DATABASE WITH DATAGUARD ROLLING UPGRADE WITHOUT DOWNTIME

Difference between Physical Standby and Logical Standby

Physical standby:

a. This standby database will be the exact replica of the primary database.
b. Block by block, the data will be copied from the primary to the standby, which means the standby database has to be ready to act as the primary database in the case of any disasters.
c. Once the archived log reaches the standby database, it will get applied through the Media Recovery Process (MRP).
d. The physical standby database will be in either mount mode or read-only mode. The archived logs/redo log entries, which have primary database transactions, will be applied to the standby database through the MRP process.
e. In an active Data Guard setup, the database can be opened in read-only mode and at the same time recovery will be running. It is especially useful for executing reporting jobs (select) or to offload the jobs from the primary.

Logical standby:

a. This standby database uses the SQL apply method. Just like a physical standby, it receives archived logs/redo logs, but it mines the received archived logs and converts its contents into SQL statements to be applied to the logical standby database.
b. As it is not a block-by-block copy method, only user transactions will be transmitted between the primary and the standby.
c. The SQL statement conversion gives the advantage of having the standby database in read write mode. Since the standby database is in read-write mode, it adds luxuries to create additional new objects in standby, which may be helpful for reporting purposes.

Difference between Physical Standby and Logical Standby

	PHYSICAL STANDBY	LOGICAL STANDBY
TRANSFER METHOD	Redo Log Apply	SQL Apply
DATABASE VERSION	Both Primary and Standby should be with same version	Primary and Standby cab be different version
ALLOWED DATATYPE	All Datatypes	Large Object is not allowed
STANDBY MODE	Mount or Read-Only	All modes including Read-Write
DBID	Same DBID for Primary and Standby	DBID is different
TRANSACTIONS	All sys and user transactions	Only user transactions
DATA UNIQUENESS	Not Required	Data uniqueness required at Table Level

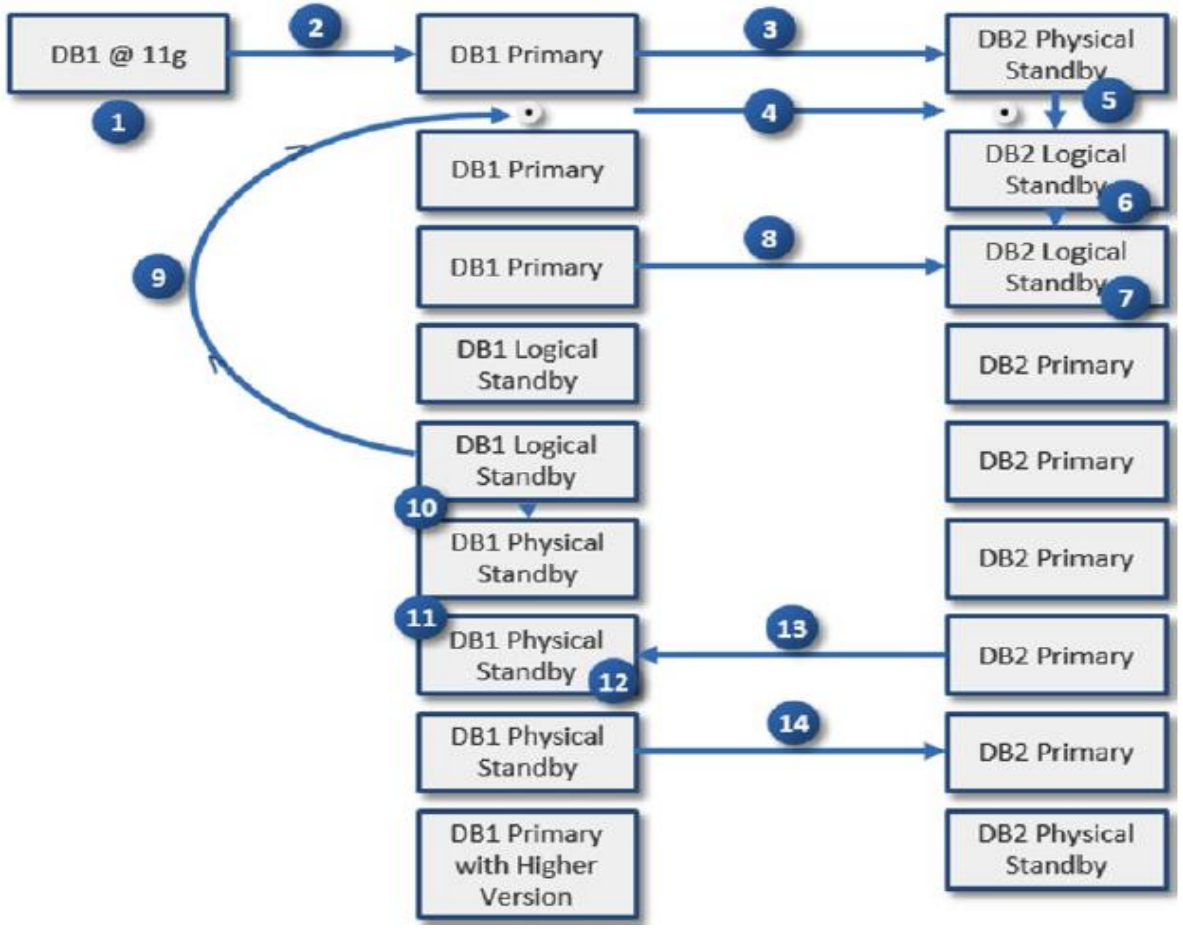
Database Rolling Upgrade using Transient Logical Database High Level Steps

