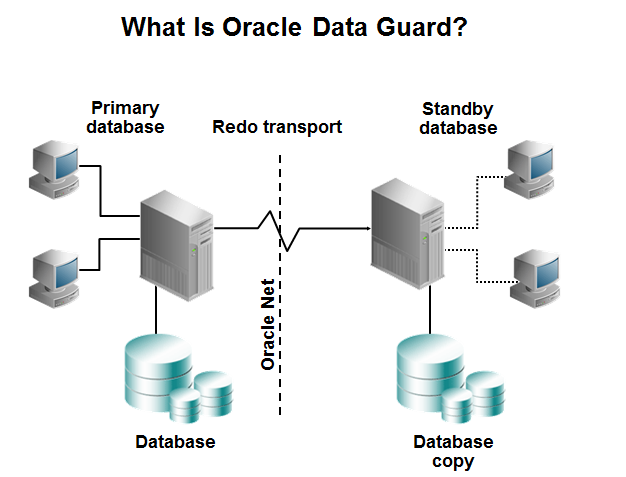
Oracle Data Guard 23ai

# Data Guard Architecture Summary

## Giới thiệu về Oracle Data Guard



Oracle Data Guard là 1 công nghệ của Oracle, cho phép bảo vệ DB khỏi những lỗi nghiêm trọng nhất có thể xảy ra.

* Nó tạo ra 1 bản sao của DB, gọi là Standby DB để lưu trữ lại những dữ liệu của Primary DB đề phòng những tình huống xấu nhất có thể xảy ra.
* Standby DB được update tất cả dữ liệu, redo log của Primary DB và chỉ có thể truy cập ở chế độ read-only để truy vấn và làm báo cáo dữ liệu

## Phân loại Standby Databases

**Physical standby database**

* Giống hệt với Primary DB ở mức dữ liệu (Database data block) nhưng có thể khác ở các thông số cấu hình instance (memory, file location)
* Đồng bộ với Primary DB bằng cách cập nhật các data redolog nhận được từ Primary DB
* Vừa có thể bảo vệ dữ liệu và trích xuất báo cáo
* Chỉ có thể đọc, không được ghi dữ liệu

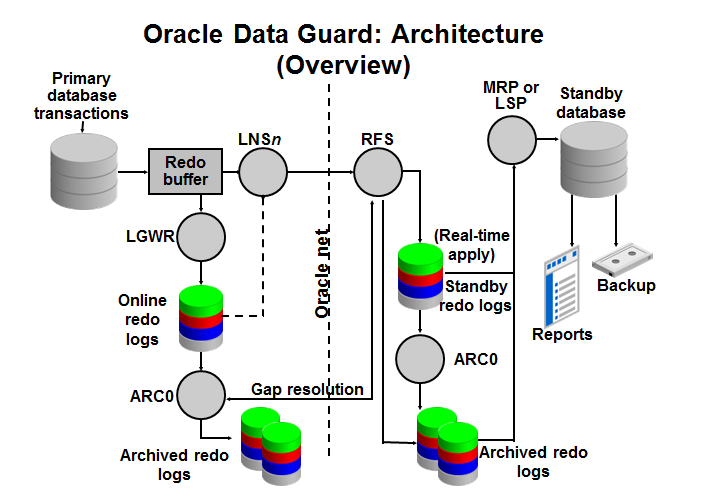
**Logical Standby Database**

* Giống vs Primary DB ở chế độ logical nhưng có thể khác nhau ở chế độ physical.
* Đồng bộ dữ liệu vs Primary DB bằng cách chuyển đổi những dữ liệu data redolog nhận được thành các câu lệnh SQL và thực thi chúng
* Có thể được sử dụng cho công việc: bảo vệ dữ liệu, trích xuất báo cáo và nâng cấp database.
* Cho phép ghi và đọc dữ liệu

