Oracle Maximum
Availability Architecture

E-Business Suite Release 12.2
Maximum Availability Architecture

ORACLE WHITE PAPER | JANUARY 2018

Table of Contents

1 Introduction

The Oracle E-Business Suite is a comprehensive suite of integrated global business applications spanning enterprise resource planning, customer relationship management, human resource management, financial management, service management, project portfolio management, value chain planning and execution, procurement, and supply chain planning. These mission-critical functions require protection to ensure they continue to be available at all times. Whether Oracle E-Business Suite is deployed in the Cloud or on-premises, planning for and implementing strategies for high availability and disaster recovery are paramount.

In this paper, we apply the principles of Oracle's Maximum Availability Architecture (MAA) to the Oracle E-Business Suite (EBS) Release 12.2 infrastructure. The major difference between the methods described here for Oracle E-Business Suite Release 12.2 and those documented elsewhere for earlier releases is the use of logical host names instead of physical host names in the EBS-managed topology. This allows us to make the configuration of the standby site the same as the primary site, from an Oracle E-Business Suite perspective, thus allowing you to switch between the two sites with minimal configuration efforts and facilitating a reduction in downtime as compared to using physical host names.

This paper provides:

- A high-level description of the Oracle E-Business Suite Release 12.2 MAA architecture and configuration best practices. In general, the MAA best practices described are generic across all platforms. We identify where there are best practices unique to deployments on Oracle engineered systems.
- A case study of Oracle E-Business Suite Release 12.2 on Oracle Exalogic and Exadata machines, showing the steps for:
  - Reconfiguring an existing Oracle E-Business Suite implementation to use logical host names
  - Establishing the disaster recovery (DR) site
  - Non-destructive testing of Oracle E-Business Suite at the DR site
  - Switchover and failover procedures
  - Simplified patching of Oracle E-Business Suite's Oracle Database homes

This MAA implementation for Oracle E-Business Suite Release 12.2 combines the proven technology used by Oracle E-Business Suite on Oracle's Exalogic and Exadata machines with tested MAA best practices. You can follow the steps in this document to implement these MAA best practices.

2 Overview of Engineered Systems

Oracle engineered systems combine best-of-breed hardware and software components with game-changing technical innovations. Designed, engineered, and tested to work best together, Oracle engineered systems can power the cloud or streamline data center operations to make traditional deployments more efficient. The components of Oracle engineered systems are preassembled for the targeted functionality and then—as a complete system—optimized for extreme performance. By taking the guesswork out of these highly available, purpose-built solutions, Oracle delivers a solution that integrates across every layer of the technology stack. This simplicity translates into less risk and lower costs for your business. Only Oracle can innovate and optimize at every layer of the stack to simplify data center operations, drive down costs, and accelerate business innovation.

2.1 Oracle Exadata Database Machine

Oracle Exadata Database Machine is Oracle's database platform delivering extreme performance for database applications including online transaction processing, data warehousing, reporting, batch processing, and consolidation of mixed database workloads. Exadata is a pre-configured, pre-tuned, and pre-tested integrated system of servers, networking, and storage all optimized around the Oracle Database. As Exadata is an integrated system, it offers superior price-performance, availability, and supportability. Exadata frees you from the need to build, test, and maintain systems and allows you to focus on higher value business problems.

Exadata uses a scale-out architecture for database servers and storage. This architecture maintains an optimal storage hierarchy spanning memory, flash storage, and disk. Features of the Exadata Database Machine include:

- Smart Scan queries offload database processing to the storage cells.
- The Exadata Smart Flash Cache feature ensures optimal use of Flash storage for reads and writes to benefit database performance.
- The high-bandwidth, low-latency InfiniBand network runs specialized database networking protocols and connects all the components inside an Exadata Database Machine.
- The database industry's best data compression, to provide a dramatic reduction in storage requirements
- In-Memory OLTP Acceleration, where data blocks are naturally aged out of the buffer cache and relocated into DRAM in the storage cells, where they can be accessed more quickly than from flash cache.

2.2 Oracle Exalogic

Oracle Exalogic is an Engineered System used by enterprises to deploy Oracle business applications, Oracle Fusion Middleware, and third-party software products. Oracle Exalogic comes pre-built with compute nodes, memory, flash storage, and centralized storage that are all connected using InfiniBand in a high redundancy architecture with in-built fault tolerance and zero-down-time maintenance.

3 Oracle E-Business Suite Release 12.2 Maximum Availability Architecture

Oracle E-Business Suite Maximum Availability Architecture (MAA) is an EBS high availability (HA) architecture layered on top of the Oracle Database, including a secondary site to provide business continuity in the event of a primary site failure.

In this section, we first present the Oracle Database Maximum Availability Architecture, then we describe how to provide high availability for Oracle E-Business Suite on top of that foundation, resulting in a full Oracle E-Business Suite MAA implementation.

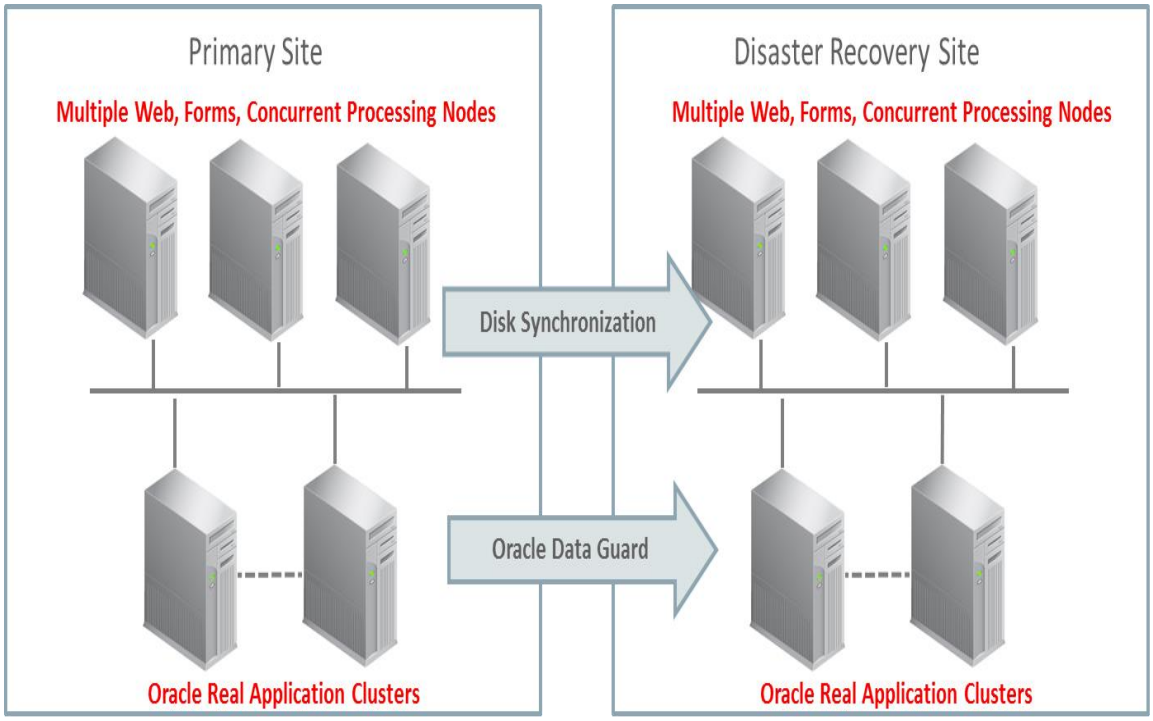

Figure 1 Oracle E-Business Suite Release 12.2 full MAA implementation

Figure 1 illustrates a full MAA implementation of an Oracle E-Business Suite Release 12.2 deployment consisting of two separate sites - a primary site and a disaster recovery site. At the primary site, Oracle Real Application Clusters (Oracle RAC) ensures high availability of the database tier, and multiple application tiers provide high availability for Oracle E-Business Suite users and applications. Database replication to the disaster recovery site is achieved using Oracle Data Guard. Application tier file system replication is achieved using storage replication, rsync, or any other common UNIX/Linux tool. The disaster recovery site is identical to the primary site in terms of system infrastructure.

3.1 Oracle Database Maximum Availability Architecture

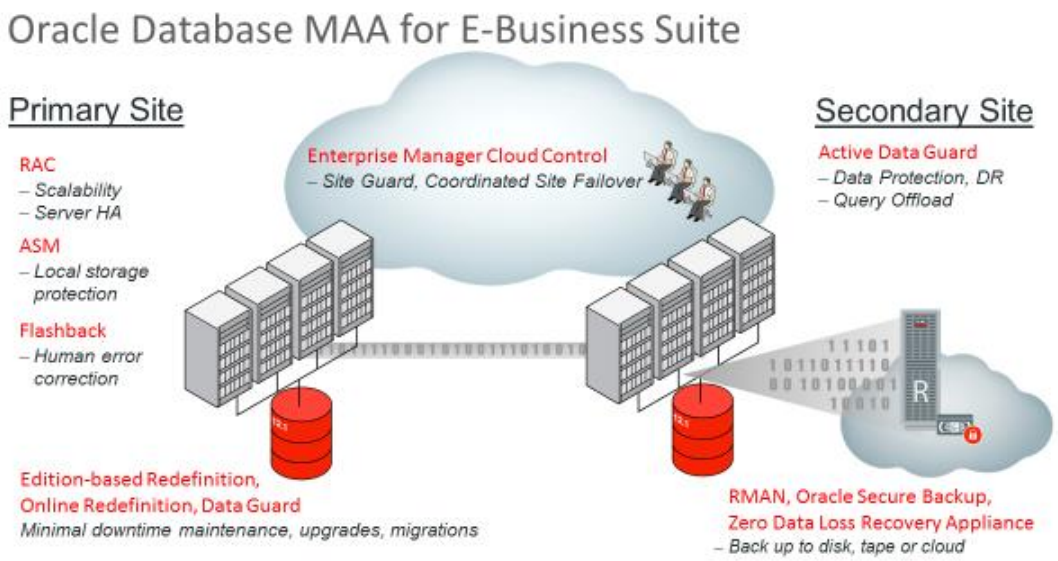

Figure 2 Oracle Database Maximum Availability Architecture

To achieve maximum Oracle E-Business Suite application availability, Oracle recommends deploying E-Business Suite on an Oracle Database MAA foundation that includes the following technologies as shown in Figure 2.

- Oracle Real Application Clusters and Oracle Clusterware
- Oracle Data Guard
- Oracle Flashback Database[1]
- Oracle Automatic Storage Management
- Oracle Recovery Manager and Oracle Secure Backup
- Oracle Online Upgrade Using Edition Based Redefinition

The rest of this section briefly describes each of these components. For further information, refer to *Oracle Database High Availability Overview* for a thorough introduction to Oracle Database high availability products, features and best practices.

3.1.1 Oracle Real Application Clusters and Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) enable the Oracle Database to run any packaged or custom application unchanged across a set of clustered nodes. This capability provides the highest levels of availability and maximum scalability. If a clustered node fails, the Oracle Database continues to run on the surviving nodes. When more processing power is needed, another node can be added without interrupting database operations or user access. For further information, refer to *Oracle Real Application Clusters Administration and Deployment Guide*.

[1] There may be performance implications when LOBs are in use with Oracle Flashback Database. This technology is used with Snapshot Standby Database, and will make re-instantiation of a production site simpler after a failover is executed to a standby site.

Oracle Clusterware is a cluster manager specifically designed for the Oracle Database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources (such as database instances and associated listeners). If a failure occurs, Oracle Clusterware automatically attempts to restart the failed resource. During outages, Oracle Clusterware will relocate the processing performed by the inoperative resource to an alternate resource. For example, if a node fails, Oracle Clusterware will relocate the database services being used by the application to a surviving node in the cluster.

For further information, refer to *Clusterware Administration and Deployment Guide*.

3.1.2 Oracle Data Guard

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle Databases to survive failures, disasters, and physical or logical block corruption. Data Guard maintains these standby databases as transactionally consistent copies of the production database. If the production database becomes unavailable due to a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus greatly reducing the application downtime caused by the outage. Data Guard can be used with traditional backup, restore, and clustering solutions to provide a high level of data protection and data availability.

Oracle E-Business Suite supports physical standby databases. A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. A physical standby database is kept synchronized with the primary database using Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

With Oracle Active Data Guard, a physical standby database can receive and apply redo while it is open for read-only access, so it may be used for other purposes as well as disaster recovery. The Oracle E-Business Suite supports a limited set of Oracle Active Data Guard functions including automatic block repair and fast incremental backups. For further information, refer to MOS Doc ID 1944539.1, "Using Active Data Guard Reporting with Oracle E-Business Suite Release 12.2 and an Oracle 11g or 12c Database".

A physical standby database can be converted into a Snapshot Standby with a single command. It will temporarily become an independent database (open read-write), which is ideal for short-term uses such as testing patches. The Snapshot Standby will continue to receive and archive (but not apply) redo data from the primary database while it is open read-write, protecting the primary data at all times. When testing is complete, a single command converts the snapshot back into a standby database, using Flashback Database functionality to roll back all changes made to the snapshot then automatically resynchronizing it with the primary by rolling forward through the redo received from the primary.

For more information about Oracle Data Guard, including Snapshot Standby, see *Data Guard Concepts and Administration*.

3.1.3 Oracle Data Guard Broker

Oracle Data Guard Broker is a command-line interface that enables easy and automated configuration of an Oracle Data Guard environment. It provides for simplified management and monitoring of the Data Guard configuraton. Orchestration of role transitions for switchover or failover are managed by Oracle Data Guard Broker with a single simple command.

For Data Guard Broker documentation, see *Data Guard Broker*.

3.1.4 Oracle Flashback Database

Oracle Flashback quickly rewinds an Oracle database, table, or transaction to a previous point in time, to help correct problems caused by logical data corruption or user error, or to reset a test database between runs. It is like a 'rewind button' for your database.

Some features of Oracle Flashback are not supported in a production Oracle E-Business Suite environment unless under direct guidance from Oracle Support. Specifically, while in some cases it might seem useful to flash a specific transaction or table back to a prior point in time, this cannot be done in a production E-Business Suite environment except under severe data recovery situations. However, Oracle Flashback Database can be particularly useful in Oracle E-Business Suite test environments, to recover from bugs discovered when testing software changes, unwind a patch, or to rewind the database between test runs. Oracle Flashback Database can also be used to rapidly return a previous failed primary database to standby operation after a Data Guard failover, eliminating the need to recopy or re-instantiate the entire database from a backup.

The use of Large Objects (LOBs) with Oracle Flashback Database can result in performance overhead. Several Oracle E-Business Suite modules use LOBs, so you should measure Oracle Flashback Database performance with expected LOB maintenance to ensure the solution performs adequately in production. Regardless, using Oracle Flashback Database is recommended for testing the disaster recovery site using Snapshot Standby.

See "Oracle Flashback Technology" in *Oracle Database Concepts* for more information. For Flashback Database best practices see MOS Doc "Flashback Database Best Practices & Performance (Doc ID 565535.1)".

3.1.5 Oracle Automatic Storage Management

Oracle Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel, resulting in:

- Significantly less work to provision database storage
- High levels of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Automatically resilvers mirrors to recover from physical hardware issues
- Proactive detection and repair of physically corrupted blocks, and of blocks that have certain logical corruptions
- Automatically rebalances contents as needed after disk group maintenance

For optimal performance, ASM distributes data files across all available storage. To protect against data loss, ASM extends the concept of SAME (Stripe And Mirror Everything) and adds more flexibility in that it can mirror at the database file level rather than the entire disk level.

For more details, see *Automatic Storage Management Administrator's Guide*.

3.1.6 Oracle Recovery Manager and Oracle Secure Backup

Oracle Recovery Manager (RMAN) is an Oracle Database utility that can back up, restore, and recover database files. It is a feature of Oracle Database and does not require a separate installation. RMAN integrates with sessions running on an Oracle database to perform a range of backup and recovery activities, including maintaining a repository of historical data about backups.

Oracle Secure Backup (OSB) is a centralized tape backup management solution providing performant, heterogeneous data protection in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. By protecting file system and Oracle Database data, OSB provides a complete tape backup solution for your IT environment. OSB is tightly integrated with RMAN to provide the media management layer for RMAN.

For more details, see *Database Backup and Recovery User's Guide*.

3.1.7 Online Upgrades Using Edition Based Redefinition

Edition-Based Redefinition (EBR) was introduced with Oracle Database 11*g*, and enables the database's participation in online application patches and upgrades in the following manner:

- Schema changes and application code changes stored in the database are installed in the privacy of a new edition.
- Data changes are made safely by writing only to new columns or new tables not seen by the old edition. An editioning view exposes a different projection of a table into each edition to allow each to see just its own columns.
- A crossedition trigger propagates data changes made by the old edition into the new edition's columns.

EBR is one of the key features utilized in Oracle E-Business Suite Release 12.2 for its online application patching. This is discussed in the section "Oracle E-Business Suite R12.2 Online Patching". Later, this paper describes how the disaster recovery site can be used to test an Oracle E-Business Suite patch or upgrade before finalizing the maintenance event in production. This allows you to make further changes or even abandon the system changes made by a patch.

For more details on Edition Based Redefinition, see the chapter: "Using Edition Based Redefinition" in the *Database Development Guide*.

3.2 Oracle E-Business Suite Release 12.2 High Availability

The distribution of Oracle E-Business Suite managed services across multiple applications nodes enables application-level Oracle E-Business Suite high availability. To accomplish this, you can add application tier nodes to scale up the application tier. The additional Oracle E-Business Suite instances are typically located on dedicated machines to increase the availability and flexibility of the system.

The Oracle E-Business Suite application tier availability features include:

- Parallel Concurrent Processing
- Multiple load balanced application tier services
- Oracle E-Business Suite Release 12.2 Online Patching
- Logical host names (new with AD/TXK Delta 9)
- Fusion MiddleWare Administration Server configuration

Each of these is expanded in the following sections. Figure 3 illustrates this architecture.

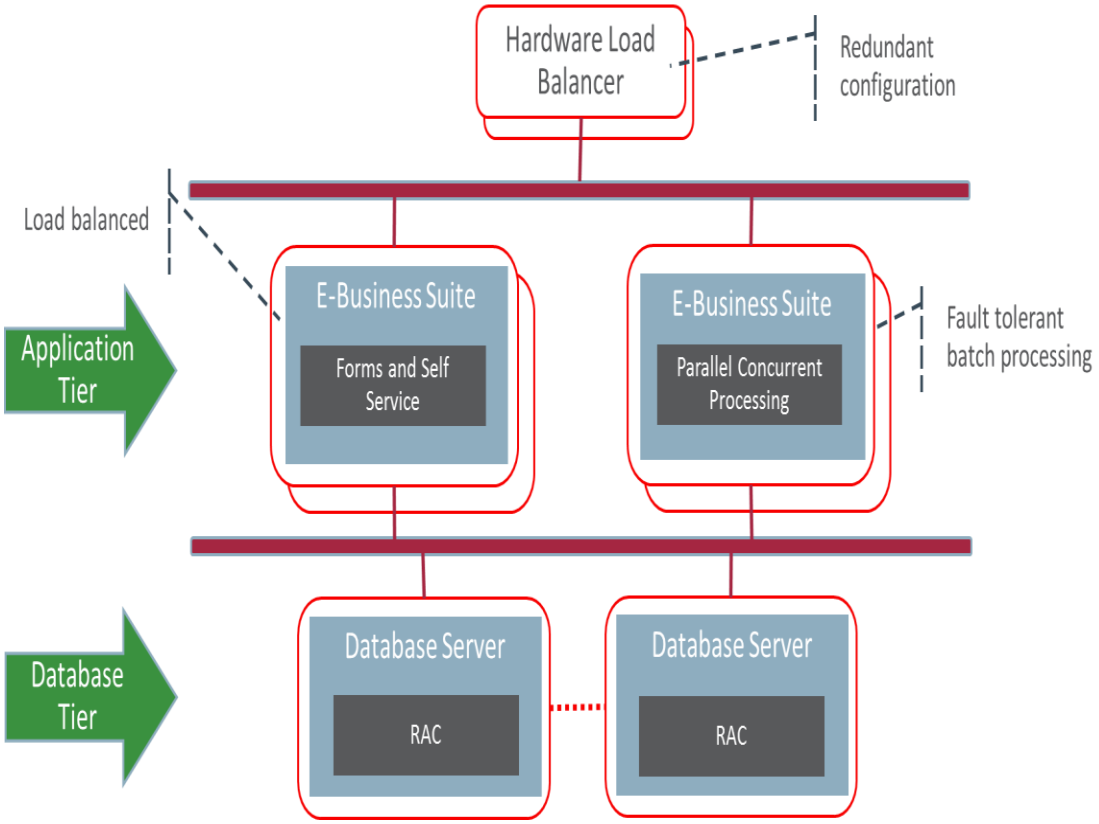

Figure 3 Oracle E-Business Suite High Availability Architecture

3.2.1 Parallel Concurrent Processing (PCP)

Parallel concurrent processing allows the distribution of concurrent managers (the core component of the batch processing application services) across multiple application nodes in a clustered, parallel, or networked environment. Instead of operating concurrent processing on a single node, the concurrent processing workload can be distributed and balanced across all available nodes.

Parallel concurrent processing provides the following availability and performance benefits for Oracle E-Business Suite:

- High performance – the ability to run concurrent processes on multiple application nodes to improve concurrent processing throughput and make concurrent processing highly scalable
- Fault Tolerance - the ability to continue running concurrent processes on available nodes even when one or more application nodes fail
- Adaptability - the ability to integrate with platform-specific batch queue and load-balancing systems to maximize concurrent processing behavior on a particular platform
- Single Point of Control - the ability to administer concurrent managers running on multiple application nodes from any node in a cluster, massively parallel, or networked environment

Concurrent programs cannot use load-balanced database services to connect to the database – all concurrent programs running on a given application server connect to a single specific database instance. Thus, to spread the workload across database instances, you must implement Parallel Concurrent Processing, which allows Concurrent Processing to run on more than one application server. You can configure all your workload to run on any

application server, thus against any database instance, or you can configure certain programs or groups of programs to run on a particular application server, thus against a specific database instance. The amount of benefit experienced by controlling which instance specific programs run against depends on the programs' data access and update profile. While there are clearly some tradeoffs such as additional planning and configuration, grouping programs onto specific servers by product can reduce the latency incurred by contention and block shipping between nodes across the Oracle RAC interconnect.

Parallel concurrent processing handles failover to a surviving database RAC instance in the same way as it handles failure of an application tier server – restarting its concurrent managers on an application tier node where there is an available connection to a surviving database instance

Note that an SMTP server is required on each application node hosting a concurrent manager, for use by the Oracle E-Business Suite Workflow Mailer.

For further information refer to MOS Docs "Concurrent Processing - Best Practices for Performance for Concurrent Managers in E-Business Suite (Doc ID 1057802.1)" and "Concurrent Processing - Product Information Center (PIC) (Doc ID 1304305.1)."

To create a high availability configuration of the Oracle E-Business Suite Workflow Mailer, define a primary and secondary node for the Workflow Mailer Service Manager using the Workflow Mailer Service form as shown in Figure 4.

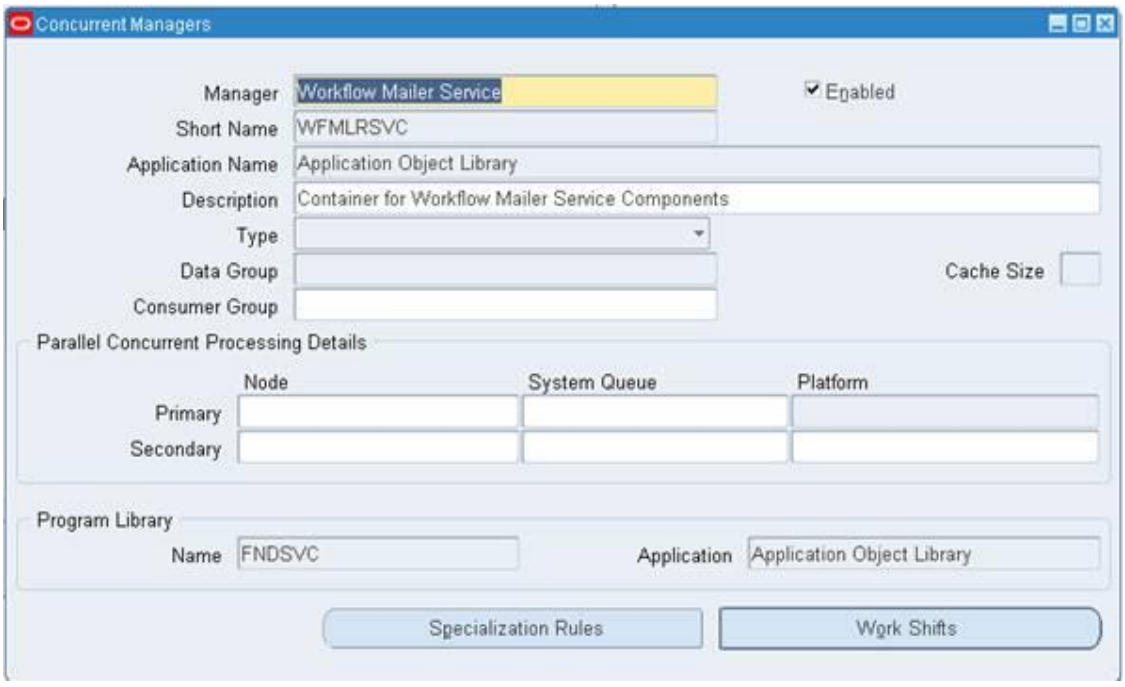

Figure 4 Oracle E-Business Suite Form for the Workflow Mailer Service

3.2.2 Multiple Load-Balanced Application Tier Services

The Oracle E-Business Suite Release 12.2 provides users with online access via the following application tier components:

- HTTP Services
- Java Services
- Forms Services

Each of the above components forms a managed service group. Each service group can be distributed across two or more physical or VM servers allowing for both high availability and scalability. Oracle E-Business Suite Release 12.2 now employs both Oracle Application Server and Fusion Middleware. Fusion Middleware is automatically configured to form a WebLogic cluster. Web and Forms services are now "managed services" under Weblogic cluster management. Managed services can be added or removed from the cluster as needed.

We recommend that more than one of each kind of managed service be deployed, on separate physical servers, so that a server outage does not affect availability. Where more than one instance of a component can serve users at one time, it is recommended that the servers have adequate capacity to run peak load even when one server is down, by provisioning at least one more server than required for peak load.

Online managed services (e.g., Forms and Java-based OACORE services) are usually load-balanced across the database RAC instances, although some customers find it beneficial to limit certain online functionality to a specific database instance, matching their Concurrent Manager configuration. For full flexibility in monitoring and managing workloads, we recommend each type of managed service is provided its own database service, and that the database service names reflect their usage or purpose rather than simply the database name. Enterprise Manager provides the ability to view connections for a given database service.

3.2.3 Oracle E-Business Suite Release 12.2 Online Patching

Beginning with Oracle E-Business Suite Release 12.2, the method for applying patches to Oracle E-Business Suite is through online patching. Release 12.2 employs the database technology Edition-Based Redefinition (EBR) to stage all database changes, and uses a dual file system approach to stage all application file system changes. This approach results in a dramatic reduction in downtime for patching as compared to prior releases.

Online patching manages two editions of the application – the run edition and the patch edition. In the database, the patch edition inherits the database structure and contents from the run edition. Schema updates are made in the privacy of the patch edition and do not disturb the running application. Data updates are accomplished using crossedition triggers, which populate new columns and/or tables with values required by the new version of the software, again without affecting or changing the running application. For the application tier file system changes, a copy is made of the directories that contain objects that are allowed to change as part of a maintenance event, and the changes are applied to the copy, without affecting or changing the running application. This is all managed by the Oracle E-Business Suite online patching tool `adop`.

When all changes required by the patching exercise have been completed and are fully staged, an `adop` process called `cutover` brings down application services then swaps the old run edition and the newly altered patch edition so that the patch edition becomes the new run edition. Once this swapping is complete, the application services are restarted. Downtime is reduced to the time it takes to shut down application services, perform the swap, then restart the application services.

When performing a cutover in production, it is highly recommended that you drain and shut down any long-running batch job queues. This can significantly reduce the time it takes to complete the cutover process.

3.2.4 Logical Host Names

Oracle E-Business Suite Release 12.2 has been enhanced, as of AD/TXK Delta 9, to provide a level of abstraction in the topology definition, known as "logical host names", which provides an alias for the physical host names. For disaster recovery, two symmetrical sites (with the same number and type of servers) are configured using logical host names – an abstraction that allows the EBS configuration to function unchanged on both the primary and the secondary site. Each server is allocated a new, neutral name that is used to configure Oracle E-Business Suite so that the installation does not need to be re-configured when switching from one site to the other. These logical host names are different from the physical host names. They are not registered in the DNS – they are used only by Oracle E-Business Suite. Other tools continue to use the physical host names.

3.2.5 Fusion Middleware Administration Server Configuration

In Oracle E-Business Suite Release 12.2, the application tier is deployed as a WebLogic Server (WLS) cluster. The first application tier to be deployed will host the WLS Administration Server. As there is only one Administration Server per WLS cluster, it is a singleton in the application architecture. The Administration Server is not required for normal production operations, but is required for changes to Oracle E-Business Suite application topology, such as adding or removing nodes or managed services, and for online (`adop`) patching.

The following options are available to provide a level of high availability should the server hosting the Administration Server fail:

- Restore the primary application server hosting the Administration Server from a backup
- Deploy an active/passive set of servers where one server is active, the second server is passive. The passive server can be brought up to assume the primary or master server role should the first active server fail. The passive server MUST have the same logical host name as the active server – therefore, only one can be up at any given time.
- If your application tier runs on VMs, start the primary Administration Server VM (which hosts the Administration Server) on a running server in the application tier cluster.
- Establish a disaster recovery site, with its own Administration Server, as described in this paper. Switch to the DR site if there are no other ways to restore the WLS Administration Server in production.

For further details regarding recovery of a lost primary WLS Administration server, see MOS Doc "Can The Primary Node Hosting the WLS Admin Server Failover To A 12.2 Slave Node, Migrating the Master WLS Admin Server To Another Machine (Doc ID 1986122.1).

4 Oracle E-Business Suite Database Configuration Best Practices

Oracle E-Business Suite MAA is based on the foundation described in section 3 of this paper. This section describes high availability and performance best practices for the entire system.

4.1 OS Role Separation

Role separation allows for separation of duties between the system administrators, the storage administrators, and the EBS database administrators. It is a configuration in which, for example, the Oracle Database software OS

owner is separate from the OS owner of the Grid Infrastructure. Prior to installing the Grid Infrastructure, OS groups and users must first be created for role separation.

Table 1 provides the description of each OS group and the OS users assigned to that group for a role separation configuration. In this table, the Grid Infrastructure and ASM are owned by the OS user `grid` and the database software is owned by the OS user `oracle`.

OS ROLE SEPARATION

Description	OS Group Name	OS Users Assigned to this Group	Oracle Privilege	Oracle Group Name
Oracle Inventory and Software Owner	oinstall	grid, oracle	N/A	N/A
Oracle Automatic Storage Management Group	asmadmin	grid	SYSASM	OSASM
ASM Database Administrator Group	asmdba	grid, oracle	SYSDBA for ASM	OSDBA for ASM
ASM Operator Group	asmoper	grid	SYSOPER for ASM	OSOPER for ASM
Database Administrator	dba	oracle	SYSDBA	OSDBA
Database Operator	oper	oracle	SYSOPER	OSOPER

Table 1: OS Role Separation

Once the OS groups and users have been created as in Table 1, Grid Infrastructure and ASM are installed using the `grid` OS user, and the Oracle database software is installed using the `oracle` OS user. Notice that the groups `oinstall` and `asmdba` are common between both `grid` and `oracle` OS users. These group inclusions are required for accessing the software installation inventory and for I/O access by the database to the ASM disk groups. However, the `oracle` user cannot perform any administrative tasks within ASM.

