Oracle Database Appliance: Implementing MAA Disaster Recovery Solutions Using Oracle Data Guard

Protect production systems while leveraging standby computing power

*Oracle* Maximum Availability Architecture

Aug 31, 2020
Copyright © 2020, Oracle and/or its affiliates
Confidential - Public
DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.
INTRODUCTION

Oracle Database Appliances are pre-built, pre-tuned, and ready-to-use non-clustered and clustered database systems that include servers, storage, networking, and software in an optimized configuration that makes them easy to deploy, operate, and manage. Oracle Database Appliance (ODA) is a complete and ideal database platform for small, medium, and large-sized database implementations and incorporates robust, time-tested Oracle technologies, including the world-leading Oracle Database, the best-selling Oracle Real Application Clusters (RAC) database option, Oracle Clusterware, and Oracle Automatic Storage Management (ASM). By integrating hardware and software, Oracle Database Appliance eliminates the complexities inherent in non-integrated, manually assembled database solutions, reducing deployment time from weeks or months to just a few hours, while preventing configuration and setup errors that often result in sub-optimal, hard-to-manage database environments.

Oracle Cloud Infrastructure
Object Storage (Backups)

Oracle Data Guard Replication

Oracle Database Appliance
Primary System

Oracle Cloud Infrastructure
Object Storage (Backups)

Oracle Database Appliance
Standby System

Oracle Maximum Availability Architecture (MAA) Using Oracle Database Appliance and Oracle Data Guard
DATA PROTECTION USING ORACLE ACTIVE DATA GUARD

While the Oracle Database Appliance is a highly available system in itself, a standby database environment can provide data protection and reduces planned and unplanned downtime in case the primary database environment becomes unavailable or corrupted. Therefore, a standby database has always been an integral component of MAA to provide additional high availability and data protection for any mission-critical production system. With Oracle Maximum Availability Architecture (MAA) Gold Tier best practices, the standby database can be synchronized with the primary database, thereby minimizing database downtime for planned maintenance activities such as database upgrades and unplanned outages such as data corruptions, database failures, cluster failures, power outage or natural disaster.

The most important two metrics that need to be considered to develop and implement the most appropriate recovery plan are Recovery Point Objective (RPO) and Recovery Time Objective (RTO). Oracle Data Guard is the most comprehensive solution available to eliminate single points of failure for mission-critical Oracle Databases. With MAA Gold Tier it prevents data loss (zero RPO) and downtime (zero RTO) in the simplest and most economical manner by maintaining a synchronized physical replica of a production database at a remote location. If the production database is unavailable for any reason, client connections can quickly, and in some configurations transparently, failover to the synchronized replica to restore service.

Oracle Active Data Guard enables administrators to improve performance by offloading processing from the primary database to a physical standby database that is open read-only while it applies updates received from the primary database. Offload capabilities of Oracle Active Data Guard include read-only reporting with the occasional write or update (via DML Re-direct in Oracle Database 19c) and ad-hoc queries (including DML to global temporary tables and unique global or session sequences), data extracts, fast incremental backups, redo transport compression, efficient servicing of multiple remote destinations, and the ability to extend zero data loss protection to a remote standby database without impacting primary database performance.

Oracle Active Data Guard also increases high availability by performing automatic block repair and enabling High Availability Upgrades (utilizing database rolling upgrade automation to bypass the need for downtime while still maintaining a highly available environment). In addition, it includes application continuity (AC) which extends data protection to in-flight transactions that may not have been committed.

Oracle recommends using a separate, dedicated Oracle Database Appliance system to host the Data Guard standby system for a mission-critical production system running on the primary Oracle Database Appliance system. The MAA best practice is to have a local (synchronous replication) standby DB in a nearby data center that has some level of isolation and a remote standby which is routinely maintained via asynchronous replication. This provides protection from disasters which may impact an entire region such as a large scale power outage while still maintaining a RPO of zero in the majority of unplanned outages cases.

BENEFITS OF USING ORACLE DATA GUARD AND ORACLE ACTIVE DATA GUARD

Oracle Data Guard provides numerous benefits and enables greater efficiency and efficacy for the deployed architecture. Even though DG itself does provide significant protection, MAA Gold Tier requires Active Data Guard as without Automatic Block Repair, Application Continuity and DBMS_ROLLING the RTO/RPO included in Maximum Availability Architecture (MAA) - On-Premises HA Reference Architectures can’t be reached. With the use of Oracle Active Data Guard, the standby database environment does not need to be idle, dark capacity. Instead, the standby database can actively serve many useful purposes. These additional uses greatly increase the overall return on effort and investment.

Migration to Oracle Database Appliance - If you plan to migrate existing databases to Oracle Database Appliance, then Oracle Data Guard enables an easy approach for migration of your databases to Oracle Database Appliance. You can simply set up a Physical Standby database on your Oracle Database Appliance and switch over operations from the legacy environment to the new Oracle Database Appliance environment. This includes migration across certain platforms as well. For example, to migrate your databases currently running on the Windows platform to Oracle Database Appliance, a Linux platform, you may simply set up Oracle Data Guard between the two environments and perform a switch over. This approach to platform migration provides the flexibility to switchback, if for any reason you choose to do so after testing. Refer to My Oracle Support (MOS) note 413484.1 Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration, for more information about platform migration using Oracle Data Guard.

Note: Oracle Data Guard also allows you to migrate across database versions using a transient logical standby database.
Disaster Recovery - Oracle Data Guard physical standby database provides an ideal solution for disaster protection. The most common example of a disaster that occurs is a regional power outage, but disaster scenarios vary from burst water or steam pipes, fire, hurricanes, vandalism, to earthquakes, floods, and acts of terrorism. Oracle Data Guard Physical Standby Database maintains a block-for-block copy of the production database. In the event the primary environment becomes unavailable due to any reason, the standby environment can be quickly activated to maintain continued database availability for your applications.

High Availability – Standby database and RAC can also be useful in maintaining availability during planned and unplanned outages and downtimes. Such events may include configuration changes, hardware replacements, and so forth as well as data corruption, failures resulting from human errors, and other unexpected system component or complete system failures.

Standby-First Patching – With Active Data Guard, the standby database can provide additional protection by first applying any hardware, operating system, Grid Infrastructure, and qualified database software updates. Validation can occur for hours, days, or even weeks, providing additional assurance before applying the same changes in RAC rolling manner on the primary database or issuing a Data Guard role transition. This additional protection can prevent an outage due to bad patch or HA or performance regression due to the patch. The only downtime for the databases is the short period of time required to change roles between primary and standby. Please refer to My Oracle Support (MOS) note 1265700.1, Oracle Patch Assurance - Data Guard Standby-First Patch Apply, for more information.

Database Rolling Upgrade – With Active Data Guard and transient logical standby, the standby database can be used to minimize downtime by applying a non-rolling software change such as a major database upgrade on the standby and then subsequently switching over. Downtime is minimized to a couple of seconds due to the Data Guard switchover. For more details, refer to Database Rolling Upgrade using Data Guard (PDF) and MAA Automated Database Upgrades using Oracle Active Data Guard and DBMS_ROLLING for 12.1 databases and higher.

Auto Block Repair – One of the other benefits of the physical standby database is its ability to automatically repair physical block corruptions. In a primary and standby configuration, a corrupt block can be automatically repaired, and this operation can be completely seamless to the application and database administrator. The Block Repair feature is part of the Oracle Active Data Guard option.

Application Continuity (AC) – This feature is available with the Oracle Real Application Clusters (RAC), Oracle RAC One Node and Oracle Active Data Guard options that masks outages from end users and applications by recovering the in-flight database sessions following recoverable outages. It masks outages from end users and applications by recovering the in-flight work for impacted database sessions following outages. Application Continuity performs this recovery beneath the application so that the outage appears to the application as a slightly delayed execution. Application Continuity improves the user experience for both unplanned outages and planned maintenance. Ultimately it enhances the fault tolerance of systems and applications that use an Oracle database.

Offloading Workload and Activities – Despite its name, the standby environment does not have to be idle. It can be actively used to maximize the overall return on your investment. With a physical standby database in place, several key activities can be offloaded to the standby environment. These include:

- **Read-Only Workload** – Using Oracle Active Data Guard option, the standby database can be open for read-only query workload while being in the standby mode and accepting redo log updates from the primary database. In many cases, offloading read-only workloads to the standby database can dramatically reduce the production workload, thereby increasing the overall available capacity for the production system.

- **Backups** – Because the Oracle Data Guard physical standby database is a physical copy of the primary database, database backups can be completely offloaded to the standby environment and these backups can be transparently used to restore and recover the primary database in the event of a failure or database loss. Note that if Oracle Active Data Guard option is licensed, then fast incremental backups can be run at the standby database, further adding to the appeal of offloading backups to the standby database.

- **Snapshot Standby** – The Snapshot Standby database is an updatable standby database that provides full data protection for the primary database. It continues to receive redo data from the primary, but the apply process is halted while the standby database is open for read/write operations for testing purposes. When testing is complete, a single command reverts the standby database to its original state, discarding the changes made while it was open in read-write mode and applying the accumulated redo logs to synchronize with the current state of primary database.
CONFIGURATION BEST PRACTICES

This section describes some of the important best practices for setting up Oracle Data Guard on Oracle Database Appliance. For a complete list of general Oracle Data Guard best practices, which also apply to the Oracle Database Appliance environment, please refer to Oracle Maximum Availability Architecture and Oracle Data Guard best practices available at https://www.oracle.com/database/technologies/high-availability/oracle-database-maa-best-practices.html

Always be on the latest and greatest ODA version – some functionality is only available in the latest ODA version, like syncing up the database related metadata. Backups and some other features might not work via ODA tooling without up-to-date metadata for standby databases.

With ODA 19.8 Release, Oracle Data Guard is integrated with ODA. You can use odacli commands to quickly setup and manage Oracle Data Guard with another ODA.

Match the primary and standby database configuration – In order to maintain consistent service levels and to use the primary and standby databases transparently, it is important to match the resources, setup, and configuration of the primary and standby systems as much as possible. Significant differences between the primary and standby database configuration can result in sub-optimal performance and unpredictable behavior when role transitions occur. Specifically, the following recommendations should be considered:

- **Run Primary and Standby Database on Separate Oracle Database Appliances** – It is recommended that the primary and the standby databases run on separate, dedicated Oracle Database Appliance units preferably located in a geographically distant location.

- **Run Primary and Standby Database in Same Configuration** – Three different database configurations are supported on Oracle Database Appliance; Oracle RAC database, Oracle RAC One, and Single-Instance Enterprise Edition database. The standby database should also be of the same configuration type as the primary database. Thus, if the primary database is configured as an Oracle RAC database, then the standby database should also be configured as an Oracle RAC database.

- **Keep symmetry between the primary and standby sites** – The instances on the primary and standby databases should be configured similar to each other in terms of database parameter settings including memory, CPU, networking, and storage. This helps avoid any unpredictability when the database switch roles. In addition, any operating system configuration customizations should be mirrored in the two environments.

- **Configure Flashback Database on both Primary and Standby Databases** – The Flashback Database feature enables rapid role transitions and reduces the effort required to re-establish database roles after a transition. As a best practice, Flashback Database should be configured on both primary and the standby databases. If FLASHBACK is only deemed necessary by you for re-instantiation, then it would be a good practice to reduce the retention time from the default 24 hours to 2 hours. It should be noted that as of the Oracle Database 19c release, all restoration points are automatically propagated to standby databases.

- **Use Dedicated Network for Standby Traffic** – Oracle Database Appliance comes pre-built with multiple redundant network interfaces. If required, a separate network path can be configured for the standby traffic to minimize any performance impact on the user and application-related workload. Note that since Oracle Data Guard needs to transport only the changes made to the primary database from the primary database to the standby database, it does not impose any unnecessary requirements on the network than is needed. Therefore, many deployments of Oracle Data Guard may not require a separate network path for redo log transport between primary and standby. However, some high volume applications or your organization’s best practices and standards may require a separate network path for redo log transport. Oracle Database Appliance does provide additional network interfaces on each server node that can be used for this purpose. Please refer to MOS note 1422563.1 for additional details on configuring a dedicated network for disaster recovery purposes on Oracle Database Appliance.

- **Utilizing Oracle Active Data Guard** – Oracle Active Data Guard allows for read-only standby of near current data since redo apply remain continuously active between primary and standby environments. This can help distribute or offload the read-only workload from the primary environment to the standby database, increasing the return on investment in the standby database. Note that with Oracle Active Data Guard, fast incremental backups can be run on the standby database. The fast incremental backups could potentially reduce backup windows from hours to minutes. Rolling upgrades can also be done using the standby database, reducing downtime to near-zero. Additionally, Active Data Guard with real time apply enables bi-directional auto-block corruption repair providing another layer of data protection for mission-critical applications.

- **Use Oracle Data Guard Broker** – Oracle Data Guard Broker's interfaces improve usability and centralize management and monitoring of an Oracle Data Guard configuration. It minimizes overall management, and it has inherent checks and balances for Data Guard configuration. Refer to Benefits of Oracle Data Guard Broker for additional details.

- **Setup Clusterware Role Based Services** – Refer to Client Failover Best Practices for Highly Available Oracle Databases
Review Oracle Maximum Availability Architecture (MAA) Best practices for Oracle Database - Depending on your deployment and usage of the Data Guard environment and other requirements, you may find many MAA Best Practices such as the following useful.

- Maximum Availability Architecture (MAA) - On-Premises HA Reference Architectures
- Client Failover Best Practices for Data Guard 12c
- Best Practices for Configuring Redo Transport for Active Data Guard 12c
- Best Practices for Asynchronous Redo Transport - Data Guard and Active Data Guard
- Best Practices for Synchronous Redo Transport - Data Guard and Active Data Guard
- Best Practices for Automatic Resolution of Outages to Resume Data Guard Zero Data Loss
- Role Transition Best Practices: Data Guard and Active Data Guard
- Preventing, Detecting, and Repairing Block Corruption - Oracle Database 12c
- Client Failover Best Practices for Highly Available Oracle Databases

ORACLE DATA GUARD SETUP BETWEEN ORACLE DATABASE APPLIANCE SYSTEMS

ORACLE DATA GUARD CONFIGURATION PROCEDURES

Depending on the version of the primary database, different methods can be used for setting up the Data Guard Physical Standby Database environment.