**Snapshot standby database**

* Được tạo ra bằng việc convert physical standby DB
* Nó cũng nhận được data redolog từ Primary database những sẽ ko chạy chúng cho đến khi được convert ngược lại physical standby DB. Tất cả mọi update ở chế độ local sẽ bị xóa khi convert ngược lại physical standby DB.
* Được sử dụng cho công việc testing
* Cho phép ghi và đọc dữ liệu

## Data guard services



Primary Database Processes

* Log writer (LGWR): ghi log của các transaction vào redo log.
  + - Đối với mỗi Standby DB động bộ, LGWR sẽ chuyển dữ liệu cho LNSn (Log Writer Network Server), cái mà sẽ chuyển dữ liệu cho RFS (remote file server) ở Stanby database. Log writer sẽ chờ đợi Log Writer Network Server xác nhận trước khi commit
    - Đối với các Stanby DB ko đồng bộ, 1 tiến trình LNS độc lập sẽ đọc dữ liệu tại redolog buffer hoặc online redolog file và chuyển tới Stanby Database
* Archiver (ARC*n*): Tạo bản sao các online redologs file để sử dụng cho quá trình recovery. Nó cũng chịu trách nghiệm chuyển dữ liệu qua RFS ở StanbyDB. Nếu có tình trạng mất kết nối xả y ra, ARCn sẽ chịu trách nghiệm chuyển lại dữ liệu khi kết nối lại được.

Standby Database Processes

* Remote file server (RFS): Nhận các redo data từ Primary DB và ghi vào redolog của StandbyDB hoặc ghi trực tiếp vào arichive reodlogs. Đối với mỗi LNS và ARC thì tương ứng có 1 RFS
* Archiver (ARC*n*): ghi dữ liệu từ redolog xuongs archivelog.
* Managed recovery (MRP): sử dụng cho physical standby DB, MRP applies các dữ liệu ở redolog nhận được PrimaryDB
* Logical standby (LSP): sử dụng cho logical standby DB, LSP quản lý cách dữ liệu được apply vào logical standby DB

Role management services

* Là dịch vụ cho phép chuyển đổi vai trò giữa PrimaryDB và StanbyDB
* Switchover: Tính năng này cho phép thay đổi vai trò giữa PrimaryDB và 1 StanbyDB được lựa chọn.
* Failover: Tính năng này cho phép loại bỏ PrimaryDB khi nó bị sự cố nghiêm trọng ko thể khôi phục và cho phép StanbyDB chỉ định thay thế nó và trở thành PrimaryDB

## Dataguard protect Mode

Là chế độ được áp dụng cho redo tranport mode:

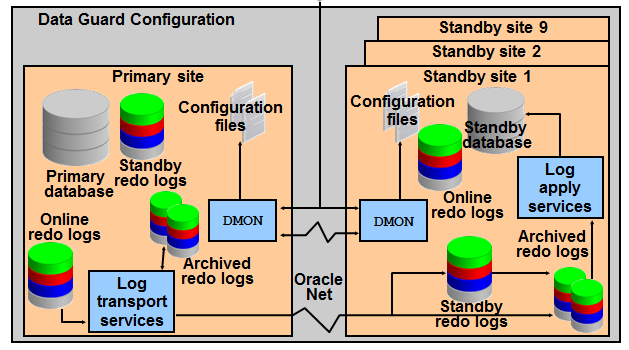
* Maximum protection:
  + Được sử dụng để không mất mát dữ liệu
  + Đảm bảo rằng trước khi transaction commit, thì cả PrimaryDB và StandbyDB đều phải ghi log vào redolog
  + Nếu như dữ liệu ko thể ghi vào redo của StandbyDB thì primary sẽ tự động shutdown để bảo đảm tính toàn vẹn cho dữ liệu
* Maximum availability
  + Được sử dụng để tránh mất mát dữ liệu mà ko ảnh hưởng tính sẵn sàng của hệ thống
  + Đảm bảo rằng trước khi transaction commit, thì cả PrimaryDB và StandbyDB đều phải ghi log vào redolog
  + Nếu như dữ liệu ko thể ghi vào redo của StandbyDB thì primary sẽ hoạt động như chế độ max performance cho đến khi có thể ghi được dữ liệu vào standby redolog của standbyDB
* Maximum performance (default)
  + Là chế độ mặc định: thường được sử dụng nhiều nhất do tốc độ cao
  + Dữ liệu ít sai lệch nhất mà ko ảnh hưởng tới hiệu năng của PrimaryDB.
  + Transaction có thể commit bất cứ lúc nào và log được ghi vào redolog. Dữ liệu ở redo được chuyển sau đó sang StandbyDB sau.