On engineered systems (Exadata and Supercluster), role separation can be configured during initial deployment using the Oracle Exadata Deployment Assistant (OEDA). For the OS Users and Groups section, select Role Separation and either define the OS users and groups or accept the default values provided. The configuration creates the OS users and groups, and then installs and deploys the Grid Infrastructure and ASM in a role separated configuration.

When running Oracle E-Business Suite Rapid Install, enter the appropriate OS user and OS groups that are required for the EBS Oracle database software. For further details about implementing role separation with EBS Rapid Install, see the MOS Doc Using Oracle 12c Release 1 (12.1) Real Application Clusters with Oracle E-Business Suite Release 12.2 (Doc ID 1626606.1).

4.2 Database Initialization Parameters

Review MOS Doc "Database Initialization Parameters for Oracle E-Business Suite Release 12 (Doc ID 396009.1)" for Oracle E-Business Suite's requirements for database initialization parameters, along with MOS Doc "Oracle Exadata Best Practices (Doc ID 757552.1)" – this is the main MOS document for Exadata best practices. Within MOS Doc 757552.1, click on the link "Verify Initialization Parameters and Diskgroup Attributes".

There will be differences between the two sets of recommendations. Discrepancies in sizing recommendations should be resolved using the largest recommendation, as indicated by your workload requirements. Settings marked "mandatory" by the EBS document 396009.1 should be used in any Oracle E-Business Suite installation.

4.3 Configure Linux HugePages

Using HugePages is a Linux 64-bit recommended best practice, and is configured by default on Exadata, but only for the default database and ASM. As Oracle E-Business Suite typically runs with many database connections and a large SGA, configuring HugePages for the database instances is essential. It is necessary to manually configure sufficient HugePages for the ASM instance and all database instances on each Linux database server node. This will result in more efficient page table memory usage, especially with a large SGA or with high numbers of concurrent database connections. HugePages can only be used for SGA memory space, so do not configure more than necessary.

MOS Doc ID 361468.1, "HugePages on Oracle Linux 64-bit" describes how to configure HugePages. Automatic Shared Memory Management (ASMM) can be used with HugePages so use the `SGA_MAX_SIZE` parameter to set the SGA size for each instance. (**Note:** database initialization parameters are discussed more fully in the section Database Initialization Parameters.)

Automatic Memory Management (AMM) cannot be used in conjunction with HugePages so the `MEMORY_TARGET` and `MEMORY_MAX_TARGET` parameters should be unset for each database instance. See MOS ID 749851.1 "HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux" for details.

Set the parameter `USE_LARGE_PAGES='only'` for each instance so that the instance only starts if sufficient HugePages are available. See "MOS ID 1392497.1 "USE_LARGE_PAGES To Enable HugePages" for details.

It might be necessary to reboot the database server to bring the new HugePages system configuration into effect. Check to make sure that sufficient HugePages exist by starting all the database instances to ensure that they all can allocate HugePages and start up successfully.

4.4 Take Backups of the Database Oracle Homes

Take regular backups of the database Oracle homes.

To be fully prepared to restore the backup to another environment if necessary, it is good practice to run `adpreclone.pl` on at least one database node before backing it up:

```
$ cd $ORACLE_HOME/appsutil/scripts/[context_name]
$ perl adpreclone.pl dbTier
```

4.5 Handle Database Password Expiration

The default behavior of Oracle Database between release 11*g* and 12*c* has changed such that database user passwords expire after 180 days. While it is important to regularly change passwords, if the Oracle E-Business Suite schema passwords are allowed to expire a system outage could occur. The Oracle E-Business Suite recommendations for password management are:

» Create two profiles: one for the Oracle E-Business Suite schemas and one for administrators accessing the database. Assign the Oracle E-Business Suite schemas to the application profile and administrators to the administrator profile as shown in Table 2: User Profiles.

USER PROFILES

Password Parameters	Application Profile	Administrator Profile
FAILED_LOGIN_ATTEMPTS	UNLIMITED	5
PASSWORD_LIFE_TIME	UNLIMITED	90
PASSWORD_REUSE_TIME	180	180
PASSWORD_REUSE_MAX	UNLIMITED	UNLIMITED
PASSWORD_LOCK_TIME	UNLIMITED	7
PASSWORD_GRACE_TIME	UNLIMITED	14
PASSWORD_VERIFY_FUNCTION	Recommended	Recommended

Table 2: User Profiles

» Assign a long (e.g., 12 characters), secure password to the APPS user and EBS product schema owners. Change these passwords regularly, but manage changing them manually. The MOS Doc "How to Change Applications Passwords using Applications Schema Password Change Utility (FNDCPASS or AFPASSWD) (Doc ID 437260.1)" provides details about how this can be accomplished.

» Give administrators their own accounts. Do not have them log in as the user APPS, so their actions can be properly audited.

For more information refer to MOS Doc ID 403537.1 "Secure Configuration Guide for Oracle E-Business Suite Release 12".

4.6 Exadata Fast Node Death Detection

As of Exadata X4-2 and X4-8, there are two active InfiniBand ports for all compute and storage servers. With that in place, Exadata Storage 12.1.2.1 and Oracle Grid Infrastructure 12.1.0.2 Bundle Patch 7 introduced the feature Fast Node Death Detection (FNDD). If both InfiniBand ports on a given compute node are no longer active or are non-responsive for more than one second to InfiniBand status queries, FNDD will force the node out of the cluster. When a node is forced out of the cluster, it will be restarted and will then rejoin the cluster. Note that this removes the need for setting the Oracle Cluster Synchronization Services (CSS) parameter `misscount` (see next section, Reduce Timeout on Oracle RAC Node Failure, for the solution with older Exadata offerings). Fast Node Death Detection is enabled by default where available on Exadata, and no manual configuration is required.

4.7 Reduce Timeout on Oracle RAC Node Failure

On older Exadata machines (prior to X4-2 and X4-8) it is possible to fail over relatively quickly in the event of an Oracle RAC node failure by reducing the CSS `misscount` parameter. The parameter defines how long to wait after a node becomes unresponsive before evicting the node from the cluster. The parameter has a default value of 60 (seconds) on Exadata and can be reduced to 30 if desired, but it should not be set to less than 30. Reducing the `misscount` parameter increases the risk that a node is evicted unnecessarily, so it should only be changed if absolutely necessary.

To update the CSS `misscount` setting, log in as the root user on one of the database servers and run the command:

```
$GRID_HOME/bin/crsctl set css misscount 30
```

4.8 Implement Flashback Database

As noted in the earlier section Oracle Flashback Database, Flashback Database allows for a quick reinstantiation of a failed primary database following a failover to its standby. With Flashback Database, the failed primary can be reinstated as the new standby, removing the need and effort to restore the database from a backup. It also reduces the time that your primary environment is not protected by a standby. We recommend you enable Flashback Database in your production database, after you verify your environment has sufficient storage capacity and performance.

Maintenance of LOBs may incur higher overhead with Flashback Database enabled. Determine which of your transaction flows heavily write LOBs to the database. Test performance of that workload with Flashback Database enabled, to ensure performance will be adequate at peak load in your production system.

The database maintains flashback logs in the Fast Recovery Area, which also provides online storage for your archived redo logs. You will need to test your peak transaction load to determine how much space will be required for your flashback logs to meet your configured retention period.

The steps for implementing Flashback Dataabase are described in our case study, in step 6 of the section Enable Data Guard Broker.

4.9 Exadata Smart Flash Logging

Exadata's Smart Flash Logging feature reduces stalls on OLTP transactions and improves overall transaction throughput. With Exadata Smart Flash Logging, the log writer process writes to both a small flash cache log and the (durable) physical disk (or separate NVMe flash devices in the case of Extreme Flash storage) simultaneously. The log writer process waits for only the first I/O completion (typically to the flash log) then moves on. The remaining outstanding I/Os are checked asynchronously for completion.

This feature is enabled by default on Exadata, requiring no configuration.

4.10 Exadata Smart Flash Cache

Exadata Smart Flash Cache provides an automated caching mechanism for frequently-accessed data in Exadata Database Machines that use disk for persistent storage. It is a write-back cache that can service extremely large numbers of random reads and writes, and improves the responsiveness of OLTP applications deployed on the Exadata Database Machine.

It is recommended that you start with the automatic caching policy, which allows the Oracle Database to decide which objects should be placed in flash cache. With further analysis, you can keep specific tables in the flash cache by using the following SQL command syntax:

```
ALTER TABLE [table_name] STORAGE(CELL_FLASH_CACHE KEEP);
```

The Exadata Smart Flash Cache can be configured to provide write-back functionality. Write-back flash cache on Exadata substantially reduces random write I/O latency and provides overall consistent performance for OLTP applications such as Oracle E-Business Suite. We recommend you enable write-back flash cache. Exadata deployments that have been imaged using recent versions of OEDA will enable write-back flash cache by default. It is also enabled by default for Exadata Extreme Flash storage cells.

For more information on write-back flash cache, including how to tell if it is enabled on your deployment and how to enable it, see Exadata Write-Back Flash Cache - FAQ (Doc ID 1500257.1).

4.11 Memory Management (Exadata Only)

We recommend upgrading to the latest available operating system for your Exadata Database Machine. Version 12.1.2.1.0 and later contain Linux kernels that address several kernel memory issues that include, for example, MTU size for InfiniBand as well as "kernel demand page 5" errors when the system is under memory pressure.

In addition, ensure that the kernel parameter `vm.min_free_kbytes` is set to 524288 (512MB) for all 2-socket compute nodes (Xn-2), and set it to 4194304 (4GB) for all 8-socket compute nodes (Xn-8).

It is also recommended that the MTU size on the InfiniBand network is set to 65520. This MTU size allows for better performance for backups and for NFS mounted file system access over InfiniBand.

For further details about Oracle Exadata best practices overall, refer to Oracle Exadata Best Practices (Doc ID 757552.1). For this specific best practice, refer to Oracle Sun Database Machine X2-2/X2-8, X3-2/X3-8 and X4-2 Performance Best Practices (Doc ID 1274475.1).

4.12 Oracle Advanced Compression

Oracle Advanced Compression saves space and improves performance for most Oracle E-Business Suite workloads. Further details on Oracle E-Business Suite compression tests can be found in the MOS Doc ID "1110648.1: "Oracle E-Business Suite Release 12.1 with Oracle Database 11g Advanced Compression". In the benchmark, an approximate 3:1 compression ratio was obtained overall. Online performance improvements varied from slightly slower to around 30% faster with a small increase in CPU usage (1-6%). Batch performance was generally faster, with exceptions seen from workloads that were I/O intensive and/or were already high in CPU usage. When Oracle implemented Advanced Compression in their production E-Business Suite database, the overall database size was reduced by 23%.

Advanced Compression also enables the use of RMAN binary compression, which can significantly reduce the size of the backup and can be achieved with only a marginal increase in the backup elapsed time. Oracle internal testing has established that with a setting of `RMAN MEDIUM` or `LOW` compression, compression ratios of between 2x and 4x of the uncompressed backup size can be obtained with less than a 1% increase in elapsed time. In some cases an actual reduction in elapsed time can be observed due to the reduction in backup size. An RMAN compression setting of `HIGH` is only recommended when the network is the most significant constraint. This setting provides the best compression ratio at the expense of high CPU consumption. This compares with the `MEDIUM` setting, which provides a good balance of CPU usage and compression ratio, while the `LOW` setting provides the fastest elapsed times and is best suited to when CPU resources are the backup constraint.

Optimal RMAN binary compression (`HIGH`, `MEDIUM` or `LOW`) is only available with the licensed Advanced Compression option. Without this option RMAN `BASIC` compression is available, but has a high CPU overhead.

4.13 Log Writer Tuning (Exadata Only)

Implement the following Exadata best practices to improve performance:

- The large Oracle Exadata Database Machine memory can accommodate a large redo log buffer (`log_buffer`); the best practice is a minimum of 128 MB.
- Ensure flash logs are configured so that Exadata smart flash logging can provide consistent low latency commits.
- Disable both `log_checkpoint_interval` and `log_checkpoint_timeout` parameters, and instead use `fast_start_mttr_target` to control checkpoint frequency. The MTTR Advisor can be used to determine an appropriate value for `fast_start_mttr_target`, for a given workload; the Exadata Health Check (`exachk`) reports an error if set to a value less than 300 (the standard database default is 60). This is an Exadata best practice and has been tested to show that a number of Oracle E-Business Suite batch programs see performance improvements with the `log_checkpoint_interval` parameter unset and `log_checkpoint_timeout` explicitly set to 0.

  For example:

  ```
  ALTER SYSTEM RESET LOG_CHECKPOINT_INTERVAL SID='*' SCOPE=SPFILE;
  ALTER SYSTEM SET LOG_CHECKPOINT_TIMEOUT=0 SID='*' SCOPE=BOTH;
  ```

  For `LOG_CHECKPOINT_INTERVAL` to take effect, database instances will need to be restarted. `LOG_CHECKPOINT_TIMEOUT` will be immediately disabled once it is set to 0.

- Log switches should occur no more than every 15 to 20 minutes under your peak load, to minimize forced checkpoints. To determine how often log switches are occurring, take AWR 1 hour snapshot reports while the system is under heavy load. Then, look for log switches in the report. For Oracle RAC, run the report on each Oracle RAC instance. If you find that the log switches occur at a higher frequency, increase the size of the online redo log files. If the primary database has a standby database, the standby redo logs must also be sized to match the new size of the primary online redo logs.

4.14 Fixed Object Statistics

The performance of queries against fixed objects can be suboptimal during high load conditions, but it can be improved considerably by gathering fixed object statistics using the following SQL statement:

```
exec dbms_stats.gather_fixed_objects_stats('ALL');
```

Note that `gather_fixed_object_stats` should be done after the instance has been "warmed up" or running for some time so the "static" data is relatively fixed, preferably during peak load. But gathering the fixed object statistics will also need resources, so you need to ensure that you have sufficient capacity to avoid impacting the other system processes. If you cannot do it during peak load, you should do it after the system has populated data for the three basic categories of fixed object tables:

- (Relatively) static data once the instance is warmed up – structural data like data files, controlfile contents, etc.
- Data that changes based on number of sessions connected
- Volatile data based on workload – e.g., vsql, vsql_plan

It is recommended that fixed object statistics be re-gathered following a major database or application upgrade, after a new EBS module is implemented, or after changes are made to the database configuration. For example, an increase in the SGA size might result in significant changes to the x$ tables that contain information about the buffer cache and shared pool, such as x$ tables used in `v$buffer_pool` or `v$shared_pool_advice`. Gathering

statistics for fixed objects is normally recommended if poor performance is encountered while querying dynamic views, for example, v$ views.

For more information refer to MOS Doc "Best Practices for Gathering Statistics with Oracle E-Business Suite (Doc ID 1586374.1)". Also, refer to MOS Doc "Fixed Objects Statistics Considerations [Doc ID 798257.1]".

4.15 Exadata I/O Resource Manager (IORM)

For Oracle E-Business Suite workloads, optimal performance is observed when the IORM Objective plan setting is "balanced". This setting balances low disk latency for small I/Os and high throughput for large I/Os. The setting limits large I/Os to 90% of disk capacity, then if there is any queuing of I/Os on a disk it puts any small I/Os at the head of the queue. This provides good performance for OLTP work while allowing excess capacity to serve DSS activity/smart scans.

On Exadata cells on release 11.2.3.2 and later, IORM is enabled by default to guard against excessively high latencies for small I/Os, using the "basic" objective. User-defined resource manager plans are not enforced in this mode. To enable IORM for user-defined resource manager plans, the objective must be set to something other than "basic". The following CellCLI command is used to change the objective to "balanced":

```
alter iormplan objective=balanced
```

See MOS Doc "Configuring Exadata I/O Resource Manager for Common Scenarios (Doc ID 1363188.1)" for more information.

4.16 Oracle Engineered Systems Best Practices and Health Check

Follow the MOS Doc "Oracle Exadata Database Machine `exachk` or HealthCheck [Doc ID 1070954.1]" to run the Oracle Exadata Database Machine `exachk` utility. `Exachk` is a valuable utility to run on Oracle engineered systems to identify potential issues at the OS, database, and storage layers, as well as in Oracle E-Business Suite. It is good practice to run this utility on a regular interval and after patching and software upgrades are performed. Any failures should be researched and addressed as appropriate.

Please note there will be differences between what EBS requires in its `init.ora` parameters and what is recommended by `exachk`. Review MOS Doc "Database Initialization Parameters for Oracle E-Business Suite Release 12 (Doc ID 396009.1) for Oracle E-Business Suite's requirements for database initialization parameters, along with MOS Doc Oracle Exadata Best Practices (Doc ID 757552.1) – this is the main MOS document for Exadata best practices. Within MOS Doc 757552.1, click on the link "Verify Initialization Parameters and Diskgroup Attributes". Discrepancies in sizing recommendations should be resolved using the largest recommendation, as indicated by your workload requirements. Settings marked "mandatory" by the EBS document 396009.1 should be used in any Oracle E-Business Suite installation.

5 Oracle E-Business Suite Release 12.2 Application Tier Configuration Best Practices

The following are best practices that are focused on the Oracle E-Business Suite application tier. While many of the following best practices are based on using Exalogic and ZFS storage appliance, many are generic in nature.

5.1 Create an Oracle E-Business Suite Central Inventory on the Application Tier

Beginning with AD / TXK C.Delta 7, Oracle E-Business Suite Release 12.2 supports creating a central inventory that is dedicated to Oracle E-Business Suite. The oraInst.loc file and the oraInventory directory contain all of the application tier Oracle homes for a specific Oracle E-Business Suite Release 12.2 installation. Creating a central inventory has the following benefits:

- The oraInst.loc file and oraInventory directory are dedicated to the Oracle E-Business Suite install and are placed within the installation
- Simplifies the cloning process
- Simplifies the instantiation of the disaster recovery (DR) site.
- Allows for consolidation of multiple Oracle E-Business Suite Release 12.2 installations on the application tier, each having its own central inventory.

For the case study discussed in sections 7 and 8 of this paper, it is essential that a central inventory is created. When setting up a central inventory, ensure the following:

- The context variable `s_ebs_central_inventory` is set to true.
- The context variable `s_invPtrLoc` specifies the directory location of the oraInst.loc file under ORACLE_BASE. For example, if ORACLE_BASE points to /u06/app/ebs/vis, then set this context variable to: /u06/app/ebs/vis/oraInventory/oraInst.loc
- Do NOT place the oraInst.loc under either the RUN, PATCH or NE (Non-Edition) (fs1, fs2 or fs_ne) directories.

Follow section 4.5 of MOS Doc Oracle E-Business Suite Applications DBA and Technology Stack Release Notes for R12.AD.C.Delta.7 and R12.TXK.C.Delta.7 (Doc ID 2033780.1) to migrate to a central inventory. Also see MOS Doc R12.2: How To Create, Update or Rebuild The Central Inventory For Oracle Applications E-Business Suite? (Doc ID 1588609.1).

5.2 Move the Apache Lock File to Local Storage

While most files on the application tier may reside on NFS, the Apache lock files used by the HTTP Server should be stored on a local disk when NFS mounts are used for the application tier file systems. This is done using the LockFile directive in the httpd.conf file located at $ORACLE_HOME/Apache/Apache/conf/ to point at a local drive. See MOS Doc "HTTP Server Is Either Slow Or Stops Responding When Installed On A NFS Mounted Drive [Doc ID 560853.1]".

5.3 Take Regular Backups of the Release 12.2 Application File System

Take regular backups of both the RUN and PATCH file systems (fs1 and fs2) as well as the non-editioned file system (fs_ne). These include the APPL_TOP, COMMON_TOP, the Oracle Fusion Middleware homes, and the instance homes.

To be fully prepared to restore the backup to another environment if necessary, run adpreclone.pl before backing it up:

```
$ cd $INST_TOP/admin/scripts/
$ perl adpreclone.pl appsTier
```

For Oracle E-Business Suite Release 12.2, once the preclone has completed, moving one directory level above where the fs1, fs2, and fs_ne are located, and then performing a backup (or zip), will capture the run, patch and non-edition directories along with all of the Fusion Middleware homes.

For specific information on backing up file systems on Exalogic ZFSSA, refer to "Exalogic Backup and Recovery Best Practices".

5.4 Use NFSv4 for NFS Mounted File Systems

The use of NFSv4 is a best practice generically, and is specifically recommended when the ZFS Storage Appliance on the Exalogic servers is in use. Do not use NFS mount options `actime=0` or `noacc` for file systems under NFSv4. Similarly, if using NFSv3 ensure that `noattr` is not set. While those options are recommended for Oracle RAC databases using NFS mounted storage, they are inappropriate for normal file system mounts and will result in poor performance.

An authentication facility such as LDAP or NIS is a prerequisite for NFSv4. Neither of these are discussed in this paper. See MOS Doc ID 1516025.1 "How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment" for details about configuring NIS on the Exalogic platform.

5.5 Disable NFS Delegation (Exalogic Only)

Under certain workloads when concurrent writes happen on the same file across multiple clients, a race condition can be induced that can cause NFSv4 mount points to hang. To avoid this hang issue NFSv4 delegation must be disabled on both active and passive ZFS Storage Appliance heads through the ZFS Browser User Interface (BUI) as follows:

1. Log in as root.
2. Navigate to Configuration -> SERVICES -> NFS.
3. Uncheck the Enable NFSv4 delegation checkbox.

For detailed information about symptoms of this known issue, and configuration steps with screenshots for disabling NFSv4 delegation using BUI mode and using the command-line interface (CLI) mode, refer to MOS Doc "NFSv4 mount directories hang on Exalogic Machine [Doc ID 1481713.1]."

5.6 Disable Failback for IPMP Setting (Exalogic Only)

In an Exalogic environment, when the active link of the ZFS Storage Appliance fails, the standby link takes over the active role. With failback enabled, when the original link comes back up, it takes over the active role again. If this happens within a short period of time, the application may still be in recovery mode due to the first failure and failback could cause a secondary failure.

To prevent the issue from occurring, the default IPMP parameters on the ZFS Storage Appliance need to be modified as follows:

- Failure Detection Latency: Change from 10000 to 5000 ms
- Failback: Change from true to false

The procedure for making the above changes to the IPMP parameters is as follows:

Using the ILOM:

1. Log in to the ILOM of the storage head as root, start the SP console and log in if required.
2. Navigate to configuration services ipmp.

```
zfssn01:> configuration services ipmp
```

3. Establish what values have been configured by issuing the following command:

```
zfssn01:configuration services ipmp> ls
Properties:
<status> = online
interval = 10000
failback = true
```

4. Check the configured value for interval. If the configured value is not 5000, set it to 5000.

```
zfssn01:configuration services ipmp> set interval=5000
interval = 5000 (uncommitted)
```

5. Check the configured value of failback. If it is true, set it to false.

```
zfssn01:configuration services ipmp> set failback=false
failback = false (uncommitted)
```

6. Commit the changes as shown here.

```
zfssn01:configuration services ipmp> commit
```

7. All of these changes can be verified by using following command.

```
zfssn01:configuration services ipmp> ls
Properties:
<status> = online
interval = 5000
failback = false
zfssn01:configuration services ipmp>
```

The IPMP configuration changes can also be made through the browser UI of the storage head.

Ensure that the above configuration changes are made to both storage heads. These configuration changes can be made to the active head without requiring a head takeover or a switch reboot.

5.7 Mount ZFS Shares over IPoIB (Exalogic Only)

For best performance, mount the ZFS shares using the IPoIB network. The network configuration performed by the Exalogic Configuration Utility (ECU) creates the system network partitions listed in the following table. These entries should not be modified.

The IP address corresponding to the *IPoIB-storage* network should be used to mount the exported shares on the bare metal compute nodes and the IP address corresponding to the *IPoIB-vserver-shared-storage* for the vServer nodes using the ZFS Storage Appliance. See Table 3: ZFS Appliance Network, for an example.

ZFS APPLIANCE NETWORK

Network Name	Default Partition Key	Device and Default IP Address	MTU	Description
IPoIB-default	Default	192.168.10.1/24 bond1 for compute nodes ipmp1 for storage appliance	65520	Default InfiniBand partition
IPoIB-admin	0x8001	192.168.20.0/24 bond2 for compute nodes ipmp2 for storage appliance	65520	This partition is used for all interconnections among the different components of Exalogic Control.
IPoIB-storage	0x8002	192.168.21.0/24 bond3 for compute nodes ipmp3 for storage appliance	65220	This partition is used internally to access shares on the storage appliance. It is different from the IPoIB-vserver-shared-storage network.
IPoIB-virt-admin	0x8003	bond4 172.16.0.0/16	65220	This partition is used by Exalogic Control for all virtualization management.
IPoIB-ovm-mgmt	0x8004	bond5 192.168.23.0/24	65220	This partition is used for all Oracle VM management. This includes heartbeat, migration, and virtualization control.
IPoIB-vserver-shared-storage	0x8005	bond6 for compute node, and ipmp4 for the storage appliance 172.17.0.0/16	65220	This partition is used to provide access to Sun ZFS Storage Appliance for customer or application vServers.
EoIB-external-mgmt	0x8006	bond7 VLAN ID, Ethernet device/connector (Ethernet port on the Sun Network QDR InfiniBand Gateway Switch), and network IP/subnet mask defined by the user	1500	This partition is used to provide external access to the vServer that hosts the Enterprise Controller component of Exalogic Control.

Table 3: ZFS Appliance Network

6 Summary of Best Practices

This section concludes with a summary of the best practices presented in this paper, providing a checklist for an Oracle E-Business Suite MAA implementation.

6.1 Best Practices – Oracle E-Business Suite Database Maximum Availability

These best practices should be applied to the primary and secondary sites to achieve highest availability:

- Deploy Oracle E-Business Suite on an Oracle RAC database to ensure the highest availability and scalability.
- Enable Oracle Flashback Database to provide the ability to reinstantiate the production database after a failoverif you have determined performance in production will be acceptable for LOB maintenance. Enable Oracle Flashback Database at the standby, to provide the ability to use a Snapshot Standby for testing.
- Use Automatic Storage Management to simplify the provisioning and management of database storage.
- Use Oracle Recovery Manager to regularly back up the Oracle E-Business Suite database.

- Take regular backups of the Oracle E-Business Suite database home. Run `adpreclone` before taking the backups.
- Monitor memory usage. Adjust parameters and re-balance the workload as needed.
- Always use HugePages for Oracle E-Business Suite databases on Linux. Set the database parameter `USE_LARGE_PAGES='ONLY'` to utilize HugePages.
- Configure database user passwords so that the seeded passwords do not automatically expire, but configure those assigned to administrators to automatically expire. Regularly change all passwords (including the seeded ones). The password expiration should be defined by your corporate security policies.
- Configure database Dead Connection Detection to actively remove dead connections in the event of application tier node failure.
- Gather fixed object statistics for better performance.

For Exadata deployments:

- Use `exachk` on a regular basis and after changes to the environment, to revalidate the configuration as well as platform and database patch levels.
- Set the IORM objective to "balanced," to ensure that the OLTP workload has good performance while allowing smart scans to be performed.
- On Exadata X3 and older generations, if you require a faster Oracle RAC node failover, change the `CSS_MISSCOUNT` from the default value of 60 to 30 seconds. It is not recommended that you set this parameter lower than 30 seconds. Exadata X4 and later will use Fast Node Detection (FNDD) and will not require `CSS_MISSCOUNT` to be set.
- Adjust `vm.min_free_kbytes`.
- Enable write-back flash cache if it is not already enabled.
- If using NFS on the database server, adjust the MTU setting on the InfiniBand network to 65520 if needed.
- Set your `init.ora` parameters as described in Exadata and Oracle E-Business Suite best practices.

6.2 Best Practices – Oracle E-Business Suite Application High Availability

These best practices should be applied to your application implementation to achieve highest availability for the primary site:

- Deploy multiple application tier servers. Spread the load across the middle tiers using a load-balanced configuration, so that work can continue in the event of an application tier node failure.
- Create load-balanced database services for the online loads to connect to, with database service names that reflect their usage or purpose in addition to the database name. Configure Oracle Forms and OACORE managed servers to connect to these load-balanced database services.
- Use a central inventory specific to Oracle E-Business Suite 12.2 on the application tier.
- Deploy the Oracle E-Business Suite file systems on a fault tolerant filer.
- Make sure the Apache lock files are located on local disk.
- Take regular backups of the application tier file systems fs1, fs2 and fs_ne. Run `adpreclone` before taking Oracle E-Business Suite backups.
- If using NFSv3, do not set the mount option `noattr`. For NFSv3, the mount option `actime=0` should not be set either. If NFSv4 is used, do not specify the mount options `actime=0` or `noac`. These options, for both NFSv3 and NFSv4, degrade performance.
- Configure the concurrent managers using Parallel Concurrent Processing, and balance the work across all Oracle RAC instances. Configure SMTP servers on all Parallel Concurrent Processing nodes. If needed to minimize interconnect traffic and block pinging across database instances, affinitize concurrent workloads to specific database instances.

For Exalogic deployments:

- NFSv4 is recommended when connecting to the ZFS Storage Appliance. Configuration of NIS is a prerequisite to implementing NFSv4.
- Disable NFSv4 delegation.
- Disable failback for the ZFS IPMP settings.
- Mount your ZFS shares using IPoIB.

6.3 Best Practices – Oracle E-Business Suite Maximum Availability

Follow these best practices for deploying a secondary site to further protect a highly-available primary site, and for switching and failing over between sites:

- Deploy a second geographically-separated site that can run the Oracle E-Business Suite workload in the event the primary site is down.
- Configure the DR site to be a symmetrical replica of the primary site, with the same number and type of servers in both locations.
- Ensure the network between the primary and secondary sites has sufficient bandwidth for the required redo transmission.
- Use Data Guard to replicate all database changes to a standby database located on the secondary site.
- Take advantage of Oracle Active Data Guard to offload a portion of the Oracle E-Business Suite reports to the standby database; for more information refer to "Offloading (Some) EBS 12 Reporting to Active Data Guard Instances". Also, use Oracle Active Data Guard for faster incremental backups and for automatic block repair.
- Enable Oracle Flashback Database on the standby and the primary database, so that the old primary database can be quickly re-instantiated in the event of site failover, so long as you have determined that the performance of the production system will be acceptable for LOB maintenance.
- Configure database Dead Connection Detection (DCD) to actively remove dead connections in the event of application node failure.
- Continuously replicate the Oracle E-Business Suite file system to the secondary site with minimal lag. Develop procedures for how to reverse the direction of replication in the event of failover or switchover, and procedures to clone the replica to create a "snapshot" for site testing.
- Conduct periodic role transitions including switchover and failover testing between the primary and DR sites to validate processes and procedures.
- Develop and document operational procedures for maintaining your secondary site and for role transitions across sites.
- Use Data Guard Broker to simplify Data Guard administration.
- Use a snapshot standby to provide an updatable replica of the primary database, to test your disaster recovery site configuration and and as a temporary environment for testing changes and issues in production.

For Exadata and Exalogic deployments:

- Use ZFS snapshot clones to provide an updateable replica of the primary application tier file system, to test your disaster recovery site configuration and as a temporary environment for testing changes and issues in production.
- Use ZFS replication to replicate Oracle E-Business Suite application tier file systems to the secondary site. This must include the software, configuration, and Concurrent Manager log and out directories.