ODA 19.8 bare metal deployments – configure Oracle Data Guard for all database versions with odacli commands regardless database Release Update, Bundle Patch, PSU versions.
Refer to Configuring Oracle Data Guard on Oracle Database Appliance

Prerequisites for Oracle Database Data Guard Configuration with this 19.8 new feature:

» Oracle recommends running the primary and the standby databases on separate Oracle Database Appliance hardware, so ensure that you have at least two separate Oracle Database Appliance machines.
» Oracle recommends running the primary and the standby databases on two homogeneous Oracle Database Appliance systems. For example, if the primary database is a High-Availability system, then the standby database must be on a similar High-Availability system.
» Oracle recommends that the primary and standby systems have the same Oracle Database Appliance configuration. The databases must have a similar configuration for database shape, version, memory, networking, and storage (both must have either Oracle ASM or Oracle ACFS storage) to avoid unpredictability with the database switch roles.
» The primary and standby systems must be the same Oracle Database Appliance release, and must be on Oracle Database Appliance release 19.8 or later.
» If you have customized the operating system, then ensure that environments on both machines are identical.

This technical brief provides guidance for configuring Data Guard on bare metal and virtualized platform ODAs. With two similarly configured bare-metal ODAs (primary and standby), and both running ODA 19.8 or higher, the recommended way to configure Oracle Data Guard is to use the built-in ODA commands as they can manage the entire lifecycle of an Oracle Data Guard configuration in an easy and efficient way including database upgrade and patching. Check the requirements for Integrated Data Guard with ODA 19.8 for any limitation that may apply.

Bare metal ODAs on older versions than 19.8 and all ODA virtualized platform versions

Database Versions 11.2 - The standard RMAN DUPLICATE method is recommended for database versions 11.2. Although, this method also works for higher versions, there are other options available which are described below. Refer to Creating a Physical Standby Database using RMAN Duplicate (RAC or Non-RAC) MOS Note 1617946.1 for details.

Note: An example step-by-step procedure for creating a primary-standby configuration for Oracle 11g databases using Oracle Database Appliance platforms is provided in Appendix C of this technical brief.

Database Versions 12.1, 12.2 and 18.x, 19.x - You can also use the RMAN ‘restore… from service ‘ method if the database version is 12.1.0.2 or higher. Refer to Creating a Physical Standby database using RMAN restore… from service MOS Note 2283978.1 for details on how to instantiate the standby database using the ‘restore… from service’ method. The RMAN ‘restore… from service’ clause enables online restore and recover of primary database files to a standby database over a network. This method also allows for utilizing the SECTION SIZE clause for parallelization of the restore over multiple RMAN channels.

Note: An example step-by-step procedure for creating a primary-standby configuration for Oracle 19c and 12c databases using Oracle Database Appliance platforms is provided in Appendix A and B of this technical brief.
As you follow the above documents for setting up your primary and standby database environments in an Oracle Data Guard configuration, adhere to the following guidelines that are specific to the Oracle Database Appliance platform.

» Due to a Bug, you may not be able to use Oracle Enterprise Manager for instantiating a standby system and using Oracle Database Appliance as the platform. This is due to a bug. You can however, follow the above-mentioned notes or examples provided in the appendix sections of this technical brief for configuring your 11g and 12c, 18c, 19c environments.

» On the old stack (OAK) if using Oracle ACFS storage, pre-create database storage on the standby Oracle Database Appliance system prior to standby database instantiation. Use the “oakcli create dbstorage” command as the root user to create ACFS storage for your standby database before you instantiate the standby database. For example:

```
# oakcli create dbstorage -db stbydb
```

Please refer to the example step-by-step configuration procedures listed in the appendix section of this technical brief.

» On the new stack (DCS), pre-create an instance only database. Use the “odacli create-database -io” command as the root user to create the storage structures for your standby database and register it in DCS before you instantiate the standby database. For example:

Database storage on ASM:
```
# odacli create-database -m -u boston -n chicago -io -dh 2d7d2b0b-8489-4e28-a151-64f6e156a512
```

Database storage on ACFS:
```
# odacli create-database -m -u boston -n chicago -r ACFS -io -dh 2d7d2b0b-8489-4e28-a151-64f6e156a512
```

Please refer to the example step-by-step configuration procedures listed in the appendix section of this technical brief.

» Oracle Data Guard can be configured between the new (DCS) and old (OAK) stacks regardless the database storage option.

» You may use the standby database deployment procedures on Oracle Database Appliance Bare Metal as well as Oracle Database Appliance Virtualized Platform deployments.

**ORACLE DATABASE APPLIANCE BARE METAL AND VIRTUALIZED PLATFORM CONFIGURATIONS**

Oracle Database Appliance can be configured as a Bare Metal (non-virtualized) platform or as a Virtualized Platform. The Oracle Data Guard Physical Standby setup process outlined in this technical brief can be used in both Oracle Database Appliance configurations, i.e., Bare Metal and Virtualized Platform. On Oracle Database Appliance Virtualized Platform, the configuration steps are executed within the ODA_BASE domain. In addition, Virtual LANs can be used on Oracle Database Appliance Virtualized Platform for configuring a logically separate network for disaster recovery purposes.

**ORACLE DATABASE APPLIANCE SMALL, MEDIUM, AND LARGE PLATFORM CONFIGURATIONS**

Oracle Real Application Clusters (RAC) and Oracle Data Guard are fundamental and essential components of Oracle Maximum Availability Architecture (MAA). While you can also setup Oracle Data Guard configuration between Oracle Database Appliance X6-2 S|M|L, X7-2 S|M, X8-2 S|M hardware models (the smaller, single node configurations), such configurations do not adhere to MAA guidelines as Oracle Real Application Clusters (RAC) runs only on Oracle Database Appliance HA hardware models (X4-2, X5-2, X6-2 HA, X7-2 HA, and X8-2 HA).

**CONCLUSION**

Oracle Data Guard enables you to instantly deploy an effective disaster recovery protection strategy right from the initial deployment of your Oracle Database Appliance. You can use the Oracle Data Guard Physical Standby environment for multiple purposes besides a disaster recovery solution. The physical standby configuration and setup process outlined in this technical brief is quick, simple, and it can be completed without any downtime incurred on the primary database. Most of the standby creation steps are automated using tools such as Oracle Appliance Manager, RMAN, and Oracle Data Guard Broker.
APPENDIX A: 19C EXAMPLE SETUP ON ODA WITH DCS STACK.

Example Environment

The following section describes the primary and standby database environment topologies used in the subsequent Data Guard setup example using Oracle Database Appliance.

Note: On ODA 19.8 BM use odacli commands to configure and manage Data Guard. For prerequisites and steps refer to Configuring Oracle Data Guard on Oracle Database Appliance

![Configuration Topology of Oracle RAC on Oracle Database Appliance](image)

**Table 1 - Example Oracle Database Primary and Standby Configuration**

<table>
<thead>
<tr>
<th></th>
<th>PRIMARY ORACLE DATABASE APPLIANCE</th>
<th>STANDBY ORACLE DATABASE APPLIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appliance Name</td>
<td>appliance#1</td>
<td>appliance#2</td>
</tr>
<tr>
<td>Host Names</td>
<td>proddb1</td>
<td>proddb1</td>
</tr>
<tr>
<td></td>
<td>stbydb1</td>
<td>stbydb2</td>
</tr>
<tr>
<td>Cluster Name</td>
<td>PCLUSTER</td>
<td>SCLUSTER</td>
</tr>
<tr>
<td>Database Name</td>
<td>chicago</td>
<td>chicago</td>
</tr>
<tr>
<td>Database Unique Name</td>
<td>chicago</td>
<td>boston</td>
</tr>
<tr>
<td>Instance Name</td>
<td>chicago1</td>
<td>chicago2</td>
</tr>
<tr>
<td></td>
<td>boston1</td>
<td>boston2</td>
</tr>
<tr>
<td>SCAN Name and IPs</td>
<td>proddb-scan (10.1.27.2, 10.1.27.3)</td>
<td>stbydb-scan (10.1.27.4, 10.1.27.5)</td>
</tr>
<tr>
<td>Grid Infrastructure</td>
<td>/u01/app/19.0.0.0/grid</td>
<td>/u01/app/19.0.0.0/grid</td>
</tr>
<tr>
<td>Software Installation</td>
<td>/u01/app/oracle/product/19.0.0.0/db_home1</td>
<td>/u01/app/oracle/product/19.0.0.0/db_home1</td>
</tr>
<tr>
<td>Database storage</td>
<td>ASM</td>
<td>ASM</td>
</tr>
<tr>
<td>ARCHIVELOG mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE LOGGING mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Primary Environment Configuration

According to Oracle best practices, it is highly recommended to configure Oracle Data Guard with Oracle Data Guard Broker.

1. Create Standby Redo Logs

Standby Redo Logs (SRLs) receives redo data from the primary database in real time minimizing transport and apply lag. In advance of the primary standby setup, Oracle recommends that standby redo logs be created on the primary database as well so that it is immediately ready to receive redo data following a Data Guard role transition.

Create Standby Redo Logs (SRL) on the primary database. Each thread of the standby redo log must have at least one more redo log group than the corresponding thread of the online redo log. For example,

```
SQL> alter database add standby logfile thread 1 group 7 size 1G, group 8 size 1G, group 9 size 1G, group 10 size 1G;
SQL> alter database add standby logfile thread 2 group 11 size 1G, group 12 size 1G, group 13 size 1G, group 14 size 1G;
```

To check the number of online redo logs & their sizes, use the following query.

```
SQL> select thread#, group#, bytes/1024/1024/1024 SIZE_IN_GB, status from v$log;
```

Note that the size of the standby redo logs should match the size of the redo logs. On the Oracle Database Appliance platform, the standby redo logs have to be created on the REDO disk group which resides on the solid state disks. On ODA Small/Medium/Large and on X8-2 HA models the controlfile, online logs are stored in RECO diskgroup as there is no REDO diskgroup.

To validate the size of each log file and number of log groups in the standby redo log, use the following query.

```
SQL> select group#, thread#, bytes/1024/1024/1024 SIZE_IN_GB from v$standby_log;
```

2. Enable archivelog mode on primary database

Archiving is the process of saving and protecting REDO information in the form of archive files before the redo logs of an active database are overwritten in a circular manner. Database created on Oracle Database Appliance have archiving turned on by default. However, it is not mandatory to run your databases in archive log mode which is the default setting on ODA.

Verify that the primary database is running in ARCHIVELOG mode.

```
SQL> archive log list
```

If the primary database is not running in ARCHIVELOG mode, then enable ARCHIVELOG mode as follows.

Shutdown both instances on Oracle Database Appliance.

```
$ srvctl stop database –d chicago
```

Startup mount one instance in exclusive mode.

```
SQL> startup mount exclusive;
```

Turn on archiving.

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

Shutdown the instance.

```
SQL> shutdown immediate;
```

Restart the database.

```
$ srvctl start database –d chicago
```
3. Enable FORCE LOGGING mode.

Force logging enables you to capture database operations performed with the NOLOGGING attribute. This ensures integrity of your standby database. Verify if FORCE LOGGING has already been enabled on your primary database.

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

If FORCE LOGGING is not enabled, then enable it using the following commands.

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

4. Configure Flashback Database feature

The Oracle Flashback Database feature provide a fast alternative to performing incomplete database recovery. Although using the Flashback Database feature is optional, it can be very useful for faster re-instatement of the old primary database after a failover. Thus, if you do a failover to the standby and the old primary can be repaired, you do not have to rebuild the old primary database as a standby database but simply flashback and let Oracle Data Guard resynchronize from that point onwards.

Check if the primary database has Flashback Database enabled, and if required enable it.

```sql
SQL> select flashback_on from v$database;
SQL> alter database flashback on;
```

Note that enabling Flashback Database will require additional space consumption in the Fast Recovery Area (RECO Disk Group). The space used by flashback logs can be controlled by setting the parameter `DB_FLASHBACK_RETENTION_TARGET` to a desired value. This value is specified in minutes. For example,

```sql
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='*';
```

5. Enable Standby File Management

When the primary database adds or drops a datafile, the corresponding action should also be automatically taken on the standby database. This operation can be enabled using automated standby file management.

```sql
SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=both sid='*';
```

6. Create the database home on the standby if it did not exist.

   ```
   [root@stbydb1]# odacli create-dbhome -v 19.8.0.0.200714
   ```

   The database home version on the standby must be identical to primary's version.

7. Setup TNS Entries and Listeners

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that in the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

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

   **Primary**

   ```
   chicago =
   (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = prod Db-scan)(PORT = 1521))
   (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = chicago.us.oracle.com))
   )
   ```
8. Setup Redo Transport Service in deferred mode. This step is not needed if DG Broker will be also configured.

   The Oracle Data Guard redo transport mechanism uses Oracle Net connections to send the redo between the databases. Redo transport is enabled by setting the LOG_ARCHIVE_DEST_n parameter. For example, the following setup enables log shipping and uses LGWR based transmission in asynchronous mode.

   SQL> alter system set log_archive_dest_2='SERVICE=boston LGWR ASYNC REGISTER VALID_FOR=(online_logfile,primary_role) REOPEN=60 DB_UNIQUE_NAME=boston' scope=both sid='*';
   SQL> alter system set log_archive_dest_state_2='defer' scope=both sid='*';

   More details about redo log transmission options can be found in Oracle Data Guard Concepts and Administration Guide.

9. Setup Fetch Archive Log Server. This step is not needed if DG Broker will be also configured.

   When the database is in standby role and the primary is unable to send any missing log files, then the standby database can use the FAL_SERVER setting to pull those missing log files. The FAL_SERVER parameter is uses the Oracle Net service name.

