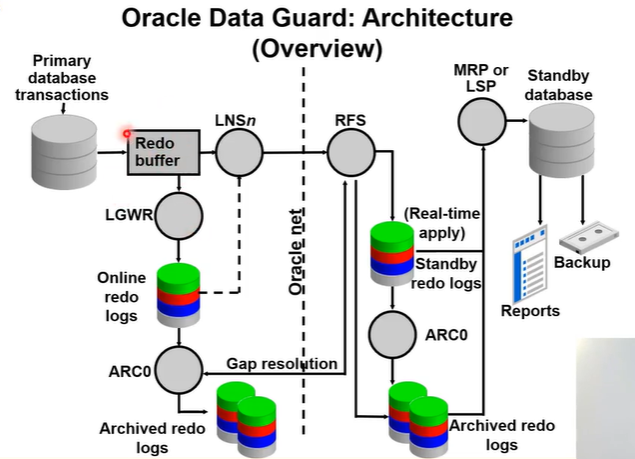
DATA GUARD

# tổng quan



Ngày nay để phòng tránh những rủi ro ngoài ý muốn gây chết hệ thống không thể khôi phụ dữ liệu như thiên tai, hỏa hoạn, … hoặc lỗi phần cứng, lỗi dữ liệu nhưng không thể khôi phục ngay lập tức thì Oracle xây dựng giải pháp sẵn sàng cao (high avaibility) Data Guard (DG). Hệ thống Data Guard gồm 1 server chính (primary) và nhiều server phụ (standby). Trong quá trình vận hành, primary thường sẽ đảm nhận vai trò cập nhật dữ liệu, còn standby đồng bộ dữ liệu từ primary về (real – time) và thường được dùng để lên báo cáo nhằm giảm tải cho primary.

Trong trường hợp primary gặp sự cố, thì standby sẽ đảm nhiệm vai trò của primary, và các ứng dụng sẽ được chuyển kết nối đến standby. Lúc này standby hoạt động không khác gì primary cho đến khi primary được phục hồi.

## khái niệm switchover và failover

### Switchover

Switchover là việc chuyển đổi vai trò giữa primary và standby. Lúc này chiều đồng bộ dữ liệu sẽ chuyển từ standby cũ (lúc này là primary) về primary cũ (lúc này là standby)

Khuyến cáo: nên swithover 6 tháng 1 lần. Chỉ switchover khi hệ thống success và đang bật tiến trình đồng bộ và tắt hết các tiến trình đang kết nối đến db.

Thực hiện:

* OS$: dgmgrl sys/oracle
* DGMGRL> switchover to standby;

### Failover

Failover là việc đổi standby thành primary, trong khi primary gặp sự cố không thể thực hiện switchover.

Thực hiện:

* OS$: dgmgrl sys/oracle
* DGMGRL> failover to standby;🡪 lúc này standby trở thành primary
* Sau khi primary cũ được hồi phục, thiết lập primary cũ thành standby và bật tiến trình đồng bộ: DGMGRL> reinstate database primary;

### Đồng bộ dữ liệu

Có 3 kiểu đồng bộ dữ liệu chính:

* Maximum protection
* Đảm bảo không có mất mát dữ liệu nào nếu primary database gặp sự cố
* Primary sẽ shutdown nếu việc đồng bộ xảy ra lỗi
* Maximum avaibility
* Đảm bảo không có mất mát dữ liệu
* Primary sẽ dừng hoạt động cho đến khi nhận được thông tin về tính trình đồng bộ sang standby
* Trong trường hợp standby không thể nhận dữ liệu từ Redo cả primary thì hệ thống sẽ tự động chuyển về hoạt động nhưng trạng thái maximum performance.
* Maximum performance (default)
* Các transaction trên primary không đợi việc ghi dữ liệu vào standby redo log

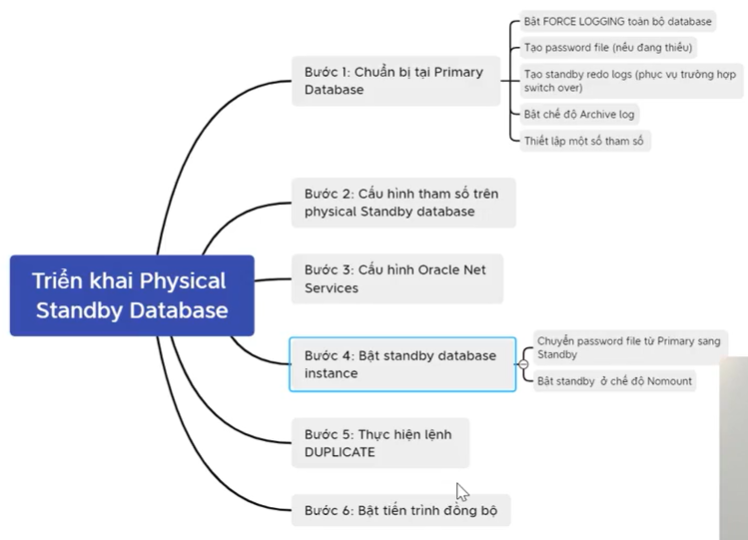
### Yêu cầu triển khai

Phần cứng giữa primary và standby có thể khác nhau về CPU, RAM, HDD, nhà sản xuất.