7 Oracle E-Business Suite MAA Case Study Architecture

7.1 Method

The MOS Doc "Business Continuity for Oracle E-Business Suite Release 12.2 using Logical Host Names (Doc ID 2246690.1)" serves as the basis for establishing a business continuity solution for Oracle E-Business Suite Release 12.2. This white paper extends the architecture further to include:

- The incorporation of ZFS mounted file systems and ZFS replication for the application tier
- Using the RMAN DUPLICATE FROM ACTIVE DATABASE capabilities in Oracle Database 12*c* to build and deploy the standby database with minimal impact to the production system on the primary site
- Using the secondary site to test patches applied to the primary site

We reconfigured our primary environment to use logical host names, and then used the same logical host names when setting up the standby environment. Note that other tools will continue to use the physical host names. Addressing the post-switchover and post-failover connection issues with third party applications are outside the scope of this paper.

In our lab, Oracle Data Guard was used to replicate database updates, and ZFS continuous replication was used to replicate application tier file system changes. Note that it is not necessary, but is beneficial, to use continuous replication for the application tier. If a continuous replication method is not used, some data may be missing on failover.

We defined read-write snapshots of the standby database and the replicated application tier data stores, using Oracle Data Guard and ZFS replication. We describe how to open the snapshot standbys for testing while the primary is still fully functional.

To avoid confusion when the databases switch roles, we adopted the database naming convention of "vis_a" at the primary site and "vis_b" at the secondary site, rather than something that would indicate a primary or standby database. It is less confusing to note that your primary is running at your DR site and is called "vis_b" than seeing your primary named "vis_stdby" or a similar name.

7.2 Case Study Topology

Oracle E-Business Suite Release 12.2.5 was deployed in a full MAA configuration on X3-2 Exalogic and X3-2 and X4-2 Exadata machines. Each site is configured such that each can assume the primary or standby role. At each site, the database servers and storage reside on the Exadata machine. The application and web servers and their associated storage reside on the Exalogic machine..

Table 4: Primary and Secondary Site Configuration describes the hardware used at the primary and secondary sites and gives the names used for each server. The difference in hardware at the database tier between the primary and secondary sites is simply due to what was available in our labs for testing.

PRIMARY AND SECONDARY SITE CONFIGURATION

Tier	Primary Site	Secondary Site
Database Tier	Quarter Rack X3-2 Exadata Database Machine » *Two Compute nodes – exaadb03, exaabd04 32 cores (SMT enabled), 128GB RAM each* » *Three Exadata storage cells*	Quarter Rack X4-2 Exadata Database Machine » *Two Compute nodes – exabadm01, exabadm02 48 cores (SMT enabled), 256GB RAM each*

	24 cores, 1.48TB flash cache, high capacity disks each	» *Three Exadata storage cells* 24 cores, 5.8TB flash cache, high capacity disks each
Application and Web Tier	X3-2 Exalogic (virtualized and bare metal) » *Two compute nodes – exalgacn23, exalgacn24(bare metal)* *32 cores, 256GB RAM each* » *Hosting Oracle E-Business Suite Web and Application Service Groups*	X3-2 Exalogic bare metal (non-virtualized) » *Two compute nodes – exalgbvm0089-eoib2, exalgbvm0090-eoib2 (Virtualized)* *8 cores, 64 GB RAM each* » *Hosting Oracle E-Business Suite Web and Application Service Groups*
Application Tier File System Storage	Sun ZFS Storage 7320 Appliance for application and web server storage -- exalgasn-fe	Sun ZFS Storage 7320 Appliance for application and web server storage – exalgbsn-fe

Table 4: Primary and Secondary Site Configuration

Figure 5 illustrates a complete MAA implementation for Oracle E-Business Suite Release 12.2 using a symmetrical Exadata and Exalogic configuration for both primary and disaster recovery sites.

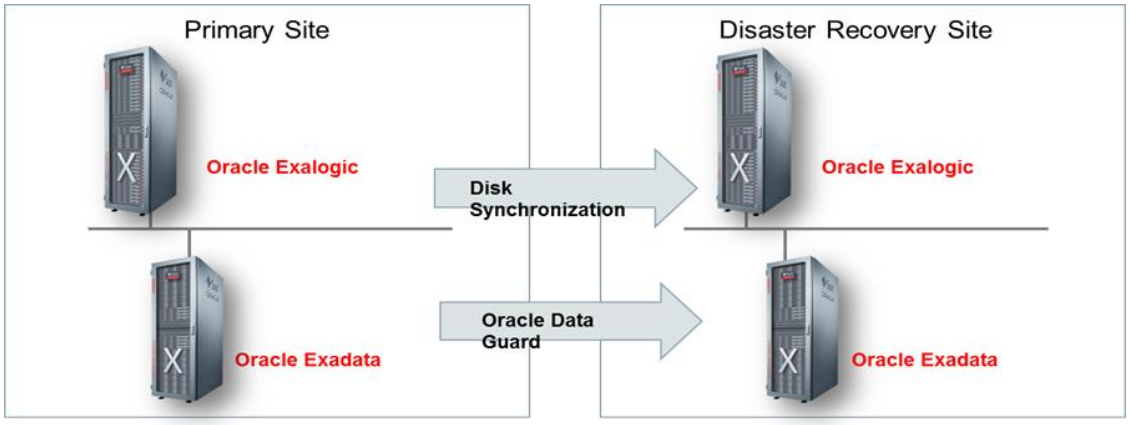

Figure 5 The complete MAA implementation of Exadata and Exalogic systems for the primary and disaster recovery sites

Figure 5 shows that both sites have an Exadata quarter rack for the Oracle E-Business Suite database tier and an Exalogic quarter rack for the application tier. Having the same number of servers at the secondary site allows us to switch quickly between sites with very little reconfiguration, relying on the logical host name implementation.

7.3 Software Versions

Table 5 shows the software and versions used for the case study. The following software and versions were used for the case study:

SOFTWARE VERSIONS

Software	Version
Exalogic	ECC 2.0.4.0.0

Oracle E-Business Suite	12.2.5, AD/TXK Delta 9 plus patches 17075726, 25882097, and 25057668
WebLogic Server	10.3.6.0
Database	12.1.0.2.170418
Exadata Storage Software	12.2.1.2.0.170815
Oracle Clusterware Grid Infrastructure	12.1.0.2.170418
ZFS 7320 Storage Appliance	2011.04.24.5.0,1-1.33

Table 5: Software Versions

7.4 Oracle E-Business Suite Application Tier – High Availability Architecture

This case study implemented a full MAA deployment as described in the previoous sections, with the following configuration:

- An f5 BigIP load balancer at both primary and secondary sites, using Local Trafic Manager (LTM)
- Two middle tier application servers on Exalogic at the primary site, each hosting:
  - All managed services (HTTP, Forms, oacore, etc), configured as active/active.
  - Parallel Concurrent Processing (PCP), with primary and secondary nodes established within the Concurrent Processing management framework.
- An Exalogic at the secondary site, to duplicate the application tier from the primary site. The application tier servers at the primary site were bare metal; at the standby they were Exalogic vServers, a configuration chosen to demonstrate either solution is viable
- A ZFS fault-tolerant filer at both sites on which the shared Oracle E-Business Suite R12.2 installation (shared APPL_TOP) resides. Continuous replication was used to keep the middle tier file systems in sync from the primary site to the secondary site.

Figure 6 shows the Application Tier and ZFS filer topology,

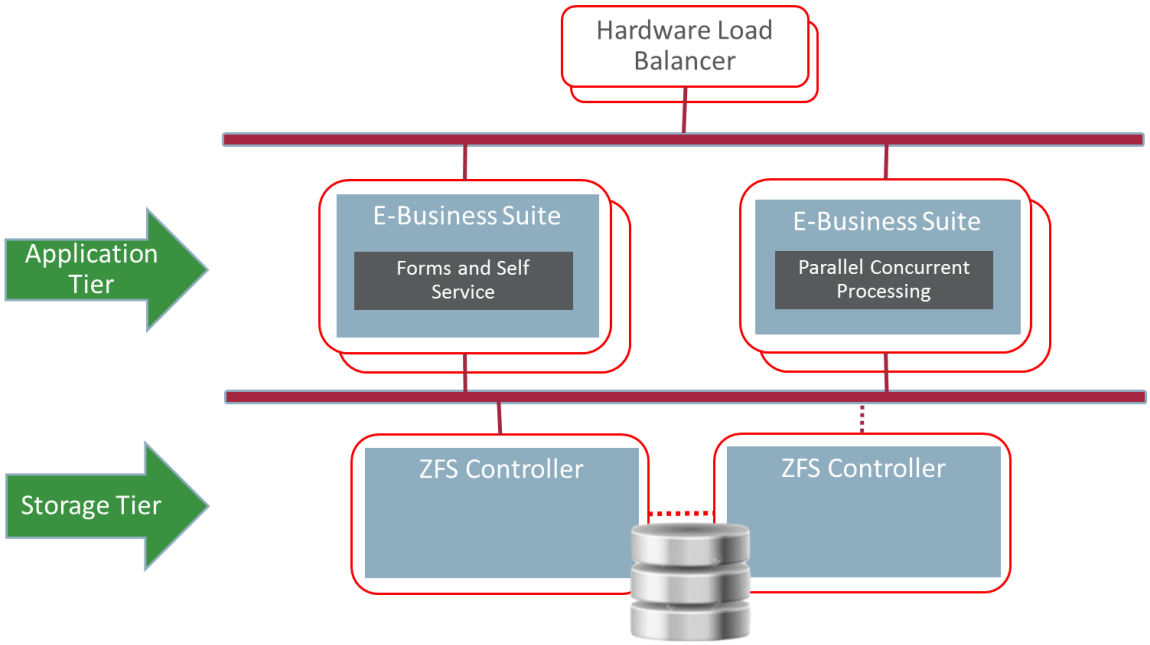

Figure 6 Two application tiers and ZFS filer

7.4.1 SMTP Server for Concurrent Processing

An SMTP server must be running on every application tier server that hosts a concurrent manager. In our project, we used postfix, a common Linux SMTP server, which is normally installed and configured by your system administrator.

Once your SMTP server is ready, perform the following tasks on each application tier server running a concurrent manager:

1. Edit the $CONTEXT_FILE and set the context variable s_smtphost to localhost.
2. Run AutoConfig.
3. Restart the application services.
4. Log into the application as SYSADMIN, then navigate to System Administrator→Oracle Application Manager then select: Workflow.
5. Verify on the display that the notification mailer is up. You should see a checkmark (green) and the word “UP” next to it.

7.5 Oracle E-Business Suite Database Tier – MAA

For this case study, similar hardware for the database tier was deployed for both the primary and secondary sites. The hardware at the primary site was a quarter-rack Exadata X3-2. The hardware at the DR site was a quarter-rack Exadata X4-2. Each quarter rack is made up of two compute nodes (database servers) and three storage cells.

Note that identical hardware was not available for these tests, but this configuration was an acceptable solution for our case study.

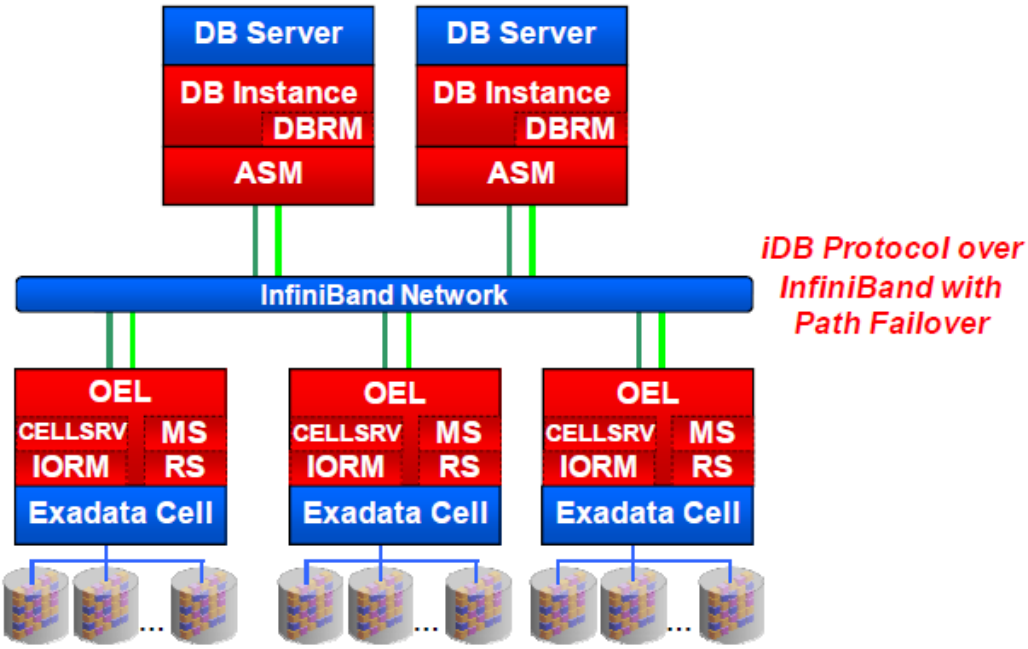

Figure 7 Block diagram showing the hardware and software layout of an Exadata quarter rack

Figure 7 shows the software stack for the database tier on Exadata. The software on the compute nodes comprises Oracle Database 12*c* (12.1.0.2) with Oracle RAC and Oracle Clusterware, Data Guard, RMAN, and Oracle ASM which manages the Exadata storage disks presented to the compute nodes. The storage cells run the Exadata storage software in an optimized Linux operating system, providing the smart scan, flash cache, flash logging, and hybrid columnar compression capabilities that differentiate the Exadata platform. iDB is a data transfer protocol implemented in the Oracle kernel and is built on the Reliable Datagram Sockets (RDS v3) protocol that runs over InfiniBand. A compute node can read data directly from memory in a given storage server over InfiniBand using Read Direct Memory Access (RDMA). This provides for zero copy of data and improves access time for overall performance improvements.

7.5.1 Database Setup

Whether you newly created or migrated to the new infrastructure, the Oracle E-Business Suite database should be configured following the Oracle E-Business Suite documentation, as well as implementing the Exadata best practices. For this case study, Oracle E-Business database for Release 12.2.5 on the primary site was cloned from the Oracle Applications Benchmark Kit onto Exadata using the cloning methods described in MOS doc "Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone (Doc ID 1383621.1)". Following that, Exadata best

practices for OLTP applications were implemented along with the recommendations and requirements specific to Oracle E-Business Suite, using the following references:

- MOS Doc "Oracle Sun Database Machine Setup/Configuration Best Practices (Doc ID1274318.1)"
- MOS Doc "Database Initialization Parameters for Oracle E-Business Suite Release 12 (Doc ID 396009.1)" The white paper Oracle E-Business Suite on Exadata Database Machine

The standard Exadata configuration was deployed as follows:

- ASM disk groups (DATA and RECO) with `HIGH` redundancy
- Oracle E-Business Suite database configured with Oracle RAC across both nodes of a quarter rack Exadata
- Database services registered in the Oracle Cluster Ready Services
- Pre-configured SCAN listener, for connection load balancing

7.5.2 Database Features for the Disaster Recovery Site

Once the primary site was established, the standby database was instantiated using the RMAN DUPLICATE feature, then Data Guard was configured with redo shipping from the primary and redo apply at the standby. In this case study, you can see that minimal configuration is required at the standby site. Using the database Snapshot Standby feature allows for testing at the secondary site without impacting the primary system.

7.6 Delegation of Roles and Administration

This section describes the operating system user accounts, groups, and the administrative role at each level: database tier on Exadata, the application, and web servers on Exalogic. These OS accounts, groups, and roles are consistent at both the primary and DR sites.

Our lab machine was not set up using full role separation as recommended earlier, a configuration we left in place as it was not important to our tests. However, many customers who do not implement full role separation do still want to have separate OS owners for their Grid Infrastructure and each database. We included that in our tests, to show it could be done, and describe those steps below.

7.6.1 Administrative Roles on Exadata Database Machine

On Oracle Exadata Database Machine, the Oracle Grid Infrastructure (Oracle Clusterware and ASM) manages all cluster and storage services. Although not required, it is recommended that the Oracle Grid Infrastructure is installed using a separate and dedicated OS user. Application databases should be installed into their own OS user account so that the Grid Infrastructure is managed separately from that of Oracle E-Business Suite application database. The following table illustrates how this was configured in this case study.

RECOMMENDED ORACLE DATABASE SERVER OS USERS

OS User	OS Groups	Role
oracle	oinstall, dba	Clusterware and ASM administrator(Grid)
oracle_ebs	oinstall, dba	Oracle E-Business Suite database administrator

Table 6: Recommended Oracle Database Server OS Users

The `oinstall` and `dba` OS groups are common between the `oracle` OS user account and that of `oracle_ebs`. These groups are required to enable the `oracle_EBS` user managing the Oracle E-Business Suite database to access the ASM services.

7.6.2 Administrative Roles on Exalogic Machine

On Exalogic, the Oracle E-Business Suite application suite was installed into the OS user account `applmgr` on both the primary and the secondary, with group `oinstall` as shown in the following table:

RECOMMENDED APPLICATION SERVER OS USERS

OS User	OS Groups	Role
applmgr	oinstall	Oracle E-Business Suite Application

Table 7: Recommended Application Server OS Users

The users and groups were registered in NIS to facilitate the NFS v4 security model. NFS v4 is recommended for use with ZFS mounts for higher performance when accessed across multiple servers.

7.7 Oracle E-Business Suite Application and Web Tier Setup

In this case study, we implemented the Oracle E-Business Suite shared APPL_TOP using ZFS shares. All application tier nodes mount the same ZFS share mount point using NFSv4. The Fusion Middleware, RUN, PATCH, and Non-Edition subdirectories all reside on these ZFS shares.

We configured the two application tier servers to each have the same number of Forms and OACORE servers. We created two database services that load-balance across the RAC instances for the online connections: VIS_OACORE_ONLINE for self-service web connections and VIS_FORMS_ONLINE for Oracle Forms connections

We also configured parallel concurrent processing (PCP) on each application tier server. As all PCP connections made from a given application server must connect to a single database instance, we created one database service per instance: VIS_PCP_BATCH1 and VIS_PCP_BATCH2.

Table 8: Database Service Mapping illustrates how Oracle Forms, Web and PCP are associated with the database services. The database services are defined in Cluster Ready Services (CRS) on the database tier and are configured to be relocated to another Oracle RAC node if an Oracle RAC instance fails.

DATABASE SERVICE MAPPING

Application Service Type	Context Variable	Database Service Name	Database Instances
Self Service Web	s_apps_jdbc_connect_descriptor	VIS_OACORE_ONLINE	VIS1, VIS2
Forms	s_tools_twotask	VIS_FORMS_ONLINE	VIS1, VIS2
PCP App Node 1	s_cp_twotask	VIS_PCP_BATCH1	VIS1
PCP App Node 2	s_cp_twotask	VIS_PCP_BATCH2	VIS2

Table 8: Database Service Mapping

To change Self-Service (OACORE) to use the database service VIS_OACORE_ONLINE, follow these steps:

1. Create the database service.

```
srvctl add service -db vis_a -service VIS_OACORE_ONLINE -preferred
  "VIS1,VIS2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -
  failovermethod BASIC -failovertype SELECT -failoverretry 10 -
  failoverdelay 3
```

2. Edit the context file on each application tier node and change change the SERVICE_NAME to VIS_OACORE_ONLINE context variable s_apps_jdbc_connect_descriptor. For example:

```
<jdbc_url
  oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(A
  DDRESS_LIST=(LOAD_BALANCE=YES)(FAILOVER=YES)(ADDRESS=(PROTOCOL=tcp)(HOST=
  exaa-
  scan3)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VIS_OACORE_ONLINE)))</jdbc
  _url>
```

3. Set the context variable `s_jdbc_connect_descriptor_generation` value to “false”.

4. Run AutoConfig.

5. Restart the application services for the changes to take effect.

To validate that the new database service is being used, connect as a DBA privileged user (SYS) in SQL*Plus and execute the following statement, which shows how many client connections there are per service name, by instance number:

```
select inst_id,service_name, count(*)
from gv$session
where service_name not like 'SYS%'
group by inst_id,service_name
order by 1
/
```

Example output:

```
   INST_ID INSTANCE_NAME    SERVICE_NAME           COUNT(*)
---------- ---------------- -------------------- ----------
         1 VIS1             VIS_OACORE_ONLINE           583
         1 VIS1             VIS_PCP_BATCH1               46
         2 VIS2             VIS_OACORE_ONLINE           567
         2 VIS2             VIS_PCP_BATCH2                7
```

To reconfigure Oracle Forms and Parallel Concurrent Processing to use non-default database services, create the database service similar to the example above, then follow sections 4 and 5 in MOS Doc Configuring and Managing Oracle E-Business Suite Release 12.2.x Forms and Concurrent Processing for Oracle RAC (Doc ID 2029173.1).

7.8 Logical Host Names

Starting with Oracle E-Business Suite Release 12.2 AD/TXK.Delta 9, EBS supports the use of logical host names as a way to reduce the need for reconfiguring the application topology when cloning the application environment, restoring from backup, or switching over to a disaster recovery site. Logical host names are aliases you can give a server by adding an alias name to the /etc/hosts file. With logical host names, you can:

- Clone an Oracle E-Business Suite installation to another environment using the same logical host names, without having to run Rapid Clone
- Restore or recover from backups to servers that use the same logical host names without having to run Rapid Clone, thereby reducing downtime
- Reduce the time required for a switchover or failover to a disaster recovery site

Logical host names are only used internally within the Oracle E-Business Suite topology configuration, and are not registered in DNS. This creates the requirement to have a load balancer in front of the the application tier servers. The load balancer can be configured with either the application tier IP addresses or their DNS-assigned host names. The Oracle E-Business Suite external URLs, web entry points, and port number are set to use the hardware load balancer as the entry point.

The use of logical host names is one of the main driving principles of this case study and is used at both the primary and secondary sites. This paper provides detailed steps for updating an existing Oracle E-Business Suite Release 12.2 to use logical host names. This process uses Oracle E-Business Rapid Clone, but does not move any components such as the database or software.

Note: The Oracle RDBMS, Grid Infrastructure, and Data Guard software will continue to use the DNS-assigned (physical) host name. This is required for registering the database with the SCAN and local listeners, and for any database links that you may require.

8 Case Study Roadmap

The following sections detail the steps and configuration used, following MAA best practices, to create the topology and test it. These high-level steps are:

- Prepare the network
- Prepare the primary
- Prepare the standby
- Test the standby
- Test switchover
- Test failover
- Test a production EBS patching event on the standby

8.1 Prepare the Network

For the case study, we verified the following was in place:

- Reliable network connectivity between the primary and standby sites for both application tier file system replication and database redo transport.
- Sufficient network bandwidth to support continuous replication from the primary to the standby ZFS storage appliance during peak periods, including concurrent manager log and out files.

» Sufficient network bandwidth to support peak Data Guard redo traffic.

8.1.1 Set Up Global or Local Traffic Managers

F5 BigIP Local Traffic Manager (LTM) hardware load balancers were installed at each site to distribute traffic across the Oracle E-Business Suite Application and Web Servers. The LTMs continuously monitor the health of the application servers at both the TCP and the application layer. They will redirect traffic if a node failure is detected by either monitor.

We used two f5 BigIP Load Balancer Switches, one for the primary and one at the standby site for testing the snapshot standby, each configured with Local Traffic Manager (LTM), to direct traffic appropriately across the environments.

1. On the primary f5 load balancer, using the Nodes option below the Local Traffic Manager, add all of the production application tier nodes at the primary site.

   Note: Add the hosts either by their DNS-assigned host names or IP addresses (not the logical host names). We used IP addresses.

2. Create a separate EBS-specific health monitor to be used by the pool being defined in the next step.
3. On the primary f5 load balancer, create a pool called "`ebs_web_prod_pool`" in which the production application tier nodes will be configured.
4. Add the nodes defined above as members of this pool and set the service port - in this case, 8002.
5. Add the health monitor to this pool.
6. On the f5 load balancer, create a Virtual Server that defines the front-end access. Ours is called "ebs_web_prod".
7. If you are using LTM rather than a GTM, repeat the previous steps at the secondary site.
8. For the virtual server, we set the source IP address to 0.0.0.0/0, which allows any remote system to access this virtual server. Refer to your system network security guide for an appropriate range.
9. Set the destination address to the IP address of the load balancer itself. Your business may require additional names that point to the load balancer to be registered in the DNS.
10. Set the Service Port to 8002, which is the same as the node's service port based on our use of EBS's port pool 2. Note that for snapshot standby testing we alter the service port to 9090 to have a clear alternate path to the test application.
11. Under the Resources tab, add the "ebs_web_prod_pool" as the default pool.

There are other steps to perform for a complete f5 setup, including recommendations for EBS WAN-optimized settings. These are documented in the f5 deployment guides, found at https://f5.com/solutions/deployment-guides.

8.2 Prepare the Primary

8.2.1 Set up the Linux Groups and Users

We set up OS users and groups on both our primary and standby database servers as described in Delegation of Roles and Administration section.

The OS groups and users on the application tier servers were provisioned using NIS. The application tier servers and the ZFS filer are configured as NIS clients. This configuration allows each applicaton tier server to use NFSv4 to mount the ZFS shares where the application installation resides. For further details, see MOS Doc: How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment (Doc ID 1516025.1).

8.2.2 Set Up Logical Host Names of Database Servers

These steps convert an existing EBS database install to one using logical host names.

The basic steps to implement logical host names are given in Business Continuity for Oracle E-Business Suite Release 12.2 Using Logical Host Names with an Oracle 12c Physical Standby Database (Doc ID 2246690.1). The logical host names in this implementation are known only to the database and applicaton servers, and Oracle E-Business Suite. They are not registered in the DNS and no external entity will need to reference the logical host names.

To set up logical host names for the database servers, perform the steps in the following table.

Step	Server, User	Action
1	All database nodes exaadb03, exaabd04, root	Three changes need to be made to /etc/hosts on each database server. a) Add aliases that translate the public IP addresses to the new logical host names. The pattern for the public IP addresses is: `<IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain>` b) Add aliases that translate the client VIP addresses to the new logical host names. The pattern for the client VIP addresses is: `<IP Address> <DNS-registered VIP Name with domain> <DNS-registered VIP Name without Domain> <Logical VIP Name with domain> <Logical VIP Name without domain>` **Note:** To find, or verify, the DNS-registered VIP's IP addresses, issue: `srvctl config vip –n <assigned node name>` For example: `srvctl config vip –n exaadb03` c) Add entries that translate the cluster SCAN addresses to a set of logical SCAN names. The pattern for the cluster SCAN addresses is: `<IP Address> <Logical SCAN name1>` `<IP Address> <Logical SCAN name2>` `<IP Address> <Logical SCAN name3>` An example from our test system: `/etc/hosts from exaadb03:`

		cat /etc/hosts #### BEGIN Generated by Exadata. DO NOT MODIFY #### 127.0.0.1 localhost.localdomain localhost 10.23.27.130 exaadb03.mycompany.com exaadb03 **ebsdb1.mycompany.com ebsdb1** 10.23.27.131 exaabd04.mycompany.com exaabd04 ebsdb2.mycompany.com ebsdb2 10.23.52.48 exaa03-vip.mycompany.com exaa03-vip **ebsdb1-vip.mycompany.com ebsdb1-vip** 10.23.52.49 exaa04-vip.mycompany.com exaa04-vip **ebsdb2-vip.mycompany.com ebsdb2-vip** #### END Generated by Exadata #### ### SCAN IP addresses for EBS 12.2 **10.23.52.53 ebsdb-scan1** **10.23.52.54 ebsdb-scan2** **10.23.52.55 ebsdb-scan3**
2	All database nodes exaadb03, exaabd04, root	Make sure you can ping all the new host aliases from each of the database nodes. For example, in our environment we ran these ping commands on each Oracle RAC node: ping ebsdb1 ping ebsdb2 ping ebsdb1-vip ping ebsdb2-vip ping ebsdb-scan1 ping ebsdb-scan2 ping ebsdb-scan3

8.2.3 Set Up Logical Host Names for the Application Tier Servers

Step	Server, User	Action
1	Primary application tier node exalgacn23, applmgr	Make sure you have at least 6G disk space available in <INST_TOP>.
2	All application tier nodes, applmgr	Apply required patches: » AD/TXK Delta 9 (for logical hostname support) » Patch 25178222 (R12.AD.C.Delta.9) » Patch 25180736 (R12.TXK.C.Delta.9) » Run ETCC on all middle tiers; apply all required patches

3	All application tier nodes, applmgr	Run AutoConfig on all app tier nodes, for both the run and the patch file systems. On our server: `. /u06/app/ebs/vis/EBSapps.env` `R` `adautocfg.sh` `. /u06/app/ebs/vis/EBSapps.env` `P` `adautocfg.sh` You will get an error when running AutoConfig against the patch file system, as it will try to connect to an Edition that is not set up. This can be ignored.
4	Primary application tier node, applmgr	Do a basic sanity check for adop – from the run file system, execute `adop -validate`
5	Primary application tier node, applmgr	Run "Update current view snapshot" in AD Administration as follows From the run file system, execute `adadmin`. Choose `2. Maintain Applications Files menu` Choose `4. Maintain snapshot information` Choose `2. Update current view snapshot`
6	Primary application tier node, applmgr	Synchronize the appsutil directory on the database tier nodes (required when upgrading to latest AD/TXK)
7	All database tier nodes, oracle_ebs	Run AutoConfig on all database tier nodes
8	Master (primary) application tier node, applmgr	Run adpreclone.pl on the RUN edition file system on the primary application tier node as follows. **Note:** This needs to be done before cloning the database tier to itself so that there are no issues connecting to the database and gathering the required data. `echo $FILE_EDITION` `run` `cd <INST_TOP>/admin/scripts` `perl adpreclone.pl appsTier`
9	All application tier nodes, applmgr	Shut the application down: `adstpall.sh`

10	All application tier nodes exalgacn23, exalgacn24, root	Three changes need to be made to /etc/hosts on each application server are: a) Add aliases that translate the public IP addresses of the app server to the new logical host names. As on the database tier, the logical host names can be before or after the physical host names as long as EBS patch 25057668 has been applied. The pattern we used for the public IP addresses is: `<IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain>` An example from our test system (exalgacn23): `10.25.29.199 exalgacn23.mycompany.com exalgacn23` **`ebsapp1.mycompany.com ebsapp1`** `10.25.29.200 exalgacn24.mycompany.com exalgacn24` **`ebsapp2.mycompany.com ebsapp2`** b) Add aliases that translate the database servers' client VIP addresses to the new logical host names. The pattern for the client VIP addresses is: `<IP Address> <Logical VIP Name with domain> <Logical VIP Name without domain>` Use the same IP addresses and VIPs as on the database tier. Note that these names are only located in /etc/hosts – they do not need to be registered in the DNS. c) Add entries that translate the database cluster SCAN addresses to a set of logical SCAN names. The pattern for the cluster SCAN addresses is: `<IP Address> <Logical SCAN name1>` `<IP Address> <Logical SCAN name2>` `<IP Address> <Logical SCAN name3>` Here is an example of the entries for 2) and 3): `### Added by Darryl Presley / Lyn Pratt -- logical host names for EBS 12.2.5` `### NOTE: The following host names are only in /etc/hosts and NOT in DNS.` `# DB Tier` `10.23.27.130 ebsdb1.mycompany.com ebsdb1` `10.23.27.131 ebsdb2.mycompany.com ebsdb2` `10.23.52.48 ebsdb1-vip.mycompany.com ebsdb1-vip` `10.23.52.49 ebsdb2-vip.mycompany.com ebsdb2-vip` `### SCAN IP addresses for EBS 12.2` **`10.23.52.53 ebsdb-scan1`** **`10.23.52.54 ebsdb-scan2`** **`10.23.52.55 ebsdb-scan3`**
11	All application tier nodes, any user	Make sure you can ping the new host alias from all noes you have added these logical names to, for example: `ping ebsapp1` `ping ebsapp2` `ping ebsdb-scan1`

		ping ebsdb-scan2 ping ebsdb-scan3

8.2.4 Clone the Database to Itself Using EBS Rapid Clone Methodology

This task is necessary for an existing installation that was created using a physical host name – not the new logical host name. The goal is to simply reconfigure the database home and contents to use the new logical host name. Using the Rapid Clone tooling enables us to configure Oracle E-Business Suite's appsutil directory in the database Oracle home for the new logical host name. Because we do not have to copy the database, we only need to refer to the Cloning the Database Separately topic in the Advanced Cloning Scenarios section of MOS Doc Cloning Oracle E-Business Suite Release 12.2 RAC Enabled Systems with Rapid Clone (Doc ID 1679270.1), and have skipped steps that are not required for this scenario. The steps that we performed are shown in the following table.

