE=LONDON LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=LONDON '
*.LOG_ARCHIVE_DEST_STATE_2=ENABLE
*.LOG_ARCHIVE_DEST_STATE_3=ALTERNATE
*.DB_FILE_NAME_CONVERT='/data/LONDON','/data/SEOUL '
*.LOG_FILE_NAME_CONVERT='/data/LONDON','/data/SEOUL '

```

FarSync Database BEIJING

```
*.sga_target=1185m
*.undo_tablespace='UNDOTBS1'

### STANDBY (FAR SYNC) ###
*.DB_UNIQUE_NAME='BEIJING'
*.FAL_SERVER='SEOUL'
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(SEOUL,BEIJING,LONDON)'
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=BEIJING'
*.LOG_ARCHIVE_DEST_2='SERVICE=LONDON LGWR VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=LONDON'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.LOG_FILE_NAME_CONVERT='/data/SEOUL','/data/BEIJING'

### STANDBY (FAR SYNC) ###
*.DB_UNIQUE_NAME='BEIJING'
*.FAL_SERVER='SEOUL'
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(SEOUL,BEIJING,LONDON)'
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=BEIJING'
*.LOG_ARCHIVE_DEST_2='SERVICE=LONDON LGWR VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=LONDON'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.LOG_FILE_NAME_CONVERT='/data/SEOUL','/data/BEIJING'
```

Physical Standby Database LONDON

```
*.sga_target=1185m
*.undo_tablespace='UNDOTBS1'

### STANDBY ###
*.DB_UNIQUE_NAME='LONDON'
*.log_archive_config='dg_config=(SEOUL,BEIJING,LONDON)'
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=LONDON'
*.LOG_ARCHIVE_DEST_2='SERVICE=SEOUL LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=SEOUL'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.FAL_SERVER=BEIJING,SEOUL
*.DB_FILE_NAME_CONVERT='/data/SEOUL','/data/LONDON'
*.LOG_FILE_NAME_CONVERT='/data/SEOUL','/data/LONDON'

### STANDBY ###
*.DB_UNIQUE_NAME='LONDON'
*.log_archive_config='dg_config=(SEOUL,BEIJING,LONDON)'
*.LOG_ARCHIVE_FORMAT=SEOUL_%t_%s_%r.arc
*.LOG_ARCHIVE_DEST_1='LOCATION=/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=LONDON'
*.LOG_ARCHIVE_DEST_2='SERVICE=SEOUL LGWR VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=SEOUL'
*.log_archive_dest_state_1='enable'
*.log_archive_dest_state_2='enable'
*.FAL_SERVER=BEIJING,SEOUL
*.DB_FILE_NAME_CONVERT='/data/SEOUL','/data/LONDON'
*.LOG_FILE_NAME_CONVERT='/data/SEOUL','/data/LONDON'
```


3.2.3 구성 확인

```
Primary Database SEOUL
SQL> select instance_name, status from v$instance;

INSTANCE_NAME  STATUS
-----
SEOUL          OPEN

SQL> select dest_id "ID",destination,status,target,
2 schedule,process,mountid mid
3 from v$archive_dest
4 where rownum < 4;

ID DESTINATION                                STATUS  TARGET          SCHEDULE
-----
PROCESS MID
-----
1 /arch                                VALID   PRIMARY         ACTIVE
ARCH      0
2 BEIJING                                VALID   STANDBY         ACTIVE
LGWR      0
3 LONDON                                 VALID   STANDBY         ACTIVE
LGWR      0

SQL> archive log list
Database log mode              Archive Mode
Automatic archival            Enabled
Archive destination           /arch
Oldest online log sequence    91
Next log sequence to archive  93
Current log sequence          93
SQL>
SQL> alter system switch logfile;

System altered.

SQL> █
```

→ 93 sequence archive 발생 시키고 Far sync 와 standby 적용 확인

FarSync Database BEIJING

```
SQL> select instance_name, status from v$instance;
```

```
INSTANCE_NAME    STATUS  
-----  
BEIJING          MOUNTED
```

```
SQL> column destination format a35 wrap  
column process format a7  
column archiver format a8  
column ID format 99  
column mid format 99
```

```
select dest_id "ID",destination,status,target,  
       schedule,process,mountid mid  
from v$archive_dest  
where rownum < 3;SQL> SQL> SQL> SQL> SQL> SQL> 2 3 4
```

```
ID DESTINATION                                STATUS    TARGET                                SCHEDULE PROCESS MID  
-----  
1 /arch                                        VALID     LOCAL                                ACTIVE   ARCH    0  
2 LONDON                                       VALID     REMOTE                                PENDING LGWR    0
```

```
SQL> █
```

```
2019-07-12T21:03:34.347999+09:00
```

```
Archived Log entry 64 added for thread 1 sequence 93 rlc 1013417767 ID 0x975b0ba7 LAD2 :
```

```
RFS[4]: No standby redo logfiles created
```

```
RFS[4]: Opened log for T-1.S-94 dbid 2539301031 branch 1013417767
```

```
█
```

Physical Standby Database LONDON

```
SQL> select instance_name, status from v$instance;

INSTANCE_NAME      STATUS
-----
LONDON              MOUNTED

SQL> column destination format a35 wrap
column process      format a7
column archiver     format a8
column ID           format 99
column mid          format 99

select dest_id "ID",destination,status,target,
       schedule,process,mountid mid
from v$archive_dest
where rownum < 3;SQL> SQL> SQL> SQL> SQL> SQL> SQL> 2 3 4

ID DESTINATION                                STATUS    TARGET          SCHEDULE PROCESS MID
-----
1 /arch                                       VALID     LOCAL           ACTIVE   ARCH      0
2 SEOUL                                       VALID     REMOTE          PENDING LGWR      0

SQL> !ps -ef |grep mrp
oracle  10430   1  0 20:27 ?          00:00:00 ora_mrp0_LONDON
oracle  11071 11061  0 21:12 pts/0        00:00:00 /bin/bash -c ps -ef |grep mrp
oracle  11073 11071  0 21:12 pts/0        00:00:00 grep mrp

SQL> █
```

```
2019-07-12T21:03:34.837679+09:00
RFS[8]: Selected log 4 for T-1.S-94 dbid 2539301031 branch 1013417767
2019-07-12T21:03:34.837897+09:00
Archived Log entry 35 added for T-1.S-93 ID 0x975b0ba7 LAD:1
2019-07-12T21:03:34.916104+09:00
Media Recovery Waiting for thread 1 sequence 94 (in transit)
2019-07-12T21:03:34.916833+09:00
Recovery of Online Redo Log: Thread 1 Group 4 Seq 94 Reading mem 0
Mem# 0: /data/LONDON/redo04.log
█
```