

Oracle E-Business Suite Release 12.2 Maximum Availability Architecture

With a case study on Oracle Private Cloud Appliance and Exadata
Database Machine

May 2021, Version 2.2
Copyright © 2021, Oracle and/or its affiliates
Public

1 Executive Overview	4
2 Introduction to Engineered Systems	5
3 Oracle E-Business Suite Release 12.2 Maximum Availability Architecture	6
3.1 Oracle Database Maximum Availability Architecture	7
3.2 Oracle E-Business Suite Release 12.2 High Availability	11
4 Oracle E-Business Suite Database Configuration Best Practices	15
4.1 OS Role Separation	15
4.2 Database Initialization Parameters	16
4.3 Configure Linux HugePages	16
4.4 Take Backups of the Database Oracle Homes	17
4.5 Handle Database Password Expiration	17
4.6 Exadata Fast Node Death Detection	17
4.7 Reduce Timeout on Oracle RAC Node Failure	18
4.8 Implement Flashback Database	18
4.9 Exadata Smart Flash Logging	18
4.10 Exadata Smart Flash Cache	18
4.11 Memory Management (Exadata Only)	Error! Bookmark not defined.
4.12 Oracle Advanced Compression	19
4.13 Log Writer Tuning (Exadata Only)	19
4.14 Fixed Object Statistics	20
4.15 Exadata I/O Resource Manager (IORM)	20
4.16 Oracle Engineered Systems Best Practices and Health Check	20
5 Oracle E-Business Suite Release 12.2 Application Tier Configuration Best Practices	21
5.1 Deploy Oracle E-Business Suite application tiers using Shared File Systems	21
5.2 Create an Oracle E-Business Suite Central Inventory on the Application Tier	21
5.3 Move the Apache Lock File to Local Storage	22
5.4 Take Regular Backups of the Release 12.2 Application File System	22
6 Summary of Best Practices	22
6.1 Best Practices – Oracle E-Business Suite Database Maximum Availability	22
6.2 Best Practices – Oracle E-Business Suite Application High Availability	23
6.3 Best Practices – Oracle E-Business Suite Maximum Availability	23
7 Oracle E-Business Suite MAA Case Study Architecture	24
7.1 Method	24
7.2 Case Study Topology	25
7.3 Software Versions	26

7.4 Oracle E-Business Suite Application Tier – High Availability Architecture	26
7.5 Oracle E-Business Suite Database Tier – MAA	28
7.6 Delegation of Roles and Administration	29
7.7 Oracle E-Business Suite Application and Web Tier Setup	30
7.8 Logical Host Names	33
8 Case Study Roadmap	33
8.1 Prepare the Network	33
8.2 Prepare the Primary	35
8.3 Prepare the Standby	52
8.4 Disaster Recovery Site Test using Snapshots	69
8.5 Test Switchover	74
8.6 Test Failover	78
8.7 Site Guard	83
8.8 Patching the Database in an Engineered Systems Environment	84
9 References	95
10 Appendix A: Verify Oracle E-Business Suite Required Packages	96
11 Appendix B: Managing SCAN for Primary and DR Sites	97
11.1 The custom.txt File	99
11.2 The ebs_scan_db.sh Script	99
11.3 The ebs_scan_app.sh Script	101
12 Appendix C: Set Web Entry Point	103
13 Appendix D: Add Managed Services	105
13.1 Review the Configuration on the Admin Server	106
13.2 Add and Start Managed Services	106
13.3 Finalize the Configuration	108
13.4 Bounce the HTTP Server	108
14 Appendix E: Perform rsync on Oracle Homes	109
14.1 Using the rsync_EBS_OH.sh Script	109
14.2 The rsync_EBS_OH.sh script	111
15 Appendix F: Scripts used by Site Guard	115

1 Executive Overview

Oracle Maximum Availability Architecture (MAA) is Oracle's best practice blueprint based on proven Oracle high availability technologies and recommendations. The goal of MAA is to achieve the optimal high availability architecture at the lowest cost and complexity. Papers are published on the Oracle Technology Network (OTN) - <http://www.oracle.com/goto/maa>.

In this paper, we apply the principles of Oracle's Maximum Availability Architecture (MAA) to the Oracle E-Business Suite (EBS) Release 12.2 (R12.2) infrastructure. Here, we describe:

- EBS R12.2 MAA architecture along with installation, configuration, and operational best practices.
- How EBS R12.2 MAA was implemented on Oracle Private Cloud Appliance (PCA) and Exadata.
- Our tests to validate our best practices and measure downtime in various outage scenarios.

When EBS R12.2 was configured with our MAA best practices on PCA and Exadata, we demonstrated how to minimize user impact during typical failure scenarios. In the event of a total site failure the disaster recovery site could be brought online in as few as 12 minutes.

This paper provides a case study of Oracle E-Business Suite Release 12.2 on Oracle Private Cloud Appliance and Exadata machines, showing the steps for:

- Reconfiguring an existing Oracle E-Business Suite R12.2 implementation to use logical host names
- Establishing the disaster recovery (DR) site
- Non-destructive testing of Oracle E-Business Suite R12.2 at the DR site
- Switching and failing over to the DR site
- Simplified patching of Oracle E-Business Suite's Oracle Database homes

This paper was developed in the Oracle Solutions Center (OSC). The OSC is a centralized global organization with twelve state-of-the-art locations around the world where customers architect, customize, and test solutions with Oracle Cloud, Cloud@Customer, and On Premises systems in a secure, scalable, and interoperable environment for all deployment models. To meet evolving business and technology challenges quickly, OSC provides a wide range of accelerated services that highlight Oracle products and complementary Partner products as needed. Key services include Architecture Reviews, TCO/ROI Analyses, Proofs of Concepts, Customized Demonstrations and Workshops to support a dynamic community of VAD/VAR, ISVs, and System Integrators.

If you are considering High Availability, Disaster Recovery, and Data Center Consolidation using Oracle's Maximum Availability Architecture (MAA), the OSC is the place to test, validate and perfect your organization's Disaster Recovery needs. At the OSC, meet the experts and leverage Oracle's best practices with any combination of technologies, hosted by Oracle subject matter experts and Solution Architects. Contact your local account manager to engage the Oracle Solutions Center and benefit from its range of capabilities and competencies to effortlessly solve your business technology challenges.

2 Introduction to Engineered Systems

Oracle's Engineered Systems combine best-of-breed hardware and software components with game-changing technical innovations. Designed, engineered, and tested to work best together, Oracle's Engineered Systems can power the cloud or streamline data center operations to make traditional deployments even more efficient. The components of Oracle's Engineered Systems are preassembled for 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 is integrated across every layer of the technology stack—a simplicity that 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.

2.1.1 Oracle Private Cloud Appliance

Oracle Private Cloud Appliance and Oracle Private Cloud at Customer are on-premise cloud-native converged infrastructures that allow customers to efficiently consolidate business-critical middleware and application workloads. Oracle Private Cloud Appliance is cost effective, easy to manage, and delivers better performance than disparate build-your-own solutions. Oracle Private Cloud Appliance and Oracle Exadata together provide a powerful, single-vendor application and database platform for today's data driven enterprise.

Oracle Private Cloud Appliance runs enterprise workloads alongside cloud-native applications to support a variety of application requirements. Its built-in secure multi tenancy, zero downtime upgradability, capacity on demand and single pane of glass management make it the ideal infrastructure for rapid deployment of mission critical workloads. Oracle Private Cloud Appliance and Oracle Cloud Infrastructure together provide customers with a complete solution to securely maintain workloads on both private and public clouds.

2.1.2 Oracle Exadata Database Machine

Oracle's Exadata Database Machine is Oracle's database platform delivering extreme performance for database applications including Online Transaction Processing, Data Warehousing, Reporting, Batch Processing, or Consolidation of mixed workload databases. Exadata is a pre-configured, pre-tuned, and pre-tested integrated system of servers, networking, and storage all optimized around the Oracle Database. Because Exadata is an integrated system, it offers superior price-performance, availability, and supportability. Exadata frees users from the need to build, test, and maintain systems and allows them to focus on higher-value business problems.

Key highlights of the Exadata architecture:

- Exadata uses a scale-out architecture for database servers and storage. This architecture maintains an optimal storage hierarchy from memory to flash to disk.
- Smart Scan query offload has been added to the storage cells to offload database processing
- Exadata implements Smart Flash Cache as part of the storage hierarchy. Exadata software determines how and when to use the Flash storage for reads and write as well as how best to incorporate Flash into the database as part of a coordinated data caching strategy.
- A high-bandwidth low-latency InfiniBand network running specialized database networking protocols connects all the components inside an Exadata Database Machine. Newer generations of Exadata, starting with X8M, now implement a much higher bandwidth and extremely lower latency network called RDMA over Converged Ethernet or RoCE. Coupled with Persistent Memory (PMEM) on the storage cells, direct memory access from a compute node using RDMA to a given storage server is as low as 19 microseconds.
- In addition to a high performance architecture and design, Exadata offers the industry's best data compression to provide a dramatic reduction in storage needs.

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 Maximum Availability Architecture, including a secondary site to provide business continuity in the event of a primary site outage.

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.

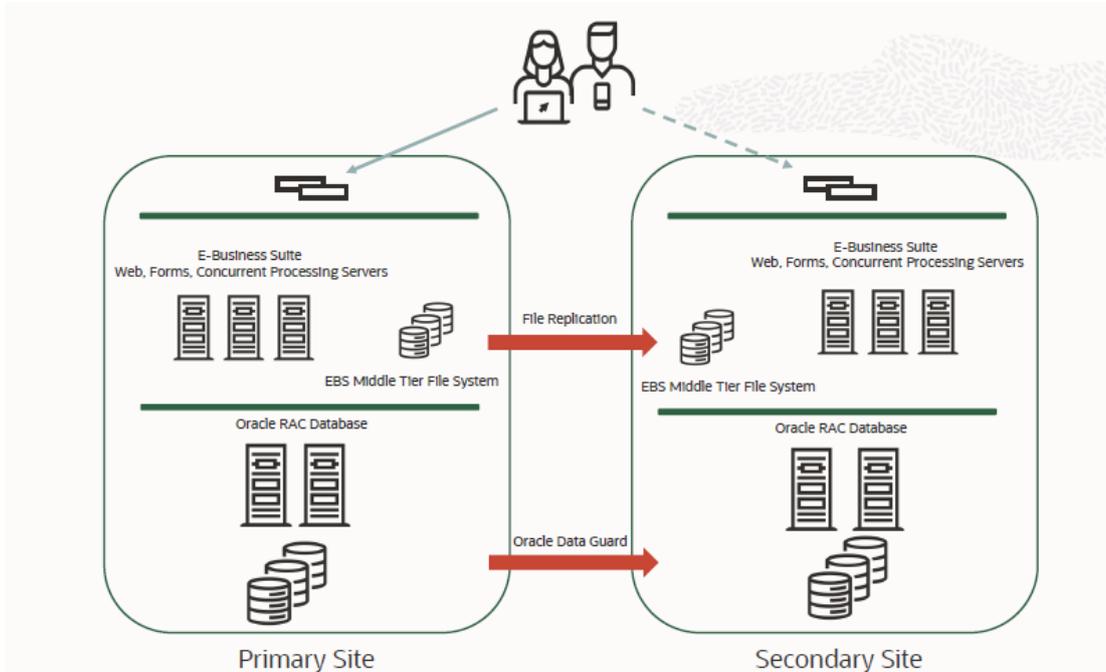


Figure 1 Oracle E-Business Suite Release 12.2 full MAA implementation

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 tier servers 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

Oracle Database MAA for E-Business Suite

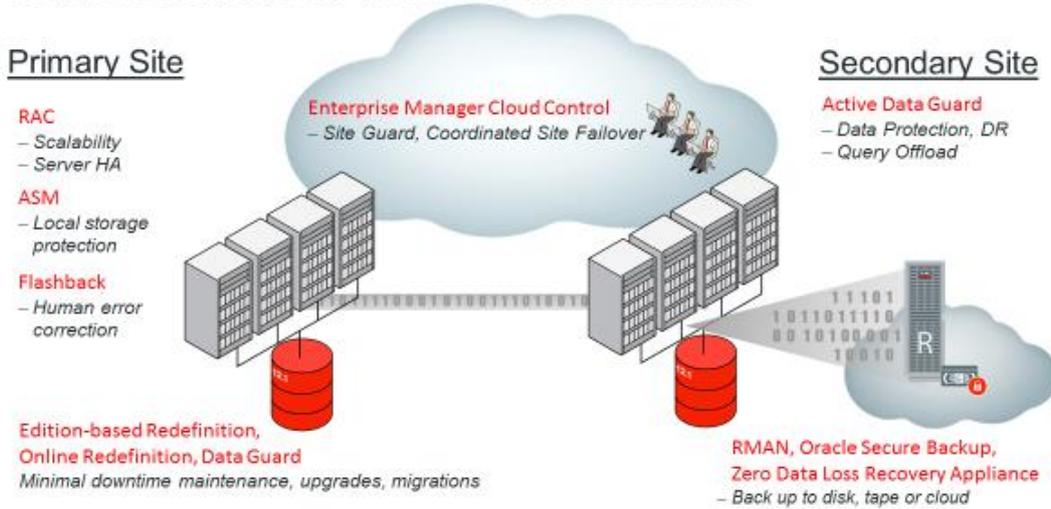


Figure 2 Oracle Database Maximum Availability Architecture

To achieve maximum Oracle E-Business Suite database 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
- 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) enables the Oracle Database to serve 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](#).

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 database 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 Multitenant

Oracle Multitenant was introduced in Oracle RDBMS 12c. Oracle Multitenant enables the segmentation of data into separate self-contained pluggable databases. A container database and the one or more pluggable database hosted by that container all share the same infrastructure resources, such as background processes, memory, CPU, IO, and networking. We can move pluggable databases from one container database to another, providing database virtualization. The following diagram illustrates a simplified form of a single container database (CDB) with three pluggable databases (PDBs).

Please note that E-Business Suite must be deployed as single PDBs into 19c or higher container databases. At present, E-Business Suite does not support its database running in a container with more than one PDB.

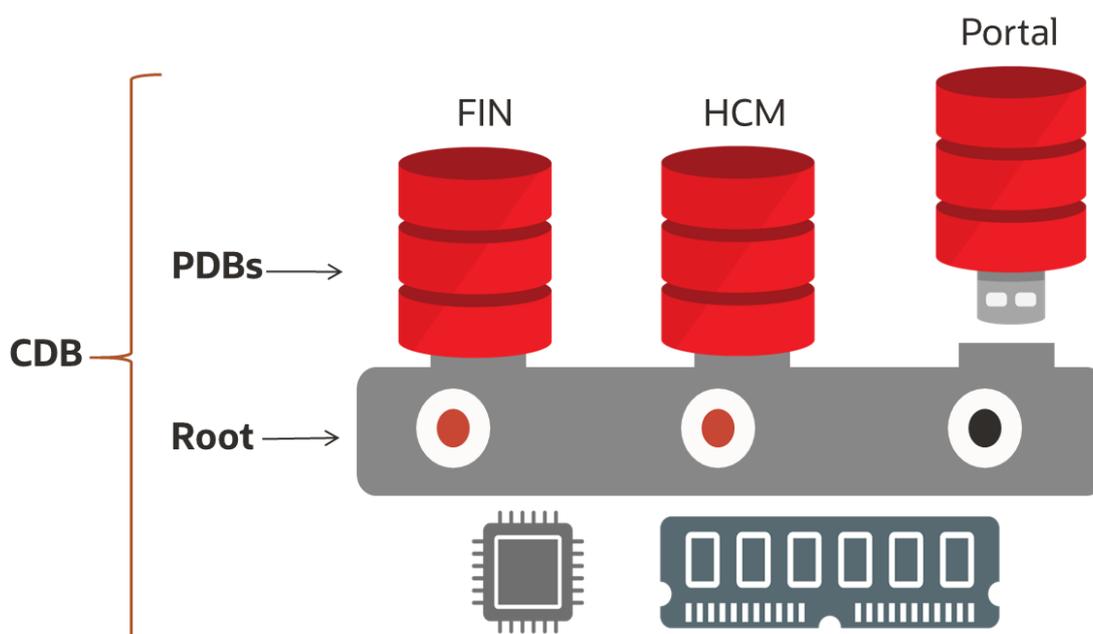


Figure 3: Container database with three pluggable databases

Figure 3 above illustrates the following:

- A single container database (CDB)
- Three pluggable databases (PDB) named: FIN, HCM and Portal (note: not supported by EBS)
- The FIN and HCM PDBs are plugged into the CDB root also known as CDB\$ROOT.
- The Portal PDB is unplugged from the CDB root.
- Pluggable databases can be unplugged from one CDB and plugged into another. This provides data portability.

Other features of Oracle Multitenant are:

- Portability via plug and unplug – the application runs unchanged.
- PDB upgrade via plug and unplug
- The database code is patched once for all PDBs in the CDB.
- Common operations such as backup, restore, and flashback can be done at the CDB or the PDB level.
- PDBs can be opened or closed on a per RAC instance basis
- Multiple PDBs hosted on a single CDB allows greater consolidation while keeping data securely segregated. Note that at this time Oracle E-Business Suite does not support multiple PDBs in a single CDB..