1. Disable Broker.
2. Create a guaranteed restore point on the primary and the standby.
3. Convert the standby to a transient logical standby.
4. Create a second guaranteed restore point.
5. Upgrade the transient logical standby.
6. Switch to the primary.
7. Flash back the original primary.
8. Mount the primary from the new home and convert it to a physical standby.
9. Start the switchover on the primary.
10. Complete the switchover on the standby.
11. Restart the old primary as the standby.
12. Drop the restore points.

Database Rolling Upgrade Diagram

Primary Database Uniue Name: DB01

Standby Database Uniue Name: DB02



Database Rolling Upgrade Step by Step

1) Ensure the database is in archivelog mode and Flashback is enabled.

2) First make the DB1 database the primary database by adding the necessary parameters such as log_archive_config and log_archive_dest_2. In other words, you are creating a Data Guard setup, and in that, DB1 becomes primary database.

3) Create the physical standby database. The standby database can be created using the RMAN duplicate method or by using the primary database backup or basic SCP. Say that the standby database's unique name is DB2. So, DB1 and DB2 are part of Data Guard with the same database version, and both are in sync through the archive log transmission.

4) Create a guaranteed restore point at the primary and physical standby database. This guaranteed restore point will be used later to flashback.

5) Convert the physical standby database to a logical standby database and start the SQL Apply process. Now both the primary and standby databases are in sync.

6) Upgrade the logical standby DB2 to a higher version. During this time, archived logs in the primary database will be queued because the standby is opened in upgrade/restricted mode.

7) Start the SQL Apply process in standby, which will apply the queued archived logs, and both databases will become synced.

8) Switch over the roles. Now DB1 will become the logical standby, and DB2 will become the primary with the higher version. During switchover, database downtime occurs. At this time, both the databases will be in sync.

9) Flashback the DB1 database to a guaranteed restore point. This is required since the DB1 database has to be upgraded. When the database goes back to the old state, it will ask the primary to provide archived logs to reach the current state. Right now the archived logs should be given by the current primary DB2, and the archived logs were generated when the upgrade was in progress. This means the archived logs have the upgrade transactions. But the current standby is a logical standby. Applying archived logs only apply NONSYS user transactions and hence convert it to physical standby.

10) Convert the DB1 database to the physical standby.

11) Shut down and start the physical standby in a higher version. In the physical standby method, both the primary and standby should be in the same version.

12) Start the MRP process in DB1, which will pull archive logs from the primary.

13) Archive logs have upgrade transactions, which will upgrade DB1 to a higher version.

14) Switch over the roles again. DB1 will become the primary database.