Step	Server, User	Action
1	One database node exaadb03, oracle_ebs	Run adpreclone.pl: `$ cd $ORACLE_HOME/appsutil/scripts/[context_name]` `$ perl adpreclone.pl dbTier`
2	One database node exaadb03, oracle_ebs	Save the contents of the $TNS_ADMIN directory, as a later step may remove the network configuration files. `cd $TNS_ADMIN/..` `cp -r $TNS_ADMIN $TNS_ADMIN.backup`
3	All database nodes exaadb03, exaabd04, oracle_ebs	Create a text file `$ORACLE_HOME/appsutil/clone/pairsfile.txt` for each node with the below contents: `s_undo_tablespace=[instance tablespace name]` `s_dbClusterInst=[total number of instances in cluster]` `s_db_oh=[value of $ORACLE_HOME]` On exaadb03: `s_undo_tablespace=APPS_UNDOTS1` `s_dbClusterInst=2` `s_db_oh=/u01/app/oracle_ebs/product/12.1.0` On exaabd04: `s_undo_tablespace=UNDOTBS2` `s_dbClusterInst=2` `s_db_oh=/u01/app/oracle_ebs/product/12.1.0`
4	All database nodes	Shut the database down.

5	1st database node	Navigate to [ORACLE_HOME]/appsutil/clone/bin and run adclonectx.pl with the following parameters: `$ perl adclonectx.pl \` `contextfile=[ORACLE_HOME]/appsutil/[current context file] \` `template=[ORACLE_HOME]/appsutil/template/adxdbctx.tmp \` `pairsfile=[ORACLE_HOME]/appsutil/clone/pairsfile.txt initialnode` Here is the conversation from our system: `$ perl adclonectx.pl \` `contextfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/VIS1_exaadb03.xml \` `template=/u01/app/oracle_ebs/product/12.1.0/appsutil/template/adxdbctx.tmp \` `pairsfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/clone/pairsfile.txt initialnode` `Provide the values required for creation of the new Database Context file.` `Target System Hostname (virtual or normal) [exaadb03] : ebsdb1` `Target Instance is RAC (y/n) [y] : Y` `Target System Database Name : VIS` `Do you want to enable SCAN addresses (y/n) [y] ? : Y` `Specify value for Scan Name : exaa-scan3` `Specify value for Scan Port : 1521` `Do you want the target system to have the same port values as the source system (y/n) [y] ? : Y` `Target System Port Pool [0-99] : 2` `Report file located at /u01/app/oracle_ebs/product/12.1.0/appsutil/out/portpool.lst` `Provide information for the initial RAC node:` `Host name [testdb1] : ebsdb1` `Virtual Host name [null] : ebsdb1-vip` `Instance number [1] : 1` `Private interconnect name [exaadb03] : exaadb03-priv` `Target System Base Directory : /u01/app/oracle_ebs` `Oracle OS User [oracle_ebs] :` `Oracle OS Group [oinstall] : dba` `Target System utl_file_dir Directory List : /u01/app/oracle_ebs/product/12.1.0/appsutil/outbound/VIS1_ebsdb1` `Number of DATA_TOP's on the Target System [4] : 1` `Target System DATA_TOP Directory 1 : +DATA_EXAA`

		Do you want to preserve the Display [scl05kss:2] (y/n) : Y New context path and file name [VIS1_ebsdb1.xml] : The new database context file has been created : /u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin/VIS1_ebsdb1.xml contextfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin/VIS1_ebsdb1.xml Check Clone Context logfile /u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin/CloneContext_1115105031.log for details.
6	1st database node	Check the new XML file, paying particular attention to the following entries. In our tests: SID = VIS1; good s_dbCluster = TRUE; good RAC prefix vis; good RAC_nodes is empty; expected SCAN name and port number should be set; good s_update_scan = true; good
7	1st database node	When adclonectx.pl is complete, copy the new context file to the $ORACLE_HOME/appsutil directory: cp VIS1_ebsdb1.xml $ORACLE_HOME/appsutil/.
8	1st database node exaadb03 oracle_ebs	Start the instance in open mode. srvctl start instance –d vis_a –i VIS1 Note that you may see errors at this point if you have Data Guard configured, because the original network configuration files may have been removed. These errors can be ignored at this time, but they will be dealt with in subsequent steps.
9	1st database node exaadb03 oracle_ebs	The generic configuration instruction for the initial node is: $ cd [RDBMS ORACLE_HOME]/appsutil/clone/bin $ perl adcfgclone.pl dbconfig [Database target context file] This is the command we executed on our system: $ cd /u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin $ perl adcfgclone.pl dbconfig /u01/app/oracle_ebs/product/12.1.0/appsutil/VIS1_ebsdb1.xml
10	1st database node exaadb03 oracle_ebs	Log out and log back in. Re-establish your Linux environment. In our environment, we need to source: . /u01/app/oracle_ebs/product/12.1.0/VIS1_ebsdb1.env
11	1st database node exaadb03 oracle_ebs	Run AutoConfig.

12	1st database node exaadb03 oracle_ebs	Archive the [ORACLE_HOME]/appsutil directory structure. This zipped archive will contain preclone content needed by the remaining nodes. Create the zip archive as follows: `$ cd $ORACLE_HOME` `$ zip –r appsutil_node1.zip appsutil`
13	All remaining Oracle RAC database nodes	Copy (transfer) the appsutil_node1.zip to the remaining Oracle RAC nodes into [ORACLE_HOME]/appsutil_node1.zip and unzip it. Doing so will overwrite existing content. If you wish to keep a copy of the current appsutil directory structure then back it up first. To extract the appsutil_node1.zip: `$ cd $ORACLE_HOME` `$ unzip –o appsutil_node1.zip`
14	2nd node exaabd04	Create a pairsfile.txt for the second node and make sure the contents of the pairsfile.txt for the second node is correct. For example: On exaabd04: `s_undo_tablespace=UNDOTBS2` `s_dbClusterInst=2` `s_db_oh=/u01/app/oracle_ebs/product/12.1.0`
15	2nd node exaabd04	Create a context file for the secondary node, specifying the `addnode` option. The generic command is: `perl adclonectx.pl \` `  contextfile=[Full Path to Existing Context File on First Node]/<contextfile>.xml \` `  template=[NEW ORACLE_HOME]/appsutil/template/adxdbctx.tmp \` `  pairsfile=[NEW ORACLE_HOME]/appsutil/clone/pairsfile.txt addnode` The command as run on our system: `perl adclonectx.pl contextfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/VIS1_ebsdb1.xml template=/u01/app/oracle_ebs/product/12.1.0/appsutil/template/adxdbctx.tmp pairsfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/clone/pairsfile.txt addnode` Answers to prompts are as follows: `Provide the values required for creation of the new Database Context file.` `Target System Hostname (virtual or normal) [exaabd04] : ebsdb2` `Target Instance is RAC (y/n) [y] : Y` `Please provide the details to connect to one of live RAC nodes`

		Host name of the live RAC node : ebsdb1 Domain name of the live RAC node : mycompany.com Database SID of the live RAC node : VIS1 Listener port number of the live RAC node : 1523 Provide information for the new Node: Host name : ebsdb2 Virtual Host name : ebsdb2-vip Instance number : 2 Private interconnect name : exaabd04-priv Current Node: Host Name : ebsdb2 SID : VIS2 Instance Name : VIS2 Instance Number : 2 Instance Thread : 2 Undo Table Space: Listener Port : 1523 Target System Base Directory : /u01/app/oracle_ebs Oracle OS User [oracle_ebs] : Oracle OS Group [oinstall] : dba Target System utl_file_dir Directory List : /u01/app/oracle_ebs/product/12.1.0/appsutil/outbound/VIS2_ebsdb 2 Do you want to preserve the Display [scl05kss:2] (y/n) : Y New context path and file name [VIS2_ebsdb2.xml] : The new database context file has been created : /u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin/VIS2_ebsd b2.xml contextfile=/u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bi n/VIS2_ebsdb2.xml Check Clone Context logfile /u01/app/oracle_ebs/product/12.1.0/appsutil/clone/bin/CloneCont ext_1115131015.log for details.
16	2nd node	Copy the context file to $ORACLE_HOME/appsutil.

17	2nd node	Start the instance on the 2nd node.
18	2nd node	Use `adcfgclone.pl dbconfig` to configure the Oracle home. We ran the command:: `perl adcfgclone.pl dbconfig /u01/app/oracle_ebs/product/12.1.0/appsutil/VIS2_ebsdb2.xml`
19	2nd node	Source the new environment file: `cd [ORACLE_HOME]` `./[CONTEXT_NAME].env`
20	2nd node	Run AutoConfig: `cd $ORACLE_HOME/appsutil/scripts/$CONTEXT_NAME` `./adautocfg.sh appspass=[APPS password]`
21	1st node	Source the new environment file again, then run AutoConfig once more. This populates the information from the second node into the TNS configuration on the file system.
22	All database nodes	Generate the tnsnames.ora ifiles. See Appendix B: Scripts for Managing SCAN for Primary and DR Sites for a script we built for this work. On each of the database nodes: `./ebs_scan_db.sh` `SCAN Name: exaa-scan3,` `Add the following entries to your /etc/hosts files on each RAC node in THIS cluster.` `## SCAN VIPs` `10.23.52.53 ebsdb-scan1` `10.23.52.54 ebsdb-scan2` `10.23.52.55 ebsdb-scan3` Then, look at the $TNS_ADMIN directory and notice that a new ifile was created for example: VIS1_ebsdb1_ifile.ora.
23	All database nodes	Create a directory under the EBS database's $TNS_ADMIN with the base logical host name pattern (in our case, ebsdb). The directory must have the same name on all databasenodes. On our environment: `mkdir /u01/app/oracle_ebs/product/12.1.0/network/admin/ebsdb` Under that directory, on each node, create three files – sqlnet.ora, listener.ora, and tnsnames.ora – that each simply point to the AutoConfig-generated-generated files for that node. Sqlnet.ora: `IFILE=/u01/app/oracle_ebs/product/121.0/network/admin/VIS[1|2]_testdb[1|2]/sqlnet.ora`

		Listener.ora: `IFILE=/u01/app/oracle_ebs/product/121.0/network/admin/VIS[1|2]_testdb[1|2]/listener.ora` tnsnames.ora: `IFILE=/u01/app/oracle_ebs/product/121.0/network/admin/VIS[1|2]_testdb[1|2]/tnsnames.ora` Configure srvctl to use these files, pointing to the new directory name: `srvctl setenv database –d vis_a –t "TNS_ADMIN=/u01/app/oracle_ebs/product/12.1.0/network/admin/ebsdb"` This configuration step should already be in place, but it is included here for your information: `srvctl setenv database –d vis_a –t "ORA_NLS10=/u01/app/oracle_ebs/product/12.1.0/nls/data/9idata"` Restart both database instances, using srvctl: `srvctl stop database –d vis_a –o immediate` `srvctl start database –d vis_a`

8.2.5 Clone the Application Tier to Itself Using EBS Rapid Clone Methodology

It is a best practice to use a new, neutral host name for your logical host name, as opposed to attempting to retain your primary's physical host name and use that as your logical host name. This section describes the steps required to reconfigure an existing installation to use a new logical host name, without having to create a new copy. In this case, we used the Rapid Clone tooling, which allows us to reconfigure the application tier host name.

Step	Server, User	Action
1	All application mid tiers (exalgacn23, exalgacn24), applmgr	In order to clone an application tier to itself, all application services must first be shut down. Log on to each application tier node, source the run environment, then shut down all application services. `$ cd $ADMIN_SCRIPTS_HOME` `$ adstpall.sh`
2	Primary application tier node exalgacn23, applmgr	There are two options here for managing the Oracle inventory: a) force EBS tooling to create a completely new inventory for this installation, or b) deregister the appropriate middle tier homes. a) Generate a completely new inventory. For this option, your inventory must be unique to the EBS application tier environment. See MOS doc R12.2 : How To Create, Update or Rebuild The Central Inventory For Oracle Applications E-Business Suite ? (Doc ID 1588609.1) and MOS doc Oracle E-Business Suite

		Applications DBA and Technology Stack Release Notes for R12.AD.C.Delta.7 and R12.TXK.C.Delta.7 (Doc ID 2033780.1) (section 4.5) for more information. Instead of detaching the 10.1.2 Oracle homes, force the EBS tooling to create a completely new inventory by "hiding" the current one: `cd <location of oraInventory>` `mv oraInventory oraInventory.backup` b) Deregister the appropriate application tier homes. If you do not use an inventory exclusive to this EBS middle tier, you must instead detach the Oracle Forms and Reports home (10.1.2) in both the run and the patch file system: **Note:** When deregistering ORACLE_HOMEs, you must run runInstaller from the ORACLE_HOME that is being detached. Make sure the ORACLE_HOME environment variable is set to that ORACLE_HOME. `cd $ORACLE_HOME/oui/bin` `./runInstaller -silent -removeHome` `ORACLE_HOME=/u06/app/ebs/vis/fs2/EBSapps/10.1.2 -invPtrLoc` `/u06/app/ebs/vis/oraInventory/oraInst.loc` `cd /u06/app/ebs/vis/fs1/EBSapps/10.1.2` `cd oui/bin` `./runInstaller -silent -removeHome` `ORACLE_HOME=/u06/app/ebs/vis/fs1/EBSapps/10.1.2 -invPtrLoc` `/u06/app/ebs/vis/oraInventory/oraInst.loc`
3	Primary application tier node, applmgr	Move the FMW Oracle homes, INST_TOP, and PATCH file system directories so they appear to be non-existant to the application. Leave the 10.1.2 home in place. To identify which directory is the PATCH directory, echo the environment variable `$PATCH_BASE`. On our system (fs1 is PATCH and fs2 is RUN): `mv /u06/app/ebs/vis/fs2/FMW_home` `/u06/app/ebs/vis/fs2/FMW_HOME_bkp` `mv /u06/app/ebs/vis/fs2/inst /u06/app/ebs/vis/fs2/inst_bkp` `mv /u06/app/ebs/vis/fs1 /u06/app/ebs/vis/fs1_bkp`
4	Primary application tier node, applmgr	Log off and back on again. Set PERL5LIB and your path to find and execute the proper version of Perl. For example: `export PERL5LIB=/usr/lib64/perl5` Make sure /usr/bin is first in the PATH to run Perl. `export PATH=/usr/bin:$PATH`
5	Primary application tier node	Configure the application tier to use the new logical host name: `$ cd <COMMON_TOP>/clone/bin` `$ perl adcfgclone.pl appsTier dualfs`

	exalgacn23, applmgr	adcfgclone Version 120.63.12020000.56 Enter the APPS password : Enter the Weblogic AdminServer password : Do you want to add a node (yes/no) [no] : Running: Context clone... Log file located at /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/CloneContext_112215 2645.log Provide the values required for creation of the new APPL_TOP Context file. Target System Hostname (virtual or normal) [ebsapp1] : Target System Database SID : VIS Target System Database Server Node [testapp1] : ebsdb1 Target System Database Domain Name [mycompany.com] : Target System Base Directory : /u06/app/ebs/vis Target System Base Directory set to /u06/app/ebs/vis Target System Current File System Base set to /u06/app/ebs/vis/fs2 Target System Other File System Base set to /u06/app/ebs/vis/fs1 Target System Fusion Middleware Home set to /u06/app/ebs/vis/fs2/FMW_Home Target System Other File System Fusion Middleware Home set to /u06/app/ebs/vis/fs1/FMW_Home Target System Web Oracle Home set to /u06/app/ebs/vis/fs2/FMW_Home/webtier Target System Other File System Web Oracle Home set to /u06/app/ebs/vis/fs1/FMW_Home/webtier Target System Appl TOP set to /u06/app/ebs/vis/fs2/EBSapps/appl Target System Other File System Appl TOP set to /u06/app/ebs/vis/fs1/EBSapps/appl Target System COMMON TOP set to /u06/app/ebs/vis/fs2/EBSapps/comn Target System Other File System COMMON TOP set to /u06/app/ebs/vis/fs1/EBSapps/comn Target System Instance Home Directory [/u06/app/ebs/vis] : Target System Current File System Instance Top set to /u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1

```
Do you want to preserve the Display [slc05kss:2] (y/n)  : Y

Target System Root Service [enabled] :

Target System Web Entry Point Services [enabled] :

Target System Web Application Services [enabled] :

Target System Batch Processing Services [enabled] :

Target System Other Services [disabled] :

Do you want the target system to have the same port values as
  the source system (y/n) [y] ? : Y
Complete port information available at
  /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/out/VIS_ebsapp1/por
  tpool.lst

UTL_FILE_DIR on database tier consists of the following
  directories.

1. /tmp
2. /tmp
3.
  /u01/app/oracle_ebs/product/12.1.0/appsutil/outbound/VIS_exaadb
  03
4. /tmp
Choose a value which will be set as APPLPTMP value on the target
  node [1] :
The new APPL_TOP context file has been created :

  /u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsap
  p1.xml
Check Clone Context logfile
  /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/CloneContext_112215
  2645.log for details.

Creating Patch file system context file.....

Log file located at
  /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/CloneContextPatch_1
  122152858.log

Target System Other File System Instance Top set to
  /u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1
Complete port information available at
  /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/out/VIS_ebsapp1/por
  tpool.lst
The new APPL_TOP context file has been created :

  /u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsap
  p1.xml
Check Clone Context logfile
  /u06/app/ebs/vis/fs2/EBSapps/comn/clone/bin/CloneContextPatch_1
  122152858.log for details.
```

		FMW Pre-requisite check log file location : /u06/app/ebs/vis/fs2/EBSapps/comn/clone/FMW/logs/prereqcheck.log Running: FMW pre-req check... Configuring: Run file system.... LogFile located at /u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/admin/log/clone/run/RCloneApplyAppstier_11221529.log
6	Primary application tier exalgacn23, applmgr	Establish the new environment by sourcing the EBSapps.env; specify the RUN environment.
7	Primary application tier node exalgacn23, applmgr	Set the web entry point. As the logical host names do not exist in the DNS, we must access the application using the load balancer. Therefore, we must re-set the web entry point in the FND tables. See Appendix C: Set Web Entry Point for an example script.
8	Primary application tier node, applmgr	Generate the tnsnames.ora ifile entries for using the database logical host names. See the ebs_scan_app.sh script in Appendix B: Scripts for Managing SCAN for Primary and DR Sites for a sample script to generate the ifiles. Do this on both the run and the patch file systems. On our system running ebs_scan_app.sh on the RUN file system: $ ./ebs_scan_app.sh Add the following entries to your /etc/hosts files on each RAC node in THIS cluster. ## SCAN VIPs 10.23.52.53 ebsdb-scan1 10.23.52.54 ebsdb-scan2 10.23.52.55 ebsdb-scan3 Then, look at the new ifile in $TNS_ADMIN. For example the new file might be VIS_ebsapp1_ifile.ora
9	Primary application tier node, applmgr	On both the run and the patch file systems, review the context file and correct the connection data if needed (**Note:** no changes were needed with AD/TXK Delta 9): <patch_jdbc_url oa_var="s_apps_jdbc_patch_connect_descriptor"> jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host name>)(PORT=<dbport>))(CONNECT_DATA=(SERVICE_NAME=ebs_patch)(INSTANCE_NAME=<RAC Instance Name>)) </patch_jdbc_url> On our system:

		Run Edition: `<jdbc_url oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=YES)(FAILOVER=YES)(ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VIS)))</jdbc_url>` `<jdbc_url_generation_check oa_var="s_jdbc_connect_descriptor_generation">true</jdbc_url_generation_check>` `<patch_jdbc_url oa_var="s_apps_jdbc_patch_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ebs_patch)(INSTANCE_NAME=VIS1)))</patch_jdbc_url>` Patch Edition: `<jdbc_url oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=YES)(FAILOVER=YES)(ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VIS)))</jdbc_url>` `<jdbc_url_generation_check oa_var="s_jdbc_connect_descriptor_generation">true</jdbc_url_generation_check>` `<patch_jdbc_url oa_var="s_apps_jdbc_patch_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ebs_patch)(INSTANCE_NAME=VIS1)))</patch_jdbc_url>`
10	One Oracle RAC database node, log into the database as the APPS user	Clean out the FND_NODES table by running the following command in SQL*Plus as the APPS user: `SQL> execute fnd_conc_clone.setup_clean;`
11	One Oracle RAC database nodes, oracle_ebs	On one database node, check the context variable `s_update_scan`. If it is set to FALSE, change it to TRUE. Then, run AutoConfig on this database node.
12	All remaining Oracle RAC database nodes	Run AutoConfig on all remaining database nodes.

13	Primary application tier node exalgacn23, applmgr	Run AutoConfig on the middle tier for both RUN and PATCH file systems. AutoConfig will complete with errors on the PATCH file system. This is expected as there is no open patch cycle.
14	Primary application tier node exalgacn23, applmgr	Validate the setup: Start the admin server on the run file system, then validate the configuration: `adadminsrvctl.sh start` `adop –validate`
15	Primary application tier node, applmgr	Start application services: `adstrtal.sh` Then log into the application to verify you can connect.
16	All slave application tier nodes	Add back any slave application tier nodes. We have provided template scripts in Appendix D: Add Managed Services that are based on instructions in the following MOS documents: Sharing the Application Tier File System in Oracle E-Business Suite Release 12.2 (Doc ID 1375769.1) Cloning Oracle E-Business Suite Release 12.2 with Rapid Clond (Doc ID 1383621.1) Managing Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 (Doc ID 1905593.1)

8.2.6 Configure the Primary Database for Oracle Data Guard

With the primary site fully configured to use logical host names, we can proceed with preparing the primary database nodes for Oracle Data Guard. Two MOS Docs are referenced in these steps: Business Continuity for Oracle E-Business Suite Release 12.2 Physical Standby using Virtual Hosts (Doc ID 2088692.1) and Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC) (Doc ID 1617946.1).

Step	Server, User	Action
1	One database node exaadb03, oracle_ebs	Enable force logging if it is not already enabled. To enable force logging, issue this command in SQL*Plus: `SQL> alter database force logging;`
2	Each Oracle RAC database node, oracle_ebs	**Note:** This step does not apply if the Grid Listener is being used instead of EBS specific local listeners. Modify the local EBS-specific listener. The process of cloning the database nodes to themselves updates the listener.ora and tnsnames.ora in the $TNS_ADMIN subdirectory to use the logical host names (ebsdb1 and ebsdb2). As we do not configure these logical names

		in the DNS, they are only resolvable on each of their respective nodes. To give remote clients the ability to connect to the database (including Data Guard Broker), both the listener and the database must use the DNS-configured host names. To modify the EBS listener and the database, log in to each Oracle RAC database node, source the EBS environment, and follow these steps: a) Copy the listener.ora to the listener_ifile.ora file. `$ cd $TNS_ADMIN` `$ cp listener.ora listener_ifile.ora` b) Edit the listener_ifile.ora and change the host name from the logical host name to the DNS assigned host name for each HOST parameter. On the first Oracle RAC node we changed ebsdb1 to exaadb03. We made similar changes on each node: `VIS =` `  (DESCRIPTION_LIST =` `    (DESCRIPTION =` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)(HOST = exaa03-vip.mycompany.com)(PORT = 1523)(IP = FIRST)))` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)(HOST = exaadb03.mycompany.com)(PORT = 1523)(IP = FIRST)))` `    )` `  )` `SID_LIST_VIS =` `  (SID_LIST =` `    (SID_DESC =` `       (ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0)(SID_NAME = VIS1)` `    )` `  )` `STARTUP_WAIT_TIME_VIS = 0` `CONNECT_TIMEOUT_VIS = 10` `TRACE_LEVEL_VIS = OFF` `LOG_DIRECTORY_VIS =` `  /u01/app/oracle_ebs/product/12.1.0/network/admin` `LOG_FILE_VIS = VIS1` `TRACE_DIRECTORY_VIS =` `  /u01/app/oracle_ebs/product/12.1.0/network/admin` `TRACE_FILE_VIS = VIS1` `ADMIN_RESTRICTIONS_VIS = ON` `SUBSCRIBE_FOR_NODE_DOWN_EVENT_VIS = OFF` c) Restart the listener on the Oracle RAC node that is being reconfigured. `$ lsnrctl stop vis` `$ lsnrctl start vis`

		d) Edit the <SID>_node_ifile.ora and edit the TNS connect string aliases <SID>_LOCAL (i.e., VIS1_LOCAL, VIS2_LOCAL) and change the HOST parameter to the assigned host name in DNS as in the following example. `VIS1_LOCAL=` `(DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)(HOST=`**`exaa03-vip`**`.mycompany.com)(PORT=1523))` `(ADDRESS=(PROTOCOL=tcp)(HOST=`**`exaa03-vip`**`.mycompany.com)(PORT=1521))` `)` `VIS2_LOCAL=` `(DESCRIPTION=` `(ADDRESS=(PROTOCOL=tcp)(HOST=`**`exaa04-vip`**`.mycompany.com)(PORT=1523))` `(ADDRESS=(PROTOCOL=tcp)(HOST=`**`exaa04-vip`**`.mycompany.com)(PORT=1521))` `)` e) Add the two TNS connect string aliases from step d) into the custom.txt file in case you run the ebs_scan_db.sh script in the future as this will preserve these entries. See Appendix A for the ebs_scan_db.sh script. f) Force the database to re-register to the local EBS listener on each Oracle RAC node. Log into SQL*Plus and issue the following commands: `SQL> alter system set local_listener='<SID_LOCAL>' sid='<SID>' scope=both;` For example: `SQL> alter system set local_listener='VIS1_LOCAL' sid='VIS1' scope=both;` `SQL> alter system set local_listener='VIS2_LOCAL' sid='VIS2' scope=both;`
3	Each Oracle RAC database node, oracle_ebs	Create password files on one of the Oracle RAC database nodes, then copy it to all of the other nodes. On our environment: `$ orapwd file=orapw$ORACLE_SID password=<your password> entries=10 ignorecase=y` Now copy it to node 2 in the cluster: `$ scp orapwVIS1 oracle_ebs@exaabd04:/u01/app/oracle_ebs/product/12.1.0/dbs/orapwVIS2`
4	Primary database node exaadb03, oracle_ebs	Review and set database parameters. Database parameters should be set per EBS requirements and also per best practice for Exadata. (**Note:** database initialization parameters are discussed more fully in the section Database Initialization Parameters.) **Note:** All Data Guard parameters will be set by Data Guard Broker after the standby database has been instantiated.

		The parameters should be set using a command such as this: `alter system set [parameter name]=[parameter value] scope=spfile` Our system parameters: `processes = 2500  -- per EBS requirements` `log_buffer=134217728` `compatible=12.1.0` `log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST'` `log_archive_format='%t_%s_%r.dbf'` `db_block_checking='MEDIUM'` `db_block_checksum=FULL` `db_lost_write_protect=TYPICAL` `db_recovery_file_dest=+RECO_EXAA` `db_file_create_dest=+DATA_EXAA` `db_file_create_dest_size=1000G` `global_names=TRUE` `fast_start_mttr_target=300` `parallel_adaptive_multi_user=FALSE` `parallel_threads_per_cpu=1` `filesystemio_options=setall` `open_cursors=1000` `use_large_pages=ONLY` `sga_target=12G` `pga_aggregate_target=10G`
5	Primary database node exaadb03, oracle_ebs	Enable ARCHIVELOG mode if it is not already active. This requires some down time as you must shut down all Oracle RAC nodes before it can be enabled, as in the following steps. a) First, shut down all Oracle RAC instances but one. b) On the remaining Oracle RAC instance, from SQL*Plus as sysdba: `shutdown immediate;` `startup mount` `alter database archivelog;` `alter database open;` c) Finish by restarting all other Oracle RAC instances.
6	Primary database node exaadb03, oracle_ebs	Add standby redo logs to the primary. RMAN duplicate clones the standby logs. Create standby logs the same size as online redo logs, with one more log group for each thread than the number of online log groups. The log files do not need to be multiplexed if the disk group is configured with high redundancy. Our primary database has 2 threads. Each thread has 4 redo log groups, and each member is 2G in size. Therefore, we need 5 standby redo logs for each thread for a total of 10. `alter database add standby logfile thread 1` `group 11 size 2G,` `group 12 size 2G,` `group 13 size 2G,` `group 14 size 2G,` `group 15 size 2G;` `alter database add standby logfile thread 2`

		group 16 size 2G, group 17 size 2G, group 18 size 2G, group 19 size 2G, group 20 size 2G;

8.3 Prepare the Standby

8.3.1 Set Up Logical Host Names for Standby Database Tier

Configure the standby database servers to use the same logical host names as set up on the primary database servers.

Step	Server, User	Action
1	All database nodes exabadm01, exabadm02 root	Three changes need to be made to /etc/hosts on each database server: a) Add aliases that translate the public IP addresses to the new logical host names. The pattern for the public IP addresses is: `<IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain>` b) Add aliases that translate the client VIP addresses to the new logical host names. The pattern for the client VIP addresses is: `<IP Address> <DNS-registered VIP Name with domain> <DNS-registered VIP Name without Domain> <Logical VIP Name with domain> <Logical VIP Name without domain>` **Note:** To find, or verify, the DNS-registered VIP's IP addresses, issue: `srvctl config vip –n <assigned node name>` For example: `srvctl config vip –n exabadm01` c) Add entries that translate the cluster SCAN addresses to a set of logical SCAN names. The pattern for the cluster SCAN addresses is: `<IP Address> <Logical SCAN name1>` `<IP Address> <Logical SCAN name2>` `<IP Address> <Logical SCAN name3>` An example from our test system: `/etc/hosts from exabadm01:` `cat /etc/hosts` `#### BEGIN Generated by Exadata. DO NOT MODIFY ####` `127.0.0.1 localhost.localdomain localhost`

		20.35.11.45 exabadm01.mycompany.com exabadm01 **ebsdb1.mycompany.com ebsdb1** 20.35.11.46 exabadm02.mycompany.com exabadm02 **ebsdb2.mycompany.com ebsdb2** ## VIPs added for EBS 12.2 20.35.22.221 exabclient01-vip.mycompany.com exabclient01-vip **ebsdb1-vip.mycompany.com ebsdb1-vip** 20.35.22.222 exabclient02-vip.mycompany.com exabclient02-vip ebsdb2-vip.mycompany.com ebsdb2-vip #### END Generated by Exadata #### #### ### SCAN IP addresses for EBS 12.2 **20.35.22.229 ebsdb-scan1** **20.35.22.230 ebsdb-scan2** **20.35.22.231 ebsdb-scan3**
2	All database nodes exabadm01, exabadm02 root	Set nsswitch.conf up to resolve network address lookups in files before the DNS, so the adjustments to /etc/hosts will take precedence. Modify the /etc/nsswitch.conf – the line that starts with "host" needs to have the directive "files" first – for example: host files dns
3	Each database node exabadm01, exabadm02 Any user	From each node, ping another node using your new logical names, to be sure that they can be resolved. E.g., from exabadm02, we ran this ping command: ping ebsdb1

8.3.2 Set Up Logical Host Names and Configure Standby Middle-Tiers

Logical host names must be configured the same as on the primary.