   SQL> alter system set FAL_SERVER=boston scope=both sid='*';

10. Create a pfile from the spfile on the primary database.

   [oracle@proddb1]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
   [oracle@proddb1]$ export ORACLE_SID=chicago1
   [oracle@proddb1]$ export PATH=$ORACLE_HOME/bin:$PATH
   [oracle@proddb1]$ sqlplus / as sysdba
   SQL> create pfile='/tmp/chicago.pfile' from spfile;
11. Add/modify the parameters on the Primary/Standby. For example:

<table>
<thead>
<tr>
<th>Primary</th>
<th>Standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.cluster_database=TRUE</td>
<td>*.cluster_database=TRUE</td>
</tr>
<tr>
<td>chicago2.instance_number=2</td>
<td>boston2.instance_number=2</td>
</tr>
<tr>
<td>chicago1.instance_number=1</td>
<td>boston1.instance_number=1</td>
</tr>
<tr>
<td>chicago2.thread=2</td>
<td>boston2.thread=2</td>
</tr>
<tr>
<td>chicago1.thread=1</td>
<td>boston1.thread=1</td>
</tr>
<tr>
<td>chicago2.undo_tablespace='UNDOTBS2'</td>
<td>boston2.undo_tablespace='UNDOTBS2'</td>
</tr>
<tr>
<td>chicago1.undo_tablespace='UNDOTBS1'</td>
<td>boston1.undo_tablespace='UNDOTBS1'</td>
</tr>
<tr>
<td>*.db_block_checking=FULL</td>
<td>*.db_block_checking=FULL</td>
</tr>
<tr>
<td>*.db_block_checksum=FULL</td>
<td>*.db_block_checksum=FULL</td>
</tr>
<tr>
<td>*.db_lost_write_protect=TYPICAL</td>
<td>*.db_lost_write_protect=TYPICAL</td>
</tr>
<tr>
<td>.db_unique_name=chicago</td>
<td>.db_unique_name=boston</td>
</tr>
<tr>
<td>.listener_networks='((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=&lt;primary node0's vip&gt;)(PORT=1521))))','((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=&lt;primary node1's vip&gt;)(PORT=1521))))')','((NAME=net1)(REMOTE_LISTENER=&lt;primary's scan name&gt;:1521))'</td>
<td>.listener_networks='((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=&lt;standby node0's vip&gt;)(PORT=1521))))','((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=&lt;standby node1's vip&gt;)(PORT=1521))))')','((NAME=net1)(REMOTE_LISTENER=&lt;standby's scan name&gt;:1521))'</td>
</tr>
<tr>
<td>.LOG_FILE_NAME_CONVERT='+REDO/BOSTON','+REDO/CHICAGO'</td>
<td>.LOG_FILE_NAME_CONVERT='+REDO/CHICAGO','+REDO/BOSTON'</td>
</tr>
<tr>
<td>.DB_FILE_NAME_CONVERT='+DATA/BOSTON','+DATA/CHICAGO'</td>
<td>.DB_FILE_NAME_CONVERT='+DATA/CHICAGO','+DATA/BOSTON'</td>
</tr>
<tr>
<td>.LOG_ARCHIVE_DEST_1=LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=chicago'</td>
<td>.LOG_ARCHIVE_DEST_1=LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston'</td>
</tr>
<tr>
<td>add the following parameters</td>
<td>add the following parameters</td>
</tr>
<tr>
<td>.audit_file_dest='/u01/app/oracle/admin/chicago/adump'</td>
<td>.audit_file_dest='/u01/app/oracle/admin/boston/adump'</td>
</tr>
<tr>
<td>.fal_server='boston'</td>
<td>.fal_server='boston'</td>
</tr>
<tr>
<td>.remote_login_passwordfile='exclusive'</td>
<td>.remote_login_passwordfile='exclusive'</td>
</tr>
</tbody>
</table>

**Notes:**

Data protection parameters should be set accordingly. Please refer to [Note 1302539.1 - Best Practices for Corruption Detection, Prevention, and Automatic Repair - in a Data Guard Configuration](https://docs.oracle.com/en/database/oracle/database/12.2.0.1/otn/doc/corrupt-recovery.html) On ODA Small/Medium/Large and on X8-2 HA models the controllfile, online logs are stored in RECO diskgroup as there is no REDO diskgroup.

Databases use listener_networks instead of local_listener and remote_listener parameters starting from ODA 19.6 on bare metal platform.
12. On all standby hosts create the audit directory for the boston database.

   [oracle@stbydb1]$ mkdir -p /u01/app/oracle/admin/boston/adump
   [oracle@stbydb2]$ mkdir -p /u01/app/oracle/admin/boston/adump

13. Create storage structures on the standby and register the standby database in DCS.

   [root@rwsodam001 ~]# odacli list dbhomes

   ID          Name                 DB Version                               Home Location
              ----------------------------------------  ----------------------------------------
    755b4b5d-6211-4d94-81e8-cf611868fe39  OraDB19000_home1  19.8.0.0.200714                          /u01/app/oracle/product/19.0.0.0/dbhome_1
                   Configured

   [root@stbydb1]$ odacli create database m-u boston -n chicago -io -dh 755b4b5d-6211-4d94-81e8-cf611868fe39

   Wait till the instance only database is ready before proceeding further. Verify the status thru odacli describe-jobs.

   [root@unaodapoc1 ~]# odacli describe job da26334e-c5ac-411b-b38d-2f6aef6ca090

   Job details
   ------------------------------------------------------------------
   ID: da26334e-c5ac-411b-b38d-2f6aef6ca090
   Description: Database service creation with db name: chicago
   Status: Success
   Created: July 3, 2020 6:02:18 PM CEST
   Message:

   Stop or kill the instance which might be running in nomount mode on both nodes.

14. Password Copy

   Copy the password file from the primary database to the first standby host.

   [oracle@proddb1]$ srvctl config database -d chicago |grep Password
   Password file: /DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847

   [grid@proddb1]$ asmcmd
   ASMCMD> pwcopy /DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847 /tmp/pwdboston
   copying /DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847 -> /tmp/pwdboston

   [grid@proddb1]$ scp /tmp/pwdboston oracle@stbydb1:/u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston

   [grid@stbydb1]$ asmcmd
   ASMCMD> mkdir /DATA/BOSTON
   ASMCMD> mkdir /DATA/BOSTON/PASSWORDFILE
   ASMCMD> pwcopy /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston /DATA/BOSTON/PASSWORDFILE/pwdboston
15. Copy the modified pfile to the first standby host and mount the standby database.

Make a note of the path where the standby control file is created.

    [oracle@proddb1]$ scp /tmp/chicago.pfile oracle@stbydb1.us.oracle.com:/tmp/boston.pfile
    [oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
    [oracle@stbydb1]$ export ORACLE_SID=boston1
    [oracle@stbydb1]$ cp /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston
                   /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston1
    [oracle@stbydb1]$ rman target /
    RMAN> startup nomount pfile='/tmp/boston.pfile';
    RMAN> restore standby controlfile from service chicago;
    Starting restore at 05-JUL-20
    using target database control file instead of recovery catalog
    allocated channel: ORA_DISK_1
    channel ORA_DISK_1: SID=26 device type=DISK
    channel ORA_DISK_1: starting datafile backup set restore
    channel ORA_DISK_1: using network backup set from service Chicago
    channel ORA_DISK_1: restoring control file
    channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
    output file name=+/REDO/BOSTON/CONTROLFILE/current.1492.986894783
    Finished restore at 05-JUL-20

16. Update the Control File parameter

Edit the pfile '/tmp/chicago.pfile' and replace the control_files parameter to show the new path from the previous output. For example:

    control_files= '+REDO/BOSTON/CONTROLFILE/current.1492.986894783'

    [oracle@stbydb1]$ vi /tmp/boston.pfile

17. Start the Standby instance

Start the standby instance in nomount mode using the modified pfile, create the spfile and restart the instance with the spfile.

    [oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
    [oracle@stbydb1]$ export ORACLE_SID=boston1
    [oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
    [oracle@stbydb1]$ sqlplus / as sysdba
    SQL> startup nomount force pfile='/tmp/boston.pfile';
    SQL> create spfile='+DATA/BOSTON/PARAMETERFILE/spfileboston' from pfile='/tmp/boston.pfile';
    SQL> echo "spfile='+DATA/BOSTON/PARAMETERFILE/spfileboston'" >
                   /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/initboston1.ora
    SQL> startup mount force;
18. Enable Parallelism and set SECTION SIZE=64MB
To take advantage of parallelism during the restore, determine the number of CPUs on your server by executing the following:

```
[oracle@stbydb1]$ grep -c ^processor /proc/cpuinfo
```

Make the following RMAN configuration changes at the Standby.
The example uses 8 preconfigured channels for RMAN to use during the recovery process.

```
[oracle@stbydb1]$ rman target /
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 8;
```

19. Restore the Standby Database from the primary database service
Backing up a single large file in parallel, RMAN's multi section backup/restore capability improves backup and recovery rates. Underneath the covers RMAN divides the work among multiple channels and each channel acts upon a file section in a file. If you specify a small section size that would produce more than 256 sections, then RMAN increases the section size to a value that results in exactly 256 sections.

Section size clause depends on various factor such as, network bandwidth, number of channels, sizes of data files and application datafile sizes.

```
[oracle@stbydb1]$ sqlplus system/welcome1@chicago
SQL> select TABLESPACE_NAME, bytes/1024/1024/1024 SIZE_IN_GB from dba_data_files;
```

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
<th>SIZE_IN_GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>.68359375</td>
</tr>
<tr>
<td>SYSAUX</td>
<td>.5859375</td>
</tr>
<tr>
<td>UNDOTBS1</td>
<td>.297851563</td>
</tr>
<tr>
<td>UNDOTBS2</td>
<td>.1953125</td>
</tr>
<tr>
<td>USERS</td>
<td>.004882813</td>
</tr>
</tbody>
</table>

For example, the following command executed on the standby host specifies a backup section size of 64MB.

```
[oracle@stbydb1]$ rman target /
RMAN> restore database from service chicago section size 64M;
RMAN> switch database to copy;
RMAN> recover database from service chicago;
RMAN> backup spfile;
```

20. Enable log shipping on the primary. This step is only needed if Data Guard Broker won't be configured.

```
[oracle@prod db1]$ sqlplus / as sysdba
SQL> alter system set log_archive_dest_state_2='enable' scope=both;
```

21. Enable Flashback Database on the standby and adjust retention as required. This step is only needed if Data Guard Broker won't be configured.

```
SQL> alter database flashback on;
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120;
```
22. Start managed recovery on the standby. This step is only needed if Data Guard Broker won’t be configured.

    [oracle@stbydb1]$ sqlplus / as sysdba
    SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

23. Register the standby database with Clusterware

    [oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
    [oracle@stbydb1]$ export ORACLE_SID=boston1
    [oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH

    Single instance example

    [oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_1 -dbtype SINGLE -instance boston1 -node stbydb1 -dbname chicago -diskgroup 'DATA,REDO,RECO' -role physical_standby -spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -pwfile '+DATA/BOSTON/PASSWORDFILE/pwdboston'

    RAC example

    [oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_1 -dbtype RAC -dbname chicago -diskgroup 'DATA,REDO,RECO' -role physical_standby -spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -pwfile '+DATA/BOSTON/PASSWORDFILE/pwdboston'
    [oracle@stbydb1]$ srvctl add instance -db boston -instance boston1 -node stbydb1
    [oracle@stbydb1]$ srvctl add instance -db boston -instance boston2 -node stbydb2
    [oracle@stbydb1]$ scp $ORACLE_HOME/dbs/initboston1.ora oracle@stbydb2:/u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/initboston2.ora
    [oracle@stbydb1]$ srvctl start instance -db boston -instance boston1 -o mount
    [oracle@stbydb1]$ srvctl start instance -db boston -instance boston2 -o mount

24. Set the parameters and create the Broker configuration.

    Modify the script below to your environment and save as PostCR.sql

    NOTE: Flashback database is required to re-instantiate a failed primary after a failover role transition. Optionally enable flashback on both primary and standby. The standby database can begin using flashback on using the PostCR script as follows.

    [oracle@stbydb1]$ cat PostCR.sql
    connect / as sysdba
    alter system set dg_broker_config_file1='+DATA/BOSTON/dr1.dat' scope=both;
    alter system set dg_broker_config_file2='+DATA/BOSTON/dr2.dat' scope=both;
    alter system set db_flashback_retention_target=120 scope=spfile;
    alter database flashback on;
    alter system set dg_broker_start=true;
    shutdown immediate
    startup mount
    alter system register;
    connect sys/welcome1@chicago as sysdba
    alter system set dg_broker_config_file1='+DATA/CHICAGO/dr1.dat' scope=both;
    alter system set dg_broker_config_file2='+DATA/CHICAGO/dr2.dat' scope=both;
    alter system set dg_broker_start=TRUE;
    host sleep 30
    host dgmgrl sys/welcome1@chicago "CREATE CONFIGURATION dgconfig AS PRIMARY DATABASE IS CHICAGO CONNECT IDENTIFIER IS CHICAGO"
    host sleep 30
    host dgmgrl sys/welcome1@chicago "ADD DATABASE BOSTON AS CONNECT IDENTIFIER IS BOSTON";
    host dgmgrl sys/welcome1@chicago "ENABLE CONFIGURATION"
    exit
Execute the script PostCR.sql on the standby database. Set your environment to standby database

```
[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> @PostCR.sql
```

In case ‘ALTER DATABASE FLASHBACK ON’ failed with ORA-38788 please let the standby sync up and execute the following steps to enable flashback after that:

```
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> alter database flashback on;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

25. Verification using sqlplus/srvctl

```
[oracle@stbydb1]$ srvctl config database -d chicago
[oracle@stbydb1]$ srvctl config database -d boston
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> select FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE, SWITCHOVER_STATUS,
DATAGUARD_BROKER, PROTECTION_MODE from v$database;
SQL> select PROCESS,PID,DELAY_MINS from V$MANAGED_STANDBY;
```

26. Verification from dg broker (using dgmgrl)

```
$ dgmgrl
DGMGRL> connect sys/welcome1@boston
DGMGRL> show configuration verbose
DGMGRL> show database verbose chicago
DGMGRL> show database verbose boston
DGMGRL> validate database chicago
dgmgrl> validate database boston
```

27. Setup Clusterware Role Based Services – Refer to [Client Failover Best Practices for Highly Available Oracle Databases](#)
28. Sync up the registry (18.7 BM new feature)

```
# odacli list-databases

+--------+-------+-------+---------+-------+------+
| ID     | DB Name | DB Type | DB Version | CDB   | Class | Shape |
|--------+-------+-------+---------+-------+------+
| e6450a56-5a7d-4dab-9ca9-25b004b66646 | chicago | Rac   | 19.8.0.0.200714 | false | Oltp  | Odb1  |
|        |        |       |          |       |       | Asm   |
|        |        |       |          |       |      | Configured |

# odacli describe-database -i e6450a56-5a7d-4dab-9ca9-25b004b66646

Database details

ID: e6450a56-5a7d-4dab-9ca9-25b004b66646
Description: chicago
DB Name: chicago
DB Version: 19.8.0.0.200714
DB Type: Rac
DB Edition: EE
DB Role:
DB Target Node Name:
DBID:
Instance Only Database: true
CDB: false
PDB Name:
PDB Admin User Name:
SEHA Enabled: false
Class: Oltp
Shape: Odb1
Storage: Asm
DB Redundancy: MIRROR
CharacterSet: AL32UTF8
National CharacterSet: AL16UTF16
Language: AMERICAN
Territory: AMERICA
Home ID: 755b4b5d-6211-4d94-81e8-cf611868fe39
Console Enabled: false
TDE Enabled: false
Level 0 Backup Day: Sunday
AutoBackup Enabled: true
Created: July 5, 2020 1:06:43 PM CEST
DB Domain Name: us.oracle.com