# Oracle Data Guard Broker

## Oracle Data Guard Broker summary

Oracle Data Guard Broker:

* Sử dụng để quản lý DataGuard
* Tự động khởi tạo, cấu hình và theo dõi DataGuard configurations
  + Cho phép tối đa 9 StandbyDB
    - Quản lý các services: redo transport services, and log apply services…
    - Thay đổi role giữa các DB …
* Hiệu quả: cho phép nhất quán dữ liệu ở mức:
  + Configuration object: database name, status…
  + Database objects
  + Instance objects: Rac…
* Broker Service:
  + Khi được yêu cầu khởi tạo, background process của Oracle Database: DMON (Data Guard monitor) sẽ được khởi động. Tiến trình DMON này sẽ Kết nối với tất cả các DMON khác được ghi trong file cấu hinh.
  + File cấu hình: 2 file



## Sử dụng Oracle Data Guard Broker

Khởi tạo và quản lý Oracle Data Guard Broker

* Set tham số DG\_BROKER\_START thành TRUE.
* Truy câp vào DGMGRL và kết nối tới PrimaryDB
* Set các tham số cấu hình cho file config
* Thêm Standby DB vào file cấu hình.
* Cho chạy file cấu hình primary database.
* Thực hiện:

# Sử dụng tên của Global\_unique\_name

# Set máy primary có role là Primary

DGMGRL> CREATE CONFIGURATION 'DGConfig1' AS

PRIMARY DATABASE IS primary

CONNECT IDENTIFIER IS primary;

#Add/ remove standby vào file config

DGMGRL> ADD DATABASE standby AS

> CONNECT IDENTIFIER IS standby;

DGMGRL> DISABLE DATABASE 'standby';

DGMGRL> REMOVE DATABASE 'standby';

#Enable/disable config

ENABLE/DISABLE CONFIGURATION;

#Kiểm tra

DGMGRL> SHOW CONFIGURATION

Một số câu lệnh thường sử dụng:

#Xem các thuộc tính

DGMGRL> SHOW DATABASE VERBOSE

#Chỉnh sửa các thuộc tính

DGMGRL> EDIT DATABASE standby

> SET PROPERTY LogXptMode='SYNC';

DGMGRL> EDIT DATABASE standby SET STATE='APPLY-OFF';

DGMGRL> EDIT DATABASE pc00prmy

> SET STATE='TRANSPORT-OFF';

**TRANSPORT-ON** : dành riêng cho primaryDB: cho phép chuyển redo khi primaryDB được open

**TRANSPORT-OFF** dành riêng cho primaryDB: dừng cho phép chuyển redo khi primaryDB được open

**APPLY-ON** dành riêng cho standbyDB: cho phép apply redo được chuyển từ primaryDB

**APPLY-OFF** dành riêng cho standbyDB: dừng cho phép apply redo được chuyển từ primaryDB

**ADD**: Adds a standby database to the broker configuration

**CONNECT**: Connects a given username to the specified instance

**CREATE**: Enables you to create broker configurations

**DISABLE**: Enables you to disable broker control of a configuration or database so that the object is no longer managed by the broker

**EDIT:** Used to edit a configuration, database, or instance

**ENABLE:** Enables you to enable broker control of a configuration or database

**EXIT/QUIT**: Exits DGMGRL

**FAILOVER:** Performs a database failover operation in which one of the standby databases changes to the role of primary database (This is an unplanned transition that may result in the loss of application data.)