3.1.3 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 applies the redo received from the primary database 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 but critical 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), creating an ideal testbed for short-term uses such as testing application upgrades. 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](#).

3.1.4 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 configuration. 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 [Oracle Data Guard Broker](#).

3.1.5 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 your expected LOB maintenance activity, to ensure the solution will perform adequately in your production environment.

Regardless of whether you enable Flashback Database in production, 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 \(Doc ID 565535.1\)](#)”.

3.1.6 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
- Automatic resilvering of mirrors to recover from physical hardware issues
- Proactive detection and repair of physically corrupted blocks, and of blocks that have certain logical corruptions
- Automatic rebalancing of 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](#).

3.1.7 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](#).

3.1.8 Online Upgrades Using Edition Based Redefinition

Edition-Based Redefinition (EBR) was introduced with Oracle Database 11g, 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 cross edition 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 an EBS 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 as of AD/TKX Delta 9)
- Fusion MiddleWare Administration Server configuration
- Each of these is expanded in the following sections. Figure 3 illustrates this architecture.

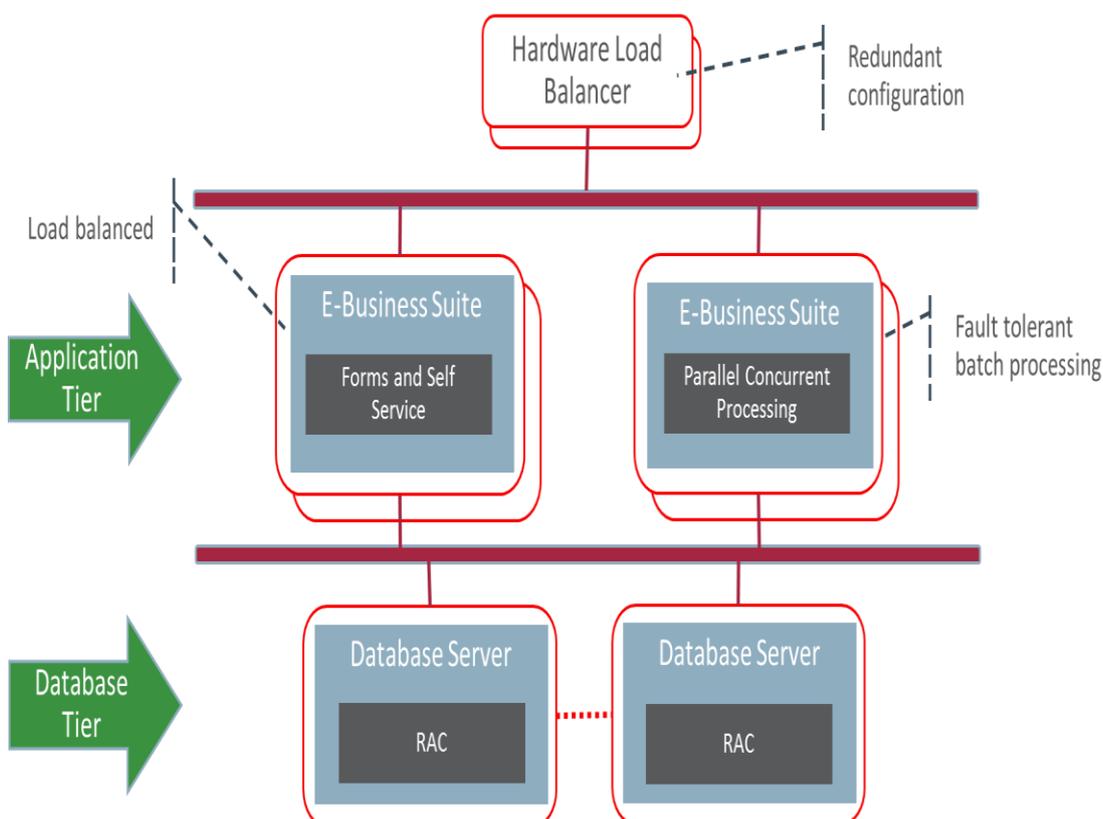


Figure 4 Oracle E-Business Suite High Availability Architecture

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 capabilities 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 the middle tier cluster.
- Database load balancing and/or load direction – the ability to spread or direct the load across the database instances. You can configure all or most of your workload to run on any application server and configure each application server to run against any database instance, or you can configure certain concurrent programs or groups of programs to run on a particular application server, and configure that application server to run against a specific database instance. The amount of benefit experienced by controlling which database instance specific programs run against depends on the programs' data access and update profile. While there are clearly some trade-offs such as additional planning and configuration, grouping certain concurrent manager 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 5.

The screenshot shows the 'Concurrent Managers' configuration form. The 'Manager' field is set to 'Workflow Mailer Service' and is checked as 'Enabled'. The 'Short Name' is 'WFMLRSVC', 'Application Name' is 'Application Object Library', and 'Description' is 'Container for Workflow Mailer Service Components'. The 'Type' is a dropdown menu. The 'Data Group' and 'Consumer Group' fields are empty. The 'Cache Size' checkbox is unchecked. The 'Parallel Concurrent Processing Details' section has a table with columns 'Node', 'System Queue', and 'Platform', and rows for 'Primary' and 'Secondary'. The 'Program Library' section has 'Name' set to 'FNDSVC' and 'Application' set to 'Application Object Library'. At the bottom, there are two buttons: 'Specialization Rules' and 'Wgrk Shifts'.

Figure 5 Oracle E-Business Suite Form for the Workflow Mailer Service

3.2.2 Multiple Load-Balanced Application Tier Services

Oracle E-Business Suite Release 12.2 uses Fusion Middleware (FMW) components, deployed as a WebLogic Server (WLS) cluster. User requests are safely routed to the appropriate FMW component in that cluster by Oracle HTTP Server (OHS).

The WLS cluster will host three application tier components, each in its own managed service group:

- OACORE services (“Self Service”)
- Forms services
- OAFM for other web services

Each managed service group can be distributed across two or more physical or VM servers, allowing for both high availability and scalability. Managed services can be added or removed from the cluster as needed.

We recommend that more than one of each type of managed service be deployed, on separate physical servers, so 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’ database sessions 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 be 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 Fusion Middleware Administration Server Configuration

The first application tier server 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 \(Doc ID 1986122.1\)”](#).

3.2.4 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. For EBS patches, this is all managed by the Oracle E-Business Suite online patching tool `adop`. Fusion Middleware patches are similarly applied to the PATCH file system, using the Fusion Middleware patch tooling.

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 before stopping online access to the application. This can significantly reduce the time it takes to complete the cutover process.

3.2.5 Logical Host Names

For disaster recovery, two symmetrical sites (with the same number and type of servers) are configured, allowing the EBS configuration to fully function on either the primary or the standby site. Starting with Oracle E-Business Suite Release 12.2, AD/TXK Delta 9, user access to the middle tier servers is provided via “logical host names”, which are aliases for the physical host names. Logical host names are site-neutral server names that are the same on both the primary and the standby, and allow the core Oracle E-Business Suite configuration to stay the same when switching between the two sites.

These logical host names are different from the physical host names. They are not registered in DNS – they are used only by Oracle E-Business Suite. Other tools continue to use the physical host names or host names assigned by DNS.

3.3 Site Outage Test and Results with Site Guard

Site Guard is an Oracle Enterprise Manager plug-in that operates under the Oracle Enterprise Manager framework. Site Guard orchestrates and fully automates switchover or failover of the entire E-Business Suite stack. Site Guard can be deployed into an existing Enterprise Management installation. The case study described in this paper implemented Site Guard and performed site role change testing. We show the timings for performing a site failover below.

During our testing, all servers on the primary site were stopped abruptly, creating a site outage. All application user sessions failed on the site outage. An administrator then used Site Guard to initiate the process for automated execution of the site failover procedure, which restored services at the standby site and allowed application users to log in and resume work against that hardware.

The failover procedure performed by Site Guard and the timing for each step are shown in the following table:

OPERATION	TYPICAL ELAPSED TIME
Failover Storage Failover	1m 10s

Mount EBS Application Shared File System on all App Tier nodes	18s
Database Failover	1m 56s
Run EBS Set Web Entry point on App Servers Scripts (Post-Scripts, run in parallel)	15s
Run EBS Autoconfig (Post-Scripts, run in parallel)	3m 45s
Start EBS WLS Admin Server (Post-Scripts, run as singleton)	2m 25s
Start all EBS Application Services (Post-Scripts, run in parallel)	5m 25s
TOTAL	12m 48s

The Weblogic Admin Server must be started first as a singleton before EBS application servers can be started unless Management Service Independent (MSI) mode is used. In our testing, we did not modify Site Guard scripts to start application services in MSI mode.

The above total timing may vary based on the implementation of EBS and site specific factors. To get to a reduced failover timing as shown above, you must implement the database and application best practices outlined in this paper. Refer to section 8.7 below for further details and other reference materials for Site Guard.

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 general high availability and performance best practices for the entire system.

4.1 OS Role Separation

Role separation allows for best-practice 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), best-practice 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. This configuration creates the OS users and groups, 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 must 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 and is recommended. To enable ASMM, 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 must 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 to ensure the instance will only start 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 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

Oracle Database Release 12c expires database user passwords after 180 days by default. 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 the 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 so their actions can be properly audited. Do not allow anyone to log in as the user APPS.

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

Exadata Fast Node Death Detection (FNDD) was introduced with Exadata Storage 12.1.2.1 and Oracle Grid Infrastructure 12.1.0.2 Bundle Patch 7. FNDD takes advantage of the two active Infiniband or (starting with the X8M series) RoCE ports for all compute and storage servers. If neither port responds to status queries for more than one second, FNDD will force the node out of the cluster. Once forced out of the cluster, the node will attempt to restart and rejoin the cluster.

Fast Node Death Detection is enabled by default where available on Exadata. No manual configuration is required.

4.7 Reduce Timeout on Oracle RAC Node Failure

In the rare case where FNDD is unable to gracefully handle a node outage, the parameter `css_miscount` is used to evict the unhealthy instance.

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

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

Later releases of Exadata software implement `css_miscount` to 30 seconds by default.

4.8 Implement Flashback Database

As noted in the earlier section [Oracle Flashback Database](#), Flashback Database can be used for the following scenarios:

- Use a physical standby database to validate changes scheduled for the production database, then resume recovery at the standby
- After a failover event, quickly reconstitute the old primary database as standby for the database now serving as production, without requiring a restore from backup.

There are two things to note for EBS databases:

- 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 those workloads with Flashback Database enabled, to ensure performance will be acceptable at peak load in your production environment. 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, then configure your storage to accommodate the requirement.

We implemented Flashback Database on both primary and standby in our environment. You may decide to only enable it on the standby, depending on the results of your performance tests. The steps for implementing Flashback Database 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 either the durable physical disk or, if Extreme Flash storage is in place, separate NVMe flash devices 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. It 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. Note that write-back flash cache is enabled by default for Exadata Extreme Flash storage cells as well as any Exadata machine having flash cache with versions 19.x and greater of the Exadata software.

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 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 testing 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. In comparison, the `MEDIUM` setting 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 greater constraint during backups.

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.

4.12 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 Exadata best practice has shown performance improvements in a number of Oracle E-Business Suite batch programs.

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.

4.13 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., `v$sql`, `v$sql_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 \[Doc ID 798257.1\]”](#).