Hệ điều hành có thể khác nhau (cả phiên bản và hệ điều hành). Nếu khác nhau cần kiểm tra ma trận tương thích của Oracle:

Database software: cả primary và standby phải cùng phiên bản và chỉ áp dụng với Enterprise Edition

## Tạo physical standby



### Quy trình

* Bật archivelog mode
* Cấu hình Force logging: SQL> alter database force logging;
* Tạo standby redo log trên cả primary và standby. Số lượng = số redo log group + 1 (với hệ thống oracle stand alone)
* SQL> alter database add standby logfile group <group\_name> size 200M
* Chuyển các file từ primary sang standby: parameter và password file
* SQL> create pfile = ‘<path>’ from spfile
* OS$: cd $ORACLE\_HOME/dbs
* OS$: ls -l |grep orapw
* OS$: cp orapw… /home/oracle
* OS$: sftp <standby\_hostname>
* Sftp> cd /home/oracle
* Sftp> put orapw… (pw file vừa copy ở trên)
* Sftp> put pfile (pfile vừa tạo ở trên)
* Chuyển sang server standby
* Sửa pfile vừa nhận
* Thêm db\_unique\_name=’standby’
* Sửa tên controlfile
* Tạo các đường dẫn xuất hiện trong pfile
* Chuyển sang primary để cấu hình oracle net service
* Sửa tnsnames.ora. Cấu hình tns cả 2 server trên tnsname. Cả standby và primary đều cấu hình như nhau
* Chuyển sang standby. Cấu hình listener tĩnh
* Export ORACLE\_SID = …
* Startup nomount và tạo spfile từ pfile
* Copy password file vào đúng path vủa Oracle và đổi tên: $ORACLE\_HOME/dbs/
* OS$: mkdir script
* OS$: cd script
* OS$: vi duplicate.sql

Run{

DUPLICATE TARGET DATABASE FOR <STANBY\_host/db???\_NAME> FROM ACTIVE DATABASE DORECOVER NOFILENAMECHECK;

}

* Trước khi chạy script thì cần test tnsping
* OS$: rman auxiliary user/pass@STANDBY. Nếu lỗi thì đổi service name trong tnsname thành sid
* OS$: cd script
* OS$: nohup rman target sys/pass@PRIMARY auxiliary sys/pass@STANDBY cmdfile=/home/oracle/script/duplicate.sql log=/home/oracle/script/duplicate.log
* Alter instance open.
* Chuyển Primary: SQL> alter system set log\_archive\_config=’dg\_config=(talentcdb,standby)’; (primary\_db\_name.standby\_host???)
* Cấu hình archivelog sang standby:
* SQL> alter system set log\_archive\_dest\_2=’service=standby valid\_for=(online\_logfiles,primary\_roles)
* SQL> alter system set log\_archive\_dest\_state\_2=enable;
* SQL> alter system set standby\_file\_management=auto;
* Chuyển standby: SQL> alter system set log\_archive\_config=’dg\_config=(talentcdb,standby)’;
* SQL> alter system set standby\_file\_management=auto;
* Bật tiến trình đồng bộ:

SQL> alter database recover managed standby database using current logfile disconnect from session;

* Check: SQL> select process, sequence#, thread#, status from v$managed\_standby; MRP0 với status applying\_log

### Real time apply

* SQL> Select database\_role, open\_mode from v$database;
* Nếu open\_mode = read only 🡪 chưa bật đồng bộ
* SQL> alter database recover managed standby database using current logfile disconnect from session;
* SQL> Select database\_role, open\_mode from v$database;
* Nếu open\_mode = read only with apply 🡪 đã bật đồng bộ
* Tắt chế độ real time: SQL> alter database recover managed standby database cancel;

### Giám sát độ trễ

SQL> select name, value, time\_computed from v$dataguard\_stats;

Transport lag: truyền dữ liệu từ primary sang standby có độ trễ k

Apply lag: standby cập nhật dữ liệu nhận từ priamry có độ trễ k

## snapshot standby

Mục đích: sử dụng standby để thực hiện test, sau khi test xong thì quay lại thời điểm trước khi test và tiếp tục đồng bộ với primary 🡪 không cần dựng lại standby