# odacli update-registry -n db -f
```
# odacli describe-job -i 24c3a6fb-0538-4571-ae38-e6cf03a6e689

Job details

----------------------------------------
ID: 24c3a6fb-0538-4571-ae38-e6cf03a6e689
Description: Discover Components : db
Status: Success
Created: July 7, 2020 9:40:41 PM CEST
Message:

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Start Time</th>
<th>End Time</th>
<th>Status</th>
</tr>
</thead>
</table>

# odacli describe-database -i e6450a56-5a7d-4dab-9ca9-25b004b66646

Database details

ID: e6450a56-5a7d-4dab-9ca9-25b004b66646
Description: chicago
DB Name: chicago
DB Version: 18.7.0.0.190716
DB Type: Rac
DB Role: Standby
DB Target Node Name: Rac
DB Edition: EE
DBID: 19.8.0.0.200714
Instance Only Database: false
CDB: false
PDB Name:
PDB Admin User Name:
SEHA Enabled: false
Class: Oltp
Shape: Odb1
Storage: Asm
DB Redundancy: MIRROR
CharacterSet: AL32UTF8
National CharacterSet: AL16UTF16
Language: AMERICAN
Territory: AMERICA
Home ID: 755b4b5d-6211-4d94-81e8-cf611868fe39
Console Enabled: false
TDE Enabled: false
Level 0 Backup Day:
AutoBackup Enabled: true
Created: July 5, 2020 1:06:43 PM CEST
DB Domain Name: us.oracle.com
29. Switchover tests

   $ dgmgrl
   DGMGRL> connect sys/welcome1@boston
   DGMGRL> switchover to boston
   DGMGRL> connect sys/welcome1@chicago
   DGMGRL> switchover to chicago;

30. Failover tests

   connect to standby before failover:

   $ dgmgrl
   DGMGRL> connect sys/welcome1@boston
   DGMGRL> failover to boston
   DGMGRL> reinstate database chicago

   connect to former primary before failover:

   DGMGRL> connect sys/welcome1@chicago
   DGMGRL> failover to chicago;
   DGMGRL> reinstate database boston
APPENDIX B: 12.1.0.2 EXAMPLE SETUP ON ODA WITH OAK STACK

Example Environment

The following section describes the primary and standby database environment topologies used in the subsequent Data Guard setup example using Oracle Database Appliance.

Figure 1 Configuration Topology of Oracle RAC on Oracle Database Appliance

<table>
<thead>
<tr>
<th>Appliance Name</th>
<th>PRIMARY ORACLE DATABASE APPLIANCE</th>
<th>STANDBY ORACLE DATABASE APPLIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appliance Name</td>
<td>appliance#1</td>
<td>appliance#2</td>
</tr>
<tr>
<td>Host Names</td>
<td>proddb1</td>
<td>proddb1</td>
</tr>
<tr>
<td></td>
<td>stbydb1</td>
<td>stbydb2</td>
</tr>
<tr>
<td>Cluster Name</td>
<td>PCLUSTER</td>
<td>SCLUSTER</td>
</tr>
<tr>
<td>Database Name</td>
<td>chicago</td>
<td>chicago</td>
</tr>
<tr>
<td>Database Unique Name</td>
<td>chicago</td>
<td>boston</td>
</tr>
<tr>
<td>Instance Name</td>
<td>chicago1</td>
<td>chicago2</td>
</tr>
<tr>
<td></td>
<td>stbydb1</td>
<td>stbydb2</td>
</tr>
<tr>
<td>SCAN Name and IPs</td>
<td>proddb-scan (10.1.27.2, 10.1.27.3)</td>
<td>stbydb-scan (10.1.27.4, 10.1.27.5)</td>
</tr>
<tr>
<td>Grid Infrastructure Software Installation</td>
<td>/u01/app/18.0.0.0/grid</td>
<td>/u01/app/18.0.0.0/grid</td>
</tr>
<tr>
<td>Oracle Database Software Installation</td>
<td>/u01/app/oracle/product/12.1.0.2/db_home1</td>
<td>/u01/app/oracle/product/12.1.0.2/db_home1</td>
</tr>
<tr>
<td>Database storage</td>
<td>ACFS</td>
<td>ACFS</td>
</tr>
<tr>
<td>ARCHIVELOG mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE LOGGING mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2 - Example Oracle Database Primary and Standby Configuration
Primary Environment Configuration

According to Oracle best practices, it is highly recommended to configure Oracle Data Guard with Oracle Data Guard Broker.

1. Create Standby Redo Logs

Standby Redo Logs (SRLs) host redo data received from the primary database. In advance of the primary standby setup, Oracle recommends that standby redo logs be created on the primary database as well so that it is immediately ready to receive redo data following a switch-over to the standby role.

Create Standby Redo Logs (SRL) on the primary database. Each thread of the standby redo log must have at least one more redo log group than the corresponding thread of the online redo log. For example,

```sql
SQL> alter database add standby logfile thread 1 group 7 size 1G, group 8 size 1G, group 9 size 1G, group 10 size 1G;
SQL> alter database add standby logfile thread 2 group 11 size 1G, group 12 size 1G, group 13 size 1G, group 14 size 1G;
```

To check the number of online redo logs & their sizes, use the following query.

```sql
SQL> select thread#, group#, bytes/1024/1024/1024 SIZE_IN_GB, status from v$log;
```

Note that the size of the standby redo logs should match the size of the redo logs. On the Oracle Database Appliance platform, the standby redo logs have to be created on the REDO disk group which resides on the solid state disks. On ODA Small/Medium/Large and on X8-2 HA models the controlfile, online logs are stored in RECO diskgroup as there is no REDO diskgroup.

To validate the size of each log file and number of log groups in the standby redo log, use the following query.

```sql
SQL> select group#, thread#, bytes/1024/1024/1024 SIZE_IN_GB from v$standby_log;
```

2. Enable archivelog mode on primary database

Archiving is the process of saving and protecting REDO information in the form of archive files before the redo logs of an active database are overwritten in a circular manner. Database created on Oracle Database Appliance have archiving turned on by default. However, it is not mandatory to run your databases in archive log mode which is the default setting on ODA.

Verify that the primary database is running in ARCHIVELOG mode.

```sql
SQL> archive log list
```

If the primary database is not running in ARCHIVELOG mode, then enable ARCHIVELOG mode as follows.

Shutdown both instances on Oracle Database Appliance.

```
$ srvctl stop database –d chicago
```

Startup mount one instance in exclusive mode.

```sql
SQL> startup mount exclusive;
```

Turn on archiving.

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

Shutdown the instance.

```sql
SQL> shutdown immediate;
```

Restart the database.

```
$ srvctl start database –d chicago
```
3. Enable FORCE LOGGING mode.

Force logging enables you to capture database operations performed with the NOLOGGING attribute. This ensures integrity of your standby database. Verify if FORCE LOGGING is already enabled on your primary database.

   SQL> select force_logging from v$database;

If FORCE LOGGING is not enabled, then enable it using the following commands.

   SQL> alter database force logging;

4. Configure Flashback Database feature

The Oracle Flashback Database feature provides a fast alternative to performing incomplete database recovery. Although using the Flashback Database feature is optional, it can be very useful for faster re-instatement of the old primary database after a failover. Thus, if you do a failover to the standby, and the old primary can be repaired, you do not have to rebuild the old primary database as a standby database but simply flashback and let Oracle Data Guard resynchronize from that point onwards.

Check if the primary database has Flashback Database enabled, and if required enable it.

   SQL> select flashback_on from v$database;
   SQL> alter database flashback on;

Note that enabling Flashback Database will require additional space consumption in the Fast Recovery Area (RECO Disk Group). The space used by flashback logs can be controlled by setting the parameter DB_FLASHBACK_RETENTION_TARGET to a desired value. This value is specified in minutes. For example,

   SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='*';

5. Enable Standby File Management

When the primary database adds or drops a datafile, the corresponding action should also be automatically taken on the standby database. This operation can be enabled using automated standby file management.

   SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=both sid='*';

6. Create the database home on the standby if it did not exist.

The database home version on the standby must be identical to primary’s version.

   [oracle@stbydb1]# oakcli create dbhome -version 12.1.0.2.180417

7. Setup TNS Entries and Listeners

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that in the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

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

   Primary

   chicago =
   (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = proddb-scan)(PORT = 1521))
      (CONNECT_DATA =
         (SERVER = DEDICATED)
         (SERVICE_NAME = chicago)
      )
   )
boston =
(DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = stbydb-scan)(PORT = 1521))
 (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = boston)
 )
)

Standby
chicago =
(DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = proddb-scan)(PORT = 1521))
 (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = chicago)
 )
)

boston =
(DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = stbydb-scan)(PORT = 1521))
 (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = boston)
 )
)

8. Setup Redo Transport Service in deferred mode. This step is not needed if DG Broker will be also configured.

The Oracle Data Guard redo transport mechanism uses Oracle Net connections to send the redo between the databases. Redo transport is enabled by setting the LOG_ARCHIVE_DEST_n parameter. For example, the following setup enables log shipping and uses LGWR based transmission in asynchronous mode.

```
SQL> alter system set log_archive_dest_2='SERVICE=boston LGWR ASYNC REGISTER VALID_FOR=(online_logfile,primary_role) REOPEN=60 DB_UNIQUE_NAME=boston' scope=both sid='*';
```

```
SQL> alter system set log_archive_dest_state_2='defer' scope=both sid='*';
```

More details about redo log transmission options can be found in Oracle Data Guard Concepts and Administration Guide.

9. Setup Fetch Archive Log Server. This step is not needed if DG Broker will be also configured.

When the database is in standby role and the primary is unable to send any missing log files, then the standby database can use the FAL_SERVER setting to pull those missing log files. The FAL_SERVER parameter is uses the Oracle Net service name.

```
SQL> alter system set FAL_SERVER=boston scope=both sid='*';
```

10. Create a pfile from the spfile on the primary database.

```
[oracle@proddb1]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
[oracle@proddb1]$ export ORACLE_SID=chicago1
[oracle@proddb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@proddb1]$ sqlplus / as sysdba
SQL> create pfile='/tmp/chicago.pfile' from spfile;
```
11. Add/modify the parameters on the Primary/Standby. For example:

### PARAMETERS TO BE MODIFIED ON THE STANDBY AS COMPARED TO THE PRIMARY

<table>
<thead>
<tr>
<th>Primary</th>
<th>Standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.cluster_database=TRUE</td>
<td>*.cluster_database=TRUE</td>
</tr>
<tr>
<td>chicago2.instance_number=2</td>
<td>boston2.instance_number=2</td>
</tr>
<tr>
<td>chicago1.instance_number=1</td>
<td>boston1.instance_number=1</td>
</tr>
<tr>
<td>chicago2.thread=2</td>
<td>boston2.thread=2</td>
</tr>
<tr>
<td>chicago1.thread=1</td>
<td>boston1.thread=1</td>
</tr>
<tr>
<td>chicago2.undo_tablespace='UNDOTBS2'</td>
<td>boston2.undo_tablespace='UNDOTBS2'</td>
</tr>
<tr>
<td>chicago1.undo_tablespace='UNDOTBS1'</td>
<td>boston1.undo_tablespace='UNDOTBS1'</td>
</tr>
<tr>
<td>*.db_block_checking=FULL</td>
<td>*.db_block_checking=FULL</td>
</tr>
<tr>
<td>*.db_block_checksum=FULL</td>
<td>*.db_block_checksum=FULL</td>
</tr>
<tr>
<td>*.db_lost_write_protect=TYPICAL</td>
<td>*.db_lost_write_protect=TYPICAL</td>
</tr>
<tr>
<td>*._cluster_flash_cache_slave_file=''</td>
<td>add the following parameters</td>
</tr>
<tr>
<td>%.audit_file_dest='/u01/app/oracle/admin/chicago/adump'</td>
<td>*.audit_file_dest='/u01/app/oracle/admin/boston/adump'</td>
</tr>
<tr>
<td>*.fal_server='boston'</td>
<td>*.fal_server='chicago'</td>
</tr>
<tr>
<td>*.remote_login_passwordfile='exclusive'</td>
<td>*.remote_login_passwordfile='exclusive'</td>
</tr>
<tr>
<td>add only on X5-2, X7-2 HA, X8-2 HA with HDDs</td>
<td>add only on X5-2, X7-2 HA, X8-2 HA with HDDs</td>
</tr>
<tr>
<td>_cluster_flash_cache_slave_file=''</td>
<td>_cluster_flash_cache_slave_file=''</td>
</tr>
<tr>
<td>_db_flash_cache_file='/u02/app/oracle/oradata/flashdata/ACFS/snaps/flashcache/chicago/flash1'</td>
<td>_db_flash_cache_file='/u02/app/oracle/oradata/flashdata/ACFS/snaps/flashcache/boston/flash1'</td>
</tr>
</tbody>
</table>

12. On all standby hosts create the audit directory for the boston database.

    [oracle@stbydb1] mkdir -p /u01/app/oracle/admin/boston/adump
    [oracle@stbydb2] mkdir -p /u01/app/oracle/admin/boston/adump
13. Create filesystem structures on the standby. This step only applies to the old stack (OAK).

```bash
[root@stbydb1]# oacil create dbstorage -db boston -storage acfs
INFO: 2017-08-12 06:16:44: Please check the logfile
/opt/oracle/oak/log/stbydb1/tools/12.1.2.10.0/createdbstorage_boston_69182.log' for more details
...
SUCCESS: All nodes in /opt/oracle/oak/onecmd/tmp/db_nodes are pingable and alive.
INFO: 2017-08-14 04:47:45: Successfully setup the storage structure for the database 'boston'
INFO: 2017-08-14 04:47:45: Set the following directory structure for the Database boston
INFO: 2017-08-14 04:47:45: DATA: /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston
INFO: 2017-08-14 04:47:45: REDO: /u01/app/oracle/oradata/datastore/boston
INFO: 2017-08-14 04:47:45: RECO: /u01/app/oracle/fast_recovery_area/datastore/boston
SUCCESS: 2017-08-14 04:47:45: Successfully setup the Storage for the Database : boston
```

14. Password Copy

Copy the password file from the primary database to the first standby host.

```bash
[oracle@prod-db1]$ $ORACLE_HOME/bin/srvctl config database -d chicago |grep -i Password
Password file: /u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/orapwchicago
[oracle@prod-db1]$ scp /u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/orapwchicago oracle@stby-db1.us.oracle.com:/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/orapwboston
```

15. Copy the modified pfile to the first standby host and mount the standby database.

Make a note of the path where is the standby control file is created.

