ORACLE

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

Protect production systems while leveraging standby computing power

Oracle Maximum Availability Architecture

Jul 14, 2022

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.

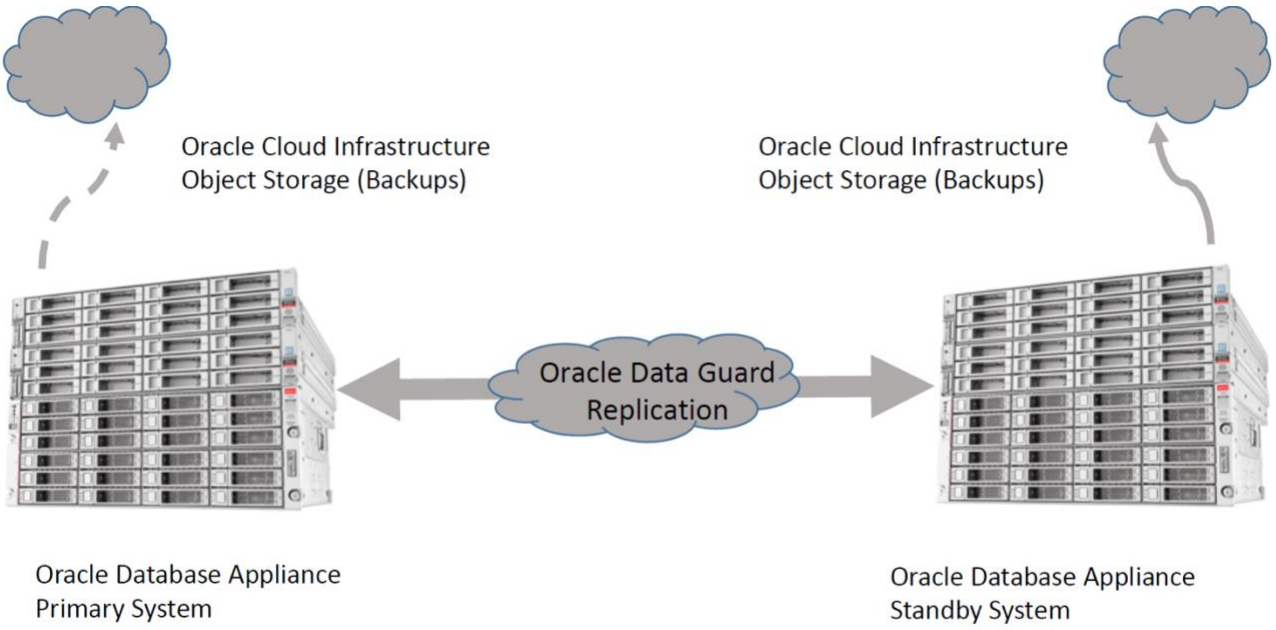
This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.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

Contents

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 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.
Note: In case OJVM is in use in the database Standby-first patching is not possible. MOS note 2217053.1 - RAC Rolling Install Process for the "Oracle JavaVM Component Database PSU/RU" (OJVM PSU/RU) Patches can help to confirm OJVM usage.

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. Utlimately 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 and later, 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 the documentation (database on bare metal, KVM based DBSystems) 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.
  Data Guard related odacli commands also use Oracle Data Guard Broker under the hood.
- **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

Please refer to https://www.oracle.com/database/technologies/high-availability/oracle-database-maa-best-practices.html for the above best practices and more.

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

Prerequisites for Oracle Database Data Guard Configuration on ODA 19.14:

- 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 that the primary and standby systems have the same Oracle Database Appliance configuration, if possible. 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. The recommended version is 19.14 due to critical bug fixes.
- If you have customized the operating system, then ensure that environments on both machines are identical.
- Ensure that your deployment follows Oracle Maximum Availability Architecture (MAA) best practices. See the Oracle Maximum Availability Architecture (MAA) page on Oracle Technology Network.
- If you decide to use Oracle ObjectStore for backup and recovery, then you must configure access for both the primary and standby systems.

This technical brief also 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 19.14 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 19.14 for any limitation that may apply.

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

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,B and C 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.

- Oracle Enterprise Manager is not integrated with ODA for instantiating a standby system. You can however, follow the above-mentioned notes or examples provided in the appendix sections of this technical brief for configuring your 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 the storage structure for your standby database with "odacli create-dbstorage" command.
  For example:

  Database storage on ASM:

  # odacli create-dbstorage -n boston -u chicago

  Database storage on ACFS:

  # odacli create-dbstorage -n boston -u chicago -r ACFS

  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 platform with KVM and DBSystem support or as an Oracle Virtual Machine (OVM) based Virtualized Platform. Integrated Data Guard configuration with odacli is the preferred way on BM and DBSystem deployments. However, the manual 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 including DBSystems and OVM based 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 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 (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 odacli, oakcli, RMAN, and Oracle Data Guard Broker.

APPENDIX A: DATA GUARD CONFIGURATION USING ODACLI

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.

	Primary Oracle Database Appliance		Standby Oracle Database Appliance	
Host Names	proddb1	proddb2	stbydb1	stbydb2
Database Name	hun		hun	
Database Unique Name	buda		pest	
Instance Name	budapest1	budapest2	budapest1	budapest2
SCAN Name and IPs	proddb-scan (10.1.27.2, 10.1.27.3)		stbydb-scan (10.1.27.4, 10.1.27.5)	
Grid Infrastructure Software Installation	/u01/app/19.14.0.0/grid		/u01/app/19.14.0.0/grid	
Oracle Database Software Installation	/u01/app/odaorahome/oracle/product/19.0.0.0/db_home1		/u01/app/odaorahome/oracle/product/19.0.0.0/db_home1	
Database storage	ASM		ASM	
ARCHIVELOG mode	Yes		Yes	
FORCE LOGGING mode	Yes		Yes	

CONFIGURING ORACLE DATA GUARD

1. Configure remote database backup for the source database either on NFS or on the cloud based Oracle Object Store

Backup location on NFS

In case NAS or external NFS server is not available you could configure NFS server by following the steps in Appendix H

- the NFS Filesystem is mounted on all source and target nodes
- for a TDE enabled database: DB and TDE backup folders are readable and writeable by oracle OS user
- for a DB without TDE encryption: DB folder is readable and writeable by oracle OS user
- NFS Filesystem is shared with no_root_squash flag

Create a backup configuration for a TDE enabled database

```
# odacli create-backupconfig -d NFS -n nfs -cr -c /odabackup/db -f /odabackup/tde -w 7
```

Create a backup configuration for a non-TDE database

```
# odacli create-backupconfig -d NFS -n nfs -cr -c /odabackup/db -w 7
```

For all odacli create-backupconfig options refer to the documentation or the online help (-h)

Backup location on Oracle Object Storage

- for a TDE enabled database: create dedicated buckets for DB and TDE backups
- for a DB without TDE encryption: create a bucket for the DB backups

Update the DCS agent configuration with the internet proxy if needed

```
[root@proddb1] # odacli update-agentconfig-parameters -n HttpProxyHost -v proxy.oracle.com -n HttpProxyPort -v 80 -u
```

Create Object Storage credential details

```
[root@proddb1] # odacli create-objectstoreswift -e https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1 -n oosswift -t mytenant -u firstname.lastname@oracle.com
```

For all odacli create-objectstoreswift options refer to the documentation or the online help (-h).

Create backup configuration

```
[root@proddb1] # odacli create-backupconfig -d ObjectStore -c dbbackups -on oosswift -w 7 -f tdebackups -cr -n backupConfig2ObjectStorage
```

For all odacli create-backupconfig options refer to the documentation or the online help (-h)

Verify that the backup configuration is available

```
[root@proddb1] # odacli list-backupconfigs
ID                                       Name                        RecoveryWindow   CrosscheckEnabled   BackupDestination
---------------------------------------- --------------------------- ---------------- ------------------- ---------------------
c0bc22a2-b9c0-4b3e-a4fb-1e69c661cfbf     backupConfig2ObjectStorage  7                true                ObjectStore
251aadf9-34ea-4579-aab7-d0e0c8f27dc7     nfs                         7                true                NFS
```

Assign the backup configuration to the source database

TDE-enabled database:

```
[root@proddb1] # odacli modify-database -in hun -bin nfs
```

Non-TDE database with an RMAN backup password:

```
[root@proddb1] # odacli modify-database -in hun -bin nfs -bp
```

2. Create a Level 0 database backup and keep archive logs

```
[root@proddb1] # odacli create-backup -in hun -bt Regular-L0 -ka
{
"jobId" : "2ff6931c-aa69-4529-92fa-379dda6e6a36",
"status" : "Created",
"message" : null,
"reports" : [ ],
"createTimestamp" : "March 18, 2022 16:15:57 PM CET",
"resourceList" : [ ],
"description" : "Create Regular-L0 Backup[TAG:auto][Db:hun][NFS:/odabackup/db/orabackups/primaryODA-c/database/2894792645/buda]",
"updatedTime" : "March 18, 2022 16:15:57 PM CET"
}
```

Verify that the job completed successfully with odacli describe-job.

```
[root@proddb1] # odacli describe-job -i 2ff6931c-aa69-4529-92fa-379dda6e6a36
Job details
----------------------------------------------------------------
                     ID:  2ff6931c-aa69-4529-92fa-379dda6e6a36
            Description:  Create Regular-L0 Backup[TAG:auto][Db:hun][NFS:/odabackup/db/orabackups/primaryODA-
c/database/2894792645/buda]
                  Status:  Success
                 Created:  March 18, 2022 4:15:57 PM CET
Task Name                            Start Time                          End Time                            Status
------------------------------------ ----------------------------------- ----------------------------------- ----------
Validate TDE Wallet Existence        March 18, 2022 4:16:00 PM CET       March 18, 2022 4:16:01 PM CET       Success
Validate backup config               March 18, 2022 4:16:01 PM CET       March 18, 2022 4:16:01 PM CET       Success
NFS location existence validation    March 18, 2022 4:16:01 PM CET       March 18, 2022 4:16:02 PM CET       Success
Backup Validations                   March 18, 2022 4:16:02 PM CET       March 18, 2022 4:16:07 PM CET       Success
Recovery Window validation           March 18, 2022 4:16:07 PM CET       March 18, 2022 4:16:10 PM CET       Success
Archivelog deletion policy configuration March 18, 2022 4:16:10 PM CET   March 18, 2022 4:16:14 PM CET       Success
Database backup                      March 18, 2022 4:16:14 PM CET       March 18, 2022 4:17:41 PM CET       Success
Password Protected TDE Wallet Backup March 18, 2022 4:17:41 PM CET       March 18, 2022 4:17:42 PM CET       Success
```