Step	Server, User	Action
1	exalgbvm0089-eoib2 and exalgbvm0090-eoib2, root	Edit /etc/hosts to add the logical host name. Make sure the logical host name is listed first as follows. For the first application tier node: # Do not remove the following line, or various programs # that require network functionality will fail. 127.0.0.1 localhost.localdomain localhost ::1 localhost6.localdomain6 localhost6 172.27.1.208 EBS-APP01-172-27-1-208

		20.23.25.55 **ebsapp1.mycompany.com ebsapp1** exalgbvm0089-eoib2.mycompany.com exalgbvm0089-eoib2 EBS-APP01-10-133-235-55 20.23.25.56 exalgbvm0090-eoib2.mycompany.com exalgbvm0090-eoib2 ebsapp2.us.oracle.clm ebsapp2 For the second application tier node: # Do not remove the following line, or various programs # that require network functionality will fail. 127.0.0.1 localhost.localdomain localhost ::1 localhost6.localdomain6 localhost6 172.27.1.209 EBS-APP02-172-27-1-209 20.23.25.56 **ebsapp2.mycompany.com ebsapp2** EBS-APP02-10-133-235-56 20.23.25.55 exalgbvm0089-eoib2.mycompany.com exalgbvm0089-eoib2 ebsapp1.mycompany.com ebsapp1 192.168.0.79 EBS-APP02-192-168-0-79
2	All standby application tier nodes, root	At the standby, add the SCAN, VIP and database node aliases into /etc/hosts. # Database logical host names and VIPs 20.35.11.45 exabadm01.mycompany.com exabadm01 ebsdb1.mycompany.com ebsdb1 20.35.11.46 exabadm02.mycompany.com exabadm02 ebsdb2.mycompany.com ebsdb2 ## VIPs added for EBS 12.2 20.35.22.221 exabclient01-vip.mycompany.com exabclient01-vip ebsdb1-vip.mycompany.com ebsdb1-vip 20.35.22.222 exabclient02-vip.mycompany.com exabclient02-vip ebsdb2-vip.mycompany.com ebsdb2-vip ### SCAN IP addresses for EBS 12.2 20.35.22.229 ebsdb-scan1 20.35.22.230 ebsdb-scan2 20.35.22.231 ebsdb-scan3 ## Please note this will not load balance. 20.35.22.229 exaa-scan3
3	All application tier nodes, any user	Make sure you can ping the new host alias e.g.: ping ebsapp1

8.3.3 Set Up the Standby Database

There are five tasks to setting up the standby database servers:

1. Copy the ORACLE_HOME
2. Configure the ORACLE_HOME

3. Instantiate the standby database using RMAN DUPLICATE
4. RAC-enable the standby
5. Enable Data Guard Broker

Our approach provides a way to establish the standby site without running any of the EBS cloning tools i.e., adpreclone, adclonectx, etc. To do this you must meet the following requirements:

- Both the primary and standby sites are symmetric for both database and app tier nodes – the same number of nodes for the database tier and for the application tier on the primary site as on the standby site.
- The database and application tiers at both the primary and the standby sites use logical host names to make the two sets of servers have exactly the same names.
- You must clone each ORACLE_HOME node for node, regardless of the method you use to copy the ORACLE_HOMEs. This means you copy the EBS database ORACLE_HOME on node1 (ebsdb1) on the primary over to node1 (ebsdb1) at the standby, you do the same for node2 (ebsdb2) and so on.

8.3.3.1 Copy the Oracle Homes

There are two methods for this task: 1) using zip and unzip, and 2) using rsync. Use the appropriate table below for the steps, depending on which method you choose.

To copy ORACLE_HOME using the ZIP method do the following steps.

Step	Server, User	Action
1	Each database node exaadb03, exaabd04, root	As root, on each of the primary database nodes, zip up the ORACLE_HOME. On our system, $ORACLE_HOME points to: /u01/app/oracle_ebs/product/12.1.0 So, we do this, zipping up the ORACLE_HOMEs on each database server: `$ cd /u01/app/oracle_ebs/product` On exaadb03: `$ zip -r 12102_BP20160719__ebsdb1_dbhome.zip 12.1.0` On exaabd04: `$ zip -r 12102_BP20160719__ebsdb2_dbhome.zip 12.1.0` **Note:** Our Oracle homes have been patched to BP20160719. **Note:** This zip file will include the appsutil and network/admin directories, which is what we want for each node.
2	Each pair of database servers exaadb03, exaabd04, exabadm01,	As the database owner, copy the zipped Oracle home from each primary database node to its partner at the standby site. Make sure that the directory structures at the standby database servers are exactly the same as that of the primary database servers. On our system, we copied from exaadb03 (ebsdb1) to the standby node exabadm01 (ebsdb1), and from exaabd04 (ebsdb2) to exabadm02 (ebsdb2).

	exabadm02, oracle_ebs	Example: `$ scp 12102_BP20160719__ebsdb1_dbhome.zip oracle_ebs@exabadm01:/u01/app/oracle_ebs/product`
3	Each standby database server exabadm01, exabadm02 oracle_ebs	On each standby database node, as the database owner, unzip the Oracle home into the corresponding directory. On our system, on sca01adm01: `$ cd /u01/app/oracle_ebs/product` `$ unzip 12102_BP20160719__ebsdb1_dbhome.zip` And on sca01adm02: `$ cd /u01/app/oracle_ebs/product` `$ unzip 12102_BP20160719__ebsdb2_dbhome.zip`

The following procedure describes the rsync method of copying the Oracle homes to the standby database servers. To copy ORACLE_HOME using the rsync method, do the following steps.

Step	Server, User	Action
1	Each pair of database servers exaadb03 & exabadm01, exaabd04 & exabadm02, root	You must perform rsync from each Oracle RAC node at the primary site to its corresponding node at the standby site. Appendix E of this document provides a script rsync_EBS_OH.sh that you can use to perform the rsync. By default, the script performs a dry run of rsync to provide you the opportunity to check that rsync will do what is expected. Specifying the --no_dry_run option executes the rsync copy. On our system, run the rsync_EBS_OH.sh script as root on exaadb03 to rsync to the target host exabadm01. As we want to clone the entire Oracle home (including the appsutil, dbs and network/admin directories), we need to use the –-with_ebs_config option: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadmo1 --os_user root --with_ebs_config` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exabadmo1` `os_user = root` `dry_run = true` `with_ebs_config = true` `About to execute the following rsync command:` `rsync -avzh --progress --dry-run /u01/app/oracle_ebs/product/12.1.0/ root@exabadmo1:/u01/app/oracle_ebs/product/12.1.0/`

		OK to continue? [Y|y|N|n] : Then sync from exaabd04 to the target exabadm02: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadmo2 --os_user root -- with_ebs_config` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exabadmo2` `os_user = root` `dry_run = true` `with_ebs_config = true` `About to execute the following rsync command:` `rsync -avzh --progress --dry-run /u01/app/oracle_ebs/product/12.1.0/ root@exabadmo2:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y|y|N|n] :`
2	Each pair of database servers exaadb03 & exabadm01, exaabd04 & exabadm02, root	Run the rsync_EBS_OH.sh script with the `--no_dry_run` option to copy over the ORACLE_HOME. On our system, as root on exaadb03 to the target host exabadm01: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadmo1 --os_user root -- no_dry_run --with_ebs_config` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exabadmo1` `os_user = root` `dry_run = false` `with_ebs_config = true` `About to execute the following rsync command:` `rsync -avzh --progress /u01/app/oracle_ebs/product/12.1.0/ root@exabadmo1:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y\|y\|N\|n] :` Then from exaabd04 to the target exabadm02: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadmo2 --os_user root -- no_dry_run --with_ebs_config` `Using the following parameters:`

		source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0 target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0 target_hosts = exabadmo2 os_user = root dry_run = false with_ebs_config = true About to execute the following rsync command: rsync -avzh --progress /u01/app/oracle_ebs/product/12.1.0/ root@exabadmo2:/u01/app/oracle_ebs/product/12.1.0/ OK to continue? [Y|y|N|n] :

8.3.3.2 Configure the Standby Database Oracle Homes

The Oracle homes must be registered with the global inventory and then relinked using rac_on and ipc_rds for Exadata. The steps are shown in the following table.

Step	Server, User	Action
1	Each standby database node exabadm01, exabadm02, oracle_ebs	Update your login profile script to source the database environment. On our environment, the script to call is ORACLE_HOME/VIS1_ebsdb1.env on node exabadm01 and ORACLE_HOME/VIS2_ebsdb2.env on node exabadm02. We placed the following in our .bash_profile login script: # Set up env for the EBS database env. ./u01/app/oracle_ebs/product/12.1.0/VIS1_ebsdb1.env
2	Each standby database node exabadm01, exabadm02, oracle_ebs	Run the database clone.pl script to update the oraInventory with the new/adjusted oracle home, using the script clone_ohome.sh provided below. On our environment, adjusting the value for the LOCAL_NODE parameter as appropriate (this example shows the command for the first node): #!/bin/sh echo "Clone started at `date`" | tee -a clone.log perl /u01/app/oracle_ebs/product/12.1.0/clone/bin/clone.pl ORACLE_BASE=/u01/app/oracle_ebs ORACLE_HOME=/u01/app/oracle_ebs/product/12.1.0 ORACLE_HOME_NAME=dbhome_ebs1 '- O"CLUSTER_NODES={exabadm01,exabadm02}"' '- O"LOCAL_NODE=exabadm01"' OSDBA_GROUP="dba" OSOPER_GROUP="dba" OSBKP_GROUP="dba" echo "Clone ended at `date`" | tee -a clone.log **Note:** If the clone.pl script fails, you must correct the errors before you attempt to run it again. In addition, you must detach the ORACLE_HOME from the oraInventory using: $ cd $ORACLE_HOME/oui/bin

		`$ ./runInstaller -silent -detachhome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" -local` The clone.pl script will re-attach the ORACLE_HOME.
3	Each standby database node exabadm01, exabadm02, oracle_ebs	Relink the ORACLE_HOME as follows: `$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk rac_on ipc_rds ioracle` **Note:** ipc_rds is for engineered systems (Exadata and Supercluster)
4	Each standby database node exabadm01, exabadm02, root	At the standby, on each database node, log in as root. Then cd to the ORACLE_HOME directory and execute root.sh: `cd /u01/app/oracle_ebs/product/12.1.0` `./root.sh`
5	Each standby database node exabadm01, exabadm02, oracle_ebs	**Note:** If the Grid Listener is being used instead of the local EBS specific listener, this step does not apply and can be skipped. Edit the listener_ifile.ora located in the EBS $TNS_ADMIN directory and change the host names in the HOST parameter to the correct DNS registered host names. On our standby database Oracle RAC server exabadm1: `VIS =` `  (DESCRIPTION_LIST =` `    (DESCRIPTION =` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)` `    (HOST = exabclient01-vip.mycompany.com)(PORT = 1523)(IP = FIRST)))` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)` `     (HOST = exabadm01.mycompany.com)(PORT = 1523)(IP = FIRST)))` `    )` `  )` `SID_LIST_VIS =` `  (SID_LIST =` `    (SID_DESC =` `       (ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0)(SID_NAME = VIS1)` `    )` `  )` `STARTUP_WAIT_TIME_VIS = 0` `CONNECT_TIMEOUT_VIS = 10` `TRACE_LEVEL_VIS = OFF` `LOG_DIRECTORY_VIS =` `  /u01/app/oracle_ebs/product/12.1.0/network/admin` `LOG_FILE_VIS = VIS1` `TRACE_DIRECTORY_VIS =` `  /u01/app/oracle_ebs/product/12.1.0/network/admin`

		TRACE_FILE_VIS = VIS1 ADMIN_RESTRICTIONS_VIS = ON SUBSCRIBE_FOR_NODE_DOWN_EVENT_VIS = OFF
6	Each standby database node exabadm01, exabadm02, oracle_ebs	Edit the <SID>_<node>_ifile.ora located in $TNS_ADMIN and change the host names in the HOST parameter to the correct DNS registered host names for the <SID>_LOCAL TNS connect string aliases. On our standby database Oracle RAC server exabadm1 the VIS1_ebsdb1.ora file has the following entries: VIS1_LOCAL= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)(HOST=**exabclient01-vip**.domain.com)(PORT=1523)) (ADDRESS=(PROTOCOL=tcp)(HOST=**exabclient01-vip**.domain.com)(PORT=1521))) VIS2_LOCAL= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)(HOST=**exabclient02-vip**.domain.com)(PORT=1523)) (ADDRESS=(PROTOCOL=tcp)(HOST=**exabclient02-vip**.domain.com)(PORT=1521))) Add the above entries to the custom.txt file that is read by ebs_scan_db.sh. **Note:** Do not run ebs_scan_db.sh.
7	Each standby database node exabadm01, exabadm02, oracle_ebs	Start the EBS listener and verify that it has started. Only the static SID will be registered with the listener. **Note:** If the Grid Listener is being used instead of the EBS listener, this step can be skipped. $ lsnrctl start vis $ lsnrctl status vis You should see the following: Service "VIS1" has 1 instance(s). Instance "VIS1", status UNKNOWN, has 1 handler(s) for this service...

8.3.3.3 Instantiate the Standby Database with RMAN DUPLICATE

This task copies the database from the primary to the standby. It is based on MOS Doc 1617946.1, to set the standby up for instantiating the database using RMAN DUPLICATE. While there are some changes made on the primary database, there should be no need to restart the database at the primary site. The instructions are expanded in the following table.

Step	Server, User	Action
1	First standby database node exabadm01, oracle_ebs	RMAN DUPLICATE needs a static SID listener to communicate with the standby. To set the SID, edit the listener_ifile.ora file. The bold text in the example below shows what should be added. Modify as necessary for your environment. **Note:** For this static SID listener, DO NOT use the logical host names; instead use the assigned host names and a different port number such as 1525. `VIS =` `  (DESCRIPTION_LIST =` `    (DESCRIPTION =` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)(HOST = exabclient01-vip.mycompany.com)(PORT = 1523)(IP = FIRST)))` `      (ADDRESS_LIST =` `        (ADDRESS = (PROTOCOL = TCP)(HOST = exabadm01.mycompany.com)(PORT = 1523)(IP = FIRST)))` `    )` `  )` **`# Begin insert #1 for static listener`** **`# Remove this once rman duplicate has completed.`** **`RMANDUP =`** **`  (DESCRIPTION_LIST =`** **`    (DESCRIPTION =`** **`      (ADDRESS_LIST =`** **`        (ADDRESS = (PROTOCOL = TCP)(HOST = exabclient01-vip.mycompany.com)(PORT = 1525)(IP = FIRST)))`** **`      (ADDRESS_LIST =`** **`        (ADDRESS = (PROTOCOL = TCP)(HOST = exabadm01)(PORT = 1525)(IP = FIRST)))`** **`    )`** **`  )`** **`# end insert #1 for static listener`** `SID_LIST_VIS =` `  (SID_LIST =` `    (SID_DESC =` `       (ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0)(SID_NAME = VIS1)` `    )` `  )` **`# Begin insert #2 for static listener`** **`#Remove this once rman duplicate has completed.`** **`SID_LIST_RMANDUP =`** **`  (SID_LIST =`** **`    (SID_DESC =`** **`       (ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0)(SID_NAME = VIS1)`** **`  (ENVS="TNS_ADMIN=/u01/app/oracle_ebs/product/12.1.0/network/admin/VIS1_ebsdb1")`** **`    )`** **`  )`** **`# end insert #2 for static listener`**

2	First standby database node exabadm01, oracle_ebs	Start the listener RMANDUP. `$ lsnrctl start RMANDUP`
3	All primary database nodes exaadb03, exaabd04, oracle_ebs	On all primary database nodes, go to the $TNS_ADMIN directory and add the following connect string alias into the IFILE used by the tnsnames.ora -- do not place this entry directly into the tnsnames.ora file. Using an ifile ensures that the entries are kept even if the tnsnames.ora file is updated. On our system, the file to edit is VIS1_ebsdb1_ifile.ora on the first Oracle RAC node. **Note:** For the RMAN DUPLICATE connect strings DO NOT use any of the logical host names in RMAN DUPLICATE. On each Oracle RAC node, edit VIS[N]_ebsdb[N]_ifile.ora and add the following connect string alias (where *n* is the instance number): `rman_dup=` `(DESCRIPTION=` `(ADDRESS=(PROTOCOL=tcp)(HOST=exabadm01.mycompany.com)(PORT=1525))` `(CONNECT_DATA=` `(SERVER = DEDICATED)` `(SERVICE_NAME=VIS1)` `(SID=VIS1)` `)` `)`
4	One primary database node exaadb03, oracle_ebs	Use tnsping to test the listener connection. `$ tnsping rman_dup` `TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 16-DEC-2016 12:35:07` `Copyright (c) 1997, 2014, Oracle. All rights reserved.` `Used parameter files:` `/u01/app/oracle_ebs/product/12.1.0/network/admin/VIS1_ebsdb1/sqlnet_ifile.ora` `Used TNSNAMES adapter to resolve the alias` `Attempting to contact (DESCRIPTION=` `(ADDRESS=(PROTOCOL=tcp)(HOST=exabadm01.mycompany.com)(PORT=1525)) (CONNECT_DATA= (SERVER - DEDICATED) (SERVICE_NAME=VIS1) (SID=VIS1)))` `OK (0 msec)`
5	First standby database node	For active database duplication, you must create a password file for the auxiliary instance and establish Oracle Net connectivity. This is a temporary password file and will be overwritten during the duplicate operation.

	exabadm01, oracle_ebs	Create the password file on the standby host in the $ORACLE_HOME/dbs directory with the same SYS (or SYSDG if you use that username in 12*c*) as the Primary database. For example: `$ orapwd file=orapwboston1 password=<primary database sys password>` Even if the password files were copied over as part of the ORACLE_HOME clone, you must still perform this step. In our case: `$ cd ORACLE_HOME/dbs` `$ orapwd file=orapwVIS1 password=welcome1`
6	First standby database node exabadm01, oracle_ebs	At the standby, create a simple init_rman.ora file in $ORACLE_HOME/dbs that only has the following parameters, for example: `db_name=vis` `sga_target=5G` `db_unique_name=vis_b`
7	Each standby database node exabadm01, exabadm02, oracle_ebs	On all standby hosts create the audit directory for the standby database. As our database db_unique_name is set to vis_b, create the subdirectory as follows: `$ mkdir -p /u01/app/oracle_ebs/admin/vis_b/adump`
8	All database nodes, primary and standby exaadb03, exaabd04, exabadm01, exabadm02, oracle_ebs	On all primary and standby database hosts, create an Oracle Net alias to reach the standby database (on our system, vis_b) on the secondary nodes. Make sure that all of the hosts have both a vis_a and a vis_b Oracle Net alias, and that all of the aliases reference the scan listener and not the node VIP. On our environment, on all Oracle RAC database nodes at both primary and standby sites, we added the following to the VIS[1|2]_ebsdb[1|2]_ifile.ora file. DO NOT use logical host names for the TNS connect string aliases. Instead, we used the SCAN listeners at each site. For Data Guard Broker: `vis_a=` `(DESCRIPTION=` `(ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521))` `(CONNECT_DATA=` `(SERVER = DEDICATED)` `(SERVICE_NAME=vis_a)` `)` `)` `vis_b=` `(DESCRIPTION=` `(ADDRESS=(PROTOCOL=tcp)(HOST=exab-scan1.mycompany.com)(PORT=1521))` `(CONNECT_DATA=`

		(SERVER = DEDICATED) (SERVICE_NAME=vis_b)))
9	First standby database node exabadm01, oracle_ebs	On the standby host ensure that ORACLE_SID is set to the standby SID (VIS1) and startup NOMOUNT the standby/auxiliary instance. The standby database environment should already be set. The init_rman.ora file was created in step 6. `SQL>  STARTUP NOMOUNT pfile=init_rman.ora`
10	One primary database node exaadb03, oracle_ebs	Before proceeding, on the primary, test with SQL*Plus that you can connect to the instance now running at the standby: `$ sqlplus sys/<password from Step 5 above>@connect_string_alias_from_step_3 as sysdba` If the connection fails **DO NOT PROCEED** until this problem is addressed. For example: `$ sqlplus sys/welcome1@rman_dup as sysdba`
11	One primary database node exaadb03, oracle_ebs	If the CLUSTER_INTERCONNECTS parameter is set on the primary it is necessary to temporarily unset it in the SPFILE only. It will be reset to its original value in step 13. **Note:** Do NOT restart any of the primary instances. The reset temporarily removes the CLUSTER_INTERCONNECTS parameter from the SPFILE so that when the SPFILE is copied to the standby system it will avoid any issues if the standby database uses the same instance names as the primary database. The values of CLUSTER_INTERCONNECTS for each instance should be noted before resetting the SPFILE because they will need to be replaced after the standby is created. For example, execute the following commands and note down the values for each instance in the Primary cluster. `SQL> select p.inst_id,instance_name, name,value from gv$parameter p, gv$instance i where p.inst_id=i.inst_id and p.name='cluster_interconnects';` Now reset the cluster_interconnects parameter temporarily in the SPFILE only: `SQL> alter system reset cluster_interconnects scope=spfile sid='VIS1';` `SQL> alter system reset cluster_interconnects scope=spfile sid='VIS2';`
12	One primary database node exaadb03, oracle_ebs	Use an RMAN DUPLICATE script to start the duplication to exabadm01. Make sure that the script includes `CLUSTER_DATABASE=FALSE`. In our script below, we use the scan listener to connect to the "target" which is the primary database. This allows the RMAN DUPLICATE process to span more than one node in an

Oracle RAC database from which the duplicate is made. See MOS Doc 1617946.1 for more details.

The degree of parallelism you use depends on the available processing power on your servers and the interconnect bandwidth between your primary and secondary sites. Setting the degree of parallelism too high will slow down the duplication process. You will need to test to determine the optimal settings for your system.

Note that the ASM disk group names at our primary site (+DATA_EXAA and +RECO_EXAA) differ from those at our secondary site (+DATAC1 and +RECOC1). The below script converts the names as appropriate.

Example of an RMAN DUPLICATE script:

```
# Run RMAN script to duplicate your database.
date

rm dup.log

time rman <<EOF! | tee -a dup.log
set echo on
connect target sys/welcome1@exaa-scan3/vis; # Primary
connect auxiliary sys/welcome1@rman_dup; # destination

CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE DEVICE TYPE DISK PARALLELISM 6;

run {
duplicate target database for standby from active database
spfile
parameter_value_convert
  '+DATA_EXAA,'+DATAC1','+RECO_EXAA,'+RECOC1'
     set db_file_name_convert='+DATA_EXAA','+DATAC1'
     set
  audit_file_dest='/u01/app/oracle_ebs/admin/vis_b/adump'
     set db_unique_name='vis_b'
     set
  log_file_name_convert='+DATA_EXAA,'+DATAC1','+RECO_EXAA','+RE
  COC1'
     set db_create_file_dest='+DATAC1'
     set db_recovery_file_dest='+RECOC1'
     set DB_RECOVERY_FILE_DEST_SIZE='100000G'
     set control_files='+DATAC1'
     set db_create_online_log_dest_1='+RECOC1'
     set db_create_online_log_dest_2='+DATAC1'
     set cluster_database='FALSE'
     set use_large_pages='TRUE'
     set standby_file_management='AUTO'
     set sga_target='12G'
     set listener_networks=''
     set local_listener='VIS1_LOCAL'
     set remote_listener='exab-scan1:1521'
nofilenamecheck;
}

EOF!
```

		`date` Run the RMAN DUPLICATE and let it complete. Fix any errors and rerun if necessary.
13	One primary database node exaadb03, oracle_ebs	On the primary, reset CLUSTER_INTERCONNECTS to its original value from step 10 in the SPFILE with the following commands. Do NOT restart any primary instance. For example: `SQL> alter system set cluster_interconnects='192.168.10.149' scope=spfile sid='chicago1';` `SQL> alter system set cluster_interconnects='192.168.10.150' scope=spfile sid='chicago2';` To set the cluster_interconnect in our case study: `alter system set cluster_interconnects='192.168.218.130' sid='VIS1' scope=spfile;` `alter system set cluster_interconnects='192.168.218.131' sid='VIS2' scope=spfile;`
14	First standby database node exabadm01, oracle_ebs	On the standby, stop and remove the listener created in MOS Doc 1617946.1 step 3. Also remove the TNS entry created in step 4. `$ lsnrctl stop rmandup` Then, remove the listener RMANDUP and its associated static SID listener from the listener_ifile.ora file.
15	First standby database node exabadm01, Oracle (or Grid)	(Oracle Release 12.1 only) Copy the password file to ASM. **Note:** If role separation has been implemented on the standby database servers, then these commands must be executed by the Grid owner. If the standby is release 11.2 then no action is required. If the standby database is at release 12.1, copy the password file to ASM. For example: `$ asmcmd -p --privilege sysdba` `ASMCMD [+] > cd +DATA` `AMSCMD [+DATA] >mkdir BOSTON/PASSWORD` `ASMCMD [+DATA] >pwcopy /u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwboston1 +DATA/BOSTON/PASSWORD/pwboston` `ASMCMD [+DATA] > exit` Remove the original password file. `$ rm /u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwboston1` On our system: `$ asmcmd -p --privilege sysdba` `ASMCMD [+] > cd +DATAC1` `AMSCMD [+DATAC1] >mkdir VIS_B/PASSWORD` `ASMCMD [+DATAC1] > pwcopy --dbuniquename vis_b /u01/app/oracle_ebs/product/12.1.0/dbs/orapwVIS1 +DATAC1/VIS_B/PASSWORD/pwvis_b`

		ASMCMD [+DATAC1] > exit

8.3.3.4 RAC-Enable the Standby

Now that the physical standby database is in place on the first standby database server, we need to set up the other standby database servers.

Step	Server, User	Action
1	First standby database node exabadm01, oracle_ebs	Create a temporary PFILE from the SPFILE on the standby. `SQL> create pfile='vis_b_pfile.ora' from spfile;`
2	First standby database node exabadm01, oracle_ebs	Modify the init.ora parameters in the PFILE just created on the standby to update the instance-specific Oracle RAC parameters. On our environment, we modified the following parameters: `*.cluster_database=TRUE` `VIS1.instance_number=1` `VIS2.instance_number=2` `VIS1.local_listener='VIS1_LOCAL' #EBS specific listener` `VIS2.local_listener='VIS2_LOCAL' #EBS specific listener` `VIS1.thread=1` `VIS2.thread=2` `# The tablespace parameters should already be set correctly.` `VIS1.undo_tablespace='APPS_UNDOTS1'` `VIS2.undo_tablespace='UNDOTBS2'`
3	First standby database node exabadm01, oracle_ebs	Create an SPFILE in +DATAC1 for the standby database: `SQL> create spfile='+DATAC1/vis_b/spfilevis_b.ora' from pfile='vis_b_pfile.ora';` `File created.`
4	Standby database nodes exabadm01,exabadm02, oracle_ebs	Conditional – Oracle Database Release 11.2 only. Copy the password file to the other nodes. If the standby database is release 11.2, you must copy the password file to the $ORACLE_HOME/dbs directory on all the other standby hosts and name it per the standby SID on each host. For example on node2 you would name it $ORACLE_HOME/dbs/orapwVIS2. If the standby database is release 12.1 then no action is needed. The password was copied to ASM in the section above, making it visible to all nodes across the cluster.
5	Each standby database node	On all standby hosts, create an initVIS<SID Number>.ora file that points to the SPFILE created in step 3.

	exabadm01,exabadm02, oracle_ebs	On our environment: `spfile='+DATAC1/VIS_B/spfilevis_b.ora'`
6	Each standby database node exabadm01,exabadm02, oracle_ebs	Before proceeding on to the next step, shut the standby database down on each node, move the spfileVIS1.ora in $ORACLE_HOME/dbs to spfileVIS1.ora.backup, and startup MOUNT (**do not open**) the instance using SQL*Plus. Do the same on exabadm02. This ensures that it will start on both nodes before registering the database and its instances with CRS. If the startup mount is successful on all instances, then shut down all instances. In our environment, on node exabadm01: `$ mv spfileVIS1.ora spfileVIS1.ora.backup` `SQL>  shutdown immediate` `SQL>  startup mount` `ORACLE instance started.` `Total System Global Area 1.2885E+10 bytes` `Fixed Size 4511656 bytes` `Variable Size 2214594648 bytes` `Database Buffers 1.0503E+10 bytes` `Redo Buffers 163258368 bytes` `Database mounted.` `SQL>  shutdown immediate` On node exabadm02: `$ mv spfileVIS2.ora spfileVIS2.ora.backup ## if this file exists` `SQL>  shutdown immediate` `SQL>  startup mount` `ORACLE instance started.` `Total System Global Area 1.2885E+10 bytes` `Fixed Size 4511656 bytes` `Variable Size 2214594648 bytes` `Database Buffers 1.0503E+10 bytes` `Redo Buffers 163258368 bytes` `Database mounted.` `SQL>  shutdown immediate`
7	First standby database node exabadm01, oracle_ebs	Register the database with CRS. The following commands register vis_b database and all instances into CRS. `srvctl add database -db vis_b -oraclehome /u01/app/oracle_ebs/product/12.1.0 -spfile +DATAC1/vis_b/spfilevis_b.ora -diskgroup "DATAC1,RECOC1"` `srvctl add instance -db vis_b -instance VIS1 -node exabadm01` `srvctl add instance -db vis_b -instance VIS2 -node exabadm02` The following SRVCTL commands set the required environment variables for EBS:

		srvctl setenv database –d vis_b –t "TNS_ADMIN=/u01/app/oracle_ebs/product/12.1.0/network/admin/ebsdb" srvctl setenv database –d vis_b –t "ORA_NLS10=/u01/app/oracle_ebs/product/12.1.0/nls/data/9idata" Specify where the password file is located using SRVCTL: srvctl modify database –d vis_b -pwfile '+DATAC1/vis_b/PASSWORD/pwvis_b'
8	First standby database node exabadm01, oracle_ebs	Restart all standby instances, using the MOUNT option. srvctl start database –d vis_b –o mount
9	First standby database node exabadm01, oracle_ebs	Add the database services for Self-Service (OACORE), Forms, and Parallel Concurrent Processing (PCP). Self-Service uses the VIS_OACORE_ONLINE service. Forms uses the service VIS_FORMS_ONLINE service. The two database services for PCP are VIS_PCP_BATCH1 and VIS_PCP_BATCH2. For Self-Service (OACORE): srvctl add service -db vis_b -service VIS_OACORE_ONLINE - preferred "VIS1,VIS2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 For Forms: srvctl add service -db vis_b -service VIS_FORMS_ONLINE - preferred "VIS1,VIS2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 For PCP: srvctl add service -db vis_b -service VIS_PCP_BATCH1 -preferred "VIS1" -available "VIS2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 srvctl add service -db vis_b -service VIS_PCP_BATCH2 -preferred "VIS2" -available "VIS1" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3

8.3.3.5 *Enable Data Guard Broker*

The following steps follow MOS Doc 1617946.1 for configuring Data Guard Broker, which in turn will enable Data Guard.