```bash
[oracle@prod-db1]$ scp /tmp/chicago.pfile oracle@stby-db1.us.oracle.com:/tmp/boston.pfile
[oracle@stby-db1]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
[oracle@stby-db1]$ export ORACLE_SID=boston1
[oracle@stby-db1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stby-db1]$ cp /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/orapwboston
/u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwboston1
[oracle@stby-db1]$ RMAN> startup nomount pfile '~/tmp/boston.pfile';
RMAN> restore standby controlfile from service chicago;
Starting restore at 12-AUG-17
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=162 instance=boston1 device type=DISK
channel ORA_DISK_1: starting datfile backup set restore
channel ORA_DISK_1: using network backup set from service Chicago
channel ORA_DISK_1: restoring control file
channel ORA_DISK_1: restore complete, elapsed time: 00:00:26
output file name=/u01/app/oracle/oradata/datastore/boston/BOSTON/controlfile/o1_mf_drw8zb81_.ctl
Finished restore at 12-AUG-17
```
16. Update the Control File parameter

Edit the pfile '/tmp/chicago.pfile' and replace the control_files parameter to show the new path from the previous output. For example:

```
control_files= '/u01/app/oracle/oradata/datastore/boston/BOSTON/controlfile/o1_mf_drw8zb81_.ctl'
```

```
[oracle@stbydb1]$ vi /tmp/boston.pfile
```

17. Start the Standby instance

Start the standby instance in nomount mode using the modified pfile, create the spfile and restart the instance with the spfile.

```
[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stbydb1]$ mkdir -p /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> startup nomount force pfile='/tmp/boston.pfile';
SQL> create spfile='/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora' from pfile='/tmp/boston.pfile';
SQL> !echo "spfile='/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora'" > /u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/initboston1.ora
SQL> startup mount force;
```

18. Enable Parallelism and set SECTION SIZE=64MB

To take advantage of parallelism during the restore, determine the number of CPUs on your server by executing the following:

```
[oracle@stbydb1]$ grep -c ^processor /proc/cpuinfo
20
```

Make the following RMAN configuration changes at the Standby.

The example uses 8 preconfigured channels for RMAN to use during the recovery process.

```
[oracle@stbydb1]$ rman target /
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 8;
```
19. Restore the Standby Database from the primary database service

To backup a single large file in parallel, RMAN's multi section backup/restore capability improves backup and recovery rates. Underneath the covers RMAN divides the work among multiple channels and each channel acts upon a file section in a file. If you specify a small section size that would produce more than 256 sections, then RMAN increases the section size to a value that results in exactly 256 sections.

Section size clause depends on various factor such as, network bandwidth, number of channels, sizes of data files and application datafile sizes.

```
[oracle@stbydb1]$ sqlplus system/welcome1@chicago
SQL> select TABLESPACE_NAME, bytes/1024/1024/1024 SIZE_IN_GB from dba_data_files;

TABLESPACE_NAME    SIZE_IN_GB
-------------------- -----------
SYSTEM              .68359375
SYSAUX              .5859375
UNDOTBS1            .297851563
UNDOTBS2            .1953125
USERS               .004882813
```

For example, the following command executed on the standby host specifies a backup section size of 64 MB.

```
[oracle@stbydb1]$ rman target /
RMAN> restore database from service chicago section size 64M;
RMAN> switch database to copy;
RMAN> recover database from service chicago;
RMAN> backup spfile;
```

20. Enable log shipping on the primary. This step is only needed if Data Guard Broker won't be configured.

```
[oracle@proddb1]$ sqlplus / as sysdba
SQL> alter system set log_archive_dest_state_2='enable' scope=both;
```

21. Enable Flashback Database on the standby and adjust retention as required. This step is only needed if Data Guard Broker won't be configured.

```
SQL> alter database recover managed standby database cancel;
SQL> alter database flashback on;
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120;
SQL> alter database recover managed standby database disconnect;
```

22. Start managed recovery on the standby. This step is only needed if Data Guard Broker won't be configured.

```
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
```

23. Register the standby database with Clusterware

```
[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
```

Single instance example

```
[oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/12.1.0.2/dbhome_1 -dbtype SINGLE -instance boston1 -node stbydb1 -dbname chicago -acspath '/u01/app/oracle/oradata/datastore,u02/app/oracle/oradata/datastore,u01/app/oracle/fast_recovery_area/datastore' -role physical_standby -spfile '/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora' -pwfile '/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/orapwboston'
```
RAC example

[oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/12.1.0.2/dbhome_1 -dbtype RAC -
dbname chicago -acfs

'/u01/app/oracle/oradata/datastore,/u02/app/oracle/oradata/datastore,/u01/app/oracle/fast_recovery_area/datastore' -role
physical_standy -spfile '/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora' -pfile
'/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/orapwboston'

[oracle@stbydb1]$ srvctl add instance -database boston -instance boston1 -node stbydb1
[oracle@stbydb1]$ srvctl add instance -database boston -instance boston2 -node stbydb2
[oracle@stbydb1]$ scp ORACLE_HOME/dbs/initboston1.ora oracle@stbydb2:/u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/initboston2.ora
[oracle@stbydb1]$ srvctl start instance -database boston -instance boston1 –o mount
[oracle@stbydb1]$ srvctl start instance -database boston -instance boston2 –o mount

24. Set parameters and create the Broker configuration.

Modify the script below to your environment and save as PostCR.sql

NOTE: Flashback database is required to re-instantiate a failed primary after a failover role transition. Optionally enable
flashback on both primary and standby. The standby database can begin using flashback on using the PostCR script as follows.

[oracle@stbydb1]$ cat PostCR.sql

connect / as sysdba
alter system set dg_broker_config_file1='/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/dr1.dat'
scope=both;
alter system set dg_broker_config_file2='/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/dr2.dat'
scope=both;
alter system set db_flashback_retention_target=120 scope=spfile;
alter database flashback on;
alter system set dg_broker_start=true scope=spfile;
shutdown immediate
startup mount
alter system register;
connect sys/welcome1@chicago as sysdba
alter system set
dg_broker_config_file1='/u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/dr1.dat'
scope=both;
alter system set
dg_broker_config_file2='/u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/dr2.dat'
scope=both;
alter system set
dg_broker_start=TRUE;
host sleep 30
host dgmgrl sys/welcome1@chicago "CREATE CONFIGURATION dgconfig AS PRIMARY
DATABASE IS CHICAGO CONNECT IDENTIFIER IS CHICAGO";
host sleep 30
host dgmgrl sys/welcome1@chicago "ADD DATABASE BOSTON AS CONNECT IDENTIFIER IS BOSTON"
exit

Execute the script PostCR.sql on the standby database. Set your environment to standby database

[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> @PostCR.sql
In case 'ALTER DATABASE FLASHBACK ON' failed with ORA-38788 please let the standby sync up and execute the following steps to enable flashback after that:

```
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> alter database flashback on;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

25. Verification using sqlplus/srvctl

```
[oracle@stbydb1]$ srvctl config database -d chicago
[oracle@stbydb1]$ srvctl config database -d boston
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> select FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE, SWITCHOVER_STATUS, DATAGUARD_BROKER, PROTECTION_MODE from v$database;
SQL> select PROCESS,PID,DELAY_MINS from V$MANAGED_STANDBY;
```

26. Verification from dg broker (using dgmgrl)

```
$ dgmgrl
DGMGRL> connect sys/welcome1@boston
DGMGRL> show configuration verbose
DGMGRL> show database verbose chicago
DGMGRL> show database verbose boston
DGMGRL> validate database chicago
DGMGRL> validate database boston
```

Validate command might report "Warning: standby redo logs not configured for thread 0" in "Current Log File Groups Configuration" section.

Please refer to MOS Note 20582405.8 - Bug 20582405 - dgmgrl "validate database" shows warning "standby redo logs not configured for thread 0"

27. Setup Clusterware Role Based Services – Refer to Client Failover Best Practices for Highly Available Oracle Databases

28. Swicthover tests

```
$ dgmgrl
DGMGRL> connect sys/welcome1@boston
DGMGRL> switchover to boston
DGMGRL> connect sys/welcome1@chicago
DGMGRL> switchover to chicago;
```

29. Failover tests

connect to standby before failover:

```
$ dgmgrl
DGMGRL> connect sys/welcome1@boston
DGMGRL> failover to boston
DGMGRL> reinstate database chicago
```

connect to former primary before failover:

```
DGMGRL> connect sys/welcome1@chicago
DGMGRL> failover to chicago;
DGMGRL> reinstate database boston
```
APPENDIX C: 11GR2 EXAMPLE SETUP ON ODA

Example Environment

The following section describes the primary and standby database environment topologies used in the subsequent Data Guard setup example using Oracle Database Appliance.

Note: On ODA 19.8 BM use odacli commands to configure and manage Data Guard. For prerequisites and steps refer to Configuring Oracle Data Guard on Oracle Database Appliance

![Configuration Topology of Oracle RAC on Oracle Database Appliance](image)

<table>
<thead>
<tr>
<th></th>
<th>PRIMARY ORACLE DATABASE APPLIANCE</th>
<th>STANDBY ORACLE DATABASE APPLIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appliance Name</td>
<td>appliance#1</td>
<td>appliance#2</td>
</tr>
<tr>
<td>Host Names</td>
<td>proddb1</td>
<td>proddb2</td>
</tr>
<tr>
<td></td>
<td>stbydb1</td>
<td>stbydb2</td>
</tr>
<tr>
<td>Cluster Name</td>
<td>PCLUSTER</td>
<td>SCLUSTER</td>
</tr>
<tr>
<td>Database Name</td>
<td>chicago</td>
<td>chicago</td>
</tr>
<tr>
<td>Database Unique Name</td>
<td>chicago</td>
<td>boston</td>
</tr>
<tr>
<td>Instance Name</td>
<td>chicago1</td>
<td>chicago2</td>
</tr>
<tr>
<td></td>
<td>boston1</td>
<td>boston2</td>
</tr>
<tr>
<td>SCAN Name and IPs</td>
<td>proddb-scan (10.1.27.2, 10.1.27.3)</td>
<td>stbydb-scan (10.1.27.4, 10.1.27.5)</td>
</tr>
<tr>
<td>Grid Infrastructure</td>
<td>/u01/app/19.0.0.0/grid</td>
<td>/u01/app/19.0.0.0/grid</td>
</tr>
<tr>
<td>Software Installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Database</td>
<td>/u01/app/oracle/product/11.2.0.4/db_home1</td>
<td>/u01/app/oracle/product/11.2.0.4/db_home1</td>
</tr>
<tr>
<td>Software Installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCHIVELOG mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE LOGGING mode</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3 - Example Oracle Database naming conventions
Primary Environment Configuration

According to Oracle best practices, it is highly recommended to configure Oracle Data Guard with Oracle Data Guard Broker.

1. Create Standby Redo Logs

Standby redo logs host redo data received from the primary database. In advance of the primary standby setup, Oracle recommends that standby redo logs be created on the primary database as well so that it is immediately ready to receive redo data following a switch-over to the standby role.

Create Standby Redo Logs (SRL) on the primary database. Each thread of the standby redo log must have at least one more redo log group than the corresponding thread of the online redo log. For example,

```
SQL> alter database add standby logfile thread 1 group 7 size 1G, group 8 size 1G, group 9 size 1G, group 10 size 1G;
SQL> alter database add standby logfile thread 2 group 11 size 1G, group 12 size 1G, group 13 size 1G, group 14 size 1G;
```

To check the number of online redo logs & their sizes, use the following query.

```
SQL> select thread#, group#, bytes/1024/1024/1024 SIZE_IN_GB, status from v$log;
```

Note that the size of the standby redo logs should match the size of the redo logs. The standby redo logs have to be created on the REDO disk group which resides on the solid state disks. On ODA Small/Medium/Large and on X8-2 HA models the controlfile, online logs are stored in RECO diskgroup as there is no REDO diskgroup.

To validate the size of each log file and number of log groups in the standby redo log, use the following query.

```
SQL> select group#, thread#, bytes/1024/1024/1024 SIZE_IN_GB from v$standby_log;
```

2. Enable archivelog mode on primary database

Archiving is the process of saving and protecting REDO information in the form of archive files before the redo logs of an active database are overwritten in a circular manner. Database created on Oracle Database Appliance have archiving turned on by default. However, it is not mandatory to run your databases in archive log mode.

Verify that the primary database is running in ARCHIVELOG mode.

```
SQL> archive log list
```

If the primary database is not running in ARCHIVELOG mode, then enable ARCHIVELOG mode as follows.

Shutdown both instances on Oracle Database Appliance.

```
$ srvctl stop database –d chicago
```

Startup mount one instance in exclusive mode.

```
SQL> startup mount exclusive;
```

Turn on archiving.

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

Shutdown the instance.

```
SQL> shutdown immediate;
```

Restart the database.

```
$ srvctl start database –d chicago
```
3. Enable FORCE LOGGING mode.

Force logging enables you to capture database operations performed with the NOLOGGING attribute. This ensures integrity of your standby database. Verify if FORCE LOGGING is already enabled on your primary database.

   SQL> select force_logging from v$database;

If FORCE LOGGING is not enabled, then enable it using the following commands.

   SQL> alter database force logging;

4. Configure Flashback Database feature

The Oracle Flashback Database feature provide a fast alternative to performing incomplete database recovery. Although using the Flashback Database feature is optional, it can be very useful for faster re-instatement of the old primary database after a failover. Thus, if you do a failover to the standby, and the old primary can be repaired, you do not have to rebuild the old primary database as a standby database but simply flashback and let Oracle Data Guard resynchronize from that point onwards.

Check if the primary database has Flashback Database enabled and if required enable it.

   SQL> select flashback_on from v$database;
   SQL> alter database flashback on;

Note that enabling Flashback Database will require additional space consumption in the Fast Recovery Area (RECO Disk Group).

The space used by flashback logs can be controlled by setting the parameter DB_FLASHBACK_RETENTION_TARGET to a desired value.

This value is specified in minutes. For example,

   SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='"';

5. Enable Standby File Management

When the primary database adds or drops a datafile, the corresponding action should also be automatically taken on the standby database.

This operation can be enabled using automated standby file management.

   SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=both sid='"';

6. Enable Remote Privileged Login

Ensure that each instance of the primary database is configured with remote login password file. Note that the Oracle Database Appliance deploys the databases with this setting. The initialization parameter REMOTE_LOGIN_PASSWORDFILE must be set to exclusive. If this parameter was reset in your environment and needs to be modified as below, it requires a database restart for it to take effect.

   [oracle@proddb1]$ sqlplus / as sysdba
   SQL> show parameter remote_login_passwordfile
   SQL> alter system set remote_login_passwordfile='exclusive' scope=spfile sid='"';

7. Setup TNS Entries

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that in the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

   chicago =
   (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = proddb-scan)(PORT = 1521))
      (CONNECT_DATA =
         (SERVER = DEDICATED) (SERVICE_NAME = chicago)
8. Setup Redo Transport Service in deferred mode

The Oracle Data Guard redo transport mechanism uses Oracle Net connections to send the redo between the databases. Redo transport is enabled by setting the LOG_ARCHIVE_DEST_n parameter. For example, the following setup enables log shipping and uses LGWR based transmission in asynchronous mode.

```sql
SQL> alter system set log_archive_dest_2='SERVICE=boston LGWR ASYNC REGISTER VALID_FOR=(online_logfile,primary_role) REOPEN=60 DB_UNIQUE_NAME=boston' scope=both sid="";
SQL> alter system set log_archive_dest_state_2='defer' scope=both sid="";
```

More details about redo log transmission options can be found in Oracle Data Guard Concepts and Administration Guide.