3. Identify the ID of the backupreport that belongs to the L0 backup:

```
[root@proddb1]# odacli list-backupreports
Backup Report Id                     Database Resource Id                Database DbId DB Name    DB Unique Name Backup Type  Backup Tag
Create Time                          Updated Time                        Status
------------------------------------ ----------------------------------- ------------ ---------- -------------- ------------ ------------ ------------------------------
13faba84-d83f-499d-ae4a-4bb451f4702c c0409b01-03da-4326-b268-29a48d8d617f 2894792645   hun        buda           Regular-L0   auto         March 18, 2022 4:16:14 PM CET       March 18, 2022 4:17:42 PM CET       Configured
```

4. Take a backup of the backup report into json format and copy it over to the standby machine

Backup on NFS:

```
[root@proddb1]# odacli describe-backupreport -i 13faba84-d83f-499d-ae4a-4bb451f4702c >
/odabackup/backupreport_hun_20220318.json
```

Backup on Oracle Object Storage:

```
[root@proddb1]# odacli describe-backupreport -i 13faba84-d83f-499d-ae4a-4bb451f4702c > /tmp/backupreport_hun_20220318.json
```

Copy the json file to the standby ODA:

```
[root@proddb1]# scp /tmp/backupreport_hun_20220318.json root@stbydb1:/tmp
```

5. Double-check if the Object Storage Swift or NFS was configured on the standby side
   - NFS: the filesystem is mounted on both target nodes (df -h)
   - Oracle Object Storage: swift credentials are configured (odacli list-objectstoreswifts)
6. Restore the database as a standby on the target – target could be Bare Metal or DBSystem
   In case the target is a DBSystem, make sure that no database is configured in it.

Identify the ID of the home in case an existing home would be used:

```
[root@stdbydb1 ~]# odacli list-dbhomes
ID                                       Name            DB Version                    Home Location                                  Status
---------------------------------------- --------------- ----------------------------- ---------------------------------------------- ----------
e8a36f29-7fcf-49fc-8575-c599dc28949d     OraDB19000_home1     19.14.0.0.220118
/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 CONFIGURED
```

Restore the database with odacli irestore-database command.

Backup on NFS using an existing DB home:

```
[root@stdbydb1 ~]# odacli irestore-database -r /odabackup/backupreport_hun_20220318.json -u pest -ro STANDBY -t -dh e8a36f29-7fcf-49fc-8575-c599dc28949d --backupLocation /odabackup/db
```

Backup on Oracle Object Storage creating a new DB home:

```
[root@stdbydb1 ~]# odacli irestore-database -r backupreport_hun_20220318.json -u pest -on odabackups -ro STANDBY -t
```

Replace -t with -bp in the above commands if database was a non-TDE one.

Example:

```
[root@stdbydb1 ~]# odacli irestore-database -r /odabackup/backupreport_hun_20220318.json -u pest -ro STANDBY -t -dh e8a36f29-7fcf-49fc-8575-c599dc28949d --backupLocation /odabackup/db
Enter SYS user password:
Retype SYS user password:
Enter TDE wallet password:
{
  "jobId" : "6d36ebdf-2b31-4d19-a75f-5d997286ed9f",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "March 18, 2022 16:32:16",
  "resourceList" : [ ],
  "description" : "Database service recovery with db name: hun",
  "updatedTime" : "March 18, 2022 16:32:16"
}
```

Verify that the job completed successfully

```
[root@stbydb1]# odacli describe-job -i "0a35a4af-13bc-4a03-bfe6-ec4ae4e43dc6"
Job details
----------------------------------------------------------------
                     ID:  6d36ebdf-2b31-4d19-a75f-5d997286ed9f
            Description:  Database service recovery with db name: hun
                 Status:  Success
                Created:  March 18, 2022 4:32:16 PM CET
Task Name                            Start Time                          End Time                            Status
------------------------------------ ----------------------------------- ----------------------------------- ----------
Check if cluster ware is running     March 18, 2022 4:32:17 PM CET       March 18, 2022 4:32:17 PM CET       Success
…
Enable New Tablespace Encryption     March 18, 2022 4:49:22 PM CET       March 18, 2022 4:49:23 PM CET       Success
```

Verify that the database is in “CONFIGURED” status.

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

ID                                       DB Name    DB Type  DB Version           CDB        Class    Shape    Storage    Status       DbHomeID
---------------------------------------- ---------- -------- -------------------- ---------- -------- -------- ---------- ------------ ----------------------------------------
9cec6f9a-5256-48c0-8386-4bda7ee6b393     hun        RAC      19.14.0.0.220118     true       OLTP     odb2     ASM        CONFIGURED
e8a36f29-7fcf-49fc-8575-c599dc28949d
```

7. Configure Data Guard from the primary ODA’s first node

If source and target DBs were on bare metal then run **odacli configure-dataguard** to configure standby

Prerequisites:

a) Listener port(s) and port 7070 must be open to create Oracle Data Guard between two ODAs.

b) Configuring Data Guard would require 19.15 ODA version when either primary or standby database (or both) were configured on a DBSystem(s)

```
[root@ proddb1]# odacli configure-dataguard
Standby site address: stbydb1
BUI username for Standby site. If Multi-user Access is disabled on Standby site, enter 'oda-admin'; otherwise, enter the name of the user
who has irestored the Standby database (default: oda-admin):
BUI password for Standby site:
root@stbydb1's password:
Database name for Data Guard configuration: hun
Primary database SYS password:
********************************************************************************************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port: 1521
Standby database listener port: 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: buda_pest
Active Data Guard: disabled
Do you want to edit this Data Guard configuration? (Y/N, default:N): y
********************************************************************************************
Primary site network for Data Guard configuration [Public-network] (default: Public-network):
Standby site network for Data Guard configuration [Public-network] (default: Public-network):
Primary database listener port (default: 1521):
Standby database listener port (default: 1521):
Transport type [ASYNC, FASTSYNC, SYNC] (default: ASYNC):
Protection mode [MAX_PROTECTION, MAX_PERFORMANCE, MAX_AVAILABILITY] (default: MAX_PERFORMANCE):
Data Guard configuration name (default: buda_pest):
Enable Active Data Guard? (Y/N, default:N): n
Standby database's SYS password will be set to Primary database's after Data Guard configuration. Ignore warning and proceed with Data
Guard configuration? (Y/N, default:N): y
********************************************************************************************
Configure Data Guard buda_pest started
********************************************************************************************
```

Step 1: Validate Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db hun
Job ID: 1cdcc4d9-f869-49ed-90a7-651a0a76db03
Started March 18, 2022 17:02:17 PM CET
Validate create Data Guard configuration request
Finished March 18, 2022 17:02:21 PM CET
****
Step 2: Validate Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db hun
Job ID: c9dcb3fc-90d7-495e-860d-d3fdd421aad0
Started March 18, 2022 17:02:22 PM CET
Validate create Data Guard configuration request
Finished March 18, 2022 17:02:27 PM CET
****
Step 3: Download password file from Primary database (Primary site)
Description: Download orapwd file from Primary database
Started March 18, 2022 17:02:27 PM CET
Prepare orapwd file for Primary database hun
Finished March 18, 2022 17:02:32 PM CET
****
Step 4: Upload password file to Standby database (Standby site)
Description: Upload orapwd file to Standby database
Started March 18, 2022 17:02:32 PM CET
Write orapwd file to Standby database hun
Finished March 18, 2022 17:02:43 PM CET
****
Step 5: Configure Primary database (Primary site)
Description: DG Config service for db hun – ConfigurePrimary
Job ID: ed2e490d-f3e4-40b5-adee-ec5a31c6cdc6
Started March 18, 2022 17:02:44 PM CET
Configure host DNS on primary env
Configure Data Guard Tns on primary env
Enable Data Guard related Db parameters for primary env
Enable force logging and archivelog mode in primary env
Enable FlashBack
Configure network parameters for local listener on primary env
Restart listener on primary env
Create services for primary db
Finished March 18, 2022 17:05:46 PM CET
****
Step 6: Configure Standby database (Standby site)
Description: DG Config service for db hun – ConfigureStandby
Job ID: 989931fb-c7ec-4f36-9e8e-7cbe932af96c
Started March 18, 2022 17:05:47 PM CET
Configure Data Guard Tns on standby env
Configure host DNS on standby env
Clear Data Guard related Db parameters for standby env
Enable Data Guard related Db parameters for standby env
Enable force logging and archivelog mode in standby env
Populate standby database metadata
Configure network parameters for local listener on standby env
Reset Db sizing and hidden parameters for ODA best practice
Restart Listener on standby env
Create services for standby db
Finished March 18, 2022 17:07:27 PM CET
****
Step 7: Configure and enable Data Guard (Primary site)
Description: DG Config service for db hun – ConfigureDg
Job ID: 0616ad61-a6fe-4e33-b9a9-f0ea1698022f
Started March 18, 2022 17:07:28 PM CET
Config and enable Data Guard
Post check Data Guard configuration
Finished March 18, 2022 17:08:03 PM CET
****
Step 8: Enable Flashback (Standby site)
Description: DG Config service for db hun – EnableFlashback
Job ID: 1104e7ab-de51-4477-9a03-0cc37fc0431f
Started March 18, 2022 17:08:04 PM CET
Enable FlashBack
Finished March 18, 2022 17:11:55 PM CET
****
Step 9: Re-enable Data Guard (Primary site)
Description: DG Config service for db hun – ReenableDg
Job ID: 6aea76eb-e51a-4517-ae85-ba6b108804a4

Started March 18, 2022 17:11:56 PM CET
Re-enable Data Guard if inconsistent properties found
Post check Data Guard configuration
Finished March 18, 2022 17:12:53 PM CET
****
Step 10: Create Data Guard status (Primary site)
Description: DG Status operation for db hun – NewDgconfig
Job ID: df82b9d3-9a7e-4545-888f-29d678879870
Started March 18, 2022 17:12:53 PM CET
Create Data Guard status
Finished March 18, 2022 17:13:00 PM CET
****
Step 11: Create Data Guard status (Standby site)
Description: DG Status operation for db hun – NewDgconfig
Job ID: 9a70c3b8-5edb-406e-99e8-e03c44000d03
Started March 18, 2022 17:13:01 PM CET
Create Data Guard status
Finished March 18, 2022 17:13:08 PM CET
****
Configure Data Guard buda_pest completed
****

In the interactive CLI configuration steps, the parameters are as follows:

- Standby site address is IP address or host name of the standby host. Provide the fully qualified domain name and hostname if the primary and the standby systems are in the same domain and DNS is configured
- Select Oracle Data Guard protection modes to meet availability, performance and data protection requirements. Oracle Data Guard Protection Modes are Maximum Availability, Maximum Performance, and Maximum Protection. The log transport modes are ASYNC, SYNC, and FASTSYNC.
- You can select the following combinations of protection modes and transport types:

Protection Mode \ Transport Type	ASYNC	FASTSYNC**	SYNC
MAXPERFORMANCE	Y*	Y	Y
MAXAVAILABILITY	N	Y	Y*
MAXPROTECTION	N	N	Y*

* the table indicates the default supported pair and ** FASTSYNC mode is available only in Oracle Database 12.1 or later.

SWITCHOVER

1. Switchover operation has to be initiated from the primary side. “odacli list-dataguardstatus” can help you to verify on which ODA the database is running as primary. The command also provides the ID of the Data Guard configuration which is needed in the switchover and failover commands.