Step	Server, User	Action
1	All primary and standby database nodes,oracle_ebs	From each primary and each standby database instance test the connections to all other primary and standby database instance (on our environment, vis_a and vis_b). These TNS connect string aliases were added to the ifile the section Instantiate the Standby Database Using RMAN DUPLICATE. **DO NOT PROCEED** if any of the connection attempts fail. Fix the connection error before going further. On our environment, on each primary database node, we entered: `sqlplus sys/<sys password>@vis_a as sysdba` `sqlplus sys/<sys password>@vis_b as sysdba` On each standby database node: `sqlplus sys/<sys password>@vis_a as sysdba` `sqlplus sys/<sys password>@vis_b as sysdba`
2	One primary and one standby database node exaadb03, exabadm01, oracle_ebs	On both the primary and standby, configure the Data Guard broker metadata files and enable the broker: You only need to run the below commands from one database node at each site. On the primary, in our environment, we ran: `set echo on` `alter system set` `  dg_broker_config_file1='+DATA_EXAA/VIS_A/dr1vis_a.dat'` `  scope=both;` `alter system set` `  dg_broker_config_file2='+RECO_EXAA/VIS_A/dr2vis_a.dat'` `  scope=both;` `alter system set dg_broker_start=true scope=both;` On the standby `set echo on` `alter system set` `  dg_broker_config_file1='+DATAC1/VIS_B/dr1vis_b.dat'` `  scope=both;` `alter system set` `  dg_broker_config_file2='+RECOC1/VIS_B/dr2vis_b.dat'` `  scope=both;` `alter system set dg_broker_start=true scope=both;`
3	One primary database node	On a primary host, connect using DGMGRL and create the Data Guard configuration.

	exaadb03, oracle_ebs	On our environment: `$ dgmgrl sys/<sys password>` `DGMGRL> create configuration dg_vis` `  as primary database is 'VIS_A'` `  connect identifier is vis_a;` `Configuration "dg_vis" created with primary database "VIS_A"` `DGMGRL> add database 'VIS_B'` `  as connect identifier is vis_b;` `Database "VIS_B" added` `DGMGRL> enable configuration;` `Enabled.`
4	One primary database node exaadb03, oracle_ebs	Verify that the configuration created successfully by using the SHOW CONFIGURATION command. On our environment: `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `  VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS  (status updated 57 seconds ago)`
5	All database nodes, primary and standby exaadb03, smam08db04, exabadm01, exabadm02,oracle_ebs	Another step to validate the Data Guard configuration is to run the DGMRL command: VALIDATE DATABASE VERBOSE '<db_unique_name>'; Log onto each node at both the primary and standby site and issue the VALIDATE DATABASE command for the remote database. If there are any residual connection issues, this command may expose them. If there are any, investigate and fix them. If there is a problem it will most likelybe a TNS configuration issue. On our environment, at the primary on nodes exaadb03 and exaabd04, on our system we logged onto DGMGRL and issued the following commands: `DGMGRL> VALIDATE DATABASE VERBOSE 'VIS_A';` `DGMGRL> VALIDATE DATABASE VERBOSE 'VIS_B';` At the standby on nodes exabadm01 and exabadm02, we logged onto DGMGRL and issued the following commands: `DGMGRL> VALIDATE DATABASE VERBOSE 'VIS_A';` `DGMGRL> VALIDATE DATABASE VERBOSE 'VIS_B';`

		Each of the above will generate a report if it can connect to both VIS_A and VIS_B databases.
6	One primary database node, one standby database node oracle_ebs	Flashback database must be enabled on the primary to quickly reinstate the primary after a failover role transition to the standby. If it is not enabled, the primary site's database must be re-instantiated after a failover by copying the database from the standby site (which would be "primary" at that time), similar to the process used above to originally instantiate the standby. While enabling Flashback Database is a recommended practice, you need to test the performance implications at your site. Optionally enable flashback on both the primary and standby. To enable flashback database on the primary: `SQL> alter database flashback on;` To enable flashback database on the standby the redo apply process must first be stopped. Once flashback has been enabled redo apply can be restarted: `SQL> recover managed standby database cancel;` `SQL> alter database flashback on;` `SQL> recover managed standby database disconnect using current logfile;` The above steps can also be accomplished using the broker with the following commands. On our environment: `DGMGRL> CONNECT sys/<password>@vis_b` `DGMGRL> EDIT DATABASE 'VIS_B' SET STATE='APPLY-OFF';` `DGMGRL> SQL 'ALTER DATABASE FLASHBACK ON';` `DGMGRL> EDIT DATABASE 'VIS_B' SET STATE='APPLY-ON';`
7	Any primary or standby database node, oracle_ebs	To test a simple database switchover from within Data Guard Broker, check that the configuration still reports SUCCESS. See step 6. On our environment, once we have checked the configuration, we ensure that the concurrent queues are drained and then issue: `DGMGR> switchover to 'VIS_B';` To switch back: `DGMGR> switchover to 'VIS_A';` **Note:** This disrupts services at the primary.

8.3.4 Set Up Standby Application Tier

To prepare the application tier servers at the secondary site, we completed the following steps:

1. Make sure the servers to be used for the standby application tier are ready.
2. Set up the OS groups and user accounts with the same names and privileges as used on the primary application tier servers.

3. Set up ZFS replication from the primary to the standby

8.3.4.1 Ensure Secondary Application Tier Servers are Ready

See Oracle E-Business Suite Installation and Upgrade Notes Release 12 (12.2) for Linux x86-64 (Doc ID 1330701.1) for the server settings required to successfully host the Oracle E-Business Suite application. Particular areas to note:

- Be sure all OS RPMs required by EBS for the application tier are installed. Appendix A provides a brief set of steps for applying the required packages to the application tier servers. Please note it is not necessary – and is not supported – to use the EBS RPM to add packages to the Exadata (database server) OS. Only the application tier servers will require these packages.
- Ensure that the swap device is at least one-half the size of the total system memory. If not, follow the specific Linux procedures for increasing or extending the swap space. For Exalogic, see: https://docs.oracle.com/cd/E18476_01/doc.220/e25258/app_lvm.htm#ELCLD77342.

8.3.4.2 Add the Application Owner OS User Account

Add the application owner OS user account ("applmgr") with the same user name, UID, group names, and GID as that of the primary.

Note: If you are using NIS or LDAP for user account provisioning, the UID and GIDs should match. This is not a hard requirement but is recommended to simplify establishing the EBS application tiers. If you are not using NIS or LDAP for OS account provisioning, then the UID and GID MUST match. These instructions are included in the following table.

Step	Server, User	Action
1	One primary application tier node, appplmgr	To find the applmgr UID and primary group GID, log into the applmgr account on a primary node and issue the ID command. On our environment: `$ id` `uid=1010(applmgr) gid=1001(oinstall)` `groups=1001(oinstall),1002(dba)`
2	Each standby application tier node, root	If you are not using NIS or LDAP to provision the OS accounts, then you can issue commands similar to these to add the groups and user accounts. On our environment: `# Add groups` `$ groupadd -g 1001 oinstall` `$ groupadd -g 1002 dba` `# Create the applmgr user account` `$ useradd --uid 1010 -g oinstall applmgr` `$ usermod -d /home/applmgr -G dba -g oinstall -s /bin/bash applmgr`

8.3.4.3 Set Up ZFS Replication to Replicate Application Tier File Systems to the DR Site

Provided that your storage and network administrators have configured the network and targets for both primary and standby ZFS storage appliance, the next task is to configure replication of the storage that contains the application tier RUN and PATCH file systems. The following steps show how we configured ZFS's continuous replication from the primary to the standby.

In the examples below, the primary ZFS is exalgasn and the standby is exalgbsn.

Step	Server, User	Action
1	Primary ZFS exalgasn-fe:215, root or administrator	Log on to the primary ZFS administrative console BUI (browser user interface) using a browser. On our environment, we use this URL: https://exalgasn-fe:215
2	Primary ZFS exalgasn-fe:215, root or administrator	Click the Shares link at the top.
3	Primary ZFS exalgasn-fe:215, root or administrator	On the left, click Projects and a side pane opens.
4	Primary ZFS exalgasn-fe:215, root or administrator	Locate your project, then click on it. The shares belonging to the project are listed in the large area to the right.
5	Primary ZFS exlgasn-fe:215, root or administrator	Click the Replication link to the right above the list of shares.
6	Primary ZFS exalgasn-fe:215, root or administrator	Click the plus sign (+) to add a replication action.
7	Primary ZFS exalgasn-fe:215, root or administrator	Select the target ZFS appliance from the drop-down list.

8	Primary ZFS exalgasn-fe:215, root or administrator	Select SSL if you need the extra layer of encryption in your environment.
9	Primary ZFS exalgasn-fe:215, root or administrator	Select Scheduled then click Apply without selecting an actual schedule.
10	Primary ZFS exalgasn-fe:215, root or administrator	Select the Continuous radio button, then click Apply. You will see a status bar that indicates continuous replication is enabled and working (looks like a horizontal bar with dark stripes) The initial synchronization will take time depending on the size of the volume. The display will show progress as percentage complete.
11	Standby ZFS exalgbsn-fe:215, root or administrator	At the standby ZFS appliance, verify that the replica is present and is being updated from the primary ZFS appliance. Log into the standby ZFS administrative console BUI with a browser.
12	Standby ZFS exalgbsn-fe:215, root or administrator	Click the Shares link near the top of the page.
13	Standby ZFS exalgbsn-fe:215, root or administrator	On the left, click Projects and a side pane opens.
14	Standby ZFS exalgbsn-fe:215, root or administrator	Click Replica in the Projects pane.
15	Standby ZFS exalgbsn-fe:215, root or administrator	Locate your project in the Projects pane and click on it. The shares that belong to this replicated project are displayed in the large area to the right.
16	Standby ZFS exalgbsn-fe:215, root or administrator	Click Replication on the right. You should see your primary host name, last sync date and time, last attempt date and time, and the status bar. The status should be "success".

8.4 Disaster Recovery Site Test using Snapshots

Using your standby environment, you can start Oracle E-Business Suite for testing at the DR site without impacting or affecting production. The Recovery Point Objective (RPO) is not compromised while you are testing because redo and application tier file changes are still being shipped to the DR site. The Recovery Time Objective (RTO) will be affected should the DR site need to assume the primary role during this period, to catch up with redo apply.

The conditions that must be true for this testing paradigm to work are:

- Your hardware provides the ability to create read-write snapshots of your applications tier file system.
- Logical host names have been configured for each of the tiers at the primary and secondary sites.
- All hosts are reachable using their physical or assigned DNS host name. Use firewall policies to prevent the application tier servers at either site from connecting to the database servers on the remote site using the physical host names. This ensures that any missed configuration in this process does not result in accidental connection from the snapshot standby servers to the primary environment.
- All application tier servers have been configured into either a global f5 load balancer (GTM), a load balancer local to the DR site (f5 LTM), or another load balancer.

There are three steps that allow EBS at the DR site to be brought up for testing without impacting production.

1. Convert the physical standby database into a snapshot standby database.
2. Create a snapshot clone of the shared file system being replicated to the DR site.
3. Configure the web entry URLs and port numbers so that they do not conflict with production, then run AutoConfig.

8.4.1 Convert the Physical Standby Database to a Snapshot Standby

The following table shows the steps to convert the standby database into a snapshot standby.

Step	Server, User	Action
1	One primary or standby database node exabadm01, oracle_ebs	Using Data Guard Broker, (DGMGRL), check the current configuration. `dgmgrl sys/welcome1` `DGMGRL for Linux: Version 12.1.0.2.0 - 64bit Production` `Copyright (c) 2000, 2013, Oracle. All rights reserved.` `Welcome to DGMGRL, type "help" for information.` `Connected as SYSDBA.` `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `    VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS   (status updated 25 seconds ago)`

		DGMGRL> validate database 'VIS_B' Database Role: Physical standby database Primary Database: VIS_A Ready for Switchover: Yes Ready for Failover: Yes (Primary Running)
2	One primary or standby database node exabadm01, oracle_ebs	Only if the SHOW CONFIGURATION status shows SUCCESS and "Ready for Switchover" says Yes, then convert the physical standby to a snapshot standby. DGMGRL> convert database 'VIS_B' to snapshot standby; Converting database "VIS_B" to a Snapshot Standby database, please wait... Database "VIS_B" converted successfully

8.4.2 Create a Snapshot Clone of the Shared File System Being Replicated to the Disaster Recovery Site

The following table shows the steps to create a read-write snapshot of the application tier file system, on a ZFS appliance.

Step	Server, User	Action
1	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	Log on to the secondary ZFS administrative console using a browser. On our environment, we use this URL: https://exalgbsn-fe:215
2	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	a) Click on Shares b) Click on Projects on the left and the left pane shows c) Click REPLICA
3	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	a) Find the replicated share in the list and click on it. It will show the share on the right. b) Click Replication on the far right.
4	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	To the right will be several icons. One of them will look like a plus-sign (+). Mouse over them to see what they are. The plus-sign should show "Clone from recent refresh…"; click on this.
5	Standby ZFS appliance:	You will be prompted for a new clone project name. Because this is site B, in our case we called it "EBS_DEV_SITE_B_SNAPSHOT".

	exalgbsn-fe:215, root or administrator	Then, click Apply.
6	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	Once the above step has completed: a) Click LOCAL on the left pane and find the newly created project, our example: EBS_DEV_SITE_B_SNAPSHOT. b) Click on the new share. The new share should show up on the right side.
7	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	a) Click General to show the general properties. b) Uncheck the box: “Inherit from project”. The items below will no longer be grayed out, indicating it now possible to change the items in the next few steps.
8	Standby ZFS appliance: exalgbsn-fe:215, root or administrator	Change the Mount point to something like: /export/test_applmgr_install_snapshot That is, add the word “snapshot” to the end of the mount point entry so the purpose is clearly represented. Do not change the rest of the options. Click Apply.
9	All standby application tier nodes exalgbvm0089-eoib2, exalgbvm0090-eoib2, root	Add an entry into /etc/fstab that mounts the snapshot share under the same mount point as production. Remember that we need to keep all of the directory structures the same on the standby as on the primary. Example: `### Snapshot used for testing.` `172.27.0.9:/export/test_applmgr_install_snapshot /u06 nfs rw,rsize=131072,wsize=131072,bg,hard,noacl,timeo=600,retrans=20`
10	All standby application tier nodes exalgbvm0089-eoib2, exalgbvm0090-eoib2, root	Mount the file system on all of the application tier nodes. In our case, its /u06. `$ mount /u06`

8.4.3 Set Up OS User Equivalency

If this is the first time that the secondary site is being configured, follow the below steps to establish user equivalency. This task must be done one time only on each standby server, at a time when the application is available on the standby either by switchover/failover testing or as a snapshot standby.

Step	Server, User	Action
1	Each standby application tier node, applmgr	Oracle E-Business Suite tooling, particularly AD Online Patching (adop), requires that user equivalency be established on all application tier nodes. Run the `txkRunSSHSetup.pl enablessh` command using the logical host names on all standby application tier nodes, changing the context path and file name to match your environment. From exalgbvm0089-eoib2 (ebsapp1) `$ perl /u06/app/ebs/vis/fs2/EBSapps/appl/ad/12.0.0/patch/115/bin/txkRunSSHSetup.pl enablessh \ -contextfile=/u06/app/ebs/vis/fs2/inst/apps/ \ VIS_ebsapp1/appl/admin/VIS_ebsapp1.xml \ -hosts=ebsapp1,ebsapp2` From exalgbvm0090 (ebsapp2) `$ perl /u06/app/ebs/vis/fs2/EBSapps/appl/ad/12.0.0/patch/115/bin/txkRunSSHSetup.pl enablessh \ -contextfile=/u06/app/ebs/vis/fs2/inst/apps/ \ VIS_ebsapp2/appl/admin/VIS_ebsapp2.xml \ -hosts=ebsapp1,ebsapp2`
2	Each standby application tier node, applmgr	On each standby node, test that SSH user equivalency has been established. From each node, test using locgical host names against all of the other nodes using a simple command such as in this example, but substituting your target node. `$ ssh applmgr@ebsapp2 "date"` The command should return the date and, importantly, you should not be prompted for the password.
3	All standby application tier nodes, applmgr	**NOTE**: The EBS script that sets up the administrator's OS environment will not succeed until the standby site's physical host names are in the node_info.txt file. This file is cumulative, with the new host information appended to the file. To add the standby site's application tier physical host names to the node_info.txt file, you need to run AutoConfig without sourcing the environment on both the RUN and PATCH file systems on all application tier nodes. This only needs to be done once in the environment if done in a way that persists – in a true switchover or failover to the standby site. Until it is done persistently, these steps will be required each time testing is done via snapshot clones. On each application tier node, in both the RUN and the PATCH file systems: Go to the directory that would be pointed to by $INST_TOP: `/<your base directory>/<run file system>/inst/admin/scripts`

		For example, on our system (fs2 is the RUN file system): `/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/admin/scripts` Execute AutoConfig from this location: `./adautocfg.sh` Now, source the environment. **Note:** Because this is on a snapshot clone, once the snapshot clone is dropped, this configuration will be lost. You will need to execute this step every time you use the environment in a snapshot clone method until you execute it once in a permanent way (with full switchover testing or a failover).

8.4.4 Configure the Web Entry URLs and Port Numbers

Configure the web entry URLs and port numbers so they do not conflict with that of production, then run AutoConfig and start services.

The following steps assume that the standby servers have already been configured into a load balancer. As none of our logical host names are registered in the DNS, the load balancer is required in order to access the EBS Web application.

Step	Server, User	Action
1	All standby application tier nodes (exalgbvm0089-eoib2, exalgbvm0090-eoib2), applmgr	Source the RUN file system.
2	All standby application tier nodes (exalgbvm0089-eoib2, exalgbvm0090-eoib2), applmgr	Change the web entry URLs, login page, and port number on the snapshot standby system, to give a different path for users to log into the new environment. We used the following script in our environment, which can be used as a basis for your environment's configuration. Note that the script copies the $CONTEXT_FILE to a backup. `# Back up the context file before running this script.` `# Backup context file` `cp $CONTEXT_FILE $CONTEXT_FILE.`date +"%Y%m%d"`` `$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar`

		oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webentryhost pd-bigip-sca-loan01 $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webentrydomain mycompany.com $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webport 9090 $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_active_webport 9090 $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_http_listen_parameter 9090 $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_login_page http://pd-bigip-sca- loan01.mycompany.com:9090/OA_HTML/AppsLogin $RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle _EBS-app1/shared-libs/ebs-appsborg/WEB- INF/lib/ebsAppsborgManifest.jar oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_external_url http://pd-bigip-sca-loan01.mycompany.com:9090
3	All standby application tier nodes (exalgbvm0089-eoib2, exalgbvm0090-eoib2), applmgr	Run AutoConfig $ cd $ADMIN_SCRIPTS_HOME $ adautocfg.sh

4	All standby application tier nodes (exalgbvm0089-eoib2, exalgbvm0090-eoib2), applmgr	Follow the solution in OPMN Fails to Start and Says to Check Adopmnctl.txt (Doc ID 2174221.1) to clear the replicated production state data for OPMN and HTTP. The production state data on the RUN file system needs to be removed or moved so OPMN does not reference it and it will then start correctly on the snapshot standby environment. There will be a subdirectory EBS_web_VIS_OHSx for each application tier node, where x corresponds to a given application tier node. For example, EBS_web_VIS_OHS1 would be for the master application tier node, EBS_web_VIS_OHS2 for the second application tier node, and so on. The state data must be moved for each active application tier node. On our system, we did the following: `$ cd /u06/app/ebs/vis/fs1/FMW_Home/webtier/instances/EBS_web_VIS_OHS1/config/OPMN/opmn` `$ mv states states_backup`
5	All standby application tier nodes (exalgbvm0089-eoib2, exalgbvm0090-eoib2), applmgr	Start the application tier services on all nodes. `$ adstrtal.sh`

8.5 Test Switchover

These are the high level steps to perform a switchover:

1. Drain the Concurrent Request queues. Shut down the EBS application services at the primary site.
2. Switch roles for the database using Data Guard Broker.
3. Reverse the replication direction on the ZFS appliance.
4. Mount the EBS application file system and start application services at the secondary site (now the primary).

Note: Executing this test will disrupt services at the primary site.

8.5.1 Shut EBS Application Services Down at the Primary Site

Step	Server, User	Action
1	All Primary application tier nodes, applmgr	Shut down the EBS application services. **Note:** You can to shut down the concurrent manager processes ahead of your scheduled switchover in order to drain the queues.

		To shut down all services, log onto each application tier node as applmgr, source the Run environment, then issue: `$ cd $ADMIN_SCRIPTS_HOME` `$ adstpall.sh`

8.5.2 Switch Database Roles

Step	Server, User	Action
1	One primary database node (this will also work on a database node at the standby)	Check the configuration and the standby database before performing a switchover. On the database node, log into Data Guard Broker to perform the switchover. On our environment: `dgmgrl sys/<password>` `DGMGRL> show configuration;` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `    VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS   (status updated 29 seconds ago)` `DGMGRL> show database  'VIS_B';` `Database - VIS_B` `  Role:               PHYSICAL STANDBY` `  Intended State:     APPLY-ON` `  Transport Lag:      0 seconds (computed 0 seconds ago)` `  Apply Lag:             0 seconds (computed 0 seconds ago)` `  Average Apply Rate: 14.47 MByte/s` `  Real Time Query:    OFF` `  Instance(s):` `    VIS1 (apply instance)` `    VIS2` `Database Status:` `SUCCESS` `DGMGRL> validate database 'VIS_B'` `  Database Role:     Physical standby database` `  Primary Database:  VIS_A` `  Ready for Switchover:  Yes`

		Ready for Failover: Yes (Primary Running) DGMGRL> switchover to 'VIS_B'; Performing switchover NOW, please wait... Operation requires a connection to instance "VIS1" on database "VIS_B" Connecting to instance "VIS1"... Connected as SYSDBA. New primary database "VIS_B" is opening... Oracle Clusterware is restarting database "VIS_A" ... Switchover succeeded, new primary is "VIS_B" DGMGRL> show configuration; Configuration - dg_vis Protection Mode: MaxPerformance Members: VIS_B - Primary database VIS_A - Physical standby database Fast-Start Failover: DISABLED Configuration Status: SUCCESS (status updated 27 seconds ago)

8.5.3 Reverse ZFS Replication Direction

Step	Server, User	Action
1	All Primary application tier nodes, root	Umount the application tier shared file system on each primary application tier node. In our case, the NFS mount point is /u06. `$ umount /u06`
2	Standby ZFS Storage Appliance exalgbsn-fe:215, root or administrative account	Log into the standby ZFS BUI using a browser. Our URL was: https://exalgbsn-fe:215 a) Click Shares. b) To the left, click Projects. c) Click Replica. d) Locate your replicated project and click on it. You will see the volumes that are being replicated and their status. e) Click Replication to the far right.

		f) Reverse direction by clicking on the U-turn looking symbol with a tool-tip that says "Reverse direction". **Note:** The U-turn icon is not ADA compliant. You must mouse over it to see the "Reverse direction" tool tip.
3	Standby ZFS, root or administrative account	When prompted, give it a project name. We called ours EBS_DEV_SITE_B.
4	Standby ZFS, root or administrative account	Ensure there are no errors or warnings. Verify that EBS_DEV_SITE_B is now a project on exalgbsn, which now becomes the new primary.
5	Old-Primary / ZFS appliance exalgasn-fe:215, root or administrative account	The old primary ZFS is now a standby. Log on to its browser-based UI and verify that the now new primary ZFS appliance is replicating successfully to the now new standby ZFS appliance. Our URL was: https://exalgasn-fe:215Follow the steps in **Set Up ZFS Replication to Replicate Application Tier File Systems to the DR Site.**

8.5.4 Mount the EBS Application File System and Start Application Services at the Standby

Step	Server, User	Action
1	All NEW primary application tier nodes	Ensure that the /etc/fstab file has the correct mount point for the application tier shared file system. The mount point will be the same as the old primary except the ZFS IP address or host name will be different. From our environment, the fstab entries for our ZFS project are, on our OLD primary (new standby): `192.168.219.207:/export/test_applmgr_install /u06 nfs rw,rsize=131072,wsize=131072,bg,hard,noacl,timeo=600,retrans=20` On our NEW primary (old standby): `172.27.0.9:/export/test_applmgr_install /u06 nfs rw,rsize=131072,wsize=131072,bg,hard,noacl,timeo=600,retrans=20`

2	All NEW primary application tier nodes, root	Now mount the /u06 NFS mount point on all application tier nodes at the NEW primary site. `$ mount /u06`
3	All NEW primary application tier nodes, applmgr	Log on to each application tier node and source the RUN file system. You can add the EBSapps.env into your .bash_profile so that it will source automatically when you log on. For example: `# Set the apps env.` `if [ -f /u06/app/ebs/vis/EBSapps.env ]; then` `. /u06/app/ebs/vis/EBSapps.env` `fi`
4	All NEW primary application tier nodes, applmgr	**IMPORTANT**: The EBS script that sets up the administrator's OS environment will not succeed until the standby site's physical host names are in the node_info.txt file. This file is cumulative, with the new host information appended to the file. To add the standby site's application tier physical host names to the node_info.txt file, you need to run AutoConfig without sourcing the environment on both the RUN and PATCH file systems on all application tier nodes. This only needs to be done once in the environment if done in a way that persists – in a true switchover or failover to the standby site. On each application tier node, in both the RUN and the PATCH file systems: Go to the directory that would be pointed to by $INST_TOP: `/<your base directory>/<run file system>/inst/admin/scripts` For example, on our system (fs2 is the run file system): `/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/admin/scripts` Execute AutoConfig from this location: `./adautocfg.sh` Source the environment. As the replication has been reversed, the updated node_info.txt file will be replicated to the standby site, making this change permanent at both sites. You can skip the next step, as AutoConfig was run in this step.
5	All NEW primary application tier nodes, applmgr	Run AutoConfig on all application tier servers in both the RUN and PATCH file systems.
6	All application tier servers – exalgbvm0089-eoib2,exalgbvm	Start the EBS application services. Start with the primary application server first as this will start the WLS administration server. `$ adstrtal.sh` Check for errors.

	0090-eoib2, applmgr	
7	At the new primary, on the primary application tier server exalgbvm0089-eoib2	Perform an adop validation, which will check both RUN and PATCH file systems on all nodes. `$ adop –validate` … `Validating credentials.` `Initializing.` `Run Edition context : /u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsapp1.xml` `Patch edition context: /u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsapp1.xml` `Node registry is valid.` `==========` `ADOP (C.Delta.8)` `Node: ebsapp1` `Command: validate` `Log: /u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/adopConsole.log` `==========` `Checking for existing patching cycle.` `No existing patching cycle exists` `Verifying SSH connection to all nodes.` `Log: /u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/logs/appl/rgf/TXK/verifyssh.log` `Output: /u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/logs/appl/rgf/TXK/out.xml` `Remote execution is operational.` `Running adop validations on Admin node: [ebsapp1].` `Log: ebsapp1:/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/ebsapp1` `Output: /u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/remote_execution_result_level1.xml` `txkADOPEvalSrvStatus.pl returned SUCCESS` `Running adop validations on node(s): [ebsapp2].` `Log: ebsapp2: /u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/ebsapp2`

		Output: /u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/remote_execution_result_level2.xml txkADOPEvalSrvStatus.pl returned SUCCESS adop exiting with status = 0 (Success)
8	Web, OAM	Log in to EBS on the new primary and ensure availability and function of the EBS application.

8.6 Test Failover

If the primary site is unexpectedly unavailable, and no production work can take place for a period of time that would significantly impact business, you must perform a failover to the secondary site. Most Oracle E-Business Suite customers choose to make failing over to the secondary site a business decision. We recommend automating the process of performing a site failover as much as possible to reduce risks of errors.

The primary and secondary sites should be configured for minimum data loss and minimal downtime during the failover process. As the primary site is no longer available, the failover steps are all performed at the secondary site.

Note: In our environment, we have enabled flashback database so that we do not have to re-copy the standby database. Instead, we can use Data Guard to re-instantiate the existing failed database, which will then serve as the new standby database. If your performance testing shows that you cannot enable flashback database in production, you must reinstantiate your primary as a standby when the site is again available, using the techniques described earlier in this paper for establishing a standby.

Note: If the EBS Suite is running in snapshot standby mode at the secondary site and a failover is required, then prior to performing the failover, do the following at the secondary site:

1. Shut down all application services.
2. Unmount and drop the file system snapshot clone.
3. Using Data Guard Broker, convert the SNAPSHOT STANDBY database back to a PHYSICAL STANDBY database.
4. Perform the failover per the following steps.

The high-level tasks for performing a site failover, which are detailed in the tables below, are:

1. Perform a Data Guard failover (not switchover).
2. Perform a role reversal of the ZFS filer at the standby to assume the primary role.
3. Perform state cleanup.
4. Start application services.

Note: To test this on our environment, we forced a shutdown abort on the primary to simulate a failure:

```
srvctl stop database -d vis_a -o abort
```

8.6.1 Perform a Data Guard Failover

Step	Server, User	Action
1	One standby database node, oracle_ebs	Perform a failover using Data Guard Broker on one database node at the secondary site. The SHOW CONFIGURATION command issued in Data Guard Broker should show errors indicating that the primary is not available. For example: `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `  **Error: ORA-12514: TNS:listener does not currently know of service requested in connect descriptor**` `    VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `ERROR   (status updated 0 seconds ago)` `DGMGRL>` `DGMGRL> failover to 'VIS_B';` `Performing failover NOW, please wait...` `Failover succeeded, new primary is "VIS_B"` `DGMGRL>`

8.6.2 Perform a Role Reversal

Reverse the role of the ZFS filer at the standby to assume the primary role.

Step	Server, User	Action
1	DR site ZFS, administrator	Log into the ZFS filer browser UI, and locate the replica for the EBS install. Click Shares, then Projects, and the left pane will open. Click REPLICA in the left pane. Your replicated EBS install will be displayed. Click on it.
2	DR site ZFS, administrator	The volumes belonging to the replicated project will be shown in the right side of the display. Click Replication.

		Depending on the nature of the failure, you may need to stop or sever the replication from the primary site. This should allow you to provide a new project name e.g., EBS_DEV_SITE_B.
3	Each DR site application tier server exalgbvm0089-eoib2,exalgbvm0090-eoib2, root	In /etc/fstab, ensure the mount point can be mounted, then mount the NFS file system. `172.27.0.9:/export/test_applmgr_install /u06 nfs rw,rsize=131072,wsize=131072,bg,hard,noacl,timeo=600,retrans=20` `$ mount /u06`

8.6.3 Perform State Cleanup

Step	Server, User	Action
1	Each DR site application tier server exalgbvm0089-eoib2,exalgbvm0090-eoib2, applmgr	If EBS was up and running at the time of the failure that lead to the failover to the secondary site, the OPMN and HTTP state cache will need to be cleaned up. The issue is that the state data for OPMN and HTTP are from the live running system on the (old) primary. The state data needs to be removed or moved so that OPMN can start correctly, otherwise, it will terminate with an internal error. Follow the solution in MOS Doc 2174221.1 to clean it up. Each application tier node has a subdirectory EBS_web_VIS_OHSx, where x corresponds to the application tier node. For example, EBS_web_VIS_OHS1 is for the master application tier node, EBS_web_VIS_OHS2 for the second application tier node, and so on. Importantly, set your environment to the RUN file system for these steps. Perform the cleanup for each application tier node, in the subdirectory for that node. For example: `$ cd /u06/app/ebs/vis/fs1/FMW_Home/webtier/instances/EBS_web_VIS_OHS1/config/OPMN/opmn` `$ mv states states_backup`
2	DR site application tier servers exalgbvm0089-eoib2,exalgbvm0090-eoib2, applmgr	To verify that the application tiers can connect to the database, attempt to use SQL*Plus to connect to the database. This is just an extra step to ensure connections can be made to the new primary database.