9. Setup Fetch Archive Log Server

When the database is in standby role and the primary is unable to send any missing log files, then the standby database can use the FAL_SERVER setting to pull those missing log files. The FAL_SERVER parameter is uses the Oracle Net service name.

```sql
SQL> alter system set FAL_SERVER=boston scope=both sid="";
```

Standby Environment Configuration

This section describes the steps that must be executed on the standby database. It is assumed that you have set up Oracle Database Appliance system in the standby environment. For setting up Oracle Database Appliance in a Bare Metal or Virtualized Platform configuration please refer to Oracle Database Appliance Setup Poster.

10. Create the database home

The database home version on the standby must be identical to primary's version.

Old stack (OAK)

```bash
[oracle@stbydb1]# oakcli create dbhome -version 11.2.0.4.200714
```

New stack (DCS)

```bash
[oracle@stbydb1]# odacli create dbhome -v 11.2.0.4.200714
```

11. Setup TNS Entries

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update the tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that on the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

```sql
chicago =
(DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = proddb-scan)(PORT = 1521))
 (CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = chicago)
 )
```
12. Create Static Listener Configuration

As the grid user, create a static listener service on the standby database for Recovery Manager (RMAN) connection during instantiation. Note that the listener home is in the Grid Infrastructure home (/u01/app/18.0.0.0/grid/network/admin)

```
boston =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = stbydb-scan)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = boston)
    )
  )
```

13. Restart Listener

After changes to the listener are made, it must be restarted.

```
[grid@stbydb1]$ lsnrctl reload listener
```

14. Create Initial Standby Parameter File

On the standby host in the ORACLE_HOME/dbs directory create a pfile (initboston1.ora) with the following parameters. It is recommended to set the sga_target same as that of the primary database. For example:

```
[oracle@stbydb1]$ vi /u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/initboston.ora

  db_name=chicago
  db_unique_name=boston
  sga_target=5G
```

15. Copy the Password File from the Primary

During the RMAN duplication process, the auxiliary instance needs to be accessed with remote authentication that requires the creation of the password file.

```
[oracle@stbydb1]$ scp oracle@proddb1:/u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/orapwchicago1 /u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/orapwboston
```

16. Create Audit Directory

Create the audit file destination directory on the standby side on both nodes.

```
[oracle@stbydb1]$ mkdir -p /u01/app/oracle/admin/boston/adump
[oracle@stbydb2]$ mkdir -p /u01/app/oracle/admin/boston/adump
```

17. Create the required ACFS storage directories on the standby Oracle Database Appliance system.

Step only applies to the old stack (OAK).

Use the “oakcli create dbstorage” as ‘root’ user command to create ACFS database storage for the standby:

```
[root@stbydb1]# oakcli create dbstorage -db boston
```
18. Create the storage structures on the standby and register the standby database in DCS. Step only applies to the new stack (DCS).

Identify the home ID that was created for standby

```
[root@stbydb1]# odacli list-dbhomes

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>DB Version</th>
<th>Home Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d7d2b0b-8489-4e28-a151-64f6e156a512</td>
<td>OraDB11204_home1</td>
<td>11.2.0.4.200714</td>
<td>/u01/app/oracle/product/11.2.0.4/dbhome_1</td>
<td>Configured</td>
</tr>
</tbody>
</table>
```

Create an instance only database:

```
[root@stbydb1]# odacli create-database -m -u boston -n chicago -r ACFS -i ACFS -io IO -dh 2d7d2b0b-8489-4e28-a151-64f6e156a512
```

Wait till the instance only database is ready before proceeding further. Verify the status thru odacli describe-jobs.

```
[root@stbydb1]# odacli describe-job -i b8d02cbb-7f0d-48aa-bba5-9d11491560ea

Job details
-------------------------------------------------------------
ID: b8d02cbb-7f0d-48aa-bba5-9d11491560ea
Description: Database service creation with db name: boston
Status: Success
Created: July 7, 2020 1:18:01 PM CEST
Message:
```

Stop or kill the instance which might be running in nomount mode on both nodes.

```
[root@stbydb1]# ps -elf |grep chic

0 S oracle 53721  1  0  80   0 1160761 SYSC_s 03:19 ? 00:00:00 ora_pmon_chicago
...
0 S oracle 54025  1  0  80   0 1160758 SYSC_s 03:19 ? 00:00:00 ora_gcr0_chicago
0 S root 59128  9659  0  80   0 25847 pipe_w 03:20 pts/2 00:00:00 grep chic
[root@scaoda7121 ~]# kill -9 53721
```

19. Startup Standby Instance

Startup the standby database instance on first standby host in NOMOUNT state to prepare for instantiation.

```
[oracle@stbydb1]$ export ORACLE_SID=boston
[oracle@stbydb1]$ sqlplus / as sysdba
SQL> startup nomount
```

20. Validate Network Connectivity

At this stage, Oracle Net should be able to resolve the TNS aliases for both primary and standby environments from the standby environment.

```
[oracle@stbydb1]$ tnsingh chicago
[oracle@stbydb1]$ tnsingh boston
[oracle@stbydb1]$ sqlplus sys/<password>@//proddb1:1521/chicago as sysdba
```
Instantiate Standby Database

This section outlines the instantiation of the standby database after the setup on the primary and standby environments is complete.

21. Duplicate Database

Using Oracle Recovery Manager (RMAN), the standby database can be created with DUPLICATE DATABASE command. As part of the duplication process, the parameter file, password file, controlfile, and database files are copied over from the primary environment to the standby environment.

The appropriate changes required to the parameter settings for standby operation also need to be specified in the RMAN DUPLICATE DATABASE command. Once RMAN copies over the primary parameter file, the parameters specified in the DUPLICATE DATABASE command are changed accordingly.

As the password file is also copied over, the standby database would have the same password as the primary database.

```
[oracle@stbydb1]$ mkdir -p /u01/app/oracle/oradata/datastore/boston/BOSTON/controlfile
[oracle@stbydb1]$ mkdir -p /u02/app/oracle/oradata/datastore/boston/BOSTON/onlinelog
[oracle@stbydb1]$ mkdir -p /u01/app/oracle/oradata/datastore/.ACFS/snaps/boston/BOSTON/datafile
[oracle@stbydb1]$ mkdir -p /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston
[oracle@stbydb1]$ mkdir -p /u01/app/oracle/fast_recovery_area/datastore/boston/BOSTON/archivelog
[oracle@stbydb1]$ mkdir -p /u01/app/oracle/fast_recovery_area/datastore/boston/BOSTON/flashback

[oracle@stbydb1]$ export NLS_DATE_FORMAT="HH24:MI:SS"
[oracle@stbydb1]$ rman
connect target sys/welcome1@//proddb1:1521/chicago
connect auxiliary sys/welcome1@//stbydb1:1521/boston
run {
allocate channel tgt1 device type disk;
allocate auxiliary channel aux1 device type disk;
allocate channel tgt2 device type disk;
allocate auxiliary channel aux2 device type disk;
allocate channel tgt3 device type disk;
allocate auxiliary channel aux3 device type disk;
DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE
SPFILE
PARAMETER_VALUE_CONVERT='chicago','boston','CHICAGO','BOSTON'
SET DB_NAME="chicago"
SET DB_UNIQUE_NAME="boston"
SET CLUSTER_DATABASE='false'
SET REMOTE_LISTENER="stbydb-scan:1521"
SET LOCAL_LISTENER=""
SET STANDBY_FILE_MANAGEMENT='AUTO'
set audit_file_dest = '/u01/app/oracle/admin/boston/adump'
set db_create_file_dest = '/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston'
set diagnostic_dest = '/u01/app/oracle'
set db_recovery_file_dest = '/u01/app/oracle/fast_recovery_area/datastore'
set db_create_online_log_dest_1 = '/u01/app/oracle/oradata/datastore/boston'
set log_archive_dest_2 = 'service=chicago lqwr async register
valid_for=(online_logfiles, primary_role) db_unique_name=chicago'
set fail_server='chicago'
NOFILENAMECHECK;
}
```

Note: Databases use listener_networks instead of local_listener and remote_listener parameters starting from ODA 19.6 on bare metal platform.
22. Modify db_file_name_convert and log_file_name_convert parameters and restart the instance

   [oracle@stbydb1]$ sqlplus / as sysdba
   SQL> alter system set db_file_name_convert='chicago','boston','CHICAGO','BOSTON' scope=spfile;
   SQL> alter system set log_file_name_convert='chicago','boston','CHICAGO','BOSTON' scope=spfile;
   SQL> shutdown immediate
   SQL> startup mount

23. Start Managed Recovery Mode

Start managed recovery on the standby database in real-time mode as follows:

   SQL> alter database recover managed standby database disconnect;

With real-time apply, redo apply services can apply redo log to the standby database as soon as it is received without having to wait for the current standby redo log to be archived. This results in faster failover and switchover times because the standby redo log files have already been applied by the time a failover or switchover occurs.

24. Enable redo transport on the primary

   SQL> alter system set log_archive_dest_state_2='enable' scope=both;

25. Enable Flashback Database

Enable Flashback Database on the standby database and adjust retention as required. For example,

   SQL> alter database recover managed standby database cancel;
   SQL> alter database flashback on;
   SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120;
   SQL> alter database recover managed standby database disconnect;

Post Instantiation Steps

The following steps are performed after the standby instantiation has been completed.

26. Register the standby database with Oracle Clusterware.

Make sure that the ORACLE_HOME environment variable is set correctly. Register the standby database with Oracle Clusterware as single instance to run from one node of the cluster.

   [oracle@stbydb1]$ mv /u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/spfileboston.ora
   /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora
   [oracle@stbydb1]$ echo "spfile='/u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora" >
   /u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/initboston1.ora

For a non-Active Data Guard configuration,

   [oracle@stbydb1]$ srvctl add database -d boston -o /u01/app/oracle/product/11.2.0.4/dbhome_1 -p
   /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora -r physical_standby
   -s mount -c SINGLE -x stbydb1
   -j '/u01/app/oracle/oradata/datastore,/u02/app/oracle/oradata/datastore,/u01/app/oracle/fast_recovery_area/datastore'

For an Active Data Guard configuration,

   [oracle@stbydb1]$ srvctl add database -d boston -o /u01/app/oracle/product/11.2.0.4/dbhome_1 -p
   /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/spfileboston.ora -r physical_standby
   -s 'read only' -c SINGLE -x stbydb1
   -j '/u01/app/oracle/oradata/datastore,/u02/app/oracle/oradata/datastore,/u01/app/oracle/fast_recovery_area/datastore'
Start up the database resource

[oracle@stbydb1]$ sqlplus "/ as sysdba"
SQL> shutdown immediate
[oracle@stbydb1]$ srvctl start database -d boston

27. Convert the standby database to Oracle RAC

This step is optional. At this stage the standby database is configured as a single instance database. If the primary database was RAC database, the standby database can also be converted into RAC standby. Appendix D provides information on using the rconfig tool to convert the single instance database to RAC standby database.

28. Setup Dedicated DR Network

This step is optional. The Redo Transport Services can be configured to use a dedicated network. A dedicated network channel can help in improving the performance of redo transmission especially when the application network traffic consumes most of available bandwidth on the public network.

29. Verify Configuration and Setup

On the standby database internal data dictionary views can be used to verify standby database operations.

[oracle@stbydb1]$ srvctl config database -d boston
SQL> select database_role, switchover_status from v$database;
SQL> select thread#, sequence#, applied from v$archived_log order by sequence#;

30. Setup Clusterware Role Based Services – Refer to Client Failover Best Practices for Highly Available Oracle Databases

31. Sync up the registry (18.7 BM new feature)

# odacli list-databases

<table>
<thead>
<tr>
<th>ID</th>
<th>DB Name</th>
<th>DB Type</th>
<th>DB Version</th>
<th>CDB</th>
<th>Class</th>
<th>Shape</th>
<th>Storage</th>
<th>Status</th>
<th>DbHomeID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d7d2b0b-8489-4e28-a151-646e156a512</td>
<td>chicago</td>
<td>Rac</td>
<td>11.2.0.4.200714</td>
<td>false</td>
<td>Oltp</td>
<td>Odb1</td>
<td>ACFS</td>
<td>Configured</td>
<td></td>
</tr>
<tr>
<td>45e829ab-bf71-48ef-8f9a-0b0397007ba3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# odacli describe-database -i 2d7d2b0b-8489-4e28-a151-64f6e156a512

Database details

ID: 2d7d2b0b-8489-4e28-a151-64f6e156a512
Description: chicago
DB Name: chicago
DB Version: 11.2.0.4.200714
DB Type: Rac
DB Role:
DB Target Node Name:
DB Edition: EE
DBID:
Instance Only Database: true
CDB: false
PDB Name:
PDB Admin User Name:
SEHA Enabled: false
Class: Oltp
Shape: Odb1
Storage: ACFS
DB Redundancy: MIRROR
CharacterSet: AL32UTF8
National CharacterSet: AL16UTF16
Language: AMERICAN
Territory: AMERICA
Home ID: 45e829ab-bf71-48ef-8f9a-0b0397007ba3
Console Enabled: false
TDE Enabled: false
Level 0 Backup Day: Sunday
AutoBackup Enabled: true
Created: November 5, 2019 6:02:19 PM CET
DB Domain Name: us.oracle.com
Associated Networks:

# odacli update-registry -n db -f
# odacli describe-job -i e367a5d4-526a-4d13-8ff2-81ab112a813a

**Job details**

ID: e367a5d4-526a-4d13-8ff2-81ab112a813a
Description: Discover Components : db
Status: Success
Created: July 7, 2020 10:32:37 PM CEST

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Start Time</th>
<th>End Time</th>
<th>Status</th>
</tr>
</thead>
</table>

# odacli describe-database -i 2d7d2b0b-8489-4e28-a151-64f6e156a512

**Database details**

ID: 2d7d2b0b-8489-4e28-a151-64f6e156a512
Description: chicago
DB Name: chicago
DB Version: 11.2.0.4.200714
DB Type: Rac
DB Role: STANDBY
DB Target Node Name:
DB Edition: EE
DBID: 1654855519
Instance Only Database: false
CDB: false
PDB Name:
PDB Admin User Name:
SEHA Enabled: false
Class: Oltp
Shape: Odb1
Storage: ACFS
DB Redundancy: MIRROR
CharacterSet: AL32UTF8
National CharacterSet: AL16UTF16
Language: AMERICAN
Territory: AMERICA
Home ID: 45e829ab-bf71-48ef-8f9a-0b0397007ba3
Console Enabled: false
TDE Enabled: false
Level 0 Backup Day:
AutoBackup Enabled: true
Created: July 7, 2020 10:32:40 PM CEST
Creating Data Guard Broker Configuration

This step is optional, but highly recommended. Creating a Data Guard Broker configuration enables easier management of the entire Data Guard environment as a single entity. It provides management, maintenance and monitoring capabilities that can be used both locally and remotely.