4.14 Exadata I/O Resource Manager (IORM)

For Oracle E-Business Suite workloads, optimal performance is observed when the IORM Objective plan setting is “auto”. 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=auto
```

See MOS Doc [“Configuring Exadata I/O Resource Manager for Common Scenarios \(Doc ID 1363188.1\)”](#) for more information.

4.15 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 Private Cloud Appliance and ZFS storage appliance, many are generic in nature.

5.1 Deploy Oracle E-Business Suite application tiers using Shared File Systems

While EBS R12.2 does support installing the application tiers on a distributed file system, Oracle recommends installing EBS R12.2 on a shared file system supported by a robust filer such as ZFS that is provided by the Private Cloud Appliance. A shared file system deployment provides the following benefits:

- Reduce space requirements since all EBS application components are shared. Some additional amount of storage is needed for each app tier node's instance directory.
- Reduce time for patching as both RUN and PATCH file systems are shared across all application tier nodes. On a distributed (non-shared) file system, each application node would have its own RUN and PATCH file system and would need to be patch individually.
- Simpler replication of the shared file system since only the shared file system and any other custom file system would need to be replicated to a DR site. On a distributed (non-shared) deployment, each file system would need to be replicated individually.

5.2 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 for the application tier. The `oralnst.loc` file and the `oralInventory` directory contain all the application tier Oracle homes for a specific Oracle E-Business Suite Release 12.2 installation, making these directories dedicated to and contained fully within a single Oracle E-Business Suite installation. Creating a central inventory has the following benefits:

- 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_ews_central_inventory` is set to true.

- The context variable `s_invPtrLoc` specifies the directory location of the `oralnst.loc` file under `ORACLE_BASE`. For example, if `ORACLE_BASE` points to `/u06/app/ebs/visprd`, then set this context variable to: `/u02/app/ebs/visprd/oralInventory/oralnst.loc`
- Do NOT place the `oralnst.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? \(Doc ID 1588609.1\)](#).

5.3 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 \[Doc ID 560853.1\]](#)".

5.4 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, move one directory level above where the `fs1`, `fs2`, and `fs_ne` are located, then perform your backup (or zip). This will capture the `run`, `patch` and non-edition directories along with all of the Fusion Middleware homes.

6 Summary of Best Practices

This section concludes the first portion of this paper with a summary of the best practices presented above, 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 failover if you have determined performance in production will be acceptable for LOB maintenance. In either case, 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 “auto,” to ensure that the OLTP workload has good performance while allowing smart scans to be performed.
- Set `CSS_MISCOUNT` to 30 if it is not already set to that value. Newer Exadata software versions will set it to 30 by default.
- Enable write-back flash cache if it is not already enabled.
- 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 work can continue in the event of an application tier node failure. For Private Cloud Appliance, ensure the nodes (VMs) are provisioned with anti-affinity so no two VMs are running on the same physical compute node.
- 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` and do not set the mount option `actime=0`. If NFSv4 is used, do not specify the mount options `actime=0` or `noac`. These options 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, use affinity to pin concurrent workloads to specific database instances.

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](#)”.
- Use Oracle Active Data Guard for faster incremental backups and for automatic block repair.
- If performance of your production system while executing LOB maintenance will be acceptable, enable Oracle Flashback Database on both the standby and the primary database, so that the old primary database can be quickly re-instantiated in the event of site failover.
- 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.
- Develop and document operational procedures for maintaining your secondary site and for role transitions across sites.
- Conduct periodic role transitions including switchover and failover testing between the primary and DR sites to validate your processes and procedures.
- 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 as a temporary environment for testing changes and issues in production.

For Exadata and Private Cloud Appliance 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: RESTORE FROM SERVICE capabilities in Oracle Database 19c to instantiate 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, 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 and/or hostnames assigned through DNS. 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 PCA ZFS Storage Appliance 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 lost following a 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 “EBSCDB_PHX3XD” at the primary site and “EBSCDB_OSC5XL” 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 “EBSCDB_OSC5XL” than seeing your primary named “visprd_stdby” or a similar name. These names were chosen to indicate the name of the database and their location.

7.2 Case Study Topology

Oracle E-Business Suite Release 12.2.7 was deployed in a full MAA configuration on X8-2 Private Cloud Appliance and X7-2 Exadata Database Machines. Each site was configured such that it could assume either the primary or the 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 PCA 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 X7-2 Oracle Exadata Database Machine <ul style="list-style-type: none"> Two Compute nodes – exa14db01, exa14db02 44 cores (88 logical cores with SMT), 528GB RAM each <ul style="list-style-type: none"> Three X6-2 Exadata storage cells 44 cores, 11TB flash cache, high capacity disks each 	Quarter Rack X7-2 Oracle Exadata Database Machine <ul style="list-style-type: none"> Two Compute nodes – exa14db03, exa14db04 44 cores (88 SMT), 528GB RAM each <ul style="list-style-type: none"> Three Exadata storage cells 44 cores, 11TB flash cache, high capacity disks each
Application and Web Tier	X8-2 Oracle Private Cloud Appliance <ul style="list-style-type: none"> Two PCA VMs (domUs) – pca3vm51, pca3vm52 16 cores, 32 GB RAM each Hosting EBS Application and Concurrent Manager 	X5-2 Oracle Private Cloud Appliance <ul style="list-style-type: none"> Two PCA VMs (domU) – pca1vm51, pca1vm52 16 cores, 32 GB RAM each Hosting EBS Application and Concurrent Manager
Application Tier File System Storage	Oracle ZFS Storage Appliance for application and web server storage -- exalgnsn-fe	Oracle ZFS Storage Appliance for application and web server storage – exalgnsn-fe

Table 3: Primary and Secondary Site Configuration

Figure 6 below illustrates a complete MAA implementation for Oracle E-Business Suite Release 12.2 using a symmetrical Exadata and PCA configuration for both primary and disaster recovery sites.

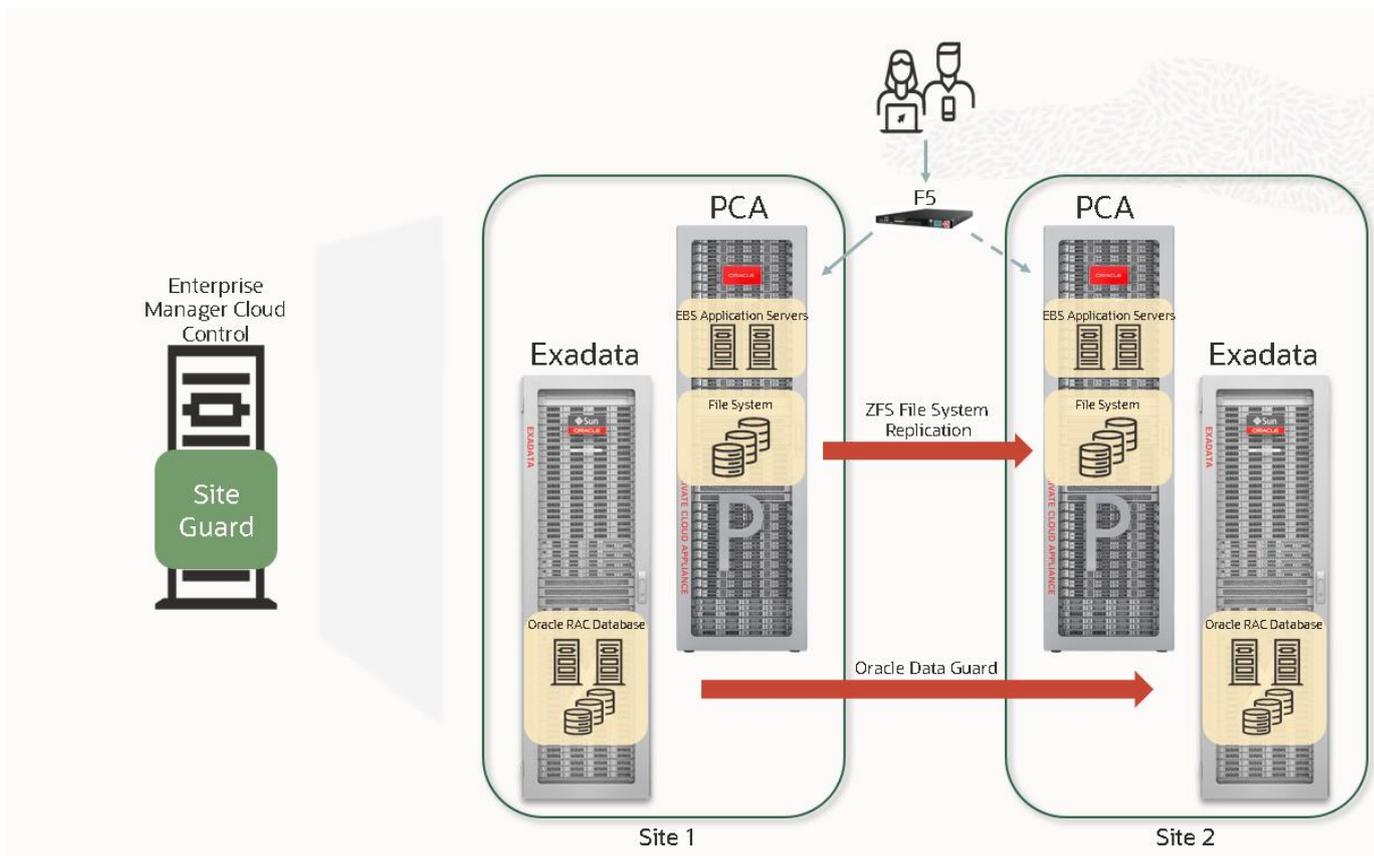


Figure 6 The complete MAA implementation of Exadata and PCA systems for the primary and disaster recovery sites

Figure 6 shows that both sites have an Exadata quarter rack for the Oracle E-Business Suite database tier and a PCA 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. Oracle Site Guard shown above was used to automate the full stack switchover and failover process.

7.3 Software Versions

Table 5 shows the software and versions used for the case study.

Software Versions

Software	Version
Private Cloud Appliance software	2.4.3
Oracle E-Business Suite	12.2.7, AD/TXK Delta 12
WebLogic Server	10.3.6.0
Database	19c – 19.8.0.0.0
Exadata Software	20.1.1.0.0.200808
Oracle Clusterware Grid Infrastructure	19c – 19.8.0.0.0
F5 Big-IP Local Traffic Manager	15.1.0 Build 0.0.31

Table 4: 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 previous sections, with the following configuration:

- An f5 BigIP load balancer at both primary and secondary sites, using Local Traffic Manager (LTM)
- Two middle tier application servers on the Private Cloud Appliance 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.
- A Private Cloud Appliance at the secondary site, to duplicate the application tier from the primary site.
- 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 7 shows the Application Tier and ZFS filer topology,

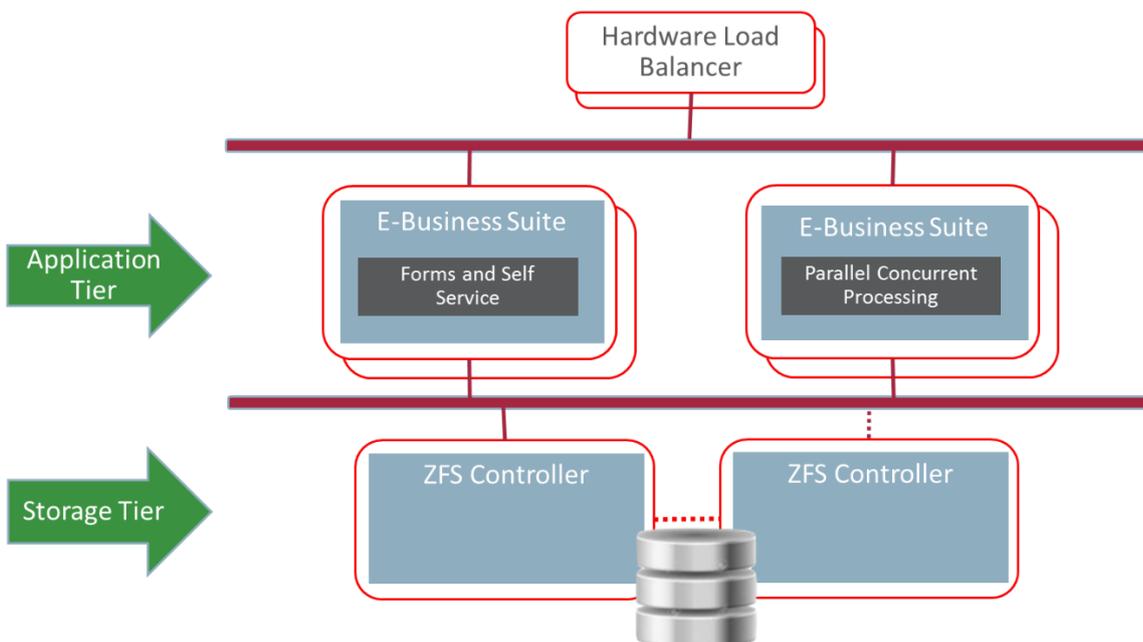


Figure 7 Two application tiers and ZFS filer

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:

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

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 Exadata 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 12.2 Maximum Availability Architecture ([PDF](#) , [HTML](#))

The standard Exadata configuration was deployed as follows:

- ASM disk groups (DATA1 and RECO1) 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 RESTORE FROM SERVICE 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. Note that 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: the database tier on Exadata, and the application and web servers on Private Cloud Appliance. 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, 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
grid	oinstall, dba	Clusterware and ASM administrator(Grid)
oracle	oinstall, dba	Oracle E-Business Suite database administrator

Table 5: Oracle Database Server OS Users

The `oinstall` and `dba` OS groups are common between the `grid` OS user account and that of `oracle`. 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 Private Cloud Appliance

On the Private Cloud Appliance, 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 6: Recommended Application Server OS Users

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 NFSv3. The Fusion Middleware, `RUN`, `PATCH`, and `Non-Edition` subdirectories all reside on the below ZFS share.

The ZFS shared file system for EBS is described in the following table:

ZFS PROJECT NAME	ZFS SHARE NAME	EXPORT MOUNT POINT
MAA_EBS	ebs	/export/ebs

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: `VISPRD_OACORE_ONLINE` for self-service web connections and `VISPRD_FORMS_ONLINE` for Oracle Forms connections

We also configured parallel concurrent processing (PCP) on each application tier server. All PCP concurrent manager connections made from a given application server were load balanced across both RAC instances using the database service: `VISPRD_PCP_BATCH`.

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	<code>s_apps_jdbc_connect_descriptor</code>	<code>VISPRD_OACORE_ONLINE</code>	EBSCDB1, EBSCDB2
Forms	<code>s_tools_twotask</code>	<code>VISPRD_FORMS_ONLINE</code>	EBSCDB1, EBSCDB2
PCP App Node 1	<code>s_cp_twotask</code>	<code>VISPRD_PCP_BATCH</code>	EBSCDB1, EBSCDB2
PCP App Node 2	<code>s_cp_twotask</code>	<code>VISPRD_PCP_BATCH</code>	EBSCDB1, EBSCDB2

Table 7: Database Service Mapping

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

1. Create the database service.

```
srvctl add service -db EBSCDB_phx3xd -pdb VISPRD -service VISPRD_OACORE_ONLINE -preferred "EBSCDB1,EBSCDB2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC -failovertime SELECT -failoverretry 10 -failoverdelay 3
```

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

```
<jdbc_url  
oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION_LIST=(LOAD_BALANCE=off)(FAILOVER=on)(DESCRIPTION=(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=exa14-scan1.us.osc.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_OACORE_ONLINE)))(DESCRIPTION=(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=exa14-scan2.us.osc.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_OACORE_ONLINE))))  
</jdbc_url>
```

```
<patch_jdbc_url  
oa_var="s_apps_jdbc_patch_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION_LIST=(LOAD_BALANCE=off)(FAILOVER=on)(DESCRIPTION=(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=exa14-scan1.us.osc.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_ebs_patch)(INSTANCE_NAME=VISPRD1)))(DESCRIPTION=(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=exa14-scan2.us.osc.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_ebs_patch)(INSTANCE_NAME=VISPRD1))))</patch_jdbc_url>
```

The JDBC URLs above have multi-address TNS connect strings. This allows the oacore managed servers to connect where the service is available. This multi-address TNS connect string is set in the s_apps_jdbc_connect_descriptor and s_apps_jdbc_patch_connect_descriptor context variables in both the RUN and PATCH file systems on the application tiers.

The above TNS connect strings provide the following advantages:

- A single connect string connects the application where the role-based database service is available
- A single tnsnames.ora file can be shared across all application servers and can be replicated from the primary to the DR site with no modifications required
- The application can fail over to the standby (or local standby) automatically, using scripts executed with Site Guard.

CAUTIONARY NOTE: There are many advantages to using the connect string as described above. However, be aware of the condition when role-based database services are available on both sites – for example, when the physical standby is in the SNAPSHOT STANDBY role, database services are up and both SCAN names are resolvable and reachable at either site. An application server or concurrent manager could connect to the database at a site you did not intend.

To remedy this situation, use the SQL*Net TCP_VALIDNODE_CHECKING feature in the sqlnet.ora files on the database tier to block application tier VMs at remote sites from accessing the database. Do this by setting the TCP.EXCLUDED_NODES and TCP.VALIDNODE_CHECKING parameters in the sqlnet.ora file for the SCAN and Grid listeners on each DB node. Put only application servers, not database servers, in this list so Data Guard traffic continues unimpeded. Note that changing this configuration requires restarting all SCAN listeners and Grid listeners.

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	EBSCDB1	VISPRD_OACORE_ONLINE	578
1	EBSCDB1	VISPRD_PCP_BATCH	24
2	EBSCDB2	VISPRD_OACORE_ONLINE	584
2	EBSCDB2	VISPRD_PCP_BATCH	28

NOTE: If the above JDBC URLs exceed 512 bytes, the EBS online patching utility adop will encounter the error “Buffer too small to get AD_APPS_JDBC_URL value” when applying online EBS patches. To work around this issue, the following steps can be performed on the PATCH file system; on each application tier node. As described below, these changes must be backed out before you execute the cutover phase of adop.

1. Source the PATCH file system.
2. Make a backup of the context file pointed to by the environment variable `$CONTEXT_FILE`.

```
$ cp $CONTEXT_FILE $CONTEXT_FILE.backup
```
3. Trim the URL length by editing the context file to remove the non-local DESCRIPTION from the DESCRIPTION_LIST. This must be done for both context variables `s_apps_jdbc_connect_descriptor` and `s_apps_jdbc_patch_connect_descriptor`.

To accomplish this on our system, we removed the entire DESCRIPTION containing the remote SCAN host from the TNS connect string:

```
(DESCRIPTION=(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=exa14-scan2.us.osc.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_OACORE_ONLINE)))
```

4. Save the changes.
5. Run Autoconfig on the PATCH file system. Since there is an open patch cycle, Autoconfig should complete successfully.
6. Re-attempt the adop operation that failed.
7. Make a backup of the context file containing the edits from step 3.
8. **IMPORTANT:** Once all online patches have been applied and BEFORE issuing “adop phase=cutover”, move the backup of the context file created in step 2 back into place
9. Run Autoconfig

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 note: [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/TKX.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 in 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 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 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 implementation 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 was configured with Local Traffic Manager (LTM) to direct traffic appropriately across the environments. Here are the steps needed to configure the f5 BigIP Load Balancers:

1. From the network administrator, obtain the following:
 - a. IP addresses for the load balancer front-end that application users will use. Ensure the IP addresses are added into DNS with the URL host and domain you wish to use for the primary site. Note the IP address should NOT be pingable at this point. If it is, then it is already in use.
 - b. The netmask and the CIDR of the network. This will be needed for setting up the internal VIP that the f5 will create and used when configuring the virtual server
2. In the primary f5 load balancer Networks area, ensure that there is at least one interface that is up and that you have a gateway route setup. Check with your network administrator for further details.
3. In the primary f5 load balancer Network area:
 - a. Select Self IPs and click Create.
 - b. Enter a name for the self-IP address. We chose to use the host.domain that was added into DNS from step 1 above, e.g., visprd.osc.oracle.com.
 - c. Enter the IP address and netmask.
 - d. Select the appropriate VLAN / Tunnel configured on your f5. Ours is set to “client”.
 - e. Click Finished. Once the Self-IP is created, it will be displayed in the list. You should now be able to ping the IP address.
4. 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. Do not use the logical host names. We used IP addresses.

5. Create a separate EBS-specific health monitor to be used by the pool being defined in the next step. The monitor settings can use the default settings of an interval of 5 seconds and a timeout of 16 seconds. The send string that can be used for EBS is:

```
GET /OA_HTML/AppsLogin\r\n
```

No Receive string is needed, this field is left blank. Set Reverse and Transparent to “No”.

6. On the primary f5 load balancer, create a pool in which the production application tier nodes will be configured, and enter a name of your choosing. We chose to name ours “ebs_site_pca3” to indicate the site and which PCA rack the app tiers were located on.
7. Add the nodes defined above as members to this pool and set the service port - in our case, 8000.
8. Add the health monitor to this pool.
9. Create and configure your Virtual Server:
 - a. On the primary f5 load balancer, create a Virtual Server that defines the front-end access. Ours is called “visprd_prod”.

- b. For the virtual server, set the source IP CIDR address. We set our 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.
- c. Set the virtual server destination address to the IP address from step 1 above. Your business may require additional names that point to the load balancer to be registered in DNS.
- d. Set the virtual server Service Port to 8000 or the appropriate port you choose. If you are using SSL and have loaded signed certificates from a certificate authority, then the default port is 443. Note that for snapshot standby testing we alter the service port to 9090 (or 9443 for SSL) to have a clear alternate path to the test application.
- e. Under the Resources tab, add the default pool created in step 6. In our case, we added “ebs_site_pca3” as the default pool.

As we used Local Traffic Managers (LTM), we repeated these steps at the standby site for its load balancer, with settings appropriate to that site. If you use a Global Traffic Manager, follow the required steps to add the second site’s nodes into a second server pool, then add that pool to the virtual server pool as a resource in step 13.

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

In the following steps, it is assumed that the EBS database is on 19c and is configured as a container database (CDB) with one pluggable database (PDB) that contains the EBS schemas. The application tier AD / TXK code level is assumed to be AD/TXK C.Delta 11 or higher.

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 the section [Delegation of Roles and Administration](#).

The OS groups and users on the application tier servers were created using the standard OS groupadd and useradd commands. We used these commands on all application tiers to create the oinstall OS group and the applmgr OS user, as noted in section 7.6.2 above:

```
groupadd -g 54321 oinstall
useradd --uid 54322 -g oinstall applmgr
usermod -d /home/applmgr -g oinstall -s /bin/bash applmgr
```

Since these environments are using NFS v3, we have specified the group ID and User ID and will do so on the application tiers at the standby so that they match.

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 application servers and to the Oracle E-Business Suite configuration. 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 exaadb01, exaabd02, root	<p>Three changes need to be made to /etc/hosts on each database server.</p> <p>a) Add aliases that translate the public IP addresses to the new logical host names. The pattern for the public IP addresses is:</p> <pre><IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain></pre> <p>b) Add aliases that translate the client VIP addresses to the new logical host names. The pattern for the client VIP addresses is:</p> <pre><IP Address> <DNS-registered VIP Name with domain> <DNS-registered VIP Name without Domain> <Logical VIP Name with domain> <Logical VIP Name without domain></pre> <p>Note: To find, or verify, the DNS-registered VIP's IP addresses, issue:</p> <pre>srvctl config vip -n <assigned node name></pre> <p>For example:</p> <pre>srvctl config vip -n exaadb03</pre> <p>c) Add entries that translate the cluster SCAN addresses to a set of logical SCAN names. The pattern for the cluster SCAN addresses is:</p> <pre><IP Address> <Logical SCAN name1> <IP Address> <Logical SCAN name2> <IP Address> <Logical SCAN name3></pre> <p>An example from our test system:</p> <pre>/etc/hosts from exaadb01: cat /etc/hosts ##### BEGIN Generated by Exadata. DO NOT MODIFY ##### 127.0.0.1 localhost.localdomain localhost 10.23.27.130 exaadb01.mycompany.com exaadb03 ebsdb1.mycompany.com ebsdb1 10.23.27.131 exaabd02.mycompany.com exaabd04 ebsdb2.mycompany.com ebsdb2 10.23.52.48 exaa01-vip.mycompany.com exaa03-vip ebsdb1-vip.mycompany.com ebsdb1-vip 10.23.52.49 exaa02-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</pre>
2	All database nodes exaadb01, exaabd02, root	<p>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:</p> <pre>ping ebsdb1 ping ebsdb2 ping ebsdb1-vip ping ebsdb2-vip ping ebsdb-scan1 ping ebsdb-scan2 ping ebsdb-scan3</pre>

8.2.3 Set Up Logical Host Names for the Application Tier Servers

STEP	SERVER, USER	ACTION
1	Primary application tier node	Make sure you have at least 6G disk space available in <INST_TOP>.

	pca3vm55, applmgr	
2	All application tier nodes, applmgr	<p>Apply patches (conditional)</p> <p>Apply the latest AD/TXK Delta 12 or higher, as this contains bug fixes and additional tooling for supporting 19c CDB/PDB for EBS:</p> <p>See MOS note Doc ID 1617461.1 for full details on applying the latest AD/TXK patch.</p>
3	All application tier nodes, applmgr	<p>Run AutoConfig on all app tier nodes, for both the RUN and the PATCH file systems.</p> <p>On our server:</p> <pre>. /u02/app/ebs/visprd/EBSapps.env R adautocfg.sh . /u02/app/ebs/visprd/EBSapps.env P adautocfg.sh</pre> <p>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.</p>
4	Primary application tier node, applmgr	Do a basic sanity check for adop – from the run file system, execute <code>adop -validate</code>
5	Primary application tier node, applmgr	<p>Run “Update current view snapshot” in AD Administration as follows</p> <p>From the run file system, execute <code>adadmin</code>.</p> <p>Choose 2. Maintain Applications Files menu</p> <p>Choose 4. Maintain snapshot information</p> <p>Choose 2. Update current view snapshot</p>
6	Primary application tier node, applmgr	When you upgrade to the latest AD/TXK, you must synchronize the appsutil directory on the database tier nodes . Follow the steps in section 5 of MOS doc 1617461.1.
7	All database tier nodes, oracle	Run AutoConfig on all database tier nodes
8	Master (primary) application tier node, applmgr	<p>Run <code>adpreclone.pl</code> on the RUN edition file system on the primary application tier node as follows.</p> <p>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.</p> <pre>echo \$FILE_EDITION run cd <INST_TOP>/admin/scripts perl adpreclone.pl appsTier</pre>
9	All application tier nodes, applmgr	<p>Shut the application down:</p> <pre>adstpall.sh</pre>
10	All application tier nodes pca3vm55, pca3vm56, root	<p>Make three changes to <code>/etc/hosts</code> on each application server:</p> <p>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. The pattern we used for the public IP addresses is:</p> <pre><IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain></pre> <p>An example from our test system (pca3vm55):</p>

		<pre> 10.25.29.199 pca3vm55.mycompany.com pca3vm55 ebsapp1.mycompany.com ebsapp1 10.25.29.200 pca3vm56.mycompany.com pca3vm56 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 must not 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 12.2.7 ### 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 </pre>
11	All application tier nodes, any user	<p>Make sure you can ping the all new host alias from all nodes you have added these logical names to. In our environment:</p> <pre> ping ebsapp1 ping ebsapp2 ping ebsdb-scan1 ping ebsdb-scan2 ping ebsdb-scan3 </pre>

8.2.4 Clone the Database to Itself Using EBS Rapid Clone Methodology

This task is only necessary when implementing logical host names in an existing installation that was created using a physical host name. The goal is to reconfigure the database home and contents to use the new logical host name. We are not using the Rapid Clone tooling to copy the database, only to configure Oracle E-Business Suite's appsutil directory in the database Oracle home for the new logical host name. Since we are reconfiguring the EBS components on a 19c RAC database, we will be making references to MOS note Cloning Oracle E-Business Suite Release 12.2 with Multitenant Database using Rapid Clone (Doc ID 2552208.1). We 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 exaad01, oracle	Source the EBS PDB environment (<i>not</i> the CDB environment). This will be the \$ORACLE_HOME/<PDB_name>_<hostname>.env file – for example, our PDB name is VISPRD: <pre> \$ cd \$ORACLE_HOME \$. ./VISPRD_exaad01.env </pre>
2	One database node	Run adpreclone.pl:

	exaadb01, oracle	<pre>\$ cd \$ORACLE_HOME/appsutil/scripts/[context_name] \$ perl adpreclone.pl dbTier</pre>
3	One database node exaadb01, oracle_ebs	<p>Save the contents of the \$TNS_ADMIN directory, as a later step may remove the network configuration files. This is a precautionary step should you need to restore them at some later time.</p> <pre>cd \$TNS_ADMIN/. cp -r \$TNS_ADMIN \$TNS_ADMIN.backup</pre>
4	All database nodes exaadb01, exaabd02, oracle	<p>Create a text file \$ORACLE_HOME/appsutil/clone/pairsfile.txt for each node with the below contents:</p> <pre>s_undo_tablespace=[Undo tablespace name in the PDB for each instance] s_dbClusterInst=[total number of instances in cluster] s_db_oh=[value of \$ORACLE_HOME]</pre> <p>On exaadb01:</p> <pre>s_undo_tablespace=APPS_UNDOTS1 s_dbClusterInst=2 s_db_oh=/u01/app/oracle_ebs/product/19.0.0.0/dbhome_2</pre> <p>On exaabd02:</p> <pre>s_undo_tablespace=UNDOTBS2 s_dbClusterInst=2 s_db_oh=/u01/app/oracle_ebs/product/19.0.0.0/dbhome_2</pre>
5	All database nodes, oracle	<p>Shut the database down.</p> <pre>\$ srvctl stop database -db EBSCDB_PHX3XD -stopoption immediate</pre>
6	1 st database node, oracle	<p>Navigate to [ORACLE_HOME]/appsutil/clone/bin and run adclonectx.pl with the following parameters:</p> <pre>\$ 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</pre> <p>Here is the conversation from our system:</p> <pre>\$ perl adclonectx.pl \ contextfile=/u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/VISPRD_ exaadb01.xml \ template=/u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/template/adxdbctx.tmp \ pairsfile=/u01/app/oracle/product/19.0.0.0/appsutil/clone/pairsfile.txt initialnode</pre> <p>Answer the prompts as follows. We provide answers to the prompts below using our environment. Please answer according to your environment. Not all prompts are listed here:</p> <p>Target System Hostname (virtual or normal) [Current hostname] : Enter the logical host name, i.e., ebsdb1</p> <p>Target Instance is RAC (y/n) [y] : y</p> <p>Target System CDB Name : EBSCDB</p> <p>Target System PDB Name : VISPRD</p> <p>Do you want to enable SCAN addresses (y/n) [y] ? : y</p> <p>Specify value for Scan Name : exaa-scan1</p> <p>Specify value for Scan Port : 1521</p> <p>Do you want the target system to have the same port values as the source system (y/n) [y] ? : Y</p> <p>Target System Port Pool [0-99] : 0</p> <p>Provide information for the initial RAC node:</p> <p>Host name : ebsdb1</p> <p>Virtual Host name [null] : ebsdb1-vip</p>

		<p>Instance number [1] : 1</p> <p>Target System Base Directory: /u01/app/oracle</p> <p>Oracle OS User [<Current OS User>]: oracle</p> <p>Oracle OS Group [<Current OS User group>]: oinstall</p> <p>Answer the remaining prompts as required. Once all questions have been answered, a new context file will be created – on our environment: VISPRD_ebsdb1.xml.</p>
7	1 st database node	<p>Check the new XML file, paying particular attention to the following entries. In our tests:</p> <pre>s_cdb_name = EBSCDB1; s_instName = EBSCDB1; s_dbCluster = TRUE; s_pdb_name = VISPRD; RAC prefix = VISPRD; [Name of the PDB] RAC_nodes is empty; expected SCAN name and port number should be set; s_update_scan = true;</pre>
8	1 st database node exadb01, oracle	<p>When adclonctx.pl is complete, copy the new context file to the \$ORACLE_HOME/appsutil directory if not already there:</p> <pre>\$ cp VISPRD_ebsdb1.xml \$ORACLE_HOME/appsutil/.</pre>
9	1 st database node exadb01 oracle	<p>Start the instance in open mode of the CDB database.</p> <pre>\$ srvctl start instance -db EBSCDB_PHX3XD -instance EBSCDB1</pre>
10	1 st database node exadb01 oracle	<p>Create the appsutil directory structure and configuration files for the logical host names.</p> <p>The generic configuration instruction for the initial node is:</p> <pre>\$ cd [RDBMS ORACLE_HOME]/appsutil/clone/bin \$ perl adcfgclone.pl dbconfig [Database target context file]</pre> <p>This is the command we executed on our system:</p> <pre>\$ cd /u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/clone/bin \$ perl adcfgclone.pl dbconfig /u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/VISPRD_ebsdb1.xml</pre>
11	1 st database node exadb01 oracle	<p>Log out and log back in. Re-establish your Linux environment.</p> <p>In our environment, we need to source this command to re-establish our Linux environment:</p> <pre>\$./u01/app/oracle_ebs/product/dbhome_2/19.0.0.0/VISPRD_ebsdb1.env</pre>
12	1 st database node exadb01 oracle	<p>If you need to make any changes to the directories used for UTIL_FILE_DIR then follow the steps in section 4.1.7 of MOS note 2552208.1.</p>
13	1 st database node exadb01 oracle	<p>Run AutoConfig. Correct any errors reported.</p>
14	1 st database node exadb01 oracle	<p>Archive the [ORACLE_HOME]/appsutil directory structure.</p> <p>This zipped archive will contain preclone content needed by the remaining nodes. Create the zip archive as follows:</p> <pre>\$ cd \$ORACLE_HOME \$ zip -r appsutil_node1.zip appsutil</pre>

15	All remaining Oracle RAC database nodes, oracle	<p>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.</p> <p>To extract the appsutil_node1.zip:</p> <pre>\$ cd \$ORACLE_HOME \$ unzip -o appsutil_node1.zip</pre>
16	2 nd node exaabd02, oracle	<p>Create a pairsfile.txt for the second node and make sure the contents of the pairsfile.txt for the second node is correct. You may have already created this pairsfile in step 4 above. As noted before, the undo tablespace that needs to be specified here resides within the EBS PDB. For example:</p> <p>On exaabd02:</p> <pre>s_undo_tablespace=UNDOTBS2 s_dbClusterInst=2 s_db_oh=/u01/app/oracle_ebs/product/19.0.0.0/dbhome_2</pre>
17	2 nd node exaabd02, oracle	<p>Create a context file for the secondary node, specifying the addnode option. Our environment is not using a dedicated EBS listener. Therefore, all of the ports are set to the grid listener ports on Exadata.</p> <p>The generic command is:</p> <pre>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</pre> <p>The command as run on our system:</p> <pre>perl adclonectx.pl contextfile=/u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/VISPRD_ebsdb1.xml template=/u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/template/adxdbctx.tmp pairsfile=/u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/clone/pairsfile.txt addnode</pre> <p>Answers to prompts are as follows:</p>

		<p>Provide the values required for creation of the new Database Context file.</p> <p>Target System Hostname (virtual or normal) [exaabd04] : ebsdb2</p> <p>Target Instance is RAC (y/n) [y] : Y</p> <p>Please provide the details to connect to one of live RAC nodes</p> <p>Host name of the live RAC node : ebsdb1</p> <p>Domain name of the live RAC node : mycompany.com</p> <p>Database SID of the live RAC node : VISPRD</p> <p>Listener port number of the live RAC node : 1521</p> <p>Provide information for the new Node:</p> <p>Host name : ebsdb2</p> <p>Virtual Host name : ebsdb2-vip</p> <p>Instance number : 2</p> <p>Private interconnect name : exaabd04-priv</p> <p>Current Node:</p> <p>Host Name : ebsdb2 ← Logical host name of 2nd RAC node</p> <p>SID : VISPRD ← PDB Name</p> <p>Instance Name : EBSCDB2 ← Instance name of the CDB</p> <p>Instance Number : 2</p> <p>Instance Thread : 2</p> <p>Undo Table Space:</p> <p>Listener Port : 1521</p> <p>Target System Base Directory : /u01/app/oracle</p> <p>Oracle OS User [oracle_ebs] :</p> <p>Oracle OS Group [oinstall] : dba</p> <p>Target System utl_file_dir Directory List : /u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/outbound/VISPRD_ebsdb2</p> <p>Do you want to preserve the Display [scl05kss:2] (y/n) : Y</p> <p>Answer any remaining prompts as required.</p>
17	2 nd node, oracle	Copy the new context file to \$ORACLE_HOME/appsutil if it's not already there.
18	2 nd node, oracle	Start the instance on the 2 nd node. \$ srvctl start instance -db EBSCDB_PHX3XD -instance EBSCDB2
19	2 nd node, oracle	Use adcfgclone.pl dbconfig to configure the Oracle home. We ran this command: perl adcfgclone.pl dbconfig /u01/app/oracle/product/19.0.0.0/dbhome_2/appsutil/VISPRD_ebsdb2.xml
20	2 nd node, oracle	Source the new environment file:

		<pre>cd [ORACLE_HOME] ./[CONTEXT_NAME].env</pre> <p>Note: The CONTEXT_NAME for 19c will be in the form <PDB_name>_<logicalhost_name>.env. For example: VISPRD_ebsdb2.env</p>
21	2 nd node, oracle	If you need to make any changes for the directories used by UIL_FILE_DIR, then follow the steps in section 4.2.11 of MOS note 2552208.1.
22	2 nd node, oracle	<p>Run AutoConfig:</p> <pre>cd \$ORACLE_HOME/appsutil/scripts/\$CONTEXT_NAME ./adautocfg.sh appspass=[APPS password]</pre>
23	1 st node, oracle	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.
24	All database nodes, oracle	<p>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.</p> <p>On each of the database nodes:</p> <pre>./ebs_scan_db.sh SCAN Name: exaa-scan3</pre> <p>Add the following entries to your /etc/hosts files on each RAC node in THIS cluster.</p> <pre>## SCAN VIPs 10.23.52.53 ebsdb-scan1 10.23.52.54 ebsdb-scan2 10.23.52.55 ebsdb-scan3</pre> <p>Then, look at the \$TNS_ADMIN directory to verify that a new ifile was created for example: VISPRD_ebsdb1_ifile.ora.</p>
25	All database nodes, oracle	<p>Create a directory under the EBS database's \$TNS_ADMIN with the same name as the CDB database name. The directory must have the same name on all database nodes. This directory is only for the container database tnsnames.ora and the sqlnet.ora. If your EBS database has implemented Transparent Data Encryption (TDE), the sqlnet.ora file with the TDE wallets specified must go into this directory.</p> <pre>mkdir /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin/EBSCDB</pre> <p>There is a separate directory created by autoconfig that contains the EBS PDB specific listener.ora, sqlnet.ora, and tnsnames.ora files. This is also where the custom tnsnames ifile will reside.</p> <p>Configure Oracle Clusterware using srvctl to set TNS_ADMIN to point to the EBSCDB subdirectory. This is required for clusterware to start and open the database:</p> <pre>srvctl setenv database -db EBSCDB_phx3xd -t "TNS_ADMIN=/u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin/EBSCDB"</pre> <p>This configuration step should already be in place, but it is included here for your information:</p> <pre>srvctl setenv database -d EBSCDB_phx3xd -t "ORA_NLS10=/u01/app/oracle/product/19.0.0.0/dbhome_2/nls/data/9idata"</pre> <p>Restart both database instances, using srvctl:</p> <pre>srvctl stop database -db EBSCDB_phx3xd -stopoption immediate srvctl start database -db EBSCDB_phx3xd</pre>

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 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. MOS note:

Cloning Oracle E-Business Suite Release 12.2 with Multitenant Database using Rapid Clone (Doc ID 2552208.1)

serves as the basis for the following steps.

STEP	SERVER, USER	ACTION
1	All application mid tiers (pca3vm55,pca3vm56), 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 run these commands to shut down all application services: <pre>\$ cd \$ADMIN_SCRIPTS_HOME \$ adstpall.sh</pre>
2	Primary application tier node pca3vm55, 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. The commands given in this step either remove or hide the oraInventory but they do not create a new oraInventory. The Rapid Clone tooling will create a new oraInventory for you. 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: <pre>cd <location of oraInventory> mv oraInventory oraInventory.backup</pre> 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. <pre>cd \$ORACLE_HOME/oui/bin ./runInstaller -silent -removeHome ORACLE_HOME=/u02/app/ebs/visprd/fs2/EBSapps/10.1.2 - invPtrLoc /u02/app/ebs/visprd/oraInventory/oraInst.loc</pre> <pre>\$ cd /u02/app/ebs/visprd/fs1/EBSapps/10.1.2 \$ cd oui/bin \$./runInstaller -silent -removeHome ORACLE_HOME=/u02/app/ebs/visprd/fs1/EBSapps/10.1.2 - invPtrLoc /u02/app/ebs/visprd/oraInventory/oraInst.loc</pre>
3	Primary application tier node pca3vm55, applmgr	Move the FMW Oracle homes, INST_TOP, and PATCH file system directories so they appear to be non-existent 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): <pre>\$ mv /u02/app/ebs/visprd/fs2/FMW_home /u02/app/ebs/visprd/fs2/FMW_HOME_bkp</pre> <pre>\$ mv /u02/app/ebs/visprd/fs2/inst /u02/app/ebs/visprd/fs2/inst_bkp</pre> <pre>\$ mv /u02/app/ebs/visprd/fs1 /u02/app/ebs/visprd/fs1_bkp</pre>
4	Primary application tier node pca3vm55, applmgr	Log off and back on again. Set PERL5LIB and your path to find and execute the proper version of Perl. For example: <pre>export PERL5LIB=/usr/lib64/perl5</pre> Make sure /usr/bin is first in the PATH to run Perl.

		<code>export PATH=/usr/bin:\$PATH</code>
5	Primary application tier node pca3vm55, applmgr	Configure the application tier to use the new logical host name. Please note that running Rapid Clone tooling “adcfgclone.pl appsTier dualfs” as described below will configure the first application tier node. You will need to add back any additional application tier nodes after the cloning process is completed and after completing all steps in this section. To configure, run the following commands:

```

$ cd <COMMON_TOP>/clone/bin
$ perl adcfgclone.pl appstier dualfs

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
/u02/app/ebs/visprd/fs2/EBSapps/comm/clone/bin/CloneContext_1122152645.log

Provide the values required for creation of the new APPL_TOP Context file.

Target System Hostname (virtual or normal) [pca3vm55] : ebsapp1

Target System Database SID : VISPRD ← Name of the EBS PDB

Target System Database Server Node [testapp1] : ebsdb1-vip

Target System Database Domain Name [mycompany.com] :

Target System Base Directory : /u02/app/ebs/visprd

Target System Base Directory set to /u02/app/ebs/visprd

Target System Current File System Base set to /u02/app/ebs/visprd/fs2
Target System Other File System Base set to /u02/app/ebs/visprd/fs1

Target System Fusion Middleware Home set to /u02/app/ebs/visprd/fs2/FMW_Home
Target System Other File System Fusion Middleware Home set to
/u02/app/ebs/visprd/fs1/FMW_Home

Target System Web Oracle Home set to /u02/app/ebs/visprd/fs2/FMW_Home/webtier
Target System Other File System Web Oracle Home set to
/u02/app/ebs/visprd/fs1/FMW_Home/webtier

Target System Appl TOP set to /u02/app/ebs/visprd/fs2/EBSapps/appl
Target System Other File System Appl TOP set to /u02/app/ebs/visprd/fs1/EBSapps/appl

Target System COMMON TOP set to /u02/app/ebs/visprd/fs2/EBSapps/comm
Target System Other File System COMMON TOP set to /u02/app/ebs/visprd/fs1/EBSapps/comm

Target System Instance Home Directory [/u02/app/ebs/visprd] :

Target System Current File System Instance Top set to
/u02/app/ebs/visprd/fs2/inst/apps/VISPRD_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
/u02/app/ebs/visprd/fs2/EBSapps/comm/clone/bin/out/VISPRD_ebsapp1/portpool.lst

UTL_FILE_DIR on database tier consists of the following directories.

```

		<pre> 1. /tmp 2. /tmp 3. /u01/app/oracle/product/19.0.0.0/appsutil/outbound/VISPRD_exadb03 4. /tmp Choose a value which will be set as APPLTMP value on the target node [1] : The new APPL_TOP context file has been created : /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/appl/admin/VISPRD_ebsapp1.xml Check Clone Context logfile /u02/app/ebs/visprd/fs2/EBSapps/comn/clone/bin/CloneContext_1122152645.log for details. Creating Patch file system context file..... Log file located at /u02/app/ebs/visprd/fs2/EBSapps/comn/clone/bin/CloneContextPatch_1122152858.log Target System Other File System Instance Top set to /u02/app/ebs/visprd/fs1/inst/apps/VISPRD_ebsapp1 Complete port information available at /u02/app/ebs/visprd/fs2/EBSapps/comn/clone/bin/out/VISPRD_ebsapp1/portpool.lst The new APPL_TOP context file has been created : /u02/app/ebs/visprd/fs1/inst/apps/VISPRD_ebsapp1/appl/admin/VISPRD_ebsapp1.xml Check Clone Context logfile /u02/app/ebs/visprd/fs2/EBSapps/comn/clone/bin/CloneContextPatch_1122152858.log for details. FMW Pre-requisite check log file location : /u02/app/ebs/visprd/fs2/EBSapps/comn/clone/FMW/logs/prereqcheck.log Running: FMW pre-req check... Configuring: Run file system.... LogFile located at /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/admin/log/clone/run/RCloneApplyApptier_ 11221529.log </pre>
6	Primary application tier pca3vm55, applmgr	Establish the new environment by sourcing the EBSapps.env. Specify the RUN environment.
7	Primary application tier pca3vm55, applmgr	As of AD / TXK C.Delta 11 or the April 2019 CPU patch that is applied to the application tier, there are changes to accessing the WLS Admin console. Within the context file, check the context variable <code>s_wls_admin_console_access_nodes</code> and ensure that the physical node name of the primary application tier is listed along with any other external nodes that should be allowed to access the WLS Admin console. See Appendix B of MOS note 1383621.1 for further details.
8	Primary application tier node pca3vm55, 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 .
9	Primary application tier node, applmgr	<p>Generate the tnsnames.ora ifile entries for using the database logical host names. See the <code>ebs_scan_app.sh</code> script in Appendix B: Scripts for Managing SCAN for Primary and DR Sites for a sample script to generate the ifiles.</p> <p>Do this on both the run and the patch file systems.</p> <p>On our system running <code>ebs_scan_app.sh</code> on the RUN file system:</p> <pre>\$./ebs_scan_app.sh</pre> <p>Add the following entries to your <code>/etc/hosts</code> files on each RAC node in THIS cluster.</p> <pre>## SCAN VIPs 10.23.52.53 ebsdb-scan1 10.23.52.54 ebsdb-scan2 10.23.52.55 ebsdb-scan3</pre> <p>Then, look at the new ifile in <code>\$TNS_ADMIN</code>. For example the new file might be <code>VISPRD_ebsapp1_ifile.ora</code></p>
10	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. Here is the generic XML:

		<pre><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=<PDB_name>ebs_patch)(INSTANCE_NAME=<RAC Instance Name>))</patch_jdbc_url></pre> <p>On our system:</p> <p>Run Edition:</p> <pre><jdbc_url oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOA D_BALANCE=YES)(FAILOVER=YES)(ADDRESS=(PROTOCOL=tcp)(HOST=exaa- scan1)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME= VISPRD_OACORE_ONLINE))</jdbc_url> <jdbc_url_generation_check oa_var="s_jdbc_connect_descriptor_generation">>false</jdbc_url_generation_check> <patch_jdbc_url oa_var="s_apps_jdbc_patch_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PR OTOCOL=tcp)(HOST=exaa- scan1)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_ebs_patch)(INSTANCE_NAME=EBSCDB1))< /patch_jdbc_url></pre> <p>Patch Edition:</p> <pre><jdbc_url oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOA D_BALANCE=YES)(FAILOVER=YES)(ADDRESS=(PROTOCOL=tcp)(HOST=exaa- scan1)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME= VISPRD_OACORE_ONLINE))</jdbc_url> <jdbc_url_generation_check oa_var="s_jdbc_connect_descriptor_generation">>false</jdbc_url_generation_check> <patch_jdbc_url oa_var="s_apps_jdbc_patch_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PR OTOCOL=tcp)(HOST=exaa- scan1)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=VISPRD_ebs_patch)(INSTANCE_NAME=EBSCDB1))< /patch_jdbc_url></pre> <p>Note: If you wish to configure a multi-address TNS connect string containing both primary and standby addresses, please read section 7.7 and change the above connect string as described in that section. You will need to know the SCAN host name at the standby site.</p>
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	On one database node, check the context variable s_update_scan. If it is set to FALSE, change it to TRUE.
12	All Oracle RAC database nodes	Run AutoConfig on the node you are configuring, then on all remaining database nodes.
13	Primary application tier node pca3vm55, 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 pca3vm55, applmgr	Validate the setup: Start the admin server on the run file system, then validate the configuration: adadminsvctl.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 \(Doc ID 1905593.1\)](#)
[Cloning Oracle E-Business Suite Release 12.2 with Multitenant Database using Rapid Clone \(Doc ID 2552208.1\)](#)

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 \(RAC or Non-RAC\) \(Doc ID 1617946.1\)](#).

ST EP	SERVER, USER	ACTION
1	One database node exadb01, oracle	<p>Enable force logging if it is not already enabled. To enable force logging, issue this command in SQL*Plus:</p> <pre>SQL> alter database force logging;</pre>
2	Each Oracle RAC database node, oracle	<p>Note: This step does not apply if the Grid Listener is being used instead of EBS specific local listeners.</p> <p>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.</p> <p>To modify the EBS listener and the database, log in to each Oracle RAC database node, source the EBS environment, and follow these steps:</p> <p>a) Copy the listener.ora to the listener_ifile.ora file.</p> <pre>\$ cd \$TNS_ADMIN \$ cp listener.ora listener_ifile.ora</pre> <p>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.</p> <p>On the first Oracle RAC node we changed ebsdb1 to exadb03. We made similar changes on each node:</p>

```

VISPRD =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = exaa01-vip.mycompany.com)(PORT =
1523)(IP = FIRST)))
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = exaadb01.mycompany.com)(PORT = 1523)(IP
= FIRST)))
    )
  )

SID_LIST_VISPRD =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2)
      (SID_NAME = EBSCDB1)
    )
  )

STARTUP_WAIT_TIME_VISPRD = 0
CONNECT_TIMEOUT_VISPRD = 10
TRACE_LEVEL_VISPRD = OFF

LOG_DIRECTORY_VISPRD = /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin
LOG_FILE_VISPRD = EBSCDB1
TRACE_DIRECTORY_VISPRD = /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin
TRACE_FILE_VISPRD = EBSCDB1
ADMIN_RESTRICTIONS_VISPRD = ON
SUBSCRIBE_FOR_NODE_DOWN_EVENT_VISPRD = OFF

```

c) Restart the listener on the Oracle RAC node that is being reconfigured.

```

$ lsnrctl stop visprd
$ lsnrctl start visprd

```

d) Edit the <SID>_node_1file.ora and edit the TNS connect string aliases <SID>_LOCAL (i.e., EBSCDB1_LOCAL, EBSCDB2_LOCAL) and change the HOST parameter to the assigned host name in DNS as in the following example.

```

EBSCDB1_LOCAL=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=exaa01-
vip.mycompany.com)(PORT=1523))
    (ADDRESS=(PROTOCOL=tcp)(HOST=exaa01-
vip.mycompany.com)(PORT=1521))
  )