3	All NEW primary application tier nodes, applmgr	**NOTE**: The EBS script that sets up the administrator's OS environment will not succeed until the standby site's physical host names are in the node_info.txt file. This file is cumulative, with the new host information appended to the file. To add the standby site's application tier physical host names to the node_info.txt file, you need to run AutoConfig without sourcing the environment on both the RUN and PATCH file systems on all application tier nodes. This only needs to be done once in the environment if done in a way that persists – in a true switchover or failover to the standby site. On each application tier node, in both the RUN and the PATCH file systems: Go to the directory that would be pointed to by $INST_TOP: `/<your base directory>/<run file system>/inst/admin/scripts` For example, on our system (fs2 is the run file system): `/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/admin/scripts` Execute AutoConfig from this location: `./adautocfg.sh` Source the environment. As the replication has been reversed, the updated node_info.txt file will be replicated to the standby site, making this change permanent at both sites. You can skip the next step, as AutoConfig was run in this step.
4	DR site application tier servers exalgbvm0089-eoib2,exalgbvm0090-eoib2, applmgr	Run AutoConfig on all application tier servers on both RUN and PATCH file systems.

8.6.4 Start Application Services

Step	Server, User	Action
1	DR site app lication tier servers exalgbvm0089-eoib2,exalgbvm0090-eoib2, applmgr	Start the EBS application services on the master application tier node, then all slave nodes. On each node: `$ adstrtal.sh` If there are errors, check the startup logs for issues.

2	DR site primary application tier server exalgbvm0089-eoib2, applmgr	To validate that the RUN and PATCH file systems on all nodes check out, perform an adop validate. `$ adop –validate` `…` `Validating credentials.` `Initializing.` `  Run Edition context  :` `/u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsapp1.xml` `  Patch edition context:` `/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp1/appl/admin/VIS_ebsapp1.xml` `Node registry is valid.` `==========` `ADOP (C.Delta.8)` `Node: ebsapp1` `Command: validate` `Log:` `/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/adopConsole.log` `==========` `Checking for existing patching cycle.` `  No existing patching cycle exists` `Verifying SSH connection to all nodes.` `  Log:` `/u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/logs/appl/rgf/TXK/verifyssh.log` `  Output:` `/u06/app/ebs/vis/fs1/inst/apps/VIS_ebsapp1/logs/appl/rgf/TXK/out.xml` `  Remote execution is operational.` `Running adop validations on Admin node: [ebsapp1].` `  Log:` `ebsapp1:/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/ebsapp1` `  Output:` `/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/remote_execution_result_level1.xml` `  txkADOPEvalSrvStatus.pl returned SUCCESS` `Running adop validations on node(s): [ebsapp2].` `  Log: ebsapp2:` `/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/ebsapp2` `  Output:` `/u06/app/ebs/vis/fs_ne/EBSapps/log/adop/65/20170216_154704/validate/remote_execution_result_level2.xml` `  txkADOPEvalSrvStatus.pl returned SUCCESS`

		`adop exiting with status = 0 (Success)`
3	Browser logged on to application at DR site	If all the services started, log into the EBS application to check that all administrative tasks and functions are running properly. Your new primary site should be up and ready for production workloads.

8.7 Patching the Database in an Engineered Systems Environment

The objective of patching the database is to minimize downtime for the production Oracle E-Business Suite system and increase accuracy when applying quarterly database patches, including all the overlay and standard database patches required by EBS.

EBS customers can apply Oracle's quarterly database patches shortly after they are available, after testing has been completed for the quarterly bundle and the additional patches required by EBS. See Database Patches Required by Oracle E-Business Suite on Oracle Engineered Systems: Exadata Database Machines and SuperClusters (Doc ID 1392527.1). As noted in that document, EBS provides a consolidated patch zip file that contains all the required patches.

In this case study, we perform Standby-First database patching. See Oracle Patch Assurance - Data Guard Standby-First Patch Apply (Doc ID 1265700.1).

After patching one of the database Oracle homes at the standby, the standby database is converted to a snapshot standby. The application can then be tested.

The high-level steps:

1. Prepare for database patching.
2. Patch one standby database instance home.
3. Convert the standby to a Snapshot Standby, then check the patched standby database home using ETCC
4. Propagate the patched database Oracle home to the other standby database homes.
5. Complete the database patching process on the snapshot standby.
6. Test the patched standby environment using ZFS snapshot and Oracle Snapshot Standby.
7. Propagate the patched database home to the primary.

8.7.1 Prepare for Database Patching

1. Download the quarterly database bundle patch and any critical database patches. See Exadata Database Machine and Exadata Storage Server Supported Versions (Doc ID 888828.1) for Exadata Database Machines; Oracle SuperCluster Supported Software Versions – All Hardware Types (Doc ID 1567979.1) for SuperCluster.
2. Using the consolidated zip file referenced in MOS Doc 1392527.1, download all database patches required by Oracle E-Business Suite for the database bundle patch to be applied.
3. Review the READMEs for all patches to be applied. Validate all patches can be applied to the standby first. Note that as of the Jan2017 PSU, the OJVM patch can be applied to the standby first (using the Oracle RAC rolling approach) given certain conditions. See Oracle RAC Rolling Install Process for the "Oracle JavaVM

Component Database PSU" (OJVM PSU) Patches (Doc ID 2217053.1) for a section describing standby-first patching.

4. Back up or zip up the primary and standby database Oracle homes to be patched before starting.

8.7.2 Patch One Standby Database Instance Home

We leave the remaining instances running so that redo shipping can continue while we are patching, keeping our Recovery Point Objective (RPO) low.

The steps in this section follow the instructions in MOS doc 1392527.1 for applying a bundle patch to the database homes.

Step	Server, User	Action
1	One standby database node – exabadm01, oracle_ebs	Shut down the standby database instance to be patched and its listener. `$ srvctl stop instance –d vis_b –I VIS1` `$ lsnrctl stop vis`
2	One standby database node – exabadm01, oracle_ebs	Follow the instructions in MOS Doc 1392527.1 to apply the bundle patch. First, identify the list of patches that conflict with the bundle patch you are applying. In our case study, we applied the January 2017 bundle patch PSU for engineered systems (17117), patch 24968615. MOS Doc 1392527.1 says to check each sub-patch for conflicts. There are four sub-patches: 21436941, 24732088, 24828643, and 25101514. Three of the sub-patches had no conflicts (21436941, 24828643, and 25101514); their OPatch pre-req check returned this: `Invoking prereq "checkconflictagainstohwithdetail"` `Prereq "checkConflictAgainstOHWithDetail" passed.` `OPatch succeeded.`
3	One standby database node – exabadm01, oracle_ebs	The conflict check we ran in step 2 for patch 24732088 did return a list of conflicting patches: `$ opatch prereq CheckConflictAgainstOHWithDetail -phBaseDir ./24732088` `Oracle Interim Patch Installer version 12.2.0.1.8` `Copyright (c) 2017, Oracle Corporation. All rights reserved.` `PREREQ session` `Oracle Home : /u01/app/oracle_ebs/product/12.1.0` `Central Inventory : /u01/app/oraInventory` `from :` `/u01/app/oracle_ebs/product/12.1.0/oraInst.loc` `OPatch version : 12.2.0.1.8` `OUI version : 12.1.0.2.0` `Log file location :` `/u01/app/oracle_ebs/product/12.1.0/cfgtoollogs/opatch/opatch2017-03-03_10-56-52AM_1.log`

		Invoking prereq "checkconflictagainstohwithdetail" ZOP-47: The patch(es) has supersets with other patches installed in the Oracle Home (or) among themselves. ZOP-40: The patch(es) has conflicts with other patches installed in the Oracle Home (or) among themselves. Prereq "checkConflictAgainstOHWithDetail" failed. Summary of Conflict Analysis: There are no patches that can be applied now. **Following patches have conflicts. Please contact Oracle Support and get the merged patch of the patches :** **21321429, 23124597, 23530796, 23595848, 24732088** Whole composite patch Conflicts/Supersets are: Composite Patch : 24732088 Conflict with 23595848 Bug Superset of 19908836 Conflict with 23530796 Bug Conflict with 23124597 Bug Superset of 22496904 Conflict with 21321429 Bug Superset of 20627866 Bug Superset of 20734332 Detail Conflicts/Supersets for each patch are: Sub-Patch : 24340679 Conflict with 23595848 Conflict details: [etc.] […] /u01/app/oracle_ebs/product/12.1.0/rdbms/admin/utluppkg.sql OPatch succeeded.
4	One standby database node – exabadm01, oracle_ebs	This list of patches to be rolled back is held in the list of patches that have conflicts, which the OPatch output above suggests getting a merged patch for: **Following patches have conflicts. Please contact Oracle Support and get the merged patch of the patches :** **21321429, 23124597, 23530796, 23595848, 24732088** You need to roll back the patches that conflict with the patch being analyzed, which is the last patch in the list above. The other patches need to be rolled back in order to apply that patch. The rest of the instructions in MOS doc 1392527.1 will take care of re-applying the missing patches. Roll back patches: 21321429, 23124597, 23530796, 23595848: $ opatch nrollback -id 21321429,23124597,23530796,23595848 -local […] Opatch succeeded.

5	One standby database node – exabadm01, root	Apply the 170117 bundle patch 24968615 using OPatchAuto which will apply it locally on THIS node only. You must be root to run OPatchAuto. Use the OPatchAuto out of the GI Home per the readme. As root, on our environment: `$ export PATH=$PATH:/u01/app/12.1.0.2/grid/OPatch` `# opatch version` `OPatch Version: 12.2.0.1.8` `OPatch succeeded.` Attempt to apply the patch: `# opatchauto apply /u01/app/oracle_ebs/product/patchdepot/170117/24968615 -oh /u01/app/oracle_ebs/product/12.1.0` Output shows that we have another conflict with patch 20476776; it too needs to be rolled back. As the database user (in our case, oracle_ebs), using the database Oracle home: `$ opatch rollback -id 20476776` The rollback succeeded. `Attempt to apply the bundle patch as root, using OPatch from the GI home:` `$ opatchauto apply /u01/app/oracle_ebs/product/patchdepot/170117/24968615 -oh /u01/app/oracle_ebs/product/12.1.0` The apply succeeded. `$ opatch lspatches` `25101514;OCW Interim patch for 25101514` `24732088;DATABASE BUNDLE PATCH: 12.1.0.2.170117 (24732088)` `21904072;` `21864513;` `21967332;` `23177536;Database PSU 12.1.0.2.160719, Oracle JavaVM Component (JUL2016)` `22731026;` `21923026;` `19472320;` `OPatch succeeded.`
6	One standby database node – exabadm01, oracle_ebs	`Apply the OJVM patch 24917972.` `$ opatch apply -local` `Succeeded.` `$ opatch lspatches` `24917972;Database PSU 12.1.0.2.170117, Oracle JavaVM Component (JAN2017)` `25101514;OCW Interim patch for 25101514` `24732088;DATABASE BUNDLE PATCH: 12.1.0.2.170117 (24732088)`

7	One standby database node – exabadm01, oracle_ebs	Apply the remaining overlay and standard patches required by EBS in the order specified per the instructions in MOS Doc 1392527.1. Make sure you are in each of the patch directories before attempting to apply that particular patch. Always use the `-local` command line option when applying the remaining patches with OPatch. Cross-check MOS Doc 1392527.1's list of patches to apply against the lsinventory above. There is no need to try to apply patches already listed in the inventory.
8	One standby database node – exabadm01, oracle_ebs	Check that all patches have been applied and that the inventory is still good: `$ opatch lspatches` `18793246;` `25203714;` `25154792;` `21321429;` `24684434;` `24326444;` `20766180;` `24917972;Database PSU 12.1.0.2.170117, Oracle JavaVM Component (JAN2017)` `25101514;OCW Interim patch for 25101514` `24732088;DATABASE BUNDLE PATCH: 12.1.0.2.170117 (24732088)` `21904072;` `21864513;` `21967332;` `22731026;` `21923026;` `19472320;` `OPatch succeeded.`

8.7.3 Check the Patched Standby Database Home

Check the patched standby database home using ETCC and Oracle Snapshot Standby.

Step	Server, User	Action
1	One standby database node – exabadm01, oracle_ebs	Using SRVCTL, shut the standby database down completely then startup MOUNT only the patched instance: `srvctl stop database –d vis_b` `srvctl start instance –d vis_b –I VIS1 –o mount` Note that redo shipping from the primary does not occur between the times the database isshut down and the instance is started.

2	Any database node, primary or standby – exabadm01, oracle_ebs	Use Data Guard Broker to check for any configuration problems. You may need to wait a few minutes before SHOW CONFIGURATION returns SUCCESS. Only one instance is up but Data Guard Broker should still show SUCCESS. `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `    VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS   (status updated 17 seconds ago)`
3	Any node at standby or primary (we used exabadm01), oracle_ebs	If all is OK within Data Guard Broker, convert the physical standby to a snapshot standby: `DGMGRL> convert database 'VIS_B' to snapshot standby;` `Converting database "VIS_B" to a Snapshot Standby database,` `  please wait...` `Database "VIS_B" converted successfully` `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `    VIS_B - Snapshot standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS   (status updated 38 seconds ago)` Note that redo shipping from the primary continues while the standby database is in snapshot standby mode. The redo is not applied, but it is available on the standby environment.
4	Both Standby Oracle RAC nodes – exabadm01,exabadm02, Oracle_ebs	If after converting to a snapshot standby the second un-patched instance starts, shut it down. `srvctl status database –d vis_b` If the second instance is running: `srvctl stop instance –d vis_b –i  VIS2 –o immediate`
5	One standby database node – exabadm01, oracle_ebs	Download the ETCC tool and run it following the instructions to validate that all patches have been applied: `./checkDBpatch.sh` Partial output from our test: `Obtained list of bugfixes to be applied and the list to be` `  rolled back.` `Now checking database ORACLE_HOME.` `The opatch utility is version 12.2.0.1.8.`

		DB-ETCC is compatible with this opatch version. Found patch records in the inventory. Checking Mapping XML file for 12.1.0.2.170117ProactiveBP **All the required one-off bugfixes are present in database ORACLE_HOME.** Stored Technology Codelevel Checker results in the database VIS successfully. Finished prerequisite patch testing : Fri Mar 3 17:18:10 PST 2017 Log file for this session: ./checkDBpatch_20625.log

8.7.4 Propagate the Patched Oracle Home

Propagate the patched database Oracle home to the other standby database homes.

Step	Server, User	Action
1	One standby database node – exabadm01, root	Using the script rsync_EBS_OH.sh from Appendix E, run rsync without the `--no_dry-run` option as root to ensure that the excluded directories not included and that all other directories are updated. Do not specify the `--with_ebs_config` option as you do not want to copy over the appsutil, dbs or the network/admin directories. For example, from exabadm01: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadm02 --os_user root` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exabadm02` `os_user = root` `dry_run = true` `with_ebs_config = false` `About to execute the following rsync command:` `rsync -avzh --progress --exclude=appsutil --exclude=network/admin --exclude=dbs --dry-run /u01/app/oracle_ebs/product/12.1.0/ root@exabadm02:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y|y|N|n] :`

2	One standby database node – exabadm01, root	If the dry run version of the command was OK, then run the rsync command with the –no_dry-run option. For example: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadm02 --os_user root -- no_dry_run` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exabadm02` `os_user = root` `dry_run = false` `with_ebs_config = false` `About to execute the following rsync command:` `rsync -avzh --progress --exclude=appsutil -- exclude=network/admin --exclude=dbs /u01/app/oracle_ebs/product/12.1.0/ root@exabadm02:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y|y|N|n] :`
3	Each target standby database node – exabadm02, oracle_ebs	As we ran rsync to our second node exabadm02, we now need to detach and re-attach the Oracle home to the inventory on that node. Make sure you use the `-local` option. To detach: `$ cd $ORACLE_HOME/oui/bin` `$ ./runInstaller -silent -detachhome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" -local` To re-attach: `$ ./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" ORACLE_HOME_NAME=dbhome_ebs1 CLUSTER_NODES="{exabadm01,exabadm02}" LOCAL_NODE="exabadm02" -local`
4	One standby database node – exabadm02, oracle_ebs	Validate that the target node's inventory is good, and shows the patches as the same as the source. For example: `$ opatch lspatches` `18793246;` `25203714;` `25154792;` `21321429;` `24684434;` `24326444;` `20766180;`

		24917972;Database PSU 12.1.0.2.170117, Oracle JavaVM Component (JAN2017) 25101514;OCW Interim patch for 25101514 24732088;DATABASE BUNDLE PATCH: 12.1.0.2.170117 (24732088) 21904072; 21864513; 21967332; 22731026; 21923026; 19472320; OPatch succeeded.

8.7.5 Complete the Database Patching Process

Complete the database patching process on the snapshot standby.

Step	Server, User	Action
1	Original standby database node – exabadm01, oracle_ebs	Run datapatch. This instance should still be up. **Note:** as of 12.1.0.2.170117, UPGRADE mode is no longer required after applying the OJVM patch, but you must specify the option `-skip_upgrade_check` when running datapatch. `$ cd $ORACLE_HOME/OPatch` `$ ./datapatch -skip_upgrade_check -verbose` When datapatch completes, review the log files listed in its output. All entries in the list should say “no errors”. If an error is indicated, read the log file and diagnose further.
2	One standby database node – exabadm01, oracle_ebs	After successfully running datapatch, it is necessary to run the adgrants.sql script located on the application tier in $APPL_TOP/admin/adgrants.sql. Copy this script to one of the database nodes and run it there. You must run it from SQL*Plus when connected as sysdba. `SQL> @adgrants.sql apps`
3	One standby database node – exabadm01, oracle_ebs	Compile any invalid packages. `$ cd $ORACLE_HOME/rdbms/admin` `$ sqlplus / as sysdba` `SQL> @utlrp.sql`
4	One standby database node – exabadm01, oracle_ebs	Start all remaining Oracle RAC instances. For example: `$ srvctl start database –d vis_b`

8.7.6 Test the Standby

Test the patched standby environment using ZFS snapshot and Oracle Snapshot Standby.

Step	Server, User	Action
1	ZFS and all standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	To test the application on the now-patched Oracle RAC database, follow all of the steps under section Create a Snapshot Clone of the Shared File System Being Replicated to the Disaster Recovery Site to create a ZFS snapshot clone for testing the application tier. Follow those steps and mount the snapshot file system on each of the application tier servers. If the application is running on the primary make sure you do step 4 in section Configure the Web Entry URLs and Port Numbers to clear the HTTP states – otherwise, the application services on the snapshot standby will fail to start.
2	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	Re-source the environment, then run AutoConfig on all standby application tier nodes for both RUN and PATCH file systems. If there are issues, for guidance refer to MOS Docs 2191284.1, 2196190.1, and 2064223.1. Remember, you are expected to get a completion error when running AutoConfig on the PATCH file system, as there will not be a PATCH edition in the database.
3	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	Reset the web entry points using the script (or similar script) found in Appendix C: Set Web Entry Point. Then, run AutoConfig again on both the RUN and PATCH file systems.
4	Standby application tier, primary node – exalgbvm0089-eoib2, applmgr	You can validate the system on the snapshot standby using adop: `$ adop –validate` If the WLS administration server is not running, adop will start it for you. You should see Success for all application tier nodes.
5	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	Start application services. `$ adstrtal.sh`

6	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	Test the application now running on the patched Oracle RAC snapshot standby database. Conduct testing of critical jobs and online workload where possible.

8.7.7 Propagate the Patch Database Homes to the Primary

Step	Server, User	Action
1	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, applmgr	When testing is complete shut down the application services on all standby application tier nodes. `$ adstpall.sh`
2	All standby application tier nodes – exalgbvm0089-eoib2, scan03vm0090-eoib2, root	Umount the snapshot clone. `$ umount /u06`
3	ZFS appliance adminstrator	Drop the snapshot clone. To drop the snapshot clone on ZFS, log into the ZFS browser UI as an administrator. a) Click Shares. b) Click Projects on the left. c) Locate the snapshot clone project (ours is called EBS_DEV_SITE_B_SNAPSHOT). d) Click the trashcan icon to delete the project. e) Confirm the operation.
4	One standby database node – exabadm01, oracle_ebs	Convert the snapshot standby database back to a physical standby using DGMGRL. `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database`

		VIS_B - Snapshot standby database Fast-Start Failover: DISABLED Configuration Status: SUCCESS (status updated 15 seconds ago) DGMGRL> convert database 'VIS_B' to physical standby; Converting database "VIS_B" to a Physical Standby database, please wait... Database "VIS_B" converted successfully
5	**Primary** application tier, all nodes – exalgacn23,exalgacn24, applmgr	Shut all application services and concurrent managers down on all **primary** application tier nodes. `$ adstpall.sh`
6	One primary database node – exaadb03, oracle_ebs	Shut all primary database instances down. `$ srvctl stop database –d vis_a –o immediate` Then either stop the listeners on all nodes using this command: `$ srvctl stop listener –l vis` Or run the following on each Oracle RAC node: `$ lsnrctl stop vis`
7	One standby database node – exabadm01, root	Using the script rsync_EBS_OH.sh from Appendix E, run rsync from one standby database Oracle home to the Oracle homes on the primary. Because we exclude the three directories--appsutil, network/admin, and dbs--we can use one ORACLE_HOME as the source. The script excludes these directories by default. Do not specify the `–with_ebs_config` option when running the script. We ran without the `–no_dry-run` option as shown in the following example. The script uses rsync to synchronize the ORACLE_HOME from exabadm01 to exaadb03: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exaadb03 --os_user root` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exaadb03` `os_user = root` `dry_run = true` `with_ebs_config = false` `About to execute the following rsync command:` `rsync -avzh --progress --dry_run --exclude=appsutil --exclude=network/admin --exclude=dbs /u01/app/oracle_ebs/product/12.1.0/ root@exaadb03:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y|y|N|n] :`

		And from exabadm01 to exaabd04: `$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exaabd04 --os_user root` `Using the following parameters:` `source ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target ORACLE_HOME = /u01/app/oracle_ebs/product/12.1.0` `target_hosts = exaabd04` `os_user = root` `dry_run = true` `with_ebs_config = false` `About to execute the following rsync command:` `rsync -avzh -progress --dry_run --exclude=appsutil --exclude=network/admin --exclude=dbs /u01/app/oracle_ebs/product/12.1.0/ root@exaabd04:/u01/app/oracle_ebs/product/12.1.0/` `OK to continue? [Y|y|N|n] :` Now, re-execute the above two commands with the `--no_dry_run` option to perform the actual rsync copy. **Note:** The complete rsync for both primary Oracle RAC nodes took 7 minutes in our environment. Your results may vary depending on network bandwidth and latency.
8	Each primary database node - exaadb03, exaabd04, oracle_ebs	Detach and re-attach the Oracle homes. Run the following commands on each Oracle RAC node. To detach: `$ cd $ORACLE_HOME/oui/bin` `$ ./runInstaller -silent -detachhome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" -local` To re-attach for exaadb03: `$ ./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" ORACLE_HOME_NAME=dbhome_ebs1 CLUSTER_NODES="{sam08db03,exaabd04}" LOCAL_NODE="exaadb03" -local` To re-attach for exaabd04: `$ ./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle_ebs/product/12.1.0" ORACLE_HOME_NAME=dbhome_ebs1 CLUSTER_NODES="{sam08db03,exaabd04}" LOCAL_NODE="exaabd04" -local`

9	Each primary database node – exaadb03, exaabd04, oracle_ebs	Run the OPatch lspatches command on each node to validate that the Oracle home inventories are good and they have the new patches. `$ opatch lspatches` `18793246;` `25203714;` `25154792;` `21321429;` `24684434;` `24326444;` `20766180;` `24917972;Database PSU 12.1.0.2.170117, Oracle JavaVM Component (JAN2017)` `25101514;OCW Interim patch for 25101514` `24732088;DATABASE BUNDLE PATCH: 12.1.0.2.170117 (24732088)` `21904072;` `21864513;` `21967332;` `22731026;` `21923026;` `19472320;` `OPatch succeeded.` Both primary Oracle RAC nodes are patched and show the same output.
10	One primary database node – exaadb03, oracle_ebs	Using SQL*Plus, start one instance and run datapatch. **Note:** As of 12.1.0.2.170117, UPGRADE mode is no longer required after applying the OJVM patch, but you must specify the option `-skip_upgrade_check` when running datapatch. `$ cd $ORACLE_HOME/OPatch` `SQL> startup` `SQL> exit` `$ ./datapatch -skip_upgrade_check -verbose` When datapatch completes, review the log files listed in its output. All entries in the list should say “no errors”. If an error is indicated, read the log file and diagnose further.
11	One primary database node – exaadb03, oracle_ebs	After all patches have been applied and datapatch successfully run, run the adgrants.sql script located on the application tier in $APPL_TOP/admin/adgrants.sql. Copy this script to one of the database nodes and run it there. You must run it from SQL*Plus connected as sysdba. `$ sqlplus / as sysdba` `SQL> @adgrants.sql apps`
12	One primary database node – exaadb03, oracle_ebs	Compile any invalid packages. `$ cd $ORACLE_HOME/rdbms/admin` `$ sqlplus / as sysdba` `SQL> @utlrp.sql`

13	One primary database node – exaadb03, oracle_ebs	Start all Oracle RAC instances and listeners at the primary. `$ srvctl start database –d vis_a` `$ srvctl start listener –l vis`
14	One primary database node – exaadb03, oracle_ebs	Check the Data Guard Broker status with SHOW CONFIGURATION. `DGMGRL> show configuration` `Configuration - dg_vis` `  Protection Mode: MaxPerformance` `  Members:` `  VIS_A - Primary database` `  VIS_B - Physical standby database` `Fast-Start Failover: DISABLED` `Configuration Status:` `SUCCESS  (status updated 49 seconds ago)` `DGMGRL> show database 'VIS_B'` `Database - VIS_B` `  Role:  PHYSICAL STANDBY` `  Intended State:  APPLY-ON` `  Transport Lag:  0 seconds (computed 1 second ago)` `  Apply Lag:  0 seconds (computed 1 second ago)` `  Average Apply Rate: 36.29 MByte/s` `  Real Time Query:  OFF` `  Instance(s):` `  VIS1 (apply instance)` `  VIS2` `Database Status:` `SUCCESS`
15	All primary database nodes – exaadb03, Exaabd04, oracle_ebs	Run the ETCC tool on all Oracle RAC nodes per MOS Doc 1594274.1. You should see the following: `All the required one-off bugfixes are present in database ORACLE_HOME.`
16	Primary application tier, all nodes – exalgacn23, exalgacn24, applmgr	Run AutoConfig on both the RUN and PATCH file systems. AutoConfig will complete with errors on the PATCH file system, but this is expected as there is no PATCH edition in the database. AutoConfig should complete without errors on the RUN file system.
17	Primary application tier, all nodes – exalgacn23, exalgacn24, applmgr	Start all application services at the primary. `$ adstrtal.sh` This ends the outage at the primary site.

9 References

This section lists the reference material that can be read to gain further knowledge in the various areas relevant to establishing a Maximum Availability Architecture for Oracle E-Business Suite, including the materials referenced to develop the processes described in this paper.

9.1 Database MAA

Oracle Database High Availability Overview

Automatic Storage Management Administrator's Guide

Database Backup and Recovery User's Guide

"Oracle Flashback Technology" in *Oracle Database Concepts*

Flashback Database Best Practices & Performance (Doc ID 565535.1)

"Using Edition Based Redefinition" in the *Database Development Guide*

9.2 Database Oracle Data Guard

Data Guard Concepts and Administration

Data Guard Broker

Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC) (Doc ID 1617946.1)

Using Active Data Guard Reporting with Oracle E-Business Suite Release 12.2 and an Oracle 11*g* or 12*c* Database (Doc ID 1944539.1)

Oracle Patch Assurance - Data Guard Standby-First Patch Apply (Doc ID 1265700.1)

9.3 Oracle Real Application Clusters (RAC)

Oracle Real Application Clusters Administration and Deployment Guide

Clusterware Administration and Deployment Guide

Using Oracle 12c Release 1 (12.1) Real Application Clusters with Oracle E-Business Suite Release R12.2 (Doc ID 1626606.1)

Oracle RAC Rolling Install Process for the "Oracle JavaVM Component Database PSU" (OJVM PSU) Patches (Doc ID 2217053.1)

9.4 Oracle E-Business Suite MAA

Business Continuity for Oracle E-Business Suite Release 12.2 Using Logical Host Names with an Oracle 12c Physical Standby Database (Doc ID 2246690.1)

Oracle E-Business Suite on Exadata Database Machine (White Paper)

9.5 Oracle E-Business Suite Parallel Concurrent Manager

Configuring and Managing Oracle E-Business Suite Release 12.2.x Forms and Concurrent Processing for Oracle RAC (Doc ID 2029173.1)

How to Activate Parallel Concurrent Processing - Background Facts and Setup Steps (Doc ID 602899.1)

Concurrent Processing - Parallel Concurrent Processing Failover/Failback Expectations (Doc ID 271090.1)

Managing Concurrent Manager Log and Out Directories (Doc ID 1616827.1)

Concurrent Processing - Purge Concurrent Request and/or Manager Data Program (FNDCPPUR) (Doc ID 104282.1)
Concurrent Processing - Best Practices for Performance for Concurrent Managers in E-Business Suite (Doc ID 1057802.1)
Concurrent Processing - Product Information Center (PIC) (Doc ID 1304305.1)

9.6 Oracle E-Business Suite Configuration and Management

Database Initialization Parameters for Oracle E-Business Suite Release 12 (Doc ID 396009.1)
Can The Primary Node Hosting the WLS Admin Server Failover To A 12.2 Slave Node, Migrating the Master WLS Admin Server To Another Machine (Doc ID 1986122.1)
Oracle Applications E-Business Suite 12.2 Fusion Middleware Log Files: Locate, View, and Control (Doc ID 1366187.1)
Managing Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 (Doc ID 1905593.1)
HTTP Server Is Either Slow Or Stops Responding When Installed On A NFS Mounted Drive [Doc ID 560853.1]
Using Load-Balancers with Oracle E-Business Suite Release 12.0 and 12.1 (Doc ID 380489.1)
E-Business Suite 12.2 Detailed Steps To Change The R12.2 Default Port To 80 (Doc ID 2072420.1)
How to Change Applications Passwords using Applications Schema Password Change Utility (FNDCPASS or AFPASSWD) (Doc ID 437260.1)
Secure Configuration Guide for Oracle E-Business Suite Release 12 (Doc ID 403537.1)
R12.2 : How To Create, Update or Rebuild The Central Inventory For Oracle Applications E-Business Suite ? (Doc ID 1588609.1)
How to Create a Clean oraInventory in Release 12.2 (Doc ID 1967205.1)
Oracle E-Business Suite Release 12.1 with Oracle Database 11g Advanced Compression (Doc ID 1110648.1)
Best Practices for Gathering Statistics with Oracle E-Business Suite (Doc ID 1586374.1)
Fixed Objects Statistics Considerations [Doc ID 798257.1]
Sharing the Application Tier File System in Oracle E-Business Suite Release 12.2 (Doc ID 1375769.1)

9.7 Cloning Oracle E-Business Suite

Cloning Oracle E-Business Suite Release 12.2 RAC Enabled Systems with Rapid Clone (Doc ID 1679270.1)
Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone (Doc ID 1383621.1)

9.8 Oracle E-Business Suite Application Tier Patching

Applying the Latest AD and TXK Release Update Packs to Oracle E-Business Suite Release 12.2 (Doc ID 1617461.1)
Oracle E-Business Suite Applications DBA and Technology Stack Release Notes for R12.AD.C.Delta.9 and R12.TXK.C.Delta.9 (Doc ID 2233485.1)
12.2 Online Patching Utility ADOP Fails During Cutover Due to Error "[UNEXPECTED] adop has detected a configured disaster recovery site" (Doc ID 2131833.1)
The adop Patch Utility in the *E-Business Suite Maintenance Guide*
Oracle E-Business Suite Applications DBA and Technology Stack Release Notes for R12.AD.C.Delta.7 and R12.TXK.C.Delta.7 (Doc ID 2033780.1)

OPMN Fails to Start and Says to Check Adopmnctl.txt (Doc ID 2174221.1)

9.9 Oracle Engineered Systems (Exadata, SuperCluster)

Exadata Database Machine and Exadata Storage Server Supported Versions (Doc ID 888828.1)

Oracle Exadata Database Machine exachk or HealthCheck (Doc ID 1070954.1)

Oracle Exadata Best Practices (Doc ID 757552.1)

Database Patches Required by Oracle E-Business Suite on Oracle Engineered Systems: Exadata Database Machines and SuperClusters (Doc ID 1392527.1)

Exadata Write-Back Flash Cache - FAQ (Doc ID 1500257.1)

Oracle Sun Database Machine X2-2/X2-8, X3-2/X3-8 and X4-2 Performance Best Practices (Doc ID 1274475.1)

Configuring Exadata I/O Resource Manager for Common Scenarios (Doc ID 1363188.1)

Exalogic Backup and Recovery Best Practices (White Paper)

How To Configure NIS Master, Slave And Client Configuration In Exalogic Virtual Environment (Doc ID 1516025.1)

NFSv4 mount directories hang on Exalogic Machine[Article ID 1481713.1]

Oracle Sun Database Machine Setup/Configuration Best Practices (Doc ID 1274318.1)

Oracle SuperCluster Supported Software Versions – All Hardware Types (Doc ID 1567979.1)

9.10 OS Required Packages and HugePages

Oracle E-Business Suite Installation and Upgrade Notes Release 12 (12.2) for Linux x86-64 (Doc ID 1330701.1)

HugePages on Oracle Linux 64-bit (Doc ID 361468.1)

HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux (Doc ID 749851.1)

USE_LARGE_PAGES To Enable HugePages (Doc ID 1392497.1)

10 Appendix A: Verify Oracle E-Business Suite Required Packages

For the application tier servers, use the Oracle E-Business Suite Pre-Install RPM to install the reqired OS packages for Oracle E-Business Suite and complete important additional OS preparation tasks such as setting security limits, setting kernel parameters, and creating the OS user and groups. If the user and groups already exist, that step is skipped.