32. Configure listeners for static registration

Configure listeners for static registration of all the instances of primary & standby databases. In the Oracle Database Appliance, listeners are running from the Grid Infrastructure home. An example of static registration for a RAC primary & standby configuration:

On node proddb1:

```sql
SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC = (GLOBAL_DBNAME = chicago_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/11.2.0.4/dbhome_1)
    (SID_NAME = chicago1)))
```

On node proddb2:

```sql
SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC = (GLOBAL_DBNAME = chicago_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/11.2.0.4/dbhome_1)
    (SID_NAME = chicago2)))
```

On node stbydb1:

```sql
SID_LIST_LISTENER =
  (SID_LIST =
   (GLOBAL_DBNAME = boston_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/11.2.0.4/dbhome_1)
    (SID_NAME = boston1)))
```

On node stbydb2:

```sql
SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC = (GLOBAL_DBNAME = boston_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/11.2.0.4/dbhome_1)
    (SID_NAME = boston2)))
```
33. Configure Broker Configuration Files

Configure location of broker configuration files at both primary and standby databases.

```
[oracle@proddb1]$ mkdir -p /u02/app/oracle/oradata/datastore/chicago/broker
SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/chicago/broker/dr1.dat' SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/chicago/broker/dr2.dat' SCOPE=BOTH SID='*';
[oracle@stbydb1]$ mkdir -p /u02/app/oracle/oradata/datastore/boston/broker
SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/boston/broker/dr1.dat' SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/boston/broker/dr2.dat' SCOPE=BOTH SID='*';
```

34. Enable Data Guard Broker

Enable Data Guard Broker on both primary and standby databases.

On node proddb1:

```
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH SID='*';
```

On node stbydb1:

```
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH SID='*';
```

35. Create Broker Configuration

Create the broker configuration on the primary using the DB_UNIQUE_NAME of the primary database and its corresponding TNS alias.

```
[oracle@proddb1]$ dgmgrl
DGMGRL> connect sys/welcome1
DGMGRL> CREATE CONFIGURATION 'ODADGConfig' AS PRIMARY DATABASE IS CHICAGO CONNECT IDENTIFIER is CHICAGO;
```

36. Add Standby Database to Data Guard Broker Configuration

Add standby database to the configuration using the DB_UNIQUE_NAME of the standby database.

```
DGMGRL> ADD DATABASE BOSTON AS CONNECT IDENTIFIER IS BOSTON;
```

37. Enable Configuration

Enable Data Guard Broker configuration as follows.

```
DGMGRL> ENABLE CONFIGURATION;
```

38. Check configuration

Run the following command to verify the established configuration.

```
DGMGRL> show configuration;
DGMGRL> show database verbose chicago;
DGMGRL> show instance verbose chicago1;
DGMGRL> show instance verbose chicago2;
DGMGRL> show database verbose boston;
DGMGRL> show instance verbose boston1;
DGMGRL> show instance verbose boston2;
```
Verify that 'StaticConnectIdentifier' parameter in "show instance verbose" output look like

```
StaticConnectIdentifier = '(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=proddb1)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=chicago_DGMGRL.us.oracle.com)(INSTANCE_NAME=chicago1)(SERVER=DEDICATED)))'
```

instead of

```
StaticConnectIdentifier = '(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=DUMMY))(CONNECT_DATA=(SERVICE_NAME=chicago_DGMGRL.us.oracle.com)(INSTANCE_NAME=chicago1)(SERVER=DEDICATED)))'
```

In case of the latter one update the parameter with the following commands:

```
DGMGRL> edit instance chicago1 set property
StaticConnectIdentifier='(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prodb1)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=chicago_DGMGRL.us.oracle.com)(INSTANCE_NAME=chicago1)(SERVER=DEDICATED)))';

DGMGRL> edit instance chicago2 set property
StaticConnectIdentifier='(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prodb2)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=chicago_DGMGRL.us.oracle.com)(INSTANCE_NAME=chicago2)(SERVER=DEDICATED)))';
```

This issue could happen if database was created on 19.6 or later ODA BM versions.

39. Switchover tests

```
$ dgmg
DGMGRL> connect sys/welcome1@boston
DGMGRL> switchover to boston
DGMGRL> connect sys/welcome1@chicago
Connected as SYSDBA.
DGMGRL> switchover to chicago;
```

40. Failover tests

```
connect to standby before failover:

$ dgmg
DGMGRL> connect sys/welcome1@boston
DGMGRL> failover to boston
DGMGRL> reinstate database chicago
connect to former primary before failover:

DGMGRL> connect sys/welcome1@chicago
DGMGRL> failover to chicago;
DGMGRL> reinstate database boston
```
APPENDIX D: CONVERTING SINGLE INSTANCE DATABASES TO ORACLE RAC

You can use the `rconfig` command line utility to convert a single-instance database to an Oracle RAC database, or to convert it to an Oracle RAC One Node.

To use this feature, complete the following steps:

1. Create Configuration XML File

A sample of the configuration XML file to be saved as convert.xml is shown below. You may modify this file as required for your system.

The sample XML files are in `$ORACLE_HOME/assistants/rconfig/sampleXML` directory.

Note: Set the convert option `Convert verify="ONLY"` initially to perform a test conversion to ensure that a conversion can be completed successfully.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<n:RConfig xmlns:n="http://www.oracle.com/rconfig"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.oracle.com/rconfig rconfig.xsd">
  <n:ConvertToRAC>
    <n:Convert verify="YES">
      <n:SourceDBHome>/u01/app/oracle/product/11.2.0.4/dbhome_1</n:SourceDBHome>
      <n:TargetDBHome>/u01/app/oracle/product/11.2.0.4/dbhome_1</n:TargetDBHome>
      <n:SourceDBInfo SID="boston">
        <n:Credentials>
          <n:User>sys</n:User>
          <n:Password>welcome1</n:Password>
          <n:Role>sysdba</n:Role>
        </n:Credentials>
      </n:SourceDBInfo>
      <n:NodeList>
        <n:Node name="stbydb1"/>
        <n:Node name="stbydb2"/>
      </n:NodeList>
      <n:InstancePrefix>boston</n:InstancePrefix>
      <n:SharedStorage type="CFS">
        <n:TargetDatabaseArea></n:TargetDatabaseArea>
        <n:TargetFlashRecoveryArea></n:TargetFlashRecoveryArea>
      </n:SharedStorage>
    </n:Convert>
  </n:ConvertToRAC>
</n:RConfig>
```
2. Run rconfig Tool

When you have completed making changes, save the file. Run the following command on the standby database. The `convert.xml` is the name of the XML input file you configured above.

```
[oracle@stbydb1]$ rconfig convert.xml
```

3. Update Cluster Ready Services Resource

The Cluster Ready Services (CRS) resource must be updated for the converted database.

```
[oracle@stbydb1]$ srvctl modify database -d boston -r physical_standby -s mount
```

Restart the standby database

```
[oracle@stbydb1]$ srvctl stop database -d boston
[oracle@stbydb1]$ srvctl start database -d boston
```

4. Validate Configuration

Validate the configuration of standby database.

```
[oracle@stbydb1]$ srvctl config database -d boston
```
APPENDIX E: UPGRADING DATABASE WITH ORACLE DATA GUARD ON ODA WITH OAK

UPGRADING ALL COMPONENTS

Upgrading an ODA environment consists of upgrading the SERVER, STORAGE, and DATABASE components. When upgrading an ODA environment where a standby system is already implemented, you can leverage the standby system to reduce downtime required for completing the upgrade activities. The purpose of this section is to provide a high-level overview of the upgrade process in a primary-standby setup.

1. Verify system is operating normally (run pre-checks, validate hardware and system processes, verify system configuration using orachk, etc.)
2. Take a backup of the OS, GI, and Oracle homes, and databases (in the primary environment). On bare metal deployments refer to MOS Note 2466177.1 - ODA: ODABR a System Backup/Restore Utility
3. Upgrade SERVER components on the standby ODA system
4. Switchover primary database role and application connections to the standby ODA system
5. Upgrade SERVER components on the original primary ODA system
6. Create new DB home on the current standby with the desired version using “oakcli create dbhome” command, if it does not exist or you do not want to use an existing one
7. Create new DB home with the desired version on the current primary, if it does not exist or you do not want to use an existing one
8. Defer redo transfer from primary to standby
9. Run “oakcli upgrade database” command on the current standby database (only binaries would be upgraded/switched and catalog scripts will not be run)
10. Stop the application traffic
11. Upgrade current primary database using “oakcli upgrade database” command
12. Start the application traffic
13. Enable redo transfer from primary to the standby database
14. Verify DB operations on primary and standby side
15. Optionally switchback roles between primary and standby environments

Using the above process, the downtime during the upgrade is minimized and system availability is affected for only the duration of the upgrade of the database component.
UPGRADING THE DATABASE ONLY

Upgrading SERVER (OS, GI, general firmware) and STORAGE (firmware on shared disks) components leveraging switchover/switchback can reduce downtime. However, if you are only upgrading the DATABASE component, then unless you are using a zero downtime solution (such as active-active GoldenGate solution), some downtime is expected for the application. The general process to execute the database upgrade when a standby configuration exists may be outlined something like as follows.

1. Verify that the system is operating normally (run pre-checks, validate hardware and system processes, verify system configuration using orachk, etc.)
2. Take a backup of the database and Oracle homes
3. Create new DB home on the standby with the desired version using “oakcli create dbhome” command, if it does not exist or you do not want to use an existing one
4. Create new DB home with the desired version on the primary, if it does not exist or you do not want to use an existing one
5. Stop the application
6. Let standby DB sync up with primary and verify last SCN generated on primary is applied on the standby
7. Defer redo transfer from primary to standby database (optional; stop redo flow)
8. Run “oakcli upgrade database” command on the standby database (only binaries would be upgraded/switched and catalog scripts will not be run)
9. Upgrade primary database using “oakcli upgrade database” command
10. Start the application
11. Enable redo transfer from the primary to the standby database
12. Verify DB operations on the primary and the standby side

The total downtime requirement is the duration of the DB upgrade. A switchover and switchback is not required during database upgrade.
APPENDIX F: UPGRADING DATABASE WITH ORACLE DATA GUARD ON ODA WITH DCS

Note: On 19.8 ODA bare metal deployments follow the documentation and completely rely on ODA tooling as it provides complete lifecycle management for Oracle Data Guard environments too including database patching and upgrade if the Data Guard was configured by odacli commands. Refer to Configuring Oracle Data Guard on Oracle Database Appliance. For Data Guard environments configured by odacli commands the following steps don’t apply.

UPGRADING ALL COMPONENTS

Upgrading an ODA environment consists of upgrading DCS, SERVER, STORAGE, and DATABASE components. When upgrading an ODA environment where a standby system is already implemented, you can leverage the standby system to reduce downtime required for completing the upgrade activities. The purpose of this section is to provide a high-level overview of the upgrade process in a primary-standby setup.

1. Verify that the system is operating normally (run pre-checks, validate hardware and system processes, verify system configuration using orachk, etc.)

2. Take a backup of the OS, GI, and Oracle homes, and databases (in the primary environment)
   Refer to MOS Note 2466177.1 - ODA (Oracle Database Appliance): ODABR a System Backup/Restore Utility

3. Upgrade DCS, SERVER components on the standby ODA system

4. Switchover primary database role and application connections to the standby ODA

5. Upgrade DCS, SERVER components on the current standby (former primary) ODA

6. Patch or upgrade the database depending on your requirements.
   On 19.8 ODA bare metal deployments follow the documentation and completely rely on ODA tooling as it provides complete lifecycle management for Oracle Data Guard environments too including database patching and upgrade if the Data Guard was configured by odacli commands. Refer to Configuring Oracle Data Guard on Oracle Database Appliance

Using the above process, the downtime during the upgrade is minimized and system availability is affected for only the duration of the upgrade of the database component.
UPGRADING THE DATABASE

Upgrading DCS, SERVER (OS, GI, general firmware) and STORAGE (firmware on shared disks) components leveraging switchover/switchback can help reduce the downtime. However, if you are only upgrading the DATABASE component, then unless you are using a zero downtime solution (such as active-active GoldenGate solution), some downtime is expected for the application. The general process to execute the database upgrade when a standby configuration exists may be outlined something like as follows.

1. Verify that the system is operating normally (run pre-checks, validate hardware and system processes, verify system configuration using orachk, etc.)
2. Take a backup of the database and Oracle homes
3. Stop the standby database
   ```bash
   [oracle@stbydb1]$ srvctl stop database -d boston
   ```
4. Create a new database home or use an existing one on the standby with the version that the database will be upgraded to on the primary
   ```bash
   [oracle@stbydb1]$ odaclic create-dbhome -v 18.7.0.0.190716
   ```
5. Stop log shipping on the primary
   ```bash
   [oracle@prod1]$ dgmgrl
   connect sys/welcome1@chicago
   DGMGR> SHOW DATABASE 'boston' 'LogShipping';
   LogShipping = 'ON'
   DGMGR> edit database 'boston' SET PROPERTY 'LogShipping'='OFF';
   Property "LogShipping" updated
   DGMGR> SHOW DATABASE 'boston' 'LogShipping';
   LogShipping = 'OFF'
   ```
6. Create a new database home or use an existing one on the primary with the version that the database will be upgraded to
   ```bash
   # odaclic create-dbhome -v 18.7.0.0.190716
   ```
7. Stop the application
8. Upgrade the primary database using "odaclic upgrade database" command
   ```bash
   [root@prod1]$ odaclic list-databases
   ID                                       DB Name    DB Type  DB Version           CDB        Class    Shape    Storage    Status        DbHomeID
   ---------------------------------------- ------------- ------------- ------------- --------------- ---------- ---------- ---------- -------------- --------------
   e97cc2f3-bdd8-4775-b959-d5f79a6c59fc     chicago    Rac      12.2.0.1.190716      false      Oltp     Odb1     Asm        Configured
   88ce2c7-1a3d-4f93-802a-bfa50d180758
   ```
[root@proddb1]# odacli list-dbhomes

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>DB Version</th>
<th>Home Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>863c8cbe-1c5f-450e-866c-15c384580ad3</td>
<td>OraDB18000_home1</td>
<td>18.7.0.0.190716</td>
<td></td>
<td>Configured</td>
</tr>
<tr>
<td>/u01/app/oracle/product/18.0.0.0/dbhome_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>288ce2c7-fa3d-4f93-802a-bfa50d180758</td>
<td>OraDB12201_home1</td>
<td>12.2.0.1.190716</td>
<td></td>
<td>Configured</td>
</tr>
<tr>
<td>/u01/app/oracle/product/12.2.0.1/dbhome_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[root@proddb1]# odacli upgrade -i 713b68d3-8c43-4d10-973e-90a3fa88a84a --destDbHomeld 863c8cbe-1c5f-450e-866c-15c384580ad3 --sourceDbHomeld 288ce2c7-fa3d-4f93-802a-bfa50d180758

[root@proddb1]# odacli list-databases

<table>
<thead>
<tr>
<th>ID</th>
<th>DB Name</th>
<th>DB Type</th>
<th>DB Version</th>
<th>CDB</th>
<th>Class</th>
<th>Shape</th>
<th>Storage</th>
<th>Status</th>
<th>DbHomeID</th>
</tr>
</thead>
<tbody>
<tr>
<td>713b68d3-8c43-4d10-973e-90a3fa88a84a</td>
<td>chicago</td>
<td>Rac</td>
<td>18.7.0.0.190716</td>
<td>false</td>
<td>Oltp</td>
<td>Odb1</td>
<td>Asm</td>
<td>Configured</td>
<td>863c8cbe-1c5f-450e-866c-15c384580ad3</td>
</tr>
</tbody>
</table>

9. Start the application

10. Copy the tnsnames.ora file on the standby from the old Oracle Home to the new on all nodes

11. Copy the password file from the primary to the standby

   [oracle@proddb1]$ srvctl config database -d chicago | grep Password
   Password file: +DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847
   [grid@proddb1]$ asmcmd
   ASMCMD> pwcopy +DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847 /tmp/pwdboston
   copying +DATA/CHICAGO/PASSWORD/pwdchicago.277.1023633847 -> /tmp/pwdboston
   [oracle@proddb1]$ scp /tmp/pwdboston oracle@stbydb1:/u01/app/oracle/product/18.0.0.0/dbhome_2/dbs/orapwboston
   [grid@stbydb1]$ asmcmd
   ASMCMD> pwcopy /u01/app/oracle/product/18.0.0.0/dbhome_2/dbs/orapwboston +DATA/BOSTON/PASSWORDFILE/pwdboston
   copying /u01/app/oracle/product/18.0.0.0/dbhome_2/dbs/orapwboston -> +DATA/BOSTON/PASSWORDFILE/pwdboston

12. Remove the 12.2 database from the Clusterware on the standby.

   [oracle@stbydb1]$ srvctl remove database -db boston
   Remove the database boston? [y/n]: y

13. Add the database back to the Clusterware on the standby. Oracle Home has to point to the new version of the home

   Single instance example

   [oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/18.0.0.0/dbhome_1 -dbtype SINGLE -instance boston1 -node stbydb1 -dbname chicago -diskgroup 'DATA,REDO,RECO' -role physical_standby -spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -pwfile '+DATA/BOSTON/PASSWORDFILE/pwdboston'

   RAC example

   [oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/oracle/product/18.0.0.0/dbhome_1 -dbtype RAC -dbname chicago -diskgroup 'DATA,REDO,RECO' -role physical_standby -spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -pwfile '+DATA/BOSTON/PASSWORDFILE/pwdboston'
   [oracle@stbydb1]$ srvctl add instance -database boston -instance boston1 -node stbydb1
   [oracle@stbydb1]$ srvctl add instance -database boston -instance boston2 -node stbydb2
   [oracle@stbydb1]$ srvctl start instance -db boston -instance boston1 -o mount
   [oracle@stbydb1]$ srvctl start instance -db boston -instance boston2 -o mount
14. Enable log shipping and validate the Data Guard configuration