**HELP**: Displays online help for the commands in DGMGRL

**REINSTATE**: Changes a disabled database into a viable standby database

**REMOVE**: Removes a broker configuration, including all of its database profiles, a specified standby database profile, or knowledge of an instance

**SHOW**: Displays either a brief or a detailed summary of information about the broker configuration, database, or instance; can also display the dependency tree and default online states for the broker configuration, as well as the configuration log or the Oracle database alert log

**SHUTDOWN:** Shuts down a currently running Oracle database instance

**START**: Starts the Fast-Start Failover Observer

**STARTUP**: Starts an Oracle instance with several options, including mounting and opening a database

**STOP:** Stops the Fast-Start Failover Observer

**SWITCHOVER**: Performs a switchover operation in which the current primary database becomes a standby database and the standby database to which the CLI is currently connected becomes the primary database

# Setup Physical StandbyDB manualy

## Chuẩn bị PrimaryDB

Bật chế độ FORCE LOGGING ở mức database.

* Chế độ này đảm bảo cho dữ liệu nhất quán
* Luôn luôn tạo redo log ngay cả khi gặp no logging

## Tạo standby redo logs.

Standby redo logs được sử dụng ở các database có vai trò là StanbyDB để lưu trữ các redo data nhận được từ PrimaryDB.

* Nên tạo cả standby redo log files trên PrimarnyDB vì đây là 1 option cho việc hỗ trợ thay đổi vai trò của các database. Nên cấu hình trên tất cả các database
* Stanby redolog file dùng để hỗ trợ quá trình cập nhật thời gian thực
* Oracle đề nghị số standby redo logs groups sẽ có cùng size và có nhiều hơn online redolgos file 1 group.
* Ví d :

Thêm

SQL> ALTER DATABASE ADD STANDBY LOGFILE

'/usr/oracle/app/oradata/oratest/stb02.log'

SIZE 50M;

Xóa

SQL> ALTER DATABASE DROP STANDBY LOGFILE

'/usr/oracle/app/oradata/oratest/stb05.log'

* Xem thông tin các Standby Redo Log

SQL> SELECT group#, type, member FROM v$logfile

WHERE type = 'STANDBY';

## Khởi tạo các tham số parameters

LOG\_ARCHIVE\_CONFIG

* Tác dụng: liệt kê tất cả các DB trong cluster
* được cấu hình ở tất cả thành phần của cluster
* Cấu hình:

LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(pc00prmy,standby)'

* Xem thống số

SQL> show parameter log\_archive\_config

NAME TYPE VALUE

------------------- ------- ---------------------------

log\_archive\_config string dg\_config=(pc00prmy,standby)

SQL> SELECT \* FROM v$dataguard\_config;

DB\_UNIQUE\_NAME

------------------------------

pc00prmy

standby

LOG\_ARCHIVE\_DEST\_*n*

* Điều khiển hoạt động của di chuyển redo
* Chỉ ra số đích mà PrimaryDB cần chuyển redo
* Bao gồm: thông tin tên, services
* Ví dụ:

LOG\_ARCHIVE\_DEST\_1=

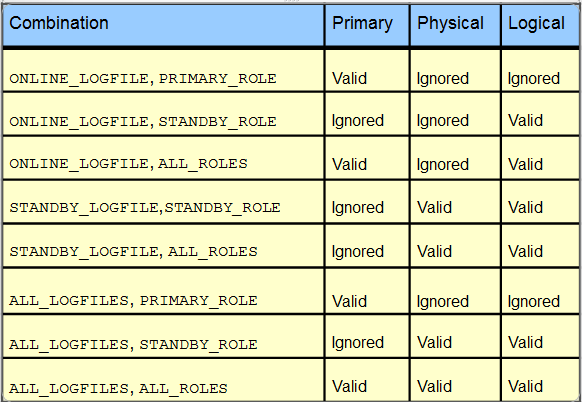
'SERVICE=standby

VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE)

DB\_UNIQUE\_NAME=standby'