To install the required Oracle E-Business Suite packages, perform these steps on each application tier server:

Note: The following is for Oracle Linux 6.

1. Log on as root to the application server
2. Move to the yum configuration directory:

```
$ cd yum.repos.d
```

3. Download the public yum configuration file:

```
$ wget http://public-yum.oracle.com/public-yum-ol6.repo
```

4. Edit the public-yum-ol6.repo and set the “enabled” to 1 the following:

```
[ol6_latest]
name=Oracle Linux $releasever Latest ($basearch)
```

```
baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL6/latest/$basearch/
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=1

[ol6_addons]
name=Oracle Linux $releasever Add ons ($basearch)
baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL6/addons/$basearch/
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=1

[ol6_UEK_latest]
name=Latest Unbreakable Enterprise Kernel for Oracle Linux $releasever
  ($basearch)
baseurl=http://public-
  yum.oracle.com/repo/OracleLinux/OL6/UEK/latest/$basearch/
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=1
```

5. Install the required packages using yum:

```
$ yum update
$ yum install oracle-ebs-server-R12-preinstall
```

For further details on the required packages for Oracle E-Business Suite Release 12.2, please see MOS Doc: Oracle E-Business Suite Installation and Upgrade Notes Release 12 (12.2) for Linux x86-64 (Doc ID 1330701.1).

11 Appendix B: Managing SCAN for Primary and DR Sites

EBS AutoConfig regenerates nearly all of the configuration files including the tnsnames.ora, sqlnet.ora, and listener.ora files every time it is executed. It generates SCAN entries that hold IP addresses, which will not work on failover. The scripts in this section will replace the AutoConfig-generated entries with logical names for the SCAN VIPs. They combine several steps that would otherwise need to be performed manually, and will:

- Generate a set of host aliases that need to be added to /etc/hosts
- Copy the tnsnames.ora file in $TNS_ADMIN to the directory where the script is run
- Replace the IP addresses specified in the HOST parameter of the connect string with an appropriate hosts alias
- Concatenate the contents of the custom.txt file into the modified tnsnames.ora file, if the custom.txt file exists in the same directory
- Copy the modified tnsnames.ora file to the location and file name specified in the IFILE directive in the original tnsnames.ora file

There are two scripts:

- ebs_scan_db.sh – to be run on each database node in the Oracle RAC cluster
- ebs_scan_app.sh – to be run on each application tier node

These scripts are intended to be run only once only at the initial primary site. These scripts do not need to be run at the secondary site, as the entries they create are by definition usable without change when switching between sites.

As these scripts generate an IFILE, they will overwrite any existing IFILE. To retain your customizations, you need to put them into a “custom.txt” file that resides in the same directory that these scripts run in. These scripts will copy

the contents of the custom.txt into the final IFILE used by the tnsnames.ora file. For example, on your database nodes you will need to preserve Data Guard Broker related connect string aliases that may be overwritten by AutoConfig or by these scripts. Add the full connect string with aliaes into the custom.txt file for any custom connect strings that you need to ensure that they will be copied into the final IFILE. This must be done on each node (database and application tier).

On each database tier node:

1. Log onto each Oracle RAC database tier node as the owner of the EBS Oracle home (i.e., oracle_ebs).
2. Create a custom.txt file in a directory of your choice, and add any custom connect strings.
3. Copy the ebs_scan_db.sh script provided below to the same directory as your custom.txt script.
4. Edit the ebs_scan_db.sh script and change the variable v_scan_prefix to a name of your choice. This variable is currently set to ebsdb-scan.
5. Copy the edited scripts to all of the Oracle RAC nodes.
6. Run the ebs_scan_db.sh script on each node.

```
./ebs_scan_db.sh
```

7. Navigate to the $TNS_ADMIN directory, look at the <SIDX>_hostname_ifile.ora file, and verify that all of the IP addresses have been replaced with the aliases. The X in SIDX corresponds to each Oracle RAC instance.

On each application tier node:

1. Log onto the application tier node and source the RUN file system.
2. Create a custom.txt file in a directory of your choice, and add any custom connect strings. You will run the ebs_scan_app.sh scripts from this directory.
3. Copy the ebs_scan_app.sh script provided below to an application tier node into the same directory as your custom.txt script.
4. Locate the scan.txt file in a directory where you ran the ebs_scan_db.sh script on one of the database nodes, and copy the scan.txt file to the directory where you copied the ebs_scan_app.sh script above.
5. Edit the ebs_scan_app.sh script and change the v_scan_prefix variable to a name of your choice. We recommend that you use the same name that you used for the database nodes. Then copy all three files to the other application tier nodes.
6. Run the ebs_scan_app.sh script on each node.

```
./ebs_scan_app.sh
```

7. Navigate to the $TNS_ADMIN directory, look at the <SIDX>_hostname_ifile.ora file, and verify that all of the IP addresses have been replaced with the aliases. The X in SIDX corresponds to each Oracle RAC instance.

11.1 The custom.txt File

Our custom.txt file on each of the Oracle RAC database nodes contains the following:

```
# For the local listener:
VIS2_LOCAL=
        (DESCRIPTION=
                (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient02-
  vip.mycompany.com)(PORT=1523))
                (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient02-
  vip.mycompany.com)(PORT=1521))
        )

VIS1_LOCAL=
        (DESCRIPTION=
                (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient01-
  vip.mycompany.com)(PORT=1523))
                (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient01-
  vip.mycompany.com)(PORT=1521))
        )

# For Data Guard Broker
vis_a=
        (DESCRIPTION=
                (ADDRESS=(PROTOCOL=tcp)(HOST=exaa-scan3)(PORT=1521))
            (CONNECT_DATA=
                (SERVER = DEDICATED)
                (SERVICE_NAME=vis_a)
            )
        )

vis_b=
        (DESCRIPTION=
                (ADDRESS=(PROTOCOL=tcp)(HOST=exab-
  scan1.mycompany.com)(PORT=1521))
            (CONNECT_DATA=
                (SERVER = DEDICATED)
                (SERVICE_NAME=vis_b)
            )
        )
```

11.2 The ebs_scan_db.sh Script

Note: This references the file “custom.txt” that you created in step 2 of Set the Primary Database Up for Oracle Data Guard above.

```
#!/bin/bash

# ebs_scan_db.sh generates the scan.txt file and updates the tnsnames.ora
  file but writes it to the directory
# that this script runs in.  Then the updated tnsnames.ora file in the local
  directory is copied to the TNS_ADMIN
# directory, renamed to be used as an ifile.  The original tnsnames.ora file
  in the TNS_ADMIN directory remains
# unchanged.
#
# On each RAC node, run this script:
#
```

```
# Usage:
# First, edit the below variable v_scan_prefix to a name of your choosing.
  We have provided a default of ebs-scan.
# Then, run the script:
# $ ./ebs_scan_db.sh

# Edit this variable to a name of your choosing.
v_scan_prefix='ebsdb-scan'

# Do not change these variables.
v_scan_name=''
v_scan_IPv4_adr_list=''
v_scan_IPv6_adr_list=''
i_file=’’

# Get the scan name detail from srvctl

srvctl config scan | egrep "SCAN name|SCAN 0|SCAN 1|SCAN 2" > scan.txt

# Extract the SCAN name, then IPv4 and IPv6 addresses

v_scan_name=`egrep "SCAN name" scan.txt | awk '{ print $3'}`

echo "SCAN Name: $v_scan_name"
# Get the IP addresses

     for i in {0..2}
     do
          v_scan_IPv4_adr_list[$i]=`egrep "SCAN $i IPv4 VIP"  scan.txt | awk
  '{ print $5'}`
          v_scan_IPv6_adr_list[$i]=`egrep "SCAN $i IPv6 VIP"  scan.txt | awk
  '{ print $5'}`

     done

     # Generate what should be added to /etc/hosts

     echo "Add the following entries to your /etc/hosts files on each RAC
  node in THIS cluster."
     echo "## SCAN VIPs"

     len=${#v_scan_IPv4_adr_list[@]}
     j=1
     for ((i=0; i<${len}; i++));
     do
          printf "%s     %s%s\n" ${v_scan_IPv4_adr_list[$i]}
  ${v_scan_prefix} ${j}
          j=`expr $j + 1`
     done

     # Edit/update the tnsnames.ora file to replace the IP addresses with
  aliases.

     cp $TNS_ADMIN/tnsnames.ora .
     cp tnsnames.ora tnsnames.ora.sed
     j=1
     for ((i=0; i<${len}; i++));
     do
```

```
        sed -i "s/${v_scan_IPv4_adr_list[$i]}/${v_scan_prefix}${j}/g"
tnsnames.ora.sed
        j=`expr $j + 1`
   done
   # Remove the line containing the IFILE directive in the resulting
ifile.

   sed -i '/IFILE/d'  tnsnames.ora.sed

   # Add any custom TNS connections from cutsom.txt.  This is useful if
you have Data Guard configured
   # and the ifile is replaced from cloning.
   if [[ -a custom.txt ]]; then
        cat custom.txt >> tnsnames.ora.sed
   fi

   # Copy the modified tnsnames.ora file to the ifile listed in the
original tnsnames.ora IFILE directive.

   i_file=`grep IFILE tnsnames.ora | sed '{ s/IFILE=//}'`
   cp tnsnames.ora.sed  ${i_file}
   echo
   echo "Created file: ${i_file}"
   echo
```

11.3 The ebs_scan_app.sh Script

```
#!/bin/bash

# The ebs_scan_app.sh script does NOT run srvctl to generate the scan.txt
  file.  Use the ebs_scan_db.sh
# script to generate the scan.txt for the cluster SCAN names.  Place this
  script and the scan.txt (from
# the database node) onto each EBS app tier node into a directory of your
  choice.  But do not place it
# in the $TNS_ADMIN directory.
#
# Run the script on each app tier node:
# $ ./ebs_scan_app.sh

# Modify the value for v_scan_prefix. It should be the same value that is
  set in the sebs_scan_db.sh script.
v_scan_prefix='ebsdb-scan'

# Do not modify these variables.
v_scan_name=''
v_scan_IPv4_adr_list=''
v_scan_IPv6_adr_list=''
i_file=''

# Get the scan name detail from srvctl

#srvctl config scan | egrep "SCAN name|SCAN 0|SCAN 1|SCAN 2" > scan.txt

# Extract the SCAN name, then IPv4 and IPv6 addresses

v_scan_name=`egrep "SCAN name" scan.txt | awk '{ print $3'}`
```

```
echo "SCAN Name: $v_scan_name"
# Get the IP addresses

     for i in {0..2}
     do
          v_scan_IPv4_adr_list[$i]=`egrep "SCAN $i IPv4 VIP"  scan.txt | awk
  '{ print $5'}`
          v_scan_IPv6_adr_list[$i]=`egrep "SCAN $i IPv6 VIP"  scan.txt | awk
  '{ print $5'}`

     done

     # Generate what should be added to /etc/hosts

     echo "Add the following entries to your /etc/hosts files on each RAC
  node in THIS cluster."
     echo "## SCAN VIPs"

     len=${#v_scan_IPv4_adr_list[@]}
     j=1
     for ((i=0; i<${len}; i++));
     do
          printf "%s     %s%s\n" ${v_scan_IPv4_adr_list[$i]}
  ${v_scan_prefix} ${j}
          j=`expr $j + 1`
     done
     # Edit/update the tnsnames.ora file

     cp $TNS_ADMIN/tnsnames.ora .
     cp tnsnames.ora tnsnames.ora.sed
     j=1
     for ((i=0; i<${len}; i++));
     do
          sed -i "s/${v_scan_IPv4_adr_list[$i]}/${v_scan_prefix}${j}/g"
  tnsnames.ora.sed
          j=`expr $j + 1`
     done

     # Remove the line containing the IFILE directive in the resulting
  ifile.

     sed -i '/IFILE/d'  tnsnames.ora.sed

     # Add any custom TNS connections from custom.txt.  This is useful if
  you have Data Guard configured
     # and the ifile is replaced from cloning.

     if [[ -a custom.txt ]]; then
          cat custom.txt >> tnsnames.ora.sed
     fi

     i_file=`grep IFILE tnsnames.ora | sed '{ s/IFILE=//}'`
     cp tnsnames.ora.sed  ${i_file}
     echo
     echo "Created file: ${i_file}"
     echo
```

12 Appendix C: Set Web Entry Point

The following script is used to change the web entry point and port number to use a load balancer. As we do not register any of the logical host names on the application tier nodes into the DNS, a known entity on the network such as a load balancer is used in order to enable web access. There are a number of MOS Docs for EBS 12.2 that describe setting the web entry point. The following is a script for implementation.

```
#!/bin/bash

# Change the web entry URL and port number
# The EBS Run env must be sourced before running this script.
# Backup the context file before running this script.

# Backup context file
cp $CONTEXT_FILE $CONTEXT_FILE.`date +"%Y%m%d"`

$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webentryhost pd-
  bigip-sca-loan02

$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webentrydomain
  mycompany.com

$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_login_page http://pd-
  bigip-sca-loan02.mycompany.com:8002/OA_HTML/AppsLogin

$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_external_url
  http://pd-bigip-sca-loan02.mycompany.com:8002
## Additional parameters but these do not affect the current configuration
  unless you do an adop cutover.
$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_webport 8002
# This is used for the front-end load balancer.  s_active_webport should be
  set to the same port number
# specified in both s_login_page and s_external_url.
$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_active_webport 8002
$RUN_BASE/EBSapps/comn/util/jdk32/jre/bin/java -classpath
  $RUN_BASE/EBSapps/comn/java/classes:$RUN_BASE/FMW_Home/Oracle_EBS-
  app1/shared-libs/ebs-appsborg/WEB-INF/lib/ebsAppsborgManifest.jar
  oracle.apps.ad.context.UpdateContext $CONTEXT_FILE s_http_listen_parameter
  8002
```

13 Appendix D: Add Managed Services

As part of the case study, we added an application tier node, then added managed servers to this new server to match the same number and type of managed servers on the "master" Admin server. These steps are based on Managing Configuration of Oracle HTTP Server and Web Application Services in Oracle E-Business Suite Release 12.2 (Doc ID 1905593.1).

The following is a set of steps and commands we used to add the managed servers. When a new "slave" node is added to the EBS WLS cluster, it automatically creates one oacore_server, one forms_server, and any other server type where that type was "enabled" in the pairs file when the node was added.

You will need the following:

- Access to the WebLogic console. You must use the physical host name (DNS registered) to access the console
  - http://<physical host name.domain name>:<port number>/console
  - For example:http://exalgacn23.mycompany.com:7002/console
- The port pool number for the RUN and PATCH file systems
- The number and type of managed services you wish to add

In our case study, we have the following information:

- The port pool for the RUN file system is 2
- The port pool for the PATCH file system is 50
- The "master" admin server has 8 oacore servers and 3 forms servers
- The new "slave" server has one oacore server and one forms server
- We need to add 7 additional oacore servers and 2 forms servers to match the "master" server

Knowing the port pool numbers for the RUN and PATCH file system allows you to prevent port conflicts between the two file systems. The port number for each managed service must be unique on a specific server and not conflict between the RUN and PATCH file systems on any individual node. Therefore, we started with port 7204 and 7404 for oacore and forms respectively on the RUN file system.

Note: The next time an adop prepare phase is run, adop will automatically perform an fs_clone and set the port numbers for the new managed services on the PATCH file system.

13.1 Review the Configuration on the Admin Server

Log onto the WebLogic Admin server as the "weblogic" user using the appropriate URL. Modify the following example as necessary:

```
http://exalgacn23.mycompany.com:7002/console
```

Once logged in, expand the Environment tree on the left and click on Servers.

Review the managed services, the node, port number, and status of each so you can familiarize yourself with their configuration. On the new "slave" node, there should only be one managed server of each type running – these are the ones that were "enabled" when the slave node was added.

13.2 Add and Start Managed Services

Add and start managed services. Use these scripts as examples, modifying them to suit your environment. Run the commands in this step ONLY on the new slave node. We added 7 more oacore services.

```
# Create the new managed servers

perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \
 ebs-create-managedserver -contextfile=$CONTEXT_FILE \
 -managedsrvname=oacore_server10 -servicetype=oacore \
 -managedsrvport=7204 -logfile=$APPLRGF/TXK/addMS_oacoreserver10.log <<!EOF
<apps-password>
<Weblogic-password>
!EOF

/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh
  start oacore_server10 <<!EOF
<Weblogic-password>
!EOF

perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \
 ebs-create-managedserver -contextfile=$CONTEXT_FILE \
 -managedsrvname=oacore_server11 -servicetype=oacore \
 -managedsrvport=7205 -logfile=$APPLRGF/TXK/addMS_oacoreserver11.log <<!EOF
<apps-password>
<Weblogic-password>
!EOF

/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh
  start oacore_server11 <<!EOF
<Weblogic-password>
!EOF

[…]..

perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \
 ebs-create-managedserver -contextfile=$CONTEXT_FILE \
 -managedsrvname=oacore_server16 -servicetype=oacore \
 -managedsrvport=7210 -logfile=$APPLRGF/TXK/addMS_oacoreserver16.log <<!EOF
<apps-password>
<Weblogic-password>
!EOF

/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh
  start oacore_server16 <<!EOF
<Weblogic-password>
!EOF
```

Here, we created and started the two forms servers. Run these commands ONLY on the new slave node.

```
perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \
 ebs-create-managedserver -contextfile=$CONTEXT_FILE \
```

```
 -managedsrvname=forms_server5 -servicetype=forms \
 -managedsrvport=7404 -logfile=$APPLRGF/TXK/addMS_formsserver5.log <<!EOF
<apps-password>
<Weblogic-password>
!EOF

/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh
  start forms_server5 <<!EOF
<Weblogic-password>
!EOF

perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \
 ebs-create-managedserver -contextfile=$CONTEXT_FILE \
 -managedsrvname=forms_server6 -servicetype=forms \
 -managedsrvport=7405 -logfile=$APPLRGF/TXK/addMS_formsserver6.log <<!EOF
<apps-password>
<Weblogic-password>
!EOF

/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh
  start forms_server6 <<!EOF
<Weblogic-password>
!EOF
```

13.3 Finalize the Configuration

On ALL application tier servers (master WLS admin and slave servers) within the EBS WLS cluster, run the following command to configure the new manages services. The node name is the name of the node where new services were added. The ports are the ones used in the prior step. In our case, the command was as follows:

```
perl $FND_TOP/patch/115/bin/txkSetAppsConf.pl \
  -contextfile=$CONTEXT_FILE \
  -configoption=addMS \
  -oacore=ebsapp2.mycompany.com:7204 \
  -oacore=ebsapp2.mycompany.com:7205 \
  -oacore=ebsapp2.mycompany.com:7206 \
  -oacore=ebsapp2.mycompany.com:7207 \
  -oacore=ebsapp2.mycompany.com:7208 \
  -oacore=ebsapp2.mycompany.com:7209 \
  -oacore=ebsapp2.mycompany.com:7210 \
  -forms=ebsapp2.mycompany.com:7404 \
  -forms=ebsapp2.mycompany.com:7405
```

13.4 Bounce the HTTP Server

On each application tier server, stop and restart the HTTP server.

```
$ADMIN_SCRIPTS_HOME/adapcctl.sh stop
$ADMIN_SCRIPTS_HOME/adapcctl.sh start
```

14 Appendix E: Perform rsync on Oracle Homes

The following script uses rsync (a UNIX/Linux command) to copy a database ORACLE_HOME from one server to another. It can be used to copy a patched ORACLE_HOME to other nodes in an Oracle RAC cluster and to nodes

on a standby site. This method eliminates the need to run OPatch on every ORACLE_HOME. It can also be used to establish the database ORACLE_HOMEs at the standby site as full duplicates of those at the primary site.

The script is a wrapper around the `rsync` command to help automate and reduce errors when Oracle homes are copied. The script must run as root or under sudo.

Note: All command line options require two dashes, to be consistent with rsync command line options.

14.1 Using the rsync_EBS_OH.sh Script

`rsync_EBS_OH.sh`

Usage

```
rsync_EBS_OH.sh [ -h | --help ] --source_oh <path to ORACLE_HOME> --target_oh
  <path to ORACLE_HOME> --target_host <host> --os_user <username> [ --no_dry_run
  ] [ --with_ebs_config ]
```

Usage Notes

- If you do not specify the `--source_OH` option, but the ORACLE_HOME environment variable is set, the script uses the ORACLE_HOME environment variable as the source Oracle home. Otherwise, the script informs the user that no source ORACLE_HOME was specified and will exit.
- By default, this script will run rsync in the dry_run mode unless you specif the –no_dry_run. This allows the user to validate that the rsync command will be generated as expected and that the correct directories will be copied, before any potentially destructive work is done.
- You must specify the `--os_user` user name and `--target_host` host name. The `--os_user` is typically root.
- The script displays the parameters being used and the rsync command that it is about to execute.
- The script always prompts whether to proceed, to give you the opportunity to check that the generated rsync command is correct.
- You can copy an ORACLE_HOME with the EBS configuration subdirectories or without them, via the `–with_ebs_config` parameter. You only want the configuration subdirectories copied when you are keeping a paired standby database ORACLE_HOME in synch with its primary. You do not want to copy the configuration subdirectories when propagating the ORACLE_HOME to other servers during patching. By default, the script will exclude (will not copy) the following three subdirectories under the ORACLE_HOME, which hold the EBS configuration files. These directories will only be included / copied if the `--with_ebs_config` command line parameter is specified:
  - appsutil
  - admin/network
  - dbs

14.1.1 Copy a patched ORACLE_HOME

To use the script to copy a patched ORACLE_HOME:

1. Copy the script to a directory of your choice on any one Oracle RAC database nodes.
2. After you have completed the patching of an ORACLE_HOME, run the script as follows:

```
$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh    $ORACLE_HOME --
  target_host exaadb03 --os_user root
```

The output will look something like the following:

```
Using the following parameters:
```

```
source ORACLE_HOME    = /u01/app/oracle_ebs/product/12.1.0
target ORACLE_HOME    = /u01/app/oracle_ebs/product/12.1.0
target_hosts          = exaadb03
os_user               = root
dry_run               = true
with_ebs_config       = false

About to execute the following rsync command:

rsync -avzh --progress --exclude=appsutil --exclude=network/admin --
  exclude=dbs  --dry-run /u01/app/oracle_ebs/product/12.1.0/
  root@exaadb03:/u01/app/oracle_ebs/product/12.1.0/

OK to continue? [Y|y|N|n] :
```

To execute the copy, specify the `--no_dry_run` option (two hyphens).

14.1.2 Copy an ORACLE_HOME to its standby

To use the script to copy the EBS ORACLE_HOME in its entirety as part of establishing a standby database for disaster recovery, use the `-with_ebs_config` parameter of this script. Example:

```
$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host
exabdb01 --os_user root -with_ebs_config
```

The output will look something like the following:

```
Using the following parameters:

source ORACLE_HOME    = /u01/app/oracle_ebs/product/12.1.0
target ORACLE_HOME    = /u01/app/oracle_ebs/product/12.1.0
target_hosts          = exabdb01
os_user               = root
dry_run               = true
with_ebs_config       = true

About to execute the following rsync command:

rsync -avzh --progress --dry-run /u01/app/oracle_ebs/product/12.1.0/
  root@exaadb03:/u01/app/oracle_ebs/product/12.1.0/

OK to continue? [Y|y|N|n] :
```

Notice that the generated `rsync` command will not exclude the configuration subdirectories.

Run the above command with the `-no_dry_run` option to actually copy the ORACLE_HOME.

14.2 The rsync_EBS_OH.sh script

Here is the script itself:

```
#!/bin/sh
#
# Use rsync to synchronize the database ORACLE_HOME to other nodes in the
  cluster or to ORACLE_HOMEs at a DR site.
```

```
# Use this script after patching one ORACLE_HOME or for setting up an
  ORACLE_HOME at a DR site.
#
# You MUST run this script as root and the target user must also be root.
  owner:group and permisions
# will be preserved.
#
# Since this should be run after patching one oracle home, do not stasrt up
  the database until after rsync
# is complete.
# MAKE SURE THAT THE OTHER DATABASE INSTANCES ARE SHUTDOWN INCLUDING ANY
  LISTENERS RUNNING OUT OF THE SPECIFIC
# ORACLE_HOME
#    -  You do not need to shutdown the Grid/CRs or ASM.
#
# The ORACLE_HOME directory structure should be the same at both the primar
  and DR site.
# For RAC, all ORACLE_HOMES need to be synched up.
# Exclude specific directories:
#    - $ORACLE_HOME/appsutil
#    - $ORACLE_HOME/network/admin
#    - $ORACLE_HOME/dbs
#
#  To override excludng the above dirfectories, use the --with_ebs_config
  command line option.
#
# To preserve the output of rsync, the following example to run this script
  can be executed:
#
#     ./rsync_EBS_OH.sh | tee -a rsync.out
#
#

# Parse the command line arguements.
TEMP=`getopt -o h --long
  help,source_oh:,target_oh:,no_dry_run,with_ebs_config,target_host:,os_user
  : -n 'test.sh' -- "$@"`

if [ $? != 0 ] ; then echo "Invalid command line option. Use the --help for
  usage help.  Exiting..." >&2 ; exit 1 ; fi

# Note the quotes around `$TEMP': they are essential!
eval set -- "$TEMP"

VERBOSE=false
DEBUG=false
DRY_RUN=true
EBS_CONFIG=false
SOURCE_OH=
TARGET_OH=
TARGET_HOST=
OS_USER=
RSYNC_COMMAND=

tmp_str=" "

print_usage()
{
```

```
echo "Usage: "
echo "  rsync_EBS_OH_after_patching.sh [ -h | --help ] --source_oh <path to
  source ORACLE_HOME> "
echo "  [ --target_oh  <path to target ORACLE_HOME> ] --target_host <host>"
echo "  --os_user <username> [ --no_dry_run ] [ --with_ebs_config ]"
echo

exit 0

}

if [[ $# -lt 2 ]]; then
     print_usage
fi

while true; do
  case "$1" in
    -h | --help ) print_usage; shift ;;
    --source_oh ) SOURCE_OH="$2"; shift 2 ;;
    --target_oh ) TARGET_OH="$2"; shift 2;;
    --no_dry_run ) DRY_RUN=false; shift ;;
    --with_ebs_config ) EBS_CONFIG=false; shift ;;
    --os_user ) OS_USER="$2"; shift 2 ;;
    --target_host ) TARGET_HOST="$2"; shift 2 ;;
    -- ) shift; break ;;
    * ) break ;;
  Esac
done

# set up the rsync command to run.

if [[  ${SOURCE_OH} = "" ]]; then
     if [[ -z ${ORACLE_HOME} ]]; then
          echo "source_oh not specifid and ORACLE_HOME not defined.  Specify
  source_oh on command line or set ORACLE_HOME."
          echo "Exiting..."
          exit 1
     else
          SOURCE_OH=${ORACLE_HOME}
     fi
else
    ORACLE_HOME=${SOURCE_OH}
fi

if [[  ${TARGET_OH} = "" ]]; then
     if [[ -z ${ORACLE_HOME} ]]; then
          echo "source_oh not specifid and ORACLE_HOME not defined.  Specify
  source_oh on command line or set ORACLE_HOME."
          echo "Exiting..."
          exit 1
     else
         TARGET_OH=${ORACLE_HOME}
    fi
fi

if [[ ${OS_USER} = "" ]]; then
     echo "os_user must be specified.  Please use --help for usage
  information.  "
     echo "Exiting...."
```

```
     exit 1
fi

if [[ ${TARGET_HOST} = "" ]]; then
     echo "target_host must be specified.  Please use --help for usage
  information.  "
     echo "Exiting...."
     exit 1
fi

echo
echo "Using the following parameters: "
echo
echo "source ORACLE_HOME    = $SOURCE_OH"
echo "target ORACLE_HOME    = $TARGET_OH"
echo "target_hosts          = $TARGET_HOST"
echo "os_user               = $OS_USER"
echo "dry_run               = $DRY_RUN"
echo "with_ebs_config       = $EBS_CONFIG"
echo

if [[ ${DRY_RUN} = "true" ]]; then
     tmp_str=" --dry-run"
fi


if [[ ${EBS_CONFIG} = "false" ]]; then
  printf -v RSYNC_COMMAND "rsync -avzh --progress --exclude=appsutil --
  exclude=network/admin --exclude=dbs %s %s/  %s@%s:%s/" "${tmp_str}"
  ${SOURCE_OH} ${OS_USER} ${TARGET_HOST} ${TARGET_OH}
else
         printf -v RSYNC_COMMAND "rsync -avzh --progress %s %s/  %s@%s:%s/"
  "${tmp_str}" ${SOURCE_OH} ${OS_USER} ${TARGET_HOST} ${TARGET_OH}
fi


echo "About to execute the following rsync command: "
echo
echo "${RSYNC_COMMAND} "
echo

while [[ -z ${proceed} ]];
do
     read -p "OK to continue? [Y|y|N|n] :" proceed

     [[ ${proceed} = 'Y' || ${proceed} = 'y' || ${proceed} = 'N' ||
  ${proceed} = 'n' ]] && break
     proceed=""
done
[[ ${proceed} = 'N' || ${proceed} = 'n' ]] && exit 0

# Run the rsync command.

time ${RSYNC_COMMAND}

exit 0
```

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

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

Integrated Cloud Applications & Platform Services

E-Business Suite Release 12.2 Maximum Availability Architecture
Maximum Availability Architecture
January 2018
Authors: Marilyn Pratt, Darryl Presley
Contributing Authors: Andy Tremayne