```
[oracle@stbydb1]$ dgmgrl
DGMGR> connect sys/welcome1@chicago
DGMGR> edit database 'boston' SET PROPERTY 'LogShipping'='ON';
Property "LogShipping" updated
DGMGR> SHOW DATABASE 'boston' 'LogShipping';
LogShipping = 'ON'
DGMGR> show configuration verbose
DGMGR> show database verbose chicago
DGMGR> show database verbose boston
DGMGR> validate database chicago
DGMGR> validate database boston
```

15. Test the switchover and failover

**Switchover tests**
```
$ dgmgrl
DGMGR> connect sys/welcome1@boston
DGMGR> switchover to boston
DGMGR> connect sys/welcome1@chicago
DGMGR> switchover to chicago;
```

**Failover tests**
Connect to standby before failover:
```
$ dgmgrl
DGMGR> connect sys/welcome1@boston
DGMGR> failover to boston
DGMGR> reinstate database chicago
```

Connect to former primary before failover:
```
DGMGR> connect sys/welcome1@chicago
DGMGR> failover to chicago;
DGMGR> reinstate database boston
```

**Health check**
```
DGMGR> show database verbose Chicago
DGMGR> show database verbose boston
DGMGR> validate database Chicago
DGMGR> validate database boston
```
16. Sync up the registry on the standby (18.7 BM new feature)

Verify the version of the DB

[root@ stbydb1-]# odacli list-databases

<table>
<thead>
<tr>
<th>ID</th>
<th>DB Name</th>
<th>DB Type</th>
<th>DB Version</th>
<th>CDB</th>
<th>Class</th>
<th>Shape</th>
<th>Storage</th>
<th>Status</th>
<th>DbHomeID</th>
</tr>
</thead>
<tbody>
<tr>
<td>e6450a56-5a7d-4dab-9ca9-25b004b66646</td>
<td>chicago</td>
<td>Rac</td>
<td>12.2.0.1.190716</td>
<td>false</td>
<td>Oltp</td>
<td>Odb1</td>
<td>Asm</td>
<td>Configured</td>
<td>755b4b5d-6211-4d94-81e9-cf611868e39</td>
</tr>
</tbody>
</table>

Sync up registry entries

[root@ stbydb1-]# odacli update-registry -n db --f
[root@ stbydb1-]# odacli describe-job -i 25ec2987-4c93-4d25-97db-bad2f6f602f6

Job details

ID: 25ec2987-4c93-4d25-97db-bad2f6f602f6

Description: Discover Components : db

Status: Success

Created: November 6, 2019 11:00:50 PM CET

Message:

Task Name            | Start Time          | End Time           | Status     |
---------------------|---------------------|--------------------|------------|
Rediscover DBHome    | November 6, 2019 11:00:54 PM CET | November 6, 2019 11:00:56 PM CET | Success    |
Rediscover DB: boston | November 6, 2019 11:00:56 PM CET | November 6, 2019 11:01:02 PM CET | Success    |

Confirm the changes in the registry

[root@ stbydb1-]# odacli list-databases

<table>
<thead>
<tr>
<th>ID</th>
<th>DB Name</th>
<th>DB Type</th>
<th>DB Version</th>
<th>CDB</th>
<th>Class</th>
<th>Shape</th>
<th>Storage</th>
<th>Status</th>
<th>DbHomeID</th>
</tr>
</thead>
<tbody>
<tr>
<td>e6450a56-5a7d-4dab-9ca9-25b004b66646</td>
<td>chicago</td>
<td>Rac</td>
<td>18.7.0.0.190716</td>
<td>false</td>
<td>Oltp</td>
<td>Odb1</td>
<td>Asm</td>
<td>Configured</td>
<td>17f68bbf-b812-42e5-96ba-1433c30f75ed</td>
</tr>
</tbody>
</table>

The total downtime requirement the duration of the DB upgrade. A switchover and switchback is not required for a database upgrade.
PATCHING THE DATABASE

Patching the databases on ODA is a completely online operation. The process to execute the database patching when a standby configuration exists may be outlined something like as follows.

1. Verify system is operating normally (run pre-checks, validate hardware and system processes, verify system configuration using orachk, etc.)
2. Take a backup of the database
3. Stop log shipping on the primary

```bash
$ dgmgrl
DGMRGL> connect sys/welcome1@chicago
DGMRGL> edit database 'CHICAGO' SET STATE="LOG-TRANSPORT-OFF";
DGMRGL> SHOW DATABASE 'boston' 'LogShipping';
    LogShipping = 'ON'
DGMRGL> edit database 'boston' SET PROPERTY 'LogShipping'='OFF';
    Property "LogShipping" updated
DGMRGL> SHOW DATABASE 'boston' 'LogShipping';
    LogShipping = 'OFF'
```
4. Stop the standby database and restart it in “read only” mode

```bash
[oracle@stbydb1]$ srvctl stop database -d boston
[oracle@stbydb1]$ srvctl start database -d boston -o "read only"
```
5. Patch the standby database first

Identify the Oracle home of the database

```bash
[root@ocboda10 ~]# odacli list-databases
ID                                        DB Name    DB Type  DB Version           CDB        Class    Shape    Storage    Status        DbHomeID
---------------------------------------- ---------- -------- ----------------- ---------- -------- -------- -------- -----------------  ---------------
667a0eec-910c-404b-9820-aecdff668d7     chicago    Rac      18.5.0.0.190115     false      Oltp     Odb1     Asm        Configured
```

```bash
[oracle@stbydb1]$ odacli list-dbhomes
ID                                        Name                 DB Version                               Home Location            Status
---------------------------------------- ------------------- -----------------  ------------------------  -----------------------------------
010b4e9c6d4-4c56-bb0c-b239a4e749f3     OraDB18000_home3     18.5.0.0.190115    /u01/app/oracle/product/18.0.0.0/dbhome_3     Configured
```

Copyright © 2020, Oracle and/or its affiliates
Run the pre-check on the Oracle home

```
[oracle@stbydb1]# odacli update-dbhome -p -i 6d05e3f1-e948-4482-bcba-c560d9c8e5e5 -v 18.7.0.0
[oracle@stbydb1]# odacli describe-job -i b4ee24d9-2b82-4c80-b789-ced90013e4b3
```

**Job details**

```
ID: b4ee24d9-2b82-4c80-b789-ced90013e4b3
Description: DB Home Prechecks
Status: Success
Created: November 7, 2019 6:26:51 PM CET
```

Apply the patches

```
[oracle@stbydb1]# odacli update-dbhome -i 6d05e3f1-e948-4482-bcba-c560d9c8e5e5 -v 18.7.0.0
[oracle@stbydb1]# odacli describe-job -i "e3556125-7ce6-4560-9f22-3fdd9738f955"
```

**Job details**

```
ID: e3556125-7ce6-4560-9f22-3fdd9738f955
Description: DB Home Patching: Home Id is e4e9fcbd-63d4-4c56-bb0c-b239a4e749f3
Status: Success
Created: November 7, 2019 7:09:52 PM CET
```

Verify the result

```
[oracle@stbydb1]# odacli list-dbhomes
ID      Name         DB Version     Home Location     Status
----------------------------------------
----------------------------------------
----------------------------------------
e4e9fcbd-63d4-4c56-bb0c-b239a4e749f3     OraDB18000_home3     18.7.0.0.190716
/u01/app/oracle/product/18.0.0.0/dbhome_3     Configured
```

```
[oracle@stbydb1]# odacli list-databases
ID      DB Name    DB Type  DB Version  CDB    Class  Shape  Storage Status  DbHomeID
-------- ------------- ----------- ----------------- ------- ------ ------ ----- --------- ------------
667a0eec-910c-404b-9920-aedcdff668d7     chicago  Rac     18.7.0.0.190716  false  Oltp  Odb1  Asm  Configured
```

6. Patch the primary database. Steps are the same as in Step 5.

7. Start log shipping on the primary and verify the Data Guard configuration

```
DGMGR> connect sys/welcome1@chicago
DGMGR> edit database 'boston' SET PROPERTY 'LogShipping'='ON';
Property "LogShipping" updated
DGMGR> SHOW DATABASE 'boston' "LogShipping";
LogShipping = 'ON'
DGMGR> show configuration verbose
dbms_connection_string
DGMGR> show database verbose chicago
dbms_connection_string
DGMGR> validate database chicago
dbms_connection_string
DGMGR> validate database boston
dbms_connection_string
```
APPENDIX G: WHY DOES THE SAME VERSION OF RDBMS HOME HAVE PSU AND BUNDLE PATCH ON SOME OLDER VERSION?

12.1 databases had different patches on the new stack (DCS) than on the old one (OAK) on a few older releases.

» S/M/L models have always been running on the new stack (DCS/odacli). Bundle Patches are applied on the 12.1 databases on these models.
» On HA models PSUs were applied on 12.1 RDBMS homes on some older ODA releases with the old stack (OAK/oakcli).
» Bundle Patches include many more fixes than PSUs. They are supersets of PSUs.
» Bundle patches have been applied on 12.1 RDBMS on all models with all stacks since 12.2.1.2 ODA release.

The aforementioned difference prevents configuring Data Guard between ODAs if RDBMS home has PSU on one and BP on the other. Please remember the patch level should be identical on the primary and the standby database. Refer to MOS Note 785347.1 - Mixed Oracle Version support with Data Guard Redo Transport Services

This issue affects those configurations where the database homes have not been patched for 3 years or more.

The solution is to patch the database to the latest version which includes BP for 12.1 RDBMS version.
FOR FURTHER READING

DOCUMENTATION

Oracle Database Appliance Website
Configuring Oracle Data Guard on Oracle Database Appliance
Oracle Maximum Availability Architecture Best Practices
Oracle Database High Availability Website
Oracle Real Application Clusters Website
Oracle Clusterware Website
Oracle Data Guard Website
Oracle Data Guard Concepts and Administration 19c, 18c, 12.2, 12.1, 11.2

TECHNICAL BRIEFS

Maximum Availability Architecture (MAA) - On-Premises HA Reference Architectures
Best Practices for Configuring Redo Transport for Active Data Guard 12c
Best Practices for Asynchronous Redo Transport - Data Guard and Active Data Guard
Best Practices for Synchronous Redo Transport - Data Guard and Active Data Guard
Best Practices for Automatic Resolution of Outages to Resume Data Guard Zero Data Loss
Preventing, Detecting, and Repairing Block Corruption - Oracle Database 12c
Role Transition Best Practices: Data Guard and Active Data Guard
Client Failover Best Practices for Highly Available Oracle Databases
Client Failover Best Practices for Data Guard 12c
Database Rolling Upgrade using Data Guard
Automated Database Upgrades using Oracle Active Data Guard and DBMS_ROLLING

MY ORACLE SUPPORT (MOS) KNOWLEDGE CONTENT NOTES

Note 2466177.1 - ODA: ODABR a System Backup/Restore Utility
Note 1265700.1 - Oracle Patch Assurance - Data Guard Standby-First Patch Apply
Note 1617946.1 - Creating a Physical Standby Database using RMAN Duplicate (RAC or Non-RAC)
Note 2283978.1 - Creating a Physical Standby database using RMAN restore from service
Note 785347.1 - Mixed Oracle Version support with Data Guard Redo Transport Services