```
[root@proddb1]# odacli list-dataguardstatus
Updated about 2 second(s) ago
ID                                       Name                 Database Name      Role       Protection Mode    Apply Lag       Transport Lag   Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ --------------- --------------- --------------- ----
be217130-633b-4eef-a4b7-3192028b853c     buda_pest            hun                PRIMARY    MAX_PERFORMANCE    0 seconds       0 seconds       14.00 KByte/s   CONFIGURED
```

2. Initiate switchover. The command requires 2 parameters. One is the Data Guard configuration ID and the second one is the new primary’s database unqiue name.

```
[root@proddb1 ~]# odacli switchover-dataguard -i be217130-633b-4eef-a4b7-3192028b853c -u pest
Password for target database:
{
  "jobId" : "02ddfc45-da95-4f70-8823-bcd30ce3b738",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "March 18, 2022 17:24:11 PM CET",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - SwitchoverDg",
  "updatedTime" : "March 18, 2022 17:24:11 PM CET"
}
```

3. odacli describe-job or list-jobs can help us to monitor the status of the switchover operation

```
[root@proddb1 ~]# odacli describe-job -i "02ddfc45-da95-4f70-8823-bcd30ce3b738"
Job details
----------------------------------------------------------------
                     ID:  02ddfc45-da95-4f70-8823-bcd30ce3b738
            Description:  Dataguard operation for buda_pest – SwitchoverDg
                 Status:  Success
                Created:  March 18, 2022 5:24:11 PM CET
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck switchover DataGuard            March 18, 2022 5:24:12 PM CET       March 18, 2022 5:24:15 PM CET       Success
Switchover DataGuard                     March 18, 2022 5:24:15 PM CET       March 18, 2022 5:25:24 PM CET       Success
Postcheck switchover DataGuard           March 18, 2022 5:25:24 PM CET       March 18, 2022 5:26:19 PM CET       Success
Check if DataGuard config is updated     March 18, 2022 5:26:29 PM CET       March 18, 2022 5:26:39 PM CET       Success
```

4. Once the job completed verify the status of the Data Guard on both nodes.
   Please note that you might need to run list-dataguardstatus or describe-dataguardstatus commands 1 to 3 times to see the changes.

```
[root@proddb1 ~]# odacli describe-dataguardstatus -i be217130-633b-4eef-a4b7-3192028b853c
Updated about 2 minute(s) ago
Dataguard Status details
----------------------------------------------------------------
                     ID: be217130-633b-4eef-a4b7-3192028b853c
```

Name: buda_pest
Database Name: c0409b01-03da-4326-b268-29a48d8d617f
Role: STANDBY
Protection Mode: MAX_PERFORMANCE
Apply Lag: 0 seconds
Transport Lag: 0 seconds
Apply Rate: 1.35 MByte/s
Status: CONFIGURED
Updated Time: March 18, 2022 5:26:26 PM CET

[root@stbydb1 ~]# odacli describe-dataguardstatus -i be217130-633b-4eef-a4b7-3192028b853c

Updated about 5 minute(s) ago
Dataguard Status details

ID: be217130-633b-4eef-a4b7-3192028b853c
Name: buda_pest
Database Name: 9cec6f9a-5256-48c0-8386-4bda7ee6b393
Role: STANDBY ← not updated yet
Protection Mode: MAX_PERFORMANCE
Apply Lag: 0 seconds
Transport Lag: 0 seconds
Apply Rate: 2.00 KByte/s
Status: CONFIGURED
Updated Time: March 18, 2022 5:23:15 PM CET

Running the same command the second time:

[root@stbydb1 ~]# odacli describe-dataguardstatus -i be217130-633b-4eef-a4b7-3192028b853c
Updated about 34 second(s) ago
Dataguard Status details

ID: be217130-633b-4eef-a4b7-3192028b853c
Name: buda_pest
Database Name: 9cec6f9a-5256-48c0-8386-4bda7ee6b393
Role: PRIMARY ← role has been updated and reflects the right status
Protection Mode: MAX_PERFORMANCE
Apply Lag: 0 seconds
Transport Lag: 0 seconds
Apply Rate: 274.00 KByte/s
Status: CONFIGURED
Updated Time: March 18, 2022 5:29:16 PM CET

FAILOVER

1. Failover operation has to be initiated from the standby side. “odacli list-dataguardstatus” can help you to verify on which ODA the database is running as standby. The command also provides the ID of the Data Guard configuration which is needed in the switchover and failover commands.

```
[root@proddb1]# odacli list-dataguardstatus
Updated about 2 second(s) ago
ID                                       Name                 Database Name      Role       Protection Mode    Apply Lag      Transport Lag  Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ -------------- -------------- -------------- ----
be217130-633b-4eef-a4b7-3192028b853c     buda_pest            hun                STANDBY    MAX_PERFORMANCE    0 seconds      0 seconds      14.00 KByte/s  CONFIGURED
```

2. Initiate failover. The command requires 2 parameters. One is the Data Guard configuration ID and the second one is the new primary’s database unqiue name.

```
[root@proddb1 ~]# odacli failover-dataguard -i be217130-633b-4eef-a4b7-3192028b853c -u buda
Password for target database:
{
  "jobId" : "3dd42271-2919-4cae-a801-1a4d635c3120",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "March 18, 2022 17:31:12 PM CET",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - FailoverDg",
  "updatedTime" : "March 18, 2022 17:31:12 PM CET"
}
```

3. Monitor the job with odacli list-jobs or odacli describe-job

```
[root@proddb1 ~]# odacli describe-job -i "3dd42271-2919-4cae-a801-1a4d635c3120"
Job details
----------------------------------------------------------------
                     ID:  3dd42271-2919-4cae-a801-1a4d635c3120
            Description:  Dataguard operation for buda_pest – FailoverDg
                 Status:  Success
                Created:  March 18, 2022 5:31:12 PM CET
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck failover DataGuard              March 18, 2022 5:31:12 PM CET       March 18, 2022 5:31:13 PM CET       Success
Failover DataGuard                       March 18, 2022 5:31:13 PM CET       March 18, 2022 5:31:45 PM CET       Success
Postcheck DataGuard status               March 18, 2022 5:31:45 PM CET       March 18, 2022 5:31:46 PM CET       Success
Check if DataGuard config is updated     March 18, 2022 5:31:56 PM CET       March 18, 2022 5:32:06 PM CET       Success
```

4. Reinstate the former primary as standby. The command requires 2 parameters. One is the Data Guard configuration ID and the second one is the former primary’s database unqiue name.

```
[root@proddb1 ~]# odacli reinstate-dataguard -i be217130-633b-4eef-a4b7-3192028b853c -u pest
Password for target database:
{
  "jobId" : "c53d2d6f-a128-4b16-a894-25fc6e73493e",
  "status" : "Created",
```

```
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "March 18, 2022 17:33:24 PM CET",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - ReinstateDg",
  "updatedTime" : "March 18, 2022 17:33:24 PM CET"
}
```

5. Monitor the reinstate job's status

```
[root@proddb1 ~]# odacli describe-job -i "c53d2d6f-a128-4b16-a894-25fc6e73493e"
Job details
----------------------------------------------------------------
                     ID:  c53d2d6f-a128-4b16-a894-25fc6e73493e
            Description:  Dataguard operation for buda_pest – ReinstateDg
                 Status:  Success
                Created:  March 18, 2022 5:33:24 PM CET
                Message:
Task Name                            Start Time                          End Time                            Status
------------------------------------ ----------------------------------- ----------------------------------- ----------
Precheck reinstate DataGuard         March 18, 2022 5:33:24 PM CET       March 18, 2022 5:33:25 PM CET       Success
Reinstate DataGuard                  March 18, 2022 5:33:25 PM CET       March 18, 2022 5:35:07 PM CET       Success
Postcheck DataGuard status           March 18, 2022 5:35:07 PM CET       March 18, 2022 5:36:30 PM CET       Success
Check if DataGuard config is updated March 18, 2022 5:36:40 PM CET       March 18, 2022 5:36:50 PM CET       Success
```

6. Once the job completed verify the status of the Data Guard on both nodes.
   Please note that you might need to run list-dataguardstatus or describe-dataguardstatus commands 1 to 3 times to see the changes.