Thực hiện:

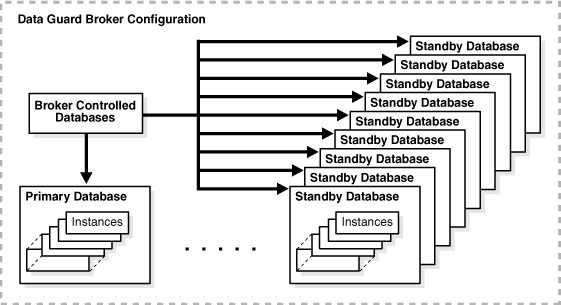
* Standby đã cấu hình FRA
* Tắt đồng bộ: SQL> alter db recover managerd standby db cancel;
* Bật db ở chế độ mount
* SQL> alter database convert to snapshot standby;
* SQL> alter database open;
* Check: SQL> select open\_mode from v$database; 🡪 read write

Đổi lại từ snapshot về apply

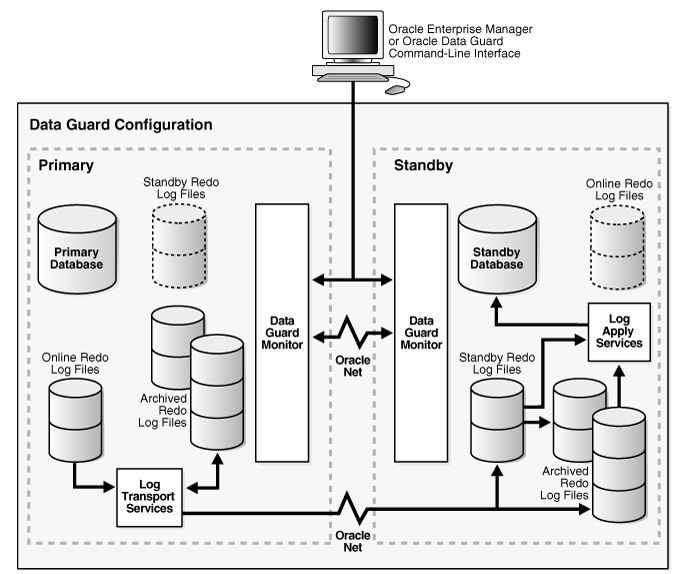
* SQL> shutdown immediate;
* SQL> startup mount;
* SQL> alter database convert to physical standby;
* SQL> alter database open;
* Check lại.

## Data guard broker

Mục đích: quản lý tập trung nhiều standby



### Kiến trúc



### Yêu cầu triển khai

* Phải dùng phiên bản enterprise và phiên bản từ 12 trở lên.
* Thiết lập tham số: dg\_broker\_start = true.
* Database bật bằng spfile, và ở chế độ mount hoặc open.
* SQL> show parameter log\_archive\_dest\_ 🡪 các value phải null

### Quy trình

* SQL> alter system set dg\_broker\_start = true; (toàn bộ **primary** và **standby**)
* SQL> alter system set log\_archive\_dest\_<xx> = ‘’; (xóa giá trị của log archive dest cả **primary** + **standby**)
* OS$: dgmgrl / (**primary**)
* DGMGRL> CREATE CONFIGURATION <BROKER\_NAME> AS PRIMARY DATABASE IS ‘<PRIMARY\_DB\_NAME>’ CONNECT IDENTIFIER IS <PRIMARY\_CONNECT\_INDENTIFIER> (TNSNAMES)
* DGMGRL> SHOW CONFIGURATION (check config)
* DGMGRL> add database ‘<standby\_unique\_name>’ as connect identifier is ‘<standby\_connect\_identifier>’; (tnsname)
* DGMGRL> enable configuration;

### Giám sát

* DGMGRL> show configuration; 🡪 config status là success thì thành công, nếu warning hoặc error thì đang có lỗi 🡪 check từng con standby: DGMGRL > show db standby;
* DGMGRL> show database verbose <db\_name> (tham số nâng cao)
* DGMGRL> edit database <db\_name> set ‘<parameter\_name>’ = <value>;

Trường hợp primary – standby đang chưa đồng bộ

* DGMGRL> show database standby; 🡪 apply – off
* DGMGRL> Edit database standby set state = ‘APPLY – ON’;

## Cấu hình data protection modes

* DGMGRL> show configuration; 🡪 protection mode: maxperformance (default)
* DGMGRL> show database verbose standby; (para: LogXptMode)
* DGMGRL> edit database standby set property LogXptMode = ‘SYNC’; (chuyển chế độ)
* DGMGRL> edit configuration set protection mode as <protection mode>;