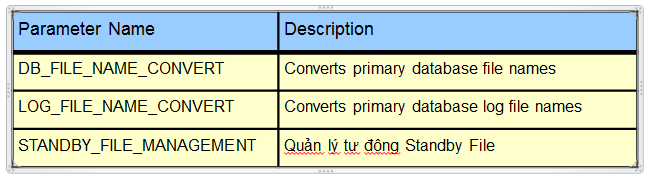
LOG\_ARCHIVE\_DEST\_STATE\_1=ENABLE

* Tham số VALID\_FOR:
  + Redolog type
    - * ONLINE\_LOGFILE: địa chỉ đích được sử dụng khi thực hiện archive ở online redo log.
      * STANDBY\_LOGFILE: địa chỉ đích được sử dụng khi thực hiện archive stanby logfile hoặc nhận được archive log từ DB khác
      * ALL\_LOGFILES: Thực hiện khi mọi archive xảy ra
  + Database\_role
    - * PRIMARY\_ROLE: địa chỉ đích được sử dụng khi DB có role là Primary
      * STANDBY\_ROLE: địa chỉ đích được sử dụng khi DB có role là Standby
      * ALL\_ROLES: địa chỉ đích được sử dụng khi DB có role bất kì
  + Sự kết hợp giữa 2 tham số này trong VALID\_FOR



* + SYNC and ASYNC:
    - * SYNC: đồng bộ dữ liệu thời gian thực, cứ khi sinh ra redolog là chuyển ngay qua StandbyDB
      * ASYNC: sau khi commit các giao dịch mới gửi redolog cho StanbyDB
      * Default: ASYNC
  + AFFIRM and NOAFFIRM
    - * AFFIRM: đảm bảo rằng địa chỉ đích biết được có redolog được chuyển đến sau khi nó được ghi thành công vào đĩa
      * NOAFFIRM: đảm bảo rằng địa chỉ đích biết được có redog được chuyển đến trước khi nó được ghi vào đĩa.

Một số tham số cần cấu hình thêm khi cấu trúc thư mục/đĩa của StanbyDB khác với PrimaryDB.



## Cấu hình thông số mạng

Cấu hình listener:

LISTENER =  
 (DESCRIPTION\_LIST =  
 (DESCRIPTION =  
 (ADDRESS=(PROTOCOL=TCP)(HOST=primary)  
 (PORT=1521)(SEND\_SDU=10485760)  
 (RECV\_SDU = 10485760)))  
 (DESCRIPTION =  
 (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521)))  
 )  
ADR\_BASE\_LISTENER = /u01/app/oracle  
LISTENER2 = …  
ADR\_BASE\_LISTENER2 = /u01/app/oracle

Cấu hình thông số tnsname

PRIMARY =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.10)

(PORT = 1521) (SEND\_BUF\_SIZE=10485760)

(RECV\_BUF\_SIZE=10485760))

)

(SDU=65535)

(CONNECT\_DATA =(SERVICE\_NAME = primary))

)

STANDBY =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.10)

(PORT = 1521) (SEND\_BUF\_SIZE=10485760)

(RECV\_BUF\_SIZE=10485760))

)

(SDU=65535)

(CONNECT\_DATA =(SERVICE\_NAME = standby))

)

## Cấu hình các thông số liên quan

Password file:

* Copy password file của PrimaryDB vào StandbyDB ở địa chỉ $ORACLE\_HOME/dbs
* Tiến hành đổi tên: orapw<SID>

Cấu hình tham số **FAL\_CLIENT và FAL\_SERVER**

* Mang lại mô hình server và client với các tính năng tốt
* Sử dung để switch over và fail over
* Đối với DB nào thì FAL\_CLIENT là nó và FAL\_SERVER là các DB còn lại:

# Tại StandbyDB

FAL\_CLIENT = 'standby'

FAL\_SERVER = 'primary'

# Tại PrimaryDB

FAL\_CLIENT = 'primary'

FAL\_SERVER = 'standby'