EBSCDB2_LOCAL=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=exaa01-
vip.mycompany.com)(PORT=1523))
    (ADDRESS=(PROTOCOL=tcp)(HOST=exaa01-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='EBSCDB1_LOCAL' sid='EBSCDB1' scope=both;
SQL> alter system set local_listener='EBSCDB2_LOCAL' sid='EBSCDB2' scope=both;

```

3

Each Oracle RAC database node, oracle

You must use password files at a standby database.

If you do NOT have password files created in either ASM or \$ORACLE_HOME/dbs,, perform this step.

		<p>Create a password file on one of the Oracle RAC database nodes. This password file will later be copied over to the standby server when we prepare the standby. Run this command to create a password file:</p> <pre>\$ orapwd file=orapw\$ORACLE_SID password=<your password> entries=10 ignorecase=y</pre> <p>Now, place the password file into ASM using commands similar to those in the example below. Our example assumes role separation between the grid and the oracle OS users:</p> <pre>\$ asmcmd -p --privilege sysdba ASMCMD [+] > cd +DATA1 ASMCMD [+DATA2] > mkdir EBSCDB_PHX3D/PASSWORD ASMCMD [+DATA1] > pwcop -dbunique EBSCDB_PHX3D /u01/app/oracle/product/19.0.0/dbhome_2/dbs/orapwEBSCDB1 +DATA1 /EBSCDB_PHX3D/PASSWORD/orapwebsc ASMCMD [+DATA2] > exit</pre> <p>Specify where the password file is located using SRVCTL:</p> <pre>srvctl modify database -db EBSCDB_PHX3D -pwfile '+DATA1/EBSCDB_PHX3D/PASSWORD/orapwebsc '</pre> <p>Proceed to step 5 below.</p>
4	Primary database node exadb01, oracle, grid	<p>If the password file is stored in ASM and is shared by all RAC instance, we only need to copy it from ASM into the \$ORACLE_HOME/dbs directory on one database node. This password file will later be copied over to the standby server when we prepare the standby.</p> <p>First, as the owner of the database home (oracle), determine the location within ASM the password file is located using srvctl:</p> <pre>\$ srvctl config database -db EBSCDB_PHX3D</pre> <p>In the output, you should see the password location similar to:</p> <pre>Password file: +DATA1/EBSCDB_PHX3D/PASSWORD/pwebsc</pre> <p>Next, copy the password file to \$ORACLE_HOME/dbs. If the grid infrastructure is owned by a separate OS user (grid), then log into that user before executing the below commands:</p> <pre>\$ asmcmd -p --privilege sysdba ASMCMD [+] > pwcop -dbunique EBSCDB_PHX3D +DATA1/EBSCDB_PHX3D/PASSWORD/pwebsc /u01/app/oracle/product/19.0.0/dbhome_2/dbs/orapw<ORACLE_SID></pre> <p>Where ORACLE_SID in our case is EBSCDB1. The password file name would then be orapwEBSCDB1.</p>
5	Primary database node exadb01, oracle	<p>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.)</p> <p>Note: All Data Guard parameters will be set by Data Guard Broker after the standby database has been instantiated. Not all parameters are listed here.</p> <p>The parameters should be set using a command such as this:</p> <pre>alter system set [parameter name]=[parameter value] scope=spfile</pre> <p>Our system parameters:</p>

		<pre> processes = 2500 -- per EBS requirements log_buffer=134217728 compatible='19.0.0.0.0' log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST' log_archive_format='%t_%s_%r.dbf' db_name='EBSCDB' db_unique_name='EBSCDB_phx3xd' db_block_checking='MEDIUM' db_block_checksum=FULL db_lost_write_protect=TYPICAL db_recovery_file_dest=+RECO1 db_file_create_dest=+DATA1 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 </pre>
6	Primary database node exaad01, oracle	<p>Enable ARCHIVELOG mode if it is not already active. This requires some downtime as you must shut down all Oracle RAC nodes before it can be enabled, as in the following steps.</p> <p>a) First, shut down all Oracle RAC instances but one.</p> <p>b) On the remaining Oracle RAC instance, from SQL*Plus as sysdba:</p> <pre> shutdown immediate; startup mount alter database archivelog; alter database open; </pre> <p>c) Finish by restarting all other Oracle RAC instances.</p>
7	Primary database node exaad01, oracle	<p>Add standby redo logs to the primary. RMAN will clone 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.</p> <p>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.</p> <pre> 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; </pre>

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 exabadm03, exabadm04 root	<p>Three changes need to be made to /etc/hosts on each database server:</p> <p>a) Add aliases that translate the public IP addresses to the new logical host names. The pattern for the public IP addresses is:</p> <pre><IP Address> <Physical Name with domain> <Physical Name without Domain> <Logical Host Name with domain> <Logical Name without domain></pre> <p>b) Add aliases that translate the client VIP addresses to the new logical host names. The pattern for the client VIP addresses is:</p> <pre><IP Address> <DNS-registered VIP Name with domain> <DNS-registered VIP Name without Domain> <Logical VIP Name with domain> <Logical VIP Name without domain></pre> <p>Note: To find, or verify, the DNS-registered VIP's IP addresses, issue:</p> <pre>srvctl config vip -n <assigned node name></pre> <p>For example:</p> <pre>srvctl config vip -n exabadm01</pre> <p>c) Add entries that translate the cluster SCAN addresses to a set of logical SCAN names. The pattern for the cluster SCAN addresses is:</p> <pre><IP Address> <Logical SCAN name1> <IP Address> <Logical SCAN name2> <IP Address> <Logical SCAN name3></pre> <p>An example from our test system:</p> <pre>/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 exabadm03.mycompany.com exabadm03 ebsdb1.mycompany.com ebsdb1 20.35.11.46 exabadm04.mycompany.com exabadm04 ebsdb2.mycompany.com ebsdb2 ## VIPs added for EBS 12.2 20.35.22.221 exabclient03-vip.mycompany.com exabclient03-vip ebsdb1-vip.mycompany.com ebsdb1-vip 20.35.22.222 exabclient04-vip.mycompany.com exabclient04-vip ebsdb2-vip.mycompany.com ebsdb2-vip #### END Generated by Exadata #### ####</pre> <pre>### 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</pre>
2	All database nodes exabadm03, exabadm04 root	<p>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:</p> <pre>host files dns</pre>
3	Each database node exabadm03, exabadm04 Any user	<p>From each node, ping another node using your new logical names, to be sure that they can be resolved.</p> <p>E.g., from exabadm02, we ran this ping command:</p> <pre>ping ebsdb1</pre>

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

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

STEP	SERVER, USER	ACTION
1	Pca1vm55 and pca1vm56, root	<p>On the standby app tier nodes, as root, edit /etc/hosts to add the logical host name.</p> <p>For the first standby application tier node:</p> <pre># 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 20.23.25.55 pca1vm55.mycompany.com pca1vm55 ebsapp1.mycompany.com ebsapp1 20.23.25.56 pva1vm56.mycompany.com pca1vm56 ebsapp2.us.oracle.cim ebsapp2</pre> <p>For the second application tier node:</p> <pre># 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 20.23.25.56 pca1vm56.mycompany.com pca1vm56 ebsapp2.mycompany.com ebsapp2 20.23.25.55 pca1vm55.mycompany.com pca1vm55 ebsapp1.mycompany.com ebsapp1</pre>
2	All standby application tier nodes, root	<p>At the standby, add the SCAN, VIP and database node aliases into /etc/hosts.</p> <pre># Database logical host names and VIPs 20.35.11.45 exabadm03.mycompany.com exabadm03 ebsdb1.mycompany.com ebsdb1 20.35.11.46 exabadm04.mycompany.com exabadm04 ebsdb2.mycompany.com ebsdb2 ## VIPs added for EBS 12.2 20.35.22.221 exabclient03-vip.mycompany.com exabclient03-vip ebsdb1- vip.mycompany.com ebsdb1-vip 20.35.22.222 exabclient04-vip.mycompany.com exabclient04-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-scan1</pre>
3	All application tier nodes, any user	<p>Make sure you can ping the new host alias, e.g.:</p> <pre>ping ebsapp1</pre>

8.3.3 Set Up the Standby Database

There are five tasks to complete to set up the standby database servers:

- Copy the ORACLE_HOME
- Configure the ORACLE_HOME
- Instantiate the standby database using RMAN RESTORE FROM SERVICE
- RAC-enable the standby
- 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, adclonctx, 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 for E-Business Suite references.
- 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 to node1 (ebsdb1) on 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 exaadb01, exaabd02, root	<p>As root, on each of the primary database nodes, zip up the ORACLE_HOME.</p> <p>On our system, \$ORACLE_HOME points to: /u01/app/oracle/product/19.0.0.0</p> <p>To zip up the ORACLE_HOMES on each database server:</p> <pre>\$ cd /u01/app/oracle/product/19.0.0.0</pre> <p>On exaadb01:</p> <pre>\$ zip -r EBSCDB_dbhome_2_ebsdb1_19800_20201015.zip dbhome_2</pre> <p>On exaabd02:</p> <pre>\$ zip -r EBSCDB_dbhome_2_ebsdb2_19800_20201015.zip dbhome_2</pre> <p>Note: This zip file will include the appsutil and network/admin directories, which is what we want for each node.</p>
2	Each pair of database servers exaadb01, exaabd02, exabadm03, exabadm04, oracle	<p>As the database owner, copy the zipped Oracle home from each primary database node to its partner at the standby site. Make sure the directory structures at the standby database servers are exactly the same as those on the primary database servers.</p> <p>On our system, we copied from exaadb03 (ebsdb1) to the standby node exabadm01 (ebsdb1), and from exaabd04 (ebsdb2) to exabadm02 (ebsdb2).</p> <p>Example:</p> <pre>\$ scp EBSCDB_dbhome_2_ebsdb1_19800_20201015.zip oracle@exabadm03:/u01/app/oracle/product/19.0.0.0</pre>
3	Each standby database server exabadm03, exabadm04 oracle_ebs	<p>On each standby database node, as the database owner, unzip the Oracle home into the corresponding directory.</p> <p>On our system, on exabadm03:</p> <pre>\$ cd /u01/app/oracle/product/19.0.0.0 \$ unzip EBSCDB_dbhome_2_ebsdb1_19800_20201015.zip</pre> <p>And on exabadm04:</p> <pre>\$ cd /u01/app/oracle/product/19.0.0.0 \$ unzip EBSCDB_dbhome_2_ebsdb1_19800_20201015.zip</pre>

To copy ORACLE_HOME using the rsync method, do the following steps.

STEP	SERVER, USER	ACTION
1	Each pair of database servers exaadb01 & exabadm02, exaabd03 & exabadm04, root	<p>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 <code>rsync_EBS_OH.sh</code> 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 <code>--no_dry_run</code> option executes the rsync copy.</p> <p>On our system, run the <code>rsync_EBS_OH.sh</code> script as root on exaadb01 to rsync to the target host exabadm03. As we want to clone the entire Oracle home (including the appsutil, dbs and network/admin directories), we need to use the <code>--with_ebs_config</code> option:</p> <pre>\$./rsync_EBS_OH.sh --source_oh \$ORACLE_HOME --target_oh \$ORACLE_HOME --target_host exabadm03 --os_user root --with_ebs_config</pre> <p>Using the following parameters:</p> <pre>source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target_hosts = exabadm03 os_user = root dry_run = true with_ebs_config = true</pre> <p>About to execute the following rsync command:</p> <pre>rsync -avzh --progress --dry-run /u01/app/oracle/product/19.0.0.0/dbhome_2 root@exabadm03:/u01/app/oracle/product/19.0.0.0/dbhome_2</pre> <p>OK to continue? [Y y N n] :</p> <p>Then sync from exaabd02 to the target exabadm04:</p> <pre>\$./rsync_EBS_OH.sh --source_oh \$ORACLE_HOME --target_oh \$ORACLE_HOME --target_host exabadm04 --os_user root --with_ebs_config</pre> <p>Using the following parameters:</p> <pre>source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target_hosts = exabadm04 os_user = root dry_run = true with_ebs_config = true</pre> <p>About to execute the following rsync command:</p> <pre>rsync -avzh --progress --dry-run /u01/app/oracle/product/19.0.0.0/dbhome_2 root@exabadm04:/u01/app/oracle/product/19.0.0.0/dbhome_2</pre> <p>OK to continue? [Y y N n] :</p>
2	Each pair of database servers exaadb01 & exabadm02, exaabd03 & exabadm04, root	<p>Run the <code>rsync_EBS_OH.sh</code> script with the <code>--no_dry_run</code> option to copy over the ORACLE_HOME.</p> <p>On our system, as root on exaadb03 to the target host exabadm01:</p>

```

$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadm03
--os_user root --no_dry_run --with_ebs_config

Using the following parameters:

source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target_hosts      = exabadm03
os_user           = root
dry_run           = false
with_ebs_config   = true

About to execute the following rsync command:

rsync -avzh --progress /u01/app/oracle/product/19.0.0.0/dbhome_2
root@exabadm03:/u01/app/oracle/product/19.0.0.0/dbhome_2

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

Then from exaabd02 to the target exabadm04:

$ ./rsync_EBS_OH.sh --source_oh $ORACLE_HOME --target_oh $ORACLE_HOME --target_host exabadm04
--os_user root --no_dry_run --with_ebs_config

Using the following parameters:

source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target_hosts      = exabadm04
os_user           = root
dry_run           = false
with_ebs_config   = true

About to execute the following rsync command:

rsync -avzh --progress /u01/app/oracle/product/19.0.0.0/dbhome_2
root@exabadm04:/u01/app/oracle/product/19.0.0.0/dbhome_2

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 exabadm03, exabadm04, oracle	<p>Update your login profile script to source the CDB (not the PDB) database environment. The file name of the CDB environment file is <code>ORACLE_SID_node.env</code>. Ours is <code>EBSCDB1_ebsdb1.env</code></p> <p>On our environment, the script to call is <code>ORACLE_HOME/EBSCDB1_ebsdb1.env</code> on node exabadm01 and <code>ORACLE_HOME/EBSCDB2_ebsdb2.env</code> on node exabadm02. We placed the following in our <code>.bash_profile</code> login script:</p> <pre># Set up env for the EBS database env. ./u01/app/oracle/product/19.0.0.0/dbhome_2/EBSCDB1_ebsdb1.env</pre>
2	Each standby database node exabadm03, exabadm04, oracle	<p>Run the database clone.pl script to update the oraInventory with the new/adjusted oracle home, using the script <code>clone_ohome.sh</code> provided below.</p> <p>On our environment, adjusting the value for the <code>LOCAL_NODE</code> parameter as appropriate (this example shows the command for the first node):</p> <pre>#!/bin/sh # The 19c version</pre>

		<pre> ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_2 ORACLE_HOME_NAME=dbhome_2_ebs ORACLE_BASE=/u01/app/oracle THISNODE=exabdm03 C01="CLUSTER_NODES={exabdm03}" C02="LOCAL_NODE=\${THISNODE}" perl \$ORACLE_HOME/clone/bin/clone.pl ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_2 ORACLE_HOME_NAME=dbhome_2_ebs ORACLE_BASE=/u01/app/oracle OSDBA_GROUP=dba \$C01 \$C02 -local echo "Clone ended at `date`" tee -a clone.log Note: When running this script on other RAC nodes, ensure you change the node name in the line C01="CLUSTER_NODES={exabdm03}" to the name of the RAC node the script is run on i.e., exabdm04. 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/product/19.0.0.0/dbhome_2" - local The clone.pl script will re-attach the ORACLE_HOME. </pre>
3	Each standby database node exabdm03, exabdm04, root	<p>At the standby, on each database node, log in as root. Then cd to the ORACLE_HOME directory and execute root.sh:</p> <pre> cd /u01/app/oracle/product/19.0.0.0/dbhome_2 ./root.sh </pre>
4	Each standby database node exabdm03, exabdm04, oracle	<p>Note: If the Grid Listener is being used instead of the local EBS specific listener, this step does not apply and can be skipped.</p> <p>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.</p> <p>On our standby database Oracle RAC server exabdm1:</p>

		<pre> VISPRD = (DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST = exabclient03-vip.mycompany.com)(PORT = 1523)(IP = FIRST))) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST = exabadm03.mycompany.com)(PORT = 1523)(IP = FIRST))))) SID_LIST_VISPRD = (SID_LIST = (SID_DESC = (ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2)(SID_NAME = EBSCDB1))) STARTUP_WAIT_TIME_VISPRD = 0 CONNECT_TIMEOUT_VISPRD = 10 TRACE_LEVEL_VISPRD = OFF LOG_DIRECTORY_VISPRD = /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin LOG_FILE_VIS = EBSCDB1 TRACE_DIRECTORY_VISPRD = /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin TRACE_FILE_VIS = EBSCDB1 ADMIN_RESTRICTIONS_VISPRD = ON SUBSCRIBE_FOR_NODE_DOWN_EVENT_VISPRD = OFF </pre>
5	Each standby database node exabadm03, exabadm04, oracle	<p>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 VISPRD_ebsdb1.ora file has the following entries:</p> <pre> EBSCDB1_LOCAL= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient03- vip.domain.com)(PORT=1523)) (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient03- vip.domain.com)(PORT=1521))) EBSCDB2_LOCAL= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient04- vip.domain.com)(PORT=1523)) (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient04- vip.domain.com)(PORT=1521))) </pre> <p>The above example lists two addresses, the first one for an EBS specific listener on port 1523 and the second one for the grid listener on port 1521. If your environment is not using an EBS specific listener then you only need to have one address that points to the grid listener. The above includes both addresses as an example.</p> <p>Add the above entries to the custom.txt file that is read by ebs_scan_db.sh. Note there is no need to run ebs_scan_db.sh at this time.</p>
6	Each standby database node exabadm03, exabadm04, oracle	<p>Start the EBS listener and verify that it has started. Only the static SID will be registered with the listener.</p> <p>Note: If the Grid Listener is being used instead of the EBS listener, this step can be skipped.</p> <pre> \$ lsnrctl start visprd \$ lsnrctl status visprd </pre> <p>You should see the following:</p> <pre> Service "VISPRD" has 1 instance(s). Instance "EBSCDB1", status UNKNOWN, has 1 handler(s) for this service... </pre>

8.3.3.3 Instantiate the Standby Database with RMAN RESTORE FROM SERVICE

This task copies the database from the primary to the standby. Instantiating the physical standby database will be accomplished using RMAN RESORE FROM SERVICE, a feature introduced as of Oracle database version 12c. This feature has been enhanced to simplify the instantiation process. The tasks are:

- Copy the password file to the standby
- Copy the TDE wallets from the primary to the standby if the primary database has implemented transparent data encryption
- Copy the parameter file (pfile) from the primary and edit it for the standby database
- Create required directories
- Update TNS connect strings
- Instantiate the physical standby database

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	Primary database node exaadb01, oracle Standby database node exabadm03, oracle	Copy the database password file from the primary database node to one of the standby database nodes. Revisit section 8.2.6 steps 3 and 4 above if you do not have a password file at the primary. On the standby database node, place the password file into the \$ORACLE_HOME/dbs directory making sure the file name is orapw<ORACLE_SID> i.e., orapwEBSCDB1. This will be the node that will be used to instantiate the physical standby database using RMAN RESTORE FROM SERVICE.
2	Primary database node exaadb01, oracle	Determine if Transparent Data Encryption (TDE) is enabled and identify the TDE wallet location. To determine if TDE is enabled and where the TDE wallets are located, on one of the DB nodes of the primary database, log onto the primary database as SYS and issue the following query: SQL> select * from v\$encryption_wallet; You can also look at the sqlnet.ora file in \$TNS_ADMIN and find an entry: ENCRYPTION_WALLET_LOCATION as in the following example: ENCRYPTION_WALLET_LOCATION = (SOURCE= (METHOD=FILE) (METHOD_DATA= (DIRECTORY=/u01/app/oracle/admin/EBSCDB/wallet_root/tde))) If there are no rows returned from the view v\$encryption_wallet, then TDE is not enabled.
3	Primary database node exaadb01, oracle All standby database nodes, oracle	If TDE is not enabled per step 2, skip this step. Since the database oracle homes were copied over from the primary database nodes, the sqlnet.ora file that specifies the TDE wallet location should already be on the standby database nodes. The TNS_ADMIN for this step is defined in the <CDB_ORACLE_SID>_logical_hostname.env and NOT the <PDB_NAME>_<logical_hostname>.env. On our system, the environment file is: EBSCDB1_ebsdb1.env and TNS_ADMIN is defined as: TNS_ADMIN=/u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin NOTE: For purposes of consolidating several databases on Exadata, we have created a subdirectory EBSCDB under \$TNS_ADMIN and set TNS_ADMIN to point to /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin/EBSCDB for the CDB database. The autoconfig utility does not regenerate this sqlnet.ora file. Copy (transfer) the TDE wallets from the primary database to the standby database nodes. Place the wallets in the correct location as defined in the sqlnet.ora. Make sure that the permissions are the same as on the primary. For example, the location of the wallets on our environment is: /u01/app/oracle/admin/EBSCDB/wallet_root/tde and the wallets are:

		<pre>-rw----- 1 oracle oinstall 6955 Sep 9 15:07 ewallet.p12 -rw----- 1 oracle oinstall 5467 Sep 9 15:07 ewallet_2020090922073104.p12 -rw----- 1 oracle oinstall 7000 Sep 9 15:07 cwallet.sso -rw----- 1 oracle oinstall 0 Sep 10 10:21 ewallet.p12.lck -rw----- 1 oracle oinstall 0 Sep 10 10:21 cwallet.sso.lck</pre> <p>If you do not have AUTO_LOGIN configured in the keystore, then the cwallet files may not be present. It is recommended to have AUTO_LOGIN configured.</p> <p>NOTE: If your environment is configured with ACFS file system that is shared by all database nodes, then the wallets can be placed in this location. Please note that if ACFS is unavailable, then the database will also become unavailable.</p>
4	<p>Primary database node exaadb01, oracle</p> <p>First standby database node exabadm03, oracle</p>	<p>Create a parameter file (PFILE) from the primary database and copy it to the standby database node. It will be edited then used on the standby.</p> <p>On the primary database, create a PFILE from the SPFILE. Log onto SQL*Plus as SYS and issue the following command:</p> <pre>SQL> create pfile='ebscdb_pfile_2021016.ora' from spfile;</pre> <p>Now, copy the pfile to the first standby database node and place it into the \$ORACLE_HOME/dbs. This should be the same database node you copied the password file to.</p>
5	<p>First standby database node exabadm03, oracle</p>	<p>On the standby, edit the PFILE from step 4 above and change init.ora parameters to reflect what should be set on the standby, as needed. For example, db_create_file_dest, db_create_online_log_dest_1 on the primary were set to +DATAC1. On the standby for this project, they were set to +DATAC2 as this is the name of the ASM diskgroup at the standby. Make sure the following parameters are set per below:</p> <p>Make sure db_name is the same as the that of the primary and db_unique_name is set differently between the primary and standby.</p> <p>Here is a list of parameters that need to be set correctly at the standby:</p> <pre>db_create_file_dest db_create_online_log_dest_1 db_file_recovery_dest db_file_recovery_dest_size db_name db_unique_name audit_file_dest diagnostic_dest remote_listener</pre> <p>Make sure the parameter control_files is NOT set.</p> <p>Here are our entries for the parameters listed above:</p> <pre>*.db_create_file_dest='+DATAC2' <- ASM disk group at standby *.db_create_online_log_dest_1='+DATAC2' <- ASM disk group at standby *.db_recovery_file_dest='+RECO2' <- ASM disk group at standby *.db_recovery_file_dest_size=10737418240000 *.db_name='EBSCDB' <- Must be the same name as the primary *.db_unique_name='EBSCDB_osc5x1' <- Must be different than the primary *.audit_file_dest='/u01/app/oracle/admin/EBSCDB_osc5x1/adump' <- Make sure this directory exists *.diagnostic_dest='/u01/app/oracle' <- Make sure this directory exists. *.remote_listener='exab-scan1:1521' <- SCAN listener at the standby</pre>
6	<p>Each standby database node exabadm03, exabadm04, oracle</p>	<p>On all standby hosts create the audit directory for the standby database. As our database db_unique_name is set to EBSCDB_osc5x1, we created the subdirectory as follows:</p> <pre>\$ mkdir -p /u01/app/oracle/admin/EBSCDB_osc5x1/adump</pre>
7	<p>All database nodes, primary and standby exaadb01, exaabd02, exabadm03, exabadm04, oracle</p>	<p>Create TNS connect string aliases for use by products external to the E-Business Suite, such as RMAN and Data Guard Broker.</p> <p>To do this, on all primary and standby database nodes, add TNS connect string aliases to reach all other primary instances as well as all standby instances. The connect strings must use the SCAN listener and not the VIP address. Do not use logical host names for these connect strings.</p>

		<p>On our environment, on all Oracle RAC database nodes at both primary and standby sites, we added the following to the tnsnames.ora file for the CDB database. In our environment, this tnsnames.ora file is located at: /u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin/EBSCDB.</p> <p>The TNS aliases we added are:</p> <pre>EBSCDB_phx3xd = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = exaa-scan1)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = EBSCDB_phx3xd)))) EBSCDB_osc5x1 = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = exab-scan2)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = EBSCDB_osc5x1))))</pre> <p>NOTE: Autoconfig will not maintain or overwrite the tnsnames.ora in this directory.</p>
8	First standby database node exabadm03, oracle	<p>On the standby host with the password file, ensure the ORACLE_SID is set to that of the primary container database – in our case, EBSCDB1 - then startup NOMOUNT the standby instance. The PFILE we created and edited earlier for the standby is referenced here:</p> <pre>\$ cd \$ORACLE_HOME/dbs \$ sqlplus / as sysdba SQL> STARTUP NOMOUNT pfile=ebscdb_pfile_2021016.ora</pre>
10	All database nodes, primary and standby exaadb01, exaabd02, exabadm03, exabadm04, oracle	<p>Before proceeding, on the primary, test with SQL*Plus that you can connect to the instance now running at the standby:</p> <p>For example:</p> <pre>\$ sqlplus sys/<sys_password>@EBSCDB_osc5x1 as sysdba</pre> <p>NOTE: You may receive an error message that the instance is blocked. This is OK and expected behavior.</p> <p>Also test with SQL*Plus that you can successfully connect from the standby to the primary. For example:</p> <pre>\$ sqlplus sys/<sys_password>@EBSCDB_phx3xd as sysdba</pre>
11	One standby database node exabadm03, oracle	<p>As of Oracle 12c (12.1.0.2), a feature was introduced to RMAN that makes it much easier to instantiate a physical standby database. While you can still use RMAN DUPLICATE, using RESTORE FROM SERVICE is a much simpler process. RESTORE FROM SERVICE does the following:</p> <ul style="list-style-type: none"> • Copies the control file • Copies the database: CDB and all PDBs • Places the PDBs into the correct location in ASM using their GUIDs • Creates the temp tablespace temp files <p>Make sure you have set the following parameters:</p> <ul style="list-style-type: none"> • db_name MUST be set to the same name as the primary database • db_unique_name MUST be set to a name different from that of the primary database. <p>The db_unique_name parameter is used by RMAN to create the appropriate directory structures within ASM. Double-check those parameters in your environment. If you need to adjust the parameters,, shut down and restart the standby instance NOMOUNT as described above.</p>
12	One standby database node exabadm03, oracle	<p>On the standby database node where the instance is started NOMOUNT, create an RMAN script to instantiate the physical standby. Our RMAN script is below. Modify it for your environment.</p> <ul style="list-style-type: none"> • Change the TNS connect string alias name used in this script for your environment • Set the PARALLELISM option according to your environment capacity

		<p>and save the script. Here is our script, which we saved to rman_restore_from_service.sh:</p> <pre># Restores the standby database from the primary over the network. # You provide a TNS connect string alias pointing to the primary # for the restore from service in the below rman command. # Run the below rman command from the standby server side. time rman target sys/sys_passowd> nocatalog <<EOF! tee -a rman_restore_db_from_service.log set echo on restore standby controlfile from service 'EBSCDB_phx3xd'; alter database mount; CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' clear; CONFIGURE DEVICE TYPE 'SBT_TAPE' clear; CONFIGURE DEFAULT DEVICE TYPE TO DISK; CONFIGURE DEVICE TYPE DISK PARALLELISM 4; restore database from service 'EBSCDB_phx3xd' section size 64G; EOF! Usage notes: In the above example, the TNS alias for the primary is 'EBSCDB_phx3xd'. If your primary database is backed up to tape or Oracle Cloud object storage using device type SBT_TAPE, both the channel and device type must first be cleared, and the default device type must then be set to DISK as shown in the above script. Not doing so could cause the restore database process to fail. This is a workaround for a known issue and will be addressed in a later release.</pre>
13	First standby database node exabadm03, oracle	<p>Run the rman script.</p> <pre>./rman_restore_from_service.sh</pre> <p>Monitor the output as the restore progresses. If the restore fails, investigate the errors and correct accordingly. Some errors may be caused by:</p> <ul style="list-style-type: none"> • The password files are not the same between the primary and standby • The wallets are not accessible, or the location is not set correctly in the sqlnet.ora file • TNS connection failures i.e, unable to resolve the TNS alias, the services at the primary are not registered with the SCAN listener.
14	First standby database node exabadm03, Oracle (or Grid)	<p>Copy the password file to ASM.</p> <p>Note: If role separation has been implemented on the standby database servers, then these commands must be executed by the Grid owner.</p> <pre>\$ asmcmd -p --privilege sysdba ASMCMD [+] > cd +DATA ASMCMD [+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</pre> <p>On our system:</p> <pre>\$ asmcmd -p --privilege sysdba ASMCMD [+] > cd +DATAC2 ASMCMD [+DATAC2] >mkdir EBSCDB_OSC5XL/PASSWORD ASMCMD [+DATAC1] > pwcopy --dbuniquename EBSCDB_OSC5XL /u01/app/oracle/product/19.0.0.0/dbhome_2/dbs/orapwEBSCDB1 +DATAC2/EBSCDB_OSC5XL/PASSWORD/orapweboscdb ASMCMD [+DATAC2] > exit</pre>
15	First standby database node exabadm03, oracle	<p>Create the SPFILE from the PFILE on the standby.</p> <pre>SQL> create spfile='+DATAC2/EBSCDB_OSC5XL/spfileebscdb.ora' from pfile=' ebscdb_pfile_2021016.ora';</pre>

16	Each standby database node exabadm03,exabadm04, oracle	On all standby hosts, create an init<ORACLE_SID Number>.ora file that points to the SPFILE created in step 15. On our environment: spfile= '+DATAC2/EBSCDB_OSC5XL/spfileebscdb.ora '
17	First standby database node exabadm03, oracle	Shut the standby database instance down and restart it with STARTUP MOUNT. Do not attempt to open the database: sqlplus / as sysdba SQL> shutdown immediate SQL> startup mount ← Specifying a PFILE should no longer be required.
18	First standby database node exabadm03, oracle	Use the following SQL script to clear all online redo and standby logs. You may see errors in the alert.log that directory locations do not exist. As long as you have set db_create_online_log_dest_1 and any other db_create_online_log_dest_n parameters per the preceding sections, these errors can be ignored. Run the following script as SYS in SQL*Plus: set pagesize 0 feedback off linesize 120 trimspool on spool clearlogs.sql select distinct 'alter database clear logfile group ' group# ';' from v\$logfile; spool off @clearlogs.sql
19	Each standby database node exabadm03,exabadm04, oracle	Before proceeding to the next step, shut down, startup MOUNT (do not open), then again shut down each instance of the standby database, using SQL*Plus. This ensures that it will start on all nodes before registering the database and its instances with CRS. In our environment, on node exabadm03: 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 exabadm04: 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

8.3.3.4 Register the Standby with Cluster Ready Services

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

STEP	SERVER, USER	ACTION

1	First standby database node exabadm03, oracle	<p>Register the database with CRS and establish the environment variables required for EBS.</p> <p>The following commands register the EBSCDB_OSC5XL database and all instances into CRS.</p> <pre> srvctl add database -db EBSCDB_osc5x1 -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_2 - spfile +DATA2/EBSCDB_OSC5XL/spfileebscdb.ora -diskgroup "DATA2,RECO2" srvctl add instance -db EBSCDB_osc5x1 -instance EBSCDB1 -node exabadm03 srvctl add instance -db EBSCDB_osc5x1 -instance EBSCDB2 -node exabadm04 </pre> <p>The following SRVCTL commands set the required environment variables for EBS:</p> <pre> srvctl setenv database -db EBSCDB_osc5x1 -t "TNS_ADMIN=/u01/app/oracle/product/19.0.0.0/dbhome_2/network/admin/EBSCDB" srvctl setenv database -db EBSCDB_osc5x1 -t "ORA_NLS10=/u01/app/oracle/product/19.0.0.0/dbhome_2/nls/data/9idata" </pre> <p>Specify where the password file is located using SRVCTL:</p> <pre> srvctl modify database -db EBSCDB_osc5x1 -pwfile '+DATA2/EBSCDB_OSC5XL/PASSWORD/orapwebscdb' </pre>
2	First standby database node exabadm03, oracle	<p>Restart all standby instances, using the MOUNT option. On each standby database server:</p> <pre> srvctl start database -db EBSCDB_osc5x1 -startoption mount </pre>
3	First standby database node exabadm03, oracle	<p>Add the database services for Self-Service (OACORE), Forms, and Parallel Concurrent Processing (PCP). Self-Service uses the VISPRD_OACORE_ONLINE service. Forms uses the service VISPRD_FORMS_ONLINE. For parallel concurrent processing (PCP) we load-balance the concurrent managers across RAC instances with the service VISPRD_PCP_BATCH.</p> <p>For Self-Service (OACORE):</p> <pre> srvctl add service -db EBSCDB_osc5x1 -pdb VISPRD -service VISPRD_OACORE_ONLINE -preferred "EBSCDB1,EBSCDB2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 </pre> <p>For Forms:</p> <pre> srvctl add service -db EBSCDB_osc5x1 -pdb VISPRD -service VISPRD_FORMS_ONLINE -preferred "EBSCDB1,EBSCDB2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 </pre> <p>For Parallel Concurrent Processing:</p> <pre> srvctl add service -db EBSCDB_osc5x1 -pdb VISPRD -service VISPRD_PCP_BATCH -preferred "EBSCDB1,EBSCDB2" -notification TRUE -role "PRIMARY,SNAPSHOT_STANDBY" -failovermethod BASIC - failovertype SELECT -failoverretry 10 -failoverdelay 3 </pre>

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	<p>From each primary and each standby database instance test the connections to all other primary and standby database instances (on our environment, EBSCDB_PHX3XD and EBSCDB_OSC5XL). These TNS connect string aliases were described in step 8 of section Instantiate the Standby Database Using RMAN RESTORE FROM SERVICE.</p> <p>DO NOT PROCEED if any of the connection attempts fail. Fix the connection error before going further.</p> <p>On our environment, on each primary database node, we entered:</p>

		<pre>sqlplus sys/<sys password>@EBSCDB_PHX3XD as sysdba sqlplus sys/<sys password>@EBSCDB_OSC5XL as sysdba</pre> <p>On each standby database node:</p> <pre>sqlplus sys/<sys password>@EBSCDB_PHX3XD as sysdba sqlplus sys/<sys password>@EBSCDB_OSC5XL as sysdba</pre>
2	One primary and one standby database node exaadb01, exabadm03, oracle	<p>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.</p> <p>On the primary, in our environment, we ran:</p> <pre>alter system set dg_broker_config_file1='+DATAC1/EBSCDB_PHX3XD/dr1EBSCDB_phx3xd.dat' sid='*' scope=both; alter system set dg_broker_config_file2='+RECO1/EBSCDB_PHX3XD/dr2EBSCDB_phx3xd.dat' sid='*' scope=both; alter system set dg_broker_start=true sid='*' scope=both;</pre> <p>On the standby</p> <pre>alter system set dg_broker_config_file1='+DATAC2/EBSCDB_OSC5XL/dr1EBSCDB_osc5x1.dat' sid='*' scope=both; alter system set dg_broker_config_file2='+RECO2/EBSCDB_OSC5XL/dr2EBSCDB_osc5x1.dat' sid='*' scope=both; alter system set dg_broker_start=true sid='*' scope=both;</pre>
3	One primary database node exaadb01, oracle	<p>On a primary host, connect using DGMGRL and create the Data Guard configuration for the primary and standby databases:</p> <p>On our environment:</p> <pre>\$ dgmgrl sys/<sys password> DGMGRL> create configuration ebscdb_dg AS primary database IS 'EBSCDB_PHX3XD' CONNECT IDENTIFIER IS EBSCDB_phx3xd; DGMGRL> add database 'EBSCDB_OSC5XL' as connect identifier is EBSCDB_osc5x1; DGMGRL> enable configuration;</pre>
4	One primary database node exaadb01, oracle	<p>Verify that the configuration created successfully by using the SHOW CONFIGURATION command. On our environment:</p> <pre>DGMGRL> show configuration</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSCDB_PHX3XD - Primary database EBSCDB_OSC5XL - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 58 seconds ago)</pre>
5	All database nodes, primary and standby exaadb01, exaadb02, exabadm03, exabadm04, oracle	<p>Another step to validate the Data Guard configuration is to run the DGMGRL command VALIDATE DATABASE VERBOSE '<db_unique_name>'. This command can be helpful exposing any residual connection issues.</p> <p>Log onto each node at both the primary and standby site and issue the VALIDATE DATABASE command for the remote database. On our environment, at the primary on nodes exaadb03 and exaadb04, we logged onto DGMGRL and issued the following commands:</p>

		<pre>DGMGRL> VALIDATE DATABASE VERBOSE 'EBSCDB_PHX3XD'; DGMGRL> VALIDATE DATABASE VERBOSE 'EBSCDB_OSC5XL';</pre> <p>At the standby on nodes exabadm03 and exabadm04, we logged onto DGMGRL and issued the following commands:</p> <pre>DGMGRL> VALIDATE DATABASE VERBOSE 'EBSCDB_PHX3XD'; DGMGRL> VALIDATE DATABASE VERBOSE 'EBSCDB_OSC5XL';</pre> <p>Each of the above will generate a report if it can connect to both databases.</p>
6	Any one primary or standby database node, oracle	<p>In recent versions of the Oracle database, there have been enhancements to Data Guard Broker. One very useful enhancement is the “show configuration lag”. Similar to the “show configuration” command, the lag modifier will also show the redo transport and redo apply lag times in seconds.</p> <pre>DGMGRL> show configuration lag</pre> <pre>Configuration - ebscdb_dg Protection Mode: MaxPerformance Members: EBSCDB_PHX3XD - Primary database EBSCDB_OSC5XL - Physical standby database Transport Lag: 6 seconds (computed 3 seconds ago) Apply Lag: 10 seconds (computed 3 seconds ago) Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 51 seconds ago)</pre>
7	One primary database node, one standby database node oracle	<p>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, paying particular attention to any processes you may have that modify LONG data types.</p> <p>It is an MAA recommendation to enable flashback database on both the primary and standby. It is particularly beneficial to enable flashback database at the standby.</p> <p>To enable flashback database on the primary:</p> <pre>SQL> alter database flashback on;</pre> <p>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:</p> <pre>SQL> recover managed standby database cancel; SQL> alter database flashback on; SQL> recover managed standby database disconnect using current logfile;</pre> <p>The above steps can also be accomplished using the broker with the following commands. On our environment:</p> <pre>DGMGRL> CONNECT sys/<password>@EBSCDB_OSC5XL DGMGRL> EDIT DATABASE 'EBSCDB_OSC5XL' SET STATE='APPLY-OFF'; DGMGRL> SQL 'ALTER DATABASE FLASHBACK ON'; DGMGRL> EDIT DATABASE 'EBSCDB_OSC5XL' SET STATE='APPLY-ON';</pre>
8	Any primary or standby database node, oracle	<p>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 there is no online activity and the concurrent queues are drained, then issue:</p> <pre>DGMGR> switchover to 'EBSCDB_OSC5XL';</pre> <p>To switch back:</p> <pre>DGMGR> switchover to 'EBSCDB_PHX3XD';</pre> <p>Note: This disrupts services at the primary and should be done at a scheduled maintenance outage window.</p>

8.3.4 Set Up Standby Application Tier

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

- Make sure the servers to be used for the standby application tier are ready.
- Set up the OS groups and user accounts with the same names and privileges as used on the primary application tier servers.
- 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 on each application tier server is at least one-half the size of the total system memory or at a minimum of 8GB. If not, follow the specific Linux procedures for increasing or extending the swap space.