```
[root@stdbydb1 ~]# odacli describe-dataguardstatus -i be217130-633b-4eef-a4b7-3192028b853c
Updated about 34 second(s) ago
Dataguard Status details
----------------------------------------------------------------
                     ID: be217130-633b-4eef-a4b7-3192028b853c
                   Name: buda_pest
          Database Name: 9cec6f9a-5256-48c0-8386-4bda7ee6b393
                   Role: PRIMARY ← the status has not been updated yet
        Protection Mode: MAX_PERFORMANCE
            Apply Lag: 0 seconds
        Transport Lag: 0 seconds
          Apply Rate: 274.00 KByte/s
             Status: CONFIGURED
       Updated Time: March 18, 2022 5:29:16 PM CET

[root@stdbydb1 ~]#  odacli describe-dataguardstatus -i be217130-633b-4eef-a4b7-3192028b853c
Updated about 3 second(s) ago
Dataguard Status details
----------------------------------------------------------------
                     ID: be217130-633b-4eef-a4b7-3192028b853c
                   Name: buda_pest
          Database Name: 9cec6f9a-5256-48c0-8386-4bda7ee6b393
                   Role: STANDBY ← updated and correct status
        Protection Mode: MAX_PERFORMANCE
            Apply Lag: 0 seconds
        Transport Lag: 0 seconds
```

Apply Rate: 386.00 KByte/s
Status: CONFIGURED
Updated Time: March 18, 2022 5:37:35 PM CET

DECONFIGURE DATA GUARD

1. Deconfiguration has to be initiated from the primary side. “odacli list-dataguardstatus” can help you to verify on which ODA the database is running as standby. The command also provides the ID of the Data Guard configuration which is needed in the switchover and failover commands. Please note that the database will be deleted/dropped on the standby side.

```
[root@proddb1]# odacli list-dataguardstatus
Updated about 2 second(s) ago
ID                                       Name                 Database Name      Role       Protection Mode    Apply Lag       Transport Lag   Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ --------------- --------------- --------------- ----
be217130-633b-4eef-a4b7-3192028b853c     buda_pest            hun                PRIMARY    MAX_PERFORMANCE    0 seconds       0 seconds       14.00 KByte/s   CONFIGURED
```

2. Running Data Guard deconfiguration is an interactive process. The flow looks like below:

```
[root@proddb1 ~]# odacli deconfigure-dataguard -i be217130-633b-4eef-a4b7-3192028b853c
Standby site address: stbydb1
BUI username for Standby site. If Multi-user Access is disabled on Standby site, enter 'oda-admin'; otherwise, enter the name of the user who has irestored the Standby database (default: oda-admin):
BUI password for Standby site:
root@stbydb1's password:
Standby database will be deleted after Data Guard configuration is removed. Ignore warning and proceed with Data Guard deconfiguration? (Y/N): y
Deconfigure Dataguard Started
********************************************************************************************
Step 1: Deconfigure Data Guard (Primary site)
Description: Deconfigure DG service
Job ID: ce9e0871-6630-452f-bf3a-44262b0d461d
Started March 18, 2022 17:39:04 PM CET
Deconfigure Data Guard service
Cleanup broker resources
Finished March 18, 2022 17:40:49 PM CET
********************************************************************************************
Step 2: Delete Data Guard status (Primary site)
Description: DG Status operation for db  - UpdateDgconfig
Job ID: 0aa8cebf-4cb5-4444-8426-991bab48eb6e
Started March 18, 2022 17:40:49 PM CET
Finished March 18, 2022 17:40:49 PM CET
********************************************************************************************
Step 3: Delete Data Guard status (Standby site)
Description: DG Status operation for db  - UpdateDgconfig
Job ID: adcd8b6d-e514-45ee-8eb9-998e4968ef97
Started March 18, 2022 17:40:50 PM CET
Update Data Guard status
Finished March 18, 2022 17:40:51 PM CET
********************************************************************************************
Step 4: Delete Standby database (Standby site)
Description: Database service deletion with db name: hun with id : 9cec6f9a-5256-48c0-8386-4bda7ee6b393
Job ID: 9fd067c3-9a51-4db9-88d2-105e673143c7
Started March 18, 2022 17:40:54 PM CET
Validate db 9cec6f9a-5256-48c0-8386-4bda7ee6b393 for deletion
Database Deletion By RHP
Unregister Db From Cluster
Kill Pmon Process
Database Files Deletion
TDE Wallet deletion
Finished March 18, 2022 17:43:16 PM CET
********************************************************************************************
Data Guard configuration is removed
```

ADDITIONAL NETWORK

According to MAA best practices, it is recommended to use a dedicated network interface for Data Guard related traffic.
odacli configure-dataguard supports that out of the box. By default the configuration uses Public-network, but a different network can be assigned to it easily.
If the database was running on bare metal then a new interface would need to be configured with "Dataguard" type and attached to the database:

1. Create a new network on the desired interface

   # odacli create-network -m <network_name> -n <interface_name> -p <ip0, ip1> -w Dataguard -no-d -s <subnet_mask> -g <gate_ip> -vs <vipname0:nodenumber0:vip0,vipname1:nodenumber1:vip1> -sn <scan_name> -sip <scanip0,scanip1> (optional: -t VLAN -v <vlan_id>)
   Example:
   # odacli create-network -m DataGuard -n bond1 -p "0:2.2.2.2,1:2.2.2.3" -w Dataguard -no-d -s 255.255.255.0 -g 2.2.2.1 -vs "dg-vip1:0:2.2.2.4,dg-vip2:1:2.2.2.5" -sn dg-scan -sip 2.2.2.6
2. Attach the network to the database

   # odacli modify-database -in <dbname> -an <network name>

   Example:
   # odacli modify-database -in testdb -an DataGuard

   Network name can be verified via "odacli list-networks" command.

If the database was running inside a DBSystem then a new virtual interface would need to be configured with "Dataguard" type and attached to the database:

1. Create a new vnetwork on the desired interface on the bare metal host

   # odacli create-vnetwork -n <vnetwork_name> -t <bridged|bridgedVLAN> -br <bridge_name> -gw <gateway> -if <interface_name> -ip

   Example:
   # odacli create-vnetwork -n DataGuard -t bridged -br DataGuard -gw 2.2.2.1 -if btbond5 -ip "2.2.2.7,2.2.2.8" -nm "255.255.255.0" -u

2. Assign the new vnetwork to the DBSystem as a "Dataguard" type network on the bare metal host

   # odacli modify-dbsystem -n <dbsystem_name> -avn <vnetwork_name> -gw <gateway> -ip <ip0,ip1> -nm <netmask> -sn <scan_name> -sip <scanip0,scanip1> -vips <vipname0:nodenumber0:vip0,vipname1:nodenumber1:vip1> -vt <network type>
   Example:
   # odacli modify-dbsystem -n scaoda818c5 -avn DataGuard -gw "2.2.2.1" -ip "2.2.2.11,2.2.2.12" -nm "255.255.255.0" -sn dg-scan -sip "2.2.2.15,2.2.2.16" -vips "dg-vip1:0:2.2.2.13,dg-vip2:1:2.2.2.14" -vt dataguard

3. Attach the network to the database on the DBSystem host

   # odacli modify-database -in <dbname> -an <network name>

   Example:
   # odacli modify-database -in testdb -an DataGuard

Note: The above steps have to be executed on the primary and the standby ODA as well regardless the DB is running on BM or inside a DBSytem.
Finally provide the network name for the Data Guard configuration in odacli configure-dataguard.
At *"Do you want to edit this Data Guard configuration?" you have to choose 'y' to change the Data Guard network.*

Example:
Do you want to edit this Data Guard configuration? (Y/N, default:N): y
…
Primary site network for Data Guard configuration [Public-network] (default: Public-network): ***DataGuard***
Standby site network for Data Guard configuration [Public-network] (default: Public-network): ***DataGuard***

APPENDIX B: REGISTERING MANUALLY CONFIGURED DATA GUARD IN DCS

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.

	Primary Oracle Database Appliance		Standby Oracle Database Appliance	
Host Names	proddb1	proddb2	stbydb1	stbydb2
Database Name	hun		hun	
Database Unique Name	buda		pest	

The goal is to register manually configured Data Guard in DCS to make all Data Guard related lifecycle operations available using odacli commands. The feature is available on 19.15 and onwards.

1. Verify that the database is registered on the primary and the standby

[root@proddb1 ~]# odacli list-databases

ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID

---

ebefcfa2-0315-4771-9881-373294a6b626 hun RAC 19.15.0.0.220419 true OLTP odb1 ASM CONFIGURED 14402597-639a-4e87-b655-aeae36cfa3a5

[root@stdbydb1 ~]# odacli list-databases

ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID

---

ebefcfa2-0315-4771-9881-373294a6b626 hun RAC 19.15.0.0.220419 true OLTP odb1 ASM CONFIGURED 575fca61-abbc-47ed-9530-37ad7ec5caa0

2. Identify the home from where the database is running on the primary

[root@proddb1 ~]# odacli list-dbhomes

ID Name DB Version Home Location Status

14402597-639a-4e87-b655-aeae36cfa3a5 OraDB19000_home1 19.15.0.0.220419 /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 CONFIGURED

3. Verify the status of the Data Guard on the primary. Status should be healthy for the registration.

```
[oracle@proddb1  ~]$ export ORACLE_HOME=/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1
[oracle@proddb1  ~]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@proddb1  ~]$ dgmgrl sys/WElcome_12##@pest
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri May 6 13:41:52 2022
Version 19.15.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Welcome to DGMGRL, type "help" for information.
Connected to "pest"
Connected as SYSDBA.
DGMGRL> show configuration

Configuration - buda_pest

  Protection Mode: MaxPerformance
  Members:
  buda - Primary database
    pest - Physical standby database

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 38 seconds ago)

DGMGRL> validate database pest

  Database Role:     Physical standby database
  Primary Database:  buda

  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)

  Managed by Clusterware:
    buda:  YES
    pest:  YES

DGMGRL> exit
```

Notes:

a. If Data Guard configuration name was other than <primary's db_unique_name>_<standby's db_unique_name> then you could consider changing it to that.
In case multiple Data Guard configurations had the same name in dgmgrl you could register the first one only in DCS and the next registration would fail as it expects unique Data Guard configuration names.

Before renaming

```
DGMGRL> show configuration

Configuration - dgconfig

  Protection Mode: MaxPerformance
  Members:
  buda  - Primary database
    pest - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS   (status updated 6 seconds ago)
```

After renaming

```
DGMGRL> EDIT CONFIGURATION RENAME TO buda_pest;
Succeeded.
DGMGRL> show configuration

Configuration - buda_pest
```

```
  Protection Mode: MaxPerformance
  Members:
  buda  - Primary database
    pest - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS   (status updated 37 seconds ago)
```

b. odacli register-database expects VIPs in the tnsnames.ora and not the SCAN

$ORACLE_HOME/network/admin/tnsnames.ora should look like the following

```
BUDA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = proddb1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = proddb2-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = buda.domain.com)
    )
  )

PEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = stdbydb1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = stdbydb2-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pest.domain.com)
    )
  )
```

4. Register the Data Guard in DCS on the first node of the primary. If primary was running on a DBSystem then the first node of the DBSystem should be used for that.

```
[root@proddb1 ~]# odacli register-dataguard
Standby site address: stdbydb1
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
root@stdbydb1 's password:
Database name for Data Guard configuration: hun
Primary database SYS password:
********************************************************************************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port (TCP): 1521
Standby database listener port (TCP): 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: buda_pest
Does the above Data Guard configuration match your actual configuration? (Y/N, default:N):
********************************************************************************
Primary site network for Data Guard configuration [Public-network] (default: Public-network):
Standby site network for Data Guard configuration [Public-network] (default: Public-network):
Primary database listener port (TCP) (default: 1521):
Standby database listener port (TCP) (default: 1521):
Transport type [ASYNC, FASTSYNC, SYNC] (default: ASYNC):
Protection mode [MAX_PROTECTION, MAX_PERFORMANCE, MAX_AVAILABILITY] (default: MAX_PERFORMANCE):
Data Guard configuration name (default: buda_pest): buda_pest
********************************************************************************
Register Data Guard buda_pest started
********************************************************************************
Step 1: Validate register Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db hun
Job ID: fc5436d2-67db-4d4c-927c-9053c56dc510
Started May 06, 2022 13:49:33 PM GMT
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate if database is configured with Data Guard already
```

```
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Precheck switchover DataGuard
Validate if input matches DGMGRL output
Validate if flashback enabled
Finished May 06, 2022 13:49:40 PM GMT
********************************************************************************************
Step 2: Validate register Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db hun
Job ID: 54224175-eb0a-4e07-a84d-b758692dc55c
Started May 06, 2022 13:49:42 PM GMT
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate if database is configured with Data Guard already
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Validate if input matches DGMGRL output
Validate if flashback enabled
Finished May 06, 2022 13:49:46 PM GMT
********************************************************************************************
Step 3: Create Data Guard status (Primary site)
Description: DG Status operation for db hun - RegisterDg
Job ID: c6dcec88-2e21-4bc0-a243-6ab61885be88
Started May 06, 2022 13:49:47 PM GMT
Create Data Guard status
Finished May 06, 2022 13:49:53 PM GMT
********************************************************************************************
Step 4: Create Data Guard status (Standby site)
Description: DG Status operation for db hun - RegisterDg
Job ID: 44f312ad-97b0-4eff-8d47-7134433011c5
Started May 06, 2022 13:49:54 PM GMT
Create Data Guard status
Finished May 06, 2022 13:50:01 PM GMT
********************************************************************************************
Register Data Guard buda_pest completed
********************************************************************************************
```

5. Verify the registration

```
[root@proddb1 ~]# odacli list-dataguardstatus
Updated about 7 minute(s) ago
ID                                       Name                 Database Name      Role      Protection Mode    Apply Lag      Transport Lag  Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ -------------- -------------- --------------
cd86f70d-31d5-4798-8abf-a8148ec2e389     buda_pest            hun                PRIMARY    MAX_PERFORMANCE    0 seconds      0
seconds      5.00 KByte/s    CONFIGURED

[root@proddb2 ~]# odacli list-dataguardstatus
Updated about 8 minute(s) ago
ID                                       Name                 Database Name      Role      Protection Mode    Apply Lag      Transport Lag  Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ -------------- -------------- --------------
cd86f70d-31d5-4798-8abf-a8148ec2e389     buda_pest            hun                PRIMARY    MAX_PERFORMANCE    0 seconds      0
seconds      5.00 KByte/s    CONFIGURED

[root@stdbydb1  ~]# odacli list-dataguardstatus
Updated about 8 minute(s) ago
ID                                       Name                 Database Name      Role      Protection Mode    Apply Lag      Transport Lag  Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ -------------- -------------- --------------
cd86f70d-31d5-4798-8abf-a8148ec2e389     buda_pest            hun                STANDBY    MAX_PERFORMANCE    0 seconds      0
seconds      5.00 KByte/s    CONFIGURED

[root@stdbydb2 ~]# odacli list-dataguardstatus
Updated about 8 minute(s) ago
ID                                       Name                 Database Name      Role      Protection Mode    Apply Lag      Transport Lag  Apply Rate
Status
---------------------------------------- -------------------- ------------------ ---------- ------------------ -------------- -------------- --------------
cd86f70d-31d5-4798-8abf-a8148ec2e389     buda_pest            hun                STANDBY    MAX_PERFORMANCE    0 seconds      0
seconds      5.00 KByte/s    CONFIGURED
```

6. It is recommended to perform switchover/failover/reinstate tests as well

```
[root@proddb1 ~]# odacli switchover-dataguard -u pest -i cd86f70d-31d5-4798-8abf-a8148ec2e389
Password for target database:
{
  "jobId" : "2821ca72-eb6e-462f-8a7b-5f976a401673",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 06, 2022 14:01:31 PM GMT",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - SwitchoverDg",
  "updatedTime" : "May 06, 2022 14:01:31 PM GMT"
}

[root@proddb1 ~]# odacli describe-job -i "2821ca72-eb6e-462f-8a7b-5f976a401673"
Job details
----------------------------------------------------------------
                     ID:  2821ca72-eb6e-462f-8a7b-5f976a401673
            Description:  Dataguard operation for buda_pest - SwitchoverDg
                 Status:  Success
                Created:  May 6, 2022 2:01:31 PM GMT
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck switchover DataGuard            May 6, 2022 2:01:31 PM GMT          May 6, 2022 2:01:34 PM GMT          Success
Switchover DataGuard                     May 6, 2022 2:01:34 PM GMT          May 6, 2022 2:02:53 PM GMT          Success
Postcheck switchover DataGuard           May 6, 2022 2:02:53 PM GMT          May 6, 2022 2:02:54 PM GMT          Success
Check if DataGuard config is updated     May 6, 2022 2:04:14 PM GMT          May 6, 2022 2:04:24 PM GMT          Success

[root@stdbydb1  ~]# odacli switchover-dataguard -u buda -i cd86f70d-31d5-4798-8abf-a8148ec2e389
Password for target database:
{
  "jobId" : "7d7ef3f3-f48b-449f-a2b8-9a5de0882ff3",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 06, 2022 14:06:28 PM GMT",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - SwitchoverDg",
  "updatedTime" : "May 06, 2022 14:06:28 PM GMT"
}

[root@stdbydb1  ~]# odacli describe-job -i "7d7ef3f3-f48b-449f-a2b8-9a5de0882ff3"
Job details
----------------------------------------------------------------
                     ID:  7d7ef3f3-f48b-449f-a2b8-9a5de0882ff3
            Description:  Dataguard operation for buda_pest - SwitchoverDg
                 Status:  Success
                Created:  May 6, 2022 2:06:28 PM GMT
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck switchover DataGuard            May 6, 2022 2:06:28 PM GMT          May 6, 2022 2:06:31 PM GMT          Success
Switchover DataGuard                     May 6, 2022 2:06:31 PM GMT          May 6, 2022 2:07:36 PM GMT          Success
Postcheck switchover DataGuard           May 6, 2022 2:07:36 PM GMT          May 6, 2022 2:07:37 PM GMT          Success
Check if DataGuard config is updated     May 6, 2022 2:08:37 PM GMT          May 6, 2022 2:08:47 PM GMT          Success

[root@stdbydb1  ~]# odacli list-dataguardstatus
Updated about 19 minute(s) ago
ID                                       Name                            Database Name       Role       Protection Mode     Apply Lag       Transport Lag   Apply Rate      Status
---------------------------------------- ------------------------------- ------------------- ---------- ------------------- --------------- --------------- --------------- ----------
cd86f70d-31d5-4798-8abf-a8148ec2e389     buda_pest                       hun                 STANDBY    MAX_PERFORMANCE     0 seconds       0 seconds       3.29 MByte/s    CONFIGURED

[root@stdbydb1  ~]# odacli failover-dataguard -u pest -i cd86f70d-31d5-4798-8abf-a8148ec2e389
Password for target database:
{
  "jobId" : "e6cd3092-94fb-4fbd-9dce-cdf9aad4638a",
  "status" : "Created",
```

```
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 06, 2022 14:20:46 PM GMT",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - FailoverDg",
  "updatedTime" : "May 06, 2022 14:20:46 PM GMT"
}
```

[root@stdbydb1 ~]# odacli describe-job -i e6cd3092-94fb-4fbd-9dce-cdf9aad4638a

```
Job details
----------------------------------------------------------------
                     ID:  e6cd3092-94fb-4fbd-9dce-cdf9aad4638a
            Description:  Dataguard operation for buda_pest - FailoverDg
                 Status:  Success
                Created:  May 6, 2022 2:20:46 PM GMT
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck failover DataGuard              May 6, 2022 2:20:46 PM GMT          May 6, 2022 2:20:47 PM GMT          Success
Failover DataGuard                       May 6, 2022 2:20:47 PM GMT          May 6, 2022 2:21:08 PM GMT          Success
Postcheck DataGuard status               May 6, 2022 2:21:08 PM GMT          May 6, 2022 2:21:09 PM GMT          Success
Check if DataGuard config is updated     May 6, 2022 2:21:19 PM GMT          May 6, 2022 2:21:29 PM GMT          Success
```

[root@stdbydb1 ~]# odacli reinstate-dataguard -u buda -i cd86f70d-31d5-4798-8abf-a8148ec2e389

```
Password for target database:
{
  "jobId" : "cb82e0ea-558d-4eb6-ad7a-82c373da7504",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 06, 2022 14:26:50 PM GMT",
  "resourceList" : [ ],
  "description" : "Dataguard operation for buda_pest - ReinstateDg",
  "updatedTime" : "May 06, 2022 14:26:50 PM GMT"
}
```

[root@stdbydb1 ~]# odacli describe-job -i "cb82e0ea-558d-4eb6-ad7a-82c373da7504"

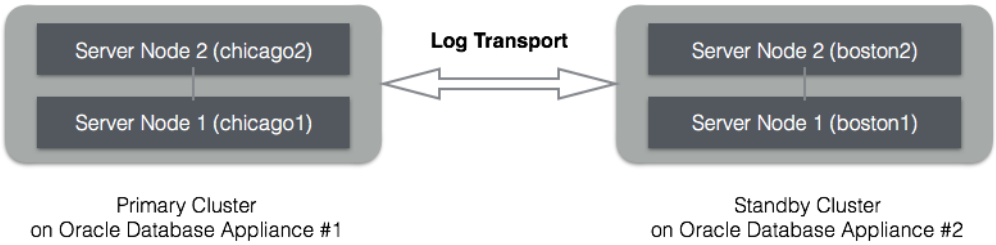
```
Job details
----------------------------------------------------------------
                     ID:  cb82e0ea-558d-4eb6-ad7a-82c373da7504
            Description:  Dataguard operation for buda_pest - ReinstateDg
                 Status:  Success
                Created:  May 6, 2022 2:26:50 PM GMT
                Message:

Task Name                                Start Time                          End Time                            Status
---------------------------------------- ----------------------------------- ----------------------------------- ----------
Precheck reinstate DataGuard             May 6, 2022 2:26:50 PM GMT          May 6, 2022 2:26:51 PM GMT          Success
Reinstate DataGuard                      May 6, 2022 2:26:51 PM GMT          May 6, 2022 2:28:36 PM GMT          Success
Postcheck DataGuard status               May 6, 2022 2:28:36 PM GMT          May 6, 2022 2:28:37 PM GMT          Success
Check if DataGuard config is updated     May 6, 2022 2:28:47 PM GMT          May 6, 2022 2:28:57 PM GMT          Success
```

APPENDIX C: MANUAL DATA GUARD CONFIGURATION 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.



	Primary Oracle Database Appliance		Standby Oracle Database Appliance	
Appliance Name	appliance#1		appliance#2	
Host Names	proddb1	proddb1	stbydb1	stbydb2
Database Name	chicago		chicago	
Database Unique Name	chicago		boston	
Instance Name	chicago1	chicago2	boston1	boston2
SCAN Name and IPs	proddb-scan (10.1.27.2, 10.1.27.3)		stbydb-scan (10.1.27.4, 10.1.27.5)	
Grid Infrastructure Software Installation	/u01/app/19.14.0.0/grid		/u01/app/19.14.0.0/grid	
Oracle Database Software Installation	/u01/app/odaorahome/oracle/product/19.0.0.0/db_home1		/u01/app/odaorahome/oracle/product/19.0.0.0/db_home1	
Database storage	ASM		ASM	
ARCHIVELOG mode	Yes		Yes	
FORCE LOGGING mode	Yes		Yes	

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

```
[root@stbydb1]# odacli create-dbhome -v 19.14.0.0.220118
```

The database home version on the standby must be identical to pimary'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 = proddb-scan)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = chicago.oracle.com)
    )
  )
```

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

Standby

```
chicago =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = proddb-scan)(PORT = 1521))
  (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = chicago.oracle.com)
  )
 )

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

Parameters to be modified on the Standby as compared to the Primary

Primary	Standby
*.cluster_database=TRUE chicago2.instance_number=2 chicago1.instance_number=1 chicago2.thread=2 chicago1.thread=1 chicago2.undo_tablespace='UNDOTBS2' chicago1.undo_tablespace='UNDOTBS1' *.db_block_checking=FULL *.db_block_checksum=FULL *.db_lost_write_protect=TYPICAL ……. *.db_unique_name=chicago *.listener_networks='((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<primary node0's vip>)(PORT=1521)))))','((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<primary node1's vip>) (PORT=1521)))))','((NAME=net1)(REMOTE_LISTENER=<primary's scan name>:1521))' *.LOG_FILE_NAME_CONVERT='+REDO/BOSTON/','+REDO/CHICAGO/' *.DB_FILE_NAME_CONVERT='+DATA/BOSTON/','+DATA/CHICAGO/' *.log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) MAX_FAILURE=1 REOPEN=5 DB_UNIQUE_NAME=chicago ALTERNATE=log_archive_dest_10' *.log_archive_dest_10='LOCATION=+DATA/db19c/arc10 VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=chicago ALTERNATE=log_archive_dest_1' <u>add the following parameters</u> *.audit_file_dest='/u01/app/oracle/admin/chicago/adump' *.fal_server='boston' *.remote_login_passwordfile='exclusive'	*.cluster_database=TRUE boston2.instance_number=2 boston1.instance_number=1 boston2.thread=2 boston1.thread=1 boston2.undo_tablespace='UNDOTBS2' boston1.undo_tablespace='UNDOTBS1' *.db_block_checking=FULL *.db_block_checksum=FULL *.db_lost_write_protect=TYPICAL ……. *.db_unique_name=boston *.listener_networks='((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<standby node0's vip>) (PORT=1521)))))','((NAME=net1)(LOCAL_LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<standby node1's vip>)(PORT=1521)))))','((NAME=net1)(REMOTE_LISTENER=<standby's scan name>:1521))' *.LOG_FILE_NAME_CONVERT='+REDO/CHICAGO/','+REDO/BOSTON/' *.DB_FILE_NAME_CONVERT='+DATA/CHICAGO/','+DATA/BOSTON/' *.log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) MAX_FAILURE=1 REOPEN=5 DB_UNIQUE_NAME=boston ALTERNATE=log_archive_dest_10' *.log_archive_dest_10='LOCATION=+DATA/db19c/arc10 VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston ALTERNATE=log_archive_dest_1' <u>add the following parameters</u> *.audit_file_dest='/u01/app/oracle/admin/boston/adump' *.fal_server='chicago' *.remote_login_passwordfile='exclusive'

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

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.

Databases use listener_networks instead of local_listener and remote_listener parameters starting from ODA 19.6 on bare metal platform.

12. Create storage structures for the database on the standby.

[root@stbydb1]$ # odacli create-dbstorage -n chicago -u boston

```
{
 "jobId" : "054dac68-9efe-4f0d-a027-5515d46ada8a",
 "status" : "Created",
 "message" : null,
 "reports" : [ ],
 "createTimestamp" : "October 18, 2021 14:14:11 PM CEST",
 "resourceList" : [ ],
 "description" : "Database storage service creation with db name: chicago",
 "updatedTime" : "October 18, 2021 14:14:11 PM CEST"
}
```

[root@stbydb1]# odacli describe-job -i "054dac68-9efe-4f0d-a027-5515d46ada8a"

```
Job details
----------------------------------------------------------------
                     ID:  054dac68-9efe-4f0d-a027-5515d46ada8a
            Description:  Database storage service creation with db name: chicago
                 Status:  Success
```

13. 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.386.1086365117

[grid@proddb1 ~]$ asmcmd
ASMCMD> pwcopy +DATA/CHICAGO/PASSWORD/pwdchicago.386.1086365117 /tmp/pwdboston
copying +DATA/CHICAGO/PASSWORD/pwdchicago.386.1086365117 -> /tmp/pwdboston
[grid@proddb1]$ scp /tmp/pwdboston oracle@stbydb1:/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston

14. 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.oracle.com:/tmp/boston.pfile
[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stbydb1]$ cp /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston
/u01/app/odaorahome/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 19-OCT-21
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=483 instance=boston1 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: 00:00:02
output file name=+FLASH/BOSTON/CONTROLFILE/current.256.1086380745
Finished restore at 19-OCT-21
```

15. 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= '+FLASH/DB19CSTBY/CONTROLFILE/current.256.1086380745'

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

16. 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/odaorahome/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> create spfile='+DATA/BOSTON/PARAMETERFILE/spfileboston' from pfile='/tmp/boston.pfile';
SQL> !echo "spfile='+DATA/BOSTON/PARAMETERFILE/spfileboston'" >
/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/initboston1.ora
SQL> !echo "spfile='+DATA/BOSTON/PARAMETERFILE/spfileboston'" >
/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/initboston2.ora
SQL> startup mount force;
```

17. 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;
```

18. 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;

TABLESPACE_NAME                SIZE_IN_GB
------------------------------ ----------
UNDOTBS1                        .102539063
SYSTEM                         .947265625
SYSAUX                          .91796875
UNDOTBS2                        .024414063
USERS                          .004882813
```

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;
RMAM> recover database from service chicago;
RMAN> backup spfile;
```

19. 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;
```

20. 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;
```

21. 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;
```

22. Register the standby database with Clusterware

```
[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/odaorahome/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/odaorahome/oracle/product/19.0.0.0/dbhome_1 -
dbtype SINGLE -instance boston1 -node stbydb1 -dbname chicago -diskgroup 'DATA,RECO,FLASH' -role physical_standby -
spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -startoption mount -acfspath
'/u01/app/odaorahome,/u01/app/odaorabase0,/u01/app/odaorabase1'
```

RAC example

```
[oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 -
dbtype RAC -dbname chicago -diskgroup 'DATA,RECO,FLASH' -role physical_standby -spfile
'+DATA/BOSTON/PARAMETERFILE/spfileboston' -startoption mount -acfspath
'/u01/app/odaorahome,/u01/app/odaorabase0,/u01/app/odaorabase1'
[oracle@stbydb1]$ srvctl add instance -db boston -instance boston1 -node stbydb1
[oracle@stbydb1]$ srvctl add instance -db boston -instance boston2 -node stbydb2
```

23. Copy the password file to ASM and verify if passwordfile pointed to ASM

```
[grid@stbydb1 ~]$ asmcmd
ASMCMD>  pwcopy /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston
+DATA/DB19CSTBY/PASSWORDFILE/pwdboston --dbuniquename boston
copying /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston ->
+DATA/DB19CSTBY/PASSWORDFILE/pwdboston

[oracle@stbydb1]$ export ORACLE_HOME=/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1
[oracle@stbydb1]$ export ORACLE_SID=boston1
[oracle@stbydb1]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@stbydb1]$ srvctl config database -db boston|grep Password
```

24. Set the parameters and create the Broker configuration.

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]$ sqlplus / as sysdba
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;
[oracle@stbydb1]$ srvctl stop database -db boston
[oracle@stbydb1]$ srvctl start database -db boston -startoption mount
[oracle@stbydb1]$ sqlplus 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;
```

Wait 1 min
[oracle@stbydb1]$ dgmgrl sys/welcome1@chicago
CREATE CONFIGURATION dgconfig AS PRIMARY DATABASE IS CHICAGO CONNECT IDENTIFIER IS CHICAGO;
ADD DATABASE BOSTON AS CONNECT IDENTIFIER IS BOSTON ;
ENABLE CONFIGURATION

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. Registration

[oracle@stbydb1]$ dgmgrl sys/welcome1@boston as sysdba
DGMGRL> edit database 'boston' set state='apply-off';
Succeeded.
DGMGRL> sql 'ALTER DATABASE OPEN READ ONLY';
Succeeded.

[oracle@stbydb1]# odacli list-databases
DCS-10032:Resource database is not found.

[oracle@stbydb1]# odacli register-database -c OLTP -s odb2 -sn boston.oracle.com -nn Public-network -t RAC
Job details
--
ID: 841f99e0-a66f-4b23-b753-b04f992a6c33
Description: Discover Components : db

[oracle@stbydb1]# odacli describe-job -i 841f99e0-a66f-4b23-b753-b04f992a6c33

Job details
--
ID: 9947df75-e9f4-4a42-bcd7-ec23561a2f3f
Description: Database service registration with db service name: zug.us.oracle.com
Status: Success
Created: February 18, 2022 12:52:04 PM CET
Message:

Task Name Start Time End Time Status
---------- ---------- ---------- ----------
Validate Hugepages For Register DB February 18, 2022 12:52:05 PM CET February 18, 2022 12:52:05 PM CET Success
Enable OMF parameters February 18, 2022 12:52:06 PM CET February 18, 2022 12:52:07 PM CET Success
Setting db character set February 18, 2022 12:52:07 PM CET February 18, 2022 12:52:07 PM CET Success
Move Spfile to right location February 18, 2022 12:52:07 PM CET February 18, 2022 12:52:15 PM CET Success
Enable DbSizing Template February 18, 2022 12:52:15 PM CET February 18, 2022 12:53:26 PM CET Success
Running DataPatch February 18, 2022 12:53:26 PM CET February 18, 2022 12:53:28 PM CET Success
Reset Associated Networks for Databse February 18, 2022 12:53:29 PM CET February 18, 2022 12:53:33 PM CET Success
Reset Associated Networks February 18, 2022 12:53:33 PM CET February 18, 2022 12:53:33 PM CET Success

[oracle@stbydb1]# odacli list-databases

ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID
---------- ---------- -------- ---------- ---------- -------- -------- ---------- ---------- ----------
9139ea53-449d-413a-841b-b157c084f3e0 bikazug RAC 19.14.0.0.220118 false OLTP odb2 ASM CONFIGURED 2afd69ed-f2cd-4345-9860-480f9e21f3ad

[oracle@stbydb1]# odacli describe-database -i fbc4a32e-fec4-403d-b7b8-b08a3c01ab46

Database details

```
----------------------------------------------------------------
                     ID: 9139ea53-449d-413a-841b-b157c084f3e0
            Description: chicago
                DB Name: chicago
            DB Version: 19.14.0.0.220118
                DB Type: RAC
                DB Role: STANDBY
     DB Target Node Name:
            DB Edition: EE
                  DBID: 1128302500
  Instance Only Database: false
                    CDB: false
               PDB Name:
     PDB Admin User Name:
          SEHA Enabled: false
                 Class: OLTP
                 Shape: odb2
               Storage: ASM
         DB Redundancy: MIRROR
          CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
              Language: AMERICAN
             Territory: AMERICA
               Home ID: 2afd69ed-f2cd-4345-9860-480f9e21f3ad
        Console Enabled: false
  TDE Wallet Management:
           TDE Enabled: false
      Level 0 Backup Day:
      AutoBackup Enabled: true
               Created: February 18, 2022 12:52:02 PM CET
        DB Domain Name:
    Associated Networks: Public-network
          CPU Pool Name:
```

29. Finally enable log shipping again and bounce the standby database.

[oracle@stbydb1]$ dgmgrl sys/welcome1@boston as sysdba
DGMGRL> edit database 'boston' set state='apply-on';

Succeeded.

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

30. Switchover tests

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

DGMGRL> connect sys/welcome1@chicago
DGMGRL> switchover to chicago;

31. Failover tests

connect to standby before failover:

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

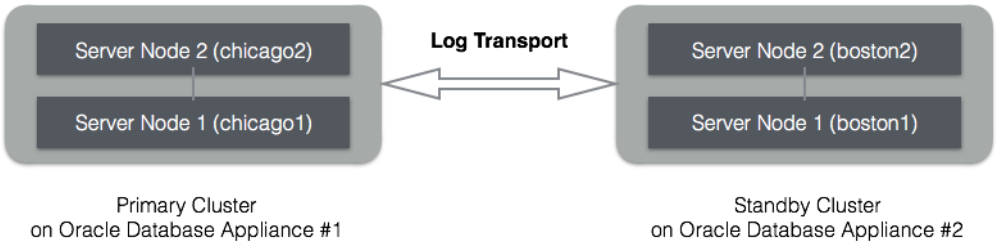
connect to former primary before failover:

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

APPENDIX D: MANUAL DATA GUARD CONFIGURATION 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.



	Primary Oracle Database Appliance		Standby Oracle Database Appliance	
Appliance Name	appliance#1		appliance#2	
Host Names	proddb1	proddb1	stbydb1	stbydb2
Database Name	chicago		chicago	
Database Unique Name	chicago		boston	
Instance Name	chicago1	chicago2	boston1	boston2
SCAN Name and IPs	proddb-scan (10.1.27.2, 10.1.27.3)		stbydb-scan (10.1.27.4, 10.1.27.5)	
Grid Infrastructure Software Installation	/u01/app/19.0.0.0/grid		/u01/app/19.0.0.0/grid	
Oracle Database Software Installation	/u01/app/oracle/product/12.1.0.2/db_home1		/u01/app/oracle/product/12.1.0.2/db_home 1	
Database storage	ACFS		ACFS	
ARCHIVELOG mode	Yes		Yes	
FORCE LOGGING mode	Yes		Yes	

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

The database home version on the standby must be identical to pimary'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

Primary	Standby
*.cluster_database=TRUE chicago2.instance_number=2 chicago1.instance_number=1 chicago2.thread=2 chicago1.thread=1 chicago2.undo_tablespace='UNDOTBS2' chicago1.undo_tablespace='UNDOTBS1' *.db_block_checking=FULL *.db_block_checksum=FULL *.db_lost_write_protect=TYPICAL ……. *.db_unique_name=Chicago *.remote_listener='proddb-scan:1521' *.DB_FILE_NAME_CONVERT='boston','chicago','BOSTON','CHICAGO' *.LOG_FILE_NAME_CONVERT='boston','chicago','BOSTON','CHICAGO' *.LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=chicago' add the following parameters *.audit_file_dest='/u01/app/oracle/admin/chicago/adump' *.fal_server='boston' *.remote_login_passwordfile='exclusive' add only on X5-2, X7-2 HA, X8-2 HA with HDDs *._cluster_flash_cache_slave_file='' *.db_flash_cache_file='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/chicago/flash1'	*.cluster_database=TRUE boston2.instance_number=2 boston1.instance_number=1 boston2.thread=2 boston1.thread=1 boston2.undo_tablespace='UNDOTBS2' boston1.undo_tablespace='UNDOTBS1' *.db_block_checking=FULL *.db_block_checksum=FULL *.db_lost_write_protect=TYPICAL ……. *.db_unique_name=boston *.remote_listener='stbydb-scan:1521' *.DB_FILE_NAME_CONVERT='chicago','boston','CHICAGO','BOSTON' *.LOG_FILE_NAME_CONVERT='chicago','boston','CHICAGO','BOSTON' *.LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston' add the following parameters *.audit_file_dest='/u01/app/oracle/admin/boston/adump' *.fal_server='chicago' *.remote_login_passwordfile='exclusive' add only on X5-2,X7-2 HA, X8-2 HA with HDDs, *._cluster_flash_cache_slave_file='' *.db_flash_cache_file='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/boston/flash1'

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).

[root@stbydb1]# oakcli 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.

[oracle@proddb1]$ $ORACLE_HOME/bin/srvctl config database -d chicago |grep -i Password
Password file: /u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/orapwchicago
[oracle@proddb1]$ scp /u02/app/oracle/oradata/datastore/.ACFS/snaps/chicago/chicago/orapwchicago
oracle@stbydb1.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.

[oracle@proddb1]$ scp /tmp/chicago.pfile oracle@stbydb1.oracle.com:/tmp/boston.pfile
[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]$ cp /u02/app/oracle/oradata/datastore/.ACFS/snaps/boston/boston/orapwboston
/u01/app/oracle/product/12.1.0.2/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 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 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: 00:00:26
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;
RMAM> 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 -acfspath
'/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 -acfspath
'/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'
[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" ;
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/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 E: UPGRADING DATABASE WITH MANUALLY CONFIGURED ORACLE DATA GUARD ON ODA WITH DCS

Note: On 19.12+ 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 below ODA release 19.12.
   On 19.12+ 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 upgrade or patching of the database component.

UPGRADING THE DATABASE BELOW ODA RELEASE 19.12

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

   [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

   [oracle@stbydb1]# odacli create-dbhome -v 19.14.0.0.220118
5. Stop log shipping on the primary

   [oracle@ proddb1] dgmgrl
   connect sys/welcome1@chicago
   DGMGRL> SHOW DATABASE 'boston' 'LogShipping';
   LogShipping = 'ON'
   DGMGRL> edit database 'boston' SET PROPERTY 'LogShipping'='OFF';
   Property "LogShipping" updated
   DGMGRL> 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

   # odacli create-dbhome -v 19.14.0.0.220118
7. Stop the application
8. Upgrade the primary database using “odacli upgrade database” command

   [root@proddb1]# odacli list-databases

   ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID
   ---------- ---------- -------- ---------- ---------- -------- -------- ---------- ---------- ----------
   e97cc2f3-bdd8-4775-b959-d5f79a6c59fc chicago Rac 18.11.0.0.200714 false Oltp Odb1 Asm Configured 88ce2c7-fa3d-4f93-802a-bfa50d180758

[root@proddb1]# odacli list-dbhomes

ID Name DB Version Home Location Status

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

863c8cbe-1c5f-450e-866c-15c384580ad3 OraDB19000_home1 19.14.0.0.220118
/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 Configured
288ce2c7-fa3d-4f93-802a-bfa50d180758 OraDB18000_home1 18.11.0.0.200714
/u01/app/oracle/product/18.0.0.0/dbhome_1 Configured

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

[root@proddb1]# odacli list-databases

ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID

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

713b68d3-8c43-4d10-973e-90a3fa88a84a chicago Rac 19.14.0.0.220118 false Oltp Odb1 Asm
Configured 863c8cbe-1c5f-450e-866c-15c384580ad3

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/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston
[grid@stbydb1 ~]$ asmcmd
ASMCMD> pwcopy /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston
+DATA/BOSTON/PASSWORDFILE/pwdboston
copying /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/dbs/orapwboston ->
+DATA/BOSTON/PASSWORDFILE/pwdboston

12. Remove the 18c 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/odaorahome/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' -startoption mount

RAC example

[oracle@stbydb1]$ srvctl add database -db boston -oraclehome /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 -dbtype RAC -dbname chicago -diskgroup 'DATA,RECO,REDO' -role physical_standby -spfile '+DATA/BOSTON/PARAMETERFILE/spfileboston' -pwfile '+DATA/BOSTON/PASSWORDFILE/pwdboston' -startoption mount
[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
DGMGRL> connect sys/welcome1@chicago
DGMGRL> edit database 'boston' SET PROPERTY 'LogShipping'='ON';
Property "LogShipping" updated
DGMGRL> SHOW DATABASE 'boston' 'LogShipping';
LogShipping = 'ON'
DGMGRL> show configuration verbose
DGMGRL> show database verbose chicago
DGMGRL> show database verbose boston
DGMGRL> validate database chicago
DGMGRL> validate database boston

15. Test the switchover and failover

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

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

Health check
DGMGRL> show database verbose chicago
DGMGRL> show database verbose boston
DGMGRL> validate database chicago
DGMGRL> validate database boston

16. Sync up the registry on the standby

Verify the version of the DB

[root@ stbydb1~]# odacli list-databases

```
ID                                       DB Name    DB Type  DB Version           CDB        Class    Shape    Storage    Status       DbHomeID
---------------------------------------- ---------- -------- -------------------- ---------- -------- -------- ---------- ------------ ----------------------------------------
e6450a56-5a7d-4dab-9ca9-25b004b66646     chicago    Rac      18.11.0.0.200714     false      Oltp     Odb1     Asm        Configured
755b4b5d-6211-4d94-81e8-cf611868fe39
```

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, 2021 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

```
ID                                       DB Name    DB Type  DB Version           CDB        Class    Shape    Storage    Status       bHomeID
---------------------------------------- ---------- -------- -------------------- ---------- -------- -------- ---------- ------------ ----------------------------------------
e6450a56-5a7d-4dab-9ca9-25b004b66646     chicago    Rac      19.14.0.0.220118     false      Oltp     Odb1     Asm        Configured
17f68bbf-b812-42e5-96ba-1433c30f75ed
```

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

Please note that update registry removes backup, dbdomain, cpupools and associated network settings for **all** databases due to Bug 33532148 - LNX64-19.12 : "ODACLI UPDATE-REGISTRY -N DB -F" WIPES OUT NETWORK, DB DOMAIN AND BACKUP SETTINGS. Backup, cpupools and associated network settings can be readded with odacli modify database command.

PATCHING THE DATABASE BELOW ODA RELEASE 19.12

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. Steps apply to databases on bare metal and databases on DBSystems as well.

Note: In case OJVM is in use in the database Standby-first patching is not possible. MOS note 2217053.1 - RAC Rolling Install Process for the "Oracle JavaVM Component Database PSU/RU" (OJVM PSU/RU) Patches can help to confirm OJVM usage. In such a case defer log shipping on the primary and patch the primary first.

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

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

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

   Identify the Oracle home of the database

   [root@ocboda10 ~]# odacli list-databases

   ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID
   ---------- ---------- -------- ---------- ---------- -------- -------- ---------- ---------- ----------
   667a0eec-910c-404b-9820-aedcddf668d7 chicago Rac 19.11.0.0.210420 false Oltp Odb1 Asm Configured 863c8cbe-1c5f-450e-866c-15c384580ad3

   [oracle@stbydb1]# odacli list-dbhomes

   ID Name DB Version Home Location Status
   ---------- ---------- ---------- ---------- ----------
   863c8cbe-1c5f-450e-866c-15c384580ad3 OraDB19000_home1 19.11.0.0.210420 /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 Configured

Run the pre-check on the Oracle home

[oracle@stbydb1]# odacli update-dbhome -p -i 6d05e3f1-e948-4482-bcba-c560d9c8e5e5 -v 19.14.0.0

[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, 2021 6:26:51 PM CET

Apply the patches

[oracle@stbydb1]# odacli update-dbhome -i 6d05e3f1-e948-4482-bcba-c560d9c8e5e5 -v 19.14.0.0

[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, 2021 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 OraDB19000_home2 19.14.0.0.220118

/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_2 Configured

863c8cbe-1c5f-450e-866c-15c384580ad3 OraDB19000_home1 19.11.0.0.210420

/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1 Configured

[oracle@stbydb1]# odacli list-databases

ID DB Name DB Type DB Version CDB Class Shape Storage Status DbHomeID

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

667a0eec-910c-404b-9820-aedcddf668d7 chicago Rac 19.14.0.0.220118 false Oltp Odb1 Asm Configured

e4e9fcbd-63d4-4c56-bb0c-b239a4e749f3

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

DGMGRL> connect sys/welcome1@chicago

DGMGRL> edit database 'boston' SET PROPERTY 'LogShipping'='ON';

Property "LogShipping" updated

DGMGRL> SHOW DATABASE 'boston' 'LogShipping';

LogShipping = 'ON'

DGMGRL> show configuration verbose

DGMGRL> show database verbose chicago

DGMGRL> show database verbose boston

DGMGRL> validate database chicago

DGMGRL> validate database boston

APPENDIX F: UPGRADING DATABASE WITH MANUALLY CONFIGURED 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 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.

APPENDIX H: CONFIGURING NFS SERVER ON ODA

NFS can be configured on one of the ODAs to take a backup of the source database and to restore it as a standby on the target machine in case NAS or Oracle Object Storage are not an option.
NFS server needs to be configured on bare metal host anyway regardless the location (e.g., BM or DBSystem) of the primary and the standby.

1. Create an ADVM volume on source BM node0 as grid OS user

[grid@odabm1 ~]$ asmcmd
asmcmd> volcreate -G data -s 100G backup

ASMCMD> volinfo -G data backup
Diskgroup Name: DATA
Volume Name: BACKUP
Volume Device: /dev/asm/backup-322
State: ENABLED
Size (MB): 102400
Resize Unit (MB): 64
Redundancy: HIGH
Stripe Columns: 8
Stripe Width (K): 4096
Usage:
Mountpath:

2. Format the volume as ACFS

[grid@odabm1 ~]$ mkfs -t acfs /dev/asm/backup-322
mkfs.acfs: version = 19.0.0.0.0
mkfs.acfs: on-disk version = 46.0
mkfs.acfs: volume = /dev/asm/backup-322
mkfs.acfs: volume size = 107374182400 (100.00 GB)
mkfs.acfs: Format complete.

3. Create a mount point on both nodes

On both source BM nodes:
 # mkdir /backup

4. Register the filesystem in the clusterware and start it up as root OS user

[root@odabm1 ~]# /u01/app/19.15.0.0/grid/bin/srvctl add filesystem -d /dev/asm/backup-322 -path /backup -mountowner oracle -mountgroup dba
[root@odabm1 ~]# /u01/app/19.15.0.0/grid/bin/srvctl start filesystem -d /dev/asm/backup-322

5. Append to /etc/exports on BM node0 and make it active

[root@odabm1 ~]# vi /etc/exports
/backup *(rw,sync,no_root_squash)

or each source and target nodes could be added separately like
/backup primary1(rw,sync,no_root_squash)
/backup primary2(rw,sync,no_root_squash)
/backup standby1(rw,sync,no_root_squash)
/backup standby2(rw,sync,no_root_squash)
where primary1, primary2 nodes refer to the nodes hosting the primary database and standby1, standby2 refer to the nodes hosting the standby

[root@odabm1 ~]# exportfs -a
[root@odabm1 ~]# exportfs -v
...
*/backup *(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)*

6. Create a mount point on the source and the target nodes using the same mount point name

 # mkdir /odabackup

7. Mount the filesystem on both nodes using the source BM node0's public IP

 # mount -t nfs 192.168.17.2:/backup /odabackup

8. As the oracle user id might be different between the source and target create a subfolder under /odabkp and change the ownership to oracle:dba on it

 # mkdir /odabackup/db
In case the DB is TDE enabled then one more folder is required:
 # mkdir /odabackup/tde
 # chown -R oracle:dba /odabackup

9. After configuring NFS on both source and target follow Appendix A up to restoring the database as a standby (step 6)
10. Before restoring the database change the ownership to oracle:dba on the target. Please recall that the user and group ids might be different between the source and target.

 # chown -R oracle:dba /odabackup

11. Continue the steps from step 6 and complete the dataguard configuration
12. After configuring the Data Guard all NFS related changes need to be reverted

a) Unmount /odabkp on source and target nodes

 # unmount /odabkp

b) Unmount /backup on all BM nodes

 # unmount /backup

c) Remove /backup from /etc/exports

d) Update the NFS configuration on the first BM node

[root@odabm1 ~]# exportfs -a

e) Delete the backup ACFS filesystem from the clusterware configuration

[root@odabm1 ~]# /u01/app/19.15.0.0/grid/bin/srvctl stop filesystem -d /dev/asm/backup-322
[root@odabm1 ~]# /u01/app/19.15.0.0/grid/bin/srvctl remove filesystem -d /dev/asm/backup-322

f) Delete the backup related ADVM volume as grid OS user on the BM node

[grid@odabm1 ~]$ asmcmd
ASMCDM> voldelete -G data backup

g) Re-assign the original backup configuration to the primary database. By default it is 'default'

[root@proddb1 ~]# odacli modify-database -in <databasename> -bin default

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 21c, 19c, 18c, 12.2, 12.1

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) KNIWLEDGE 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

Note 2217053.1 - RAC Rolling Install Process for the "Oracle JavaVM Component Database PSU/RU" (OJVM PSU/RU) Patches

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Oracle Database Appliance: Implementing MAA Disaster Recovery Solutions Using Oracle Data Guard
August, 2022
Author: Krisztian Fekete, Sanjay Singh
Contributing Authors: Oracle RAC Pack and MAA Team