8.3.4.2 Add the Application Owner OS User Account

Add the application owner OS user account (“applmgr”) to the application tier servers 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, applmgr	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: <pre>\$ id uid=54322(applmgr) gid=54321(oinstall) groups=54321(oinstall)</pre>
2	Each standby application tier node pca1vm55, pca1vm56, 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: <pre># Add groups \$ groupadd -g 54321 oinstall # Create the applmgr user account \$ useradd --uid 54322 -g oinstall applmgr \$ usermod -d /home/applmgr -g oinstall -s /bin/bash applmgr</pre>

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 ZFS replication 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.

To configure continuous replication from the primary site to the standby, follow the steps at the following documentation link:

https://docs.oracle.com/cd/F13758_01/html/F13769/gokub.html#scrolltoc

When selecting the frequency of replication, select the Continuous radio button option. Once the action is enabled, you should see a status bar showing that replication has started. Wait until the target ZFS has been fully synchronized with the primary before testing the application at the standby site.

8.4 Disaster Recovery Site Test using Snapshots

You can start the E-Business Suite at your standby environment to test major changes in your environment 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 roll back any local changes then 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 BIG IP 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.

- Convert the physical standby database into a snapshot standby database.
- Create a snapshot clone of the shared file system being replicated to the DR site.
- 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 exabadm03, oracle	Using Data Guard Broker (DGMRGL), check the current configuration.

		<pre> dgmgrl sys/<sys password> DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Dec 12 15:25:26 2020 Version 19.8.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for information. Connected to "EBSCDB_osc5x1" Connected as SYSDBA. DGMGRL> show configuration lag Configuration - ebscdb_dg Protection Mode: MaxPerformance Members: EBSCDB_PHX3XD - Primary database EBSCDB_OSC5XL - Physical standby database Transport Lag: 4 seconds (computed 0 seconds ago) Apply Lag: 9 seconds (computed 0 seconds ago) Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 41 seconds ago) DGMGRL> validate database 'EBSCDB_OSC5XL'; Database Role: Physical standby database Primary Database: EBSCDB_PHX3XD Ready for Switchover: Yes Ready for Failover: Yes (Primary Running) Managed by Clusterware: EBSCDB_OSC5XL: YES EBSCDB_PHX3XD: YES Standby Apply-Related Information: Apply State: Running Apply Lag: 6 seconds (computed 2 seconds ago) Apply Delay: 0 minutes </pre>
2	One primary or standby database node, oracle	<p>Only if the SHOW CONFIGURATION status shows SUCCESS and “Ready for Switchover” says Yes, then convert the physical standby to a snapshot standby.</p> <pre> DGMGRL> convert database 'EBSCDB_OSC5XL' to snapshot standby; Converting database "EBSCDB_OSC5XL" to a Snapshot Standby database, please wait... Database "EBSCDB_OSC5XL" converted successfully </pre>

8.4.2 Create a Snapshot Clone of the Shared Application Tier 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. Please note that the steps in the below table refer to using the ZFS Appliance web based user interface, which does not meet accessibility guidelines and therefore is “Not Accessible”.

STEP	SERVER, USER	ACTION
1	Standby ZFS appliance: pca1zfs, root or administrator	Log on to the secondary ZFS administrative console using a browser. On our environment, we use this URL: https://pca1zfs:215
2	Standby ZFS appliance:	a) Click on Shares

	pca1zfs, root or administrator	b) Click on Projects on the left and the left pane shows c) Click REPLICA
3	Standby ZFS appliance: pca1zfs, 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: pca1zfs, 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: pca1zfs, root or administrator	You will be prompted for a new clone project name. Because this is site PCA1, in our case we called it "EBS_PCA1_SNAPSHOT". Then, click Apply.
6	Standby ZFS appliance: pca1zfs, root or administrator	Once the above step has completed: a) Click LOCAL on the left pane and find the newly created project, our example: MAA_EBS_SNAPSHOT. b) Click on the new share. The new share should show up on the right side.
7	Standby ZFS appliance: pca1zfs, 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 greyed out, indicating it now possible to change the items in the next few steps.
8	Standby ZFS appliance: pca1zfs, root or administrator	Change the Mount point to something like: /export/ebs_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 pca1vm55, pca1vm56, 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 the directory structures the same on the standby as on the primary. Example: ### Snapshot used for testing. pca1zfs:/export/ebs_snapshot /u02 nfs rw,rsize=131072,wsiz=131072,bg,hard,noacl,timeo=600,retrans=20
10	All standby application tier nodes pca1vm55, pca1vm56, root	Mount the file system on all the application tier nodes. In our case, its /u02. \$ mount /u02

8.4.3 Set Up OS User Equivalency at the Application Tier

There are two tasks that must be done once only in a persistent way, to finalize setup of the secondary site. The first task, in steps one and two, are done at the OS level and will persist beyond this test. The second task is in the EBS directory structure and, to persist, must be done during the first true switchover or failover event. This section is in the flow describing site testing via snapshot standby, where the change to the EBS directory structure will not persist.

We present both steps here in case this is the first time the secondary site is being configured.

STEP	SERVER, USER	ACTION
1	Each standby application tier node, applmgr	<p>Oracle E-Business Suite tooling, particularly AD Online Patching (adop), requires that user equivalency be established on all application tier nodes.</p> <p>Run the <code>txkRunSSHSetup.pl enablessh</code> command using the logical host names on all standby application tier nodes, changing the context path and file name to match your environment.</p> <p>From <code>pca1vm55 (ebsapp1)</code></p> <pre>\$ per1 /u02/app/ebs/visprd/fs2/EBSapps/app1/ad/12.0.0/patch/115/bin/txkRunSSHSetup.pl enablessh \ \ -contextfile=/u02/app/ebs/visprd/fs2/inst/apps/ \ VISPRD_ebsapp1/app1/admin/VISPRD_ebsapp1.xml \ -hosts=ebsapp1,ebsapp2</pre> <p>From <code>pca1vm56 (ebsapp2)</code></p> <pre>\$ per1 /u06/app/ebs/vis/fs2/EBSapps/app1/ad/12.0.0/patch/115/bin/txkRunSSHSetup.pl enablessh \ -contextfile=/u02/app/ebs/visprd/fs2/inst/apps/ \ VISPRD_ebsapp2/app1/admin/VISPRD_ebsapp2.xml \ -hosts=ebsapp1,ebsapp2</pre>
2	Each standby application tier node, applmgr	<p>On each standby node, test that SSH user equivalency has been established.</p> <p>From each node, test using logical host names against all of the other nodes using a simple command such as in this example, substituting your target node.</p> <pre>\$ ssh applmgr@ebsapp2 "date"</pre> <p>The command should return the date and, importantly, you should not be prompted for the password.</p>
3	All standby application tier nodes, applmgr	<p>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 <code>node_info.txt</code> file. This file is cumulative, with the new host information appended to the file.</p> <p>To add the standby site's application tier physical host names to the <code>node_info.txt</code> file, you need to run <code>AutoConfig</code> without sourcing the environment on both the <code>RUN</code> and <code>PATCH</code> 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 (e.g., not via snapshot standby). Until it is done persistently, these steps will be required each time testing is done via snapshot clones.</p> <p>On each application tier node, in both the <code>RUN</code> and the <code>PATCH</code> file systems:</p> <p>Go to the directory that would be pointed to by <code>\$INST_TOP</code>:</p> <pre>/<your base directory>/<run file system>/inst/admin/scripts</pre> <p>For example, on our system (fs2 is the <code>RUN</code> file system):</p> <pre>/u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/admin/scripts</pre> <p>Execute <code>AutoConfig</code> from this location:</p> <pre>./adautocfg.sh</pre> <p>Now, source the environment.</p> <p>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).</p>

8.4.4 Configure the Web Entry URLs and Test Application Services

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 (pca1vm55, pca1vm56), applmgr	Source the RUN file system.
2	All standby application tier nodes (pca1vm55, pca1vm56), applmgr	<p>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.</p> <p>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.</p> <pre> # 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. # # NOTE: For enabling SSL terminated at the load balancer, you MUST edit the context file and # and remove the '#' in the context variable: s_enable_sslterminator. # Change the context variable s_webentryurlprotocol from http to https. if [-f /u02/app/ebs/visprd/EBSapps.env]; then . /u02/app/ebs/visprd/EBSapps.env run else echo "EBS environment not available to source. Please check that /u02/app/ebs/visprd/EBSapps.env exists and the file system is mounted." exit -1 fi # 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 ebsnapdr \$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 cloud.osc.oracle.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://ebsnapdr.cloud.osc.oracle.com:8000/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://ebsnapdr.cloud.osc.oracle.com:8000 </pre>
3	All standby application tier nodes (pca1vm55, pca1vm56), applmgr	<p>Run AutoConfig</p> <pre> \$ cd \$ADMIN_SCRIPTS_HOME \$ adautocfg.sh </pre>

4	All standby application tier nodes (pca1vm55, pca1vm56), applmgr	<p>Follow the solution in OPMN Fails to Start and Says to Check Adopmnc1.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.</p> <p>There will be a subdirectory EBS_web_VISPRD_OHSx for each application tier node, where x corresponds to a given application tier node. For example, EBS_web_VISPRD_OHS1 would be for the master application tier node, EBS_web_VISPRD_OHS2 for the second application tier node, and so on. The state data must be moved for each active application tier node.</p> <p>On our system, we did the following:</p> <pre>\$ cd /u02/app/ebs/visprd/fs1/FMW_Home/webtier/instances/EBS_web_VISPRD_OHS1/config/OPMN/opmn \$ mv states states_backup</pre>
5	All standby application tier nodes (pca1vm55, pca1vm56), applmgr	<p>Run your tests of the secondary site.</p> <p>Start the application tier services on all nodes.</p> <pre>\$ adstrtal.sh</pre> <p>You should now be able to log into EBS Application and conduct any testing.</p>
6	All standby application tier nodes (pca1vm55, pca1vm56), applmgr	<p>When you are done testing, shut down the application services, unmount and drop the ZFS snapshot.</p> <p>Using DGMGRL, convert the database back to a PHYSICAL STANDBY.</p>

8.5 Test Switchover

These are the high level steps to perform a switchover:

- Drain the Concurrent Request queues. Shut down the EBS application services at the primary site.
- Switch roles for the database using Data Guard Broker.
- Reverse the replication direction on the ZFS appliance.
- 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	<p>Shut down the EBS application services.</p> <p>Note: You can to shut down the concurrent manager processes ahead of your scheduled switchover in order to drain the queues.</p> <p>To shut down all services, log onto each application tier node as applmgr, source the Run environment, then issue:</p> <pre>\$ cd \$ADMIN_SCRIPTS_HOME \$ adstpall.sh</pre>

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 lag

Configuration - ebscdb_dg

Protection Mode: MaxPerformance
Members:
EBSCDB_PHX3XD - Primary database
EBSCDB_OSC5XL - Physical standby database
Transport Lag:      0 seconds (computed 3 seconds ago)
Apply Lag:          6 seconds (computed 3 seconds ago)

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 47 seconds ago)

DGMGRL> show database 'EBSCDB_OSC5XL';

Database - EBSCDB_OSC5XL

Role:                PHYSICAL STANDBY
Intended State:      APPLY-ON
Transport Lag:       0 seconds (computed 4 seconds ago)
Apply Lag:           8 seconds (computed 4 seconds ago)
Average Apply Rate: 306.00 KByte/s
Real Time Query:    ON
Instance(s):
EBSCDB1
EBSCDB2 (apply instance)

Database Status:
SUCCESS

DGMGRL> validate database 'EBSCDB_OSC5XL';

Database Role:      Physical standby database
Primary Database:  EBSCDB_PHX3XD

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

Managed by Clusterware:
EBSCDB_PHX3XD: YES
EBSCDB_OSC5XL: YES

Standby Apply-Related Information:
Apply State:       Running
Apply Lag:         9 seconds (computed 4 seconds ago)
Apply Delay:       0 minutes

DGMGRL> switchover to 'EBSCDB_OSC5XL';
Performing switchover NOW, please wait...
Operation requires a connection to database "EBSCDB_OSC5XL"
Connecting ...
Connected to "EBSCDB_osc5x1"
Connected as SYSDBA.
New primary database "EBSCDB_OSC5XL" is opening...
Oracle Clusterware is restarting database "EBSCDB_PHX3XD" ...
Connected to "EBSCDB_phx3xd"
Connected to "EBSCDB_phx3xd"
Switchover succeeded, new primary is "EBSCDB_OSC5XL"

NOTE: Right after the switchover by Data Guard Broker is complete, if the "show configuration;" is executed, you may see an error: ORA-16525: The Oracle Data Guard broker is not yet available. Wait a few minutes as this will clear:
```

```

DGMGRL> show configuration

Configuration - ebscdb_dg

Protection Mode: MaxPerformance
Members:
EBSCDB_OSC5XL - Primary database
EBSCDB_PHX3XD - Physical standby database
Error: ORA-16525: The Oracle Data Guard broker is not yet available.

Fast-Start Failover: Disabled

Configuration Status:
ERROR (status updated 46 seconds ago)

### Wait a few minutes.

DGMGRL> show configuration lag;

Configuration - ebscdb_dg

Protection Mode: MaxPerformance
Members:
EBSCDB_OSC5XL - Primary database
EBSCDB_PHX3XD - Physical standby database
Transport Lag: 0 seconds (computed 2 seconds ago)
Apply Lag: 0 seconds (computed 2 seconds ago)

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 65 seconds ago)

```

8.5.3 Reverse ZFS Replication Direction

To reverse ZFS replication direction also called role reversal, please follow the steps in the ZFS documentation at: https://docs.oracle.com/cd/F13758_01/html/F13769/gppni.html#scrolltoc. Before you switch replication direction, make sure you first unmount the NFS file system as root on all concurrent primary application tiers:

```
# /umount /u02
```

Once the ZFS replication has been reversed so that the standby ZFS is now the primary, follow the steps in the next sections.

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	<p>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):</p> <pre>Pca3zfs.cloud.osc.oracle.com:/export/ebs /u02 nfs _netdev,hard,intr,noatime,rsize=32768,wsiz=32768 0 0</pre> <p>On our NEW primary (old standby):</p> <pre>pca1zfs.cloud.osc.oracle.com:/export/ebs /u02 nfs _netdev,hard,intr,noatime,rsize=32768,wsiz=32768 0 0</pre>
2	All NEW primary	Now mount the /u02 NFS mount point on all application tier nodes at the NEW primary site.

	application tier nodes, root	\$ mount /u02
3	All NEW primary application tier nodes, applmgr	<p>Log on to each application tier node and source the RUN file system.</p> <p>You can add the EBSapps.env into your .bash_profile so that it will source automatically when you log on. For example:</p> <pre># Set the apps env. if [-f /u02/app/ebs/visprd/EBSapps.env]; then . /u02/app/ebs/visprd/EBSapps.env fi</pre>
4	All NEW primary application tier nodes, applmgr	<p>If this is the FIRST time these applications these application tiers have assumed the primary role, then do THIS step. Otherwise, skip to step 5 below.</p> <p>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.</p> <p>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.</p> <p>On each application tier node, in both the RUN and the PATCH file systems:</p> <p>Go to the directory that would be pointed to by \$INST_TOP:</p> <pre><your base directory>/<run file system>/inst/admin/scripts</pre> <p>For example, on our system (fs2 is the run file system):</p> <pre>/u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/admin/scripts</pre> <p>Execute AutoConfig from this location:</p> <pre>./adautocfg.sh</pre> <p>Source the environment.</p> <p>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.</p> <p>You can skip the next step, as AutoConfig was run in this step.</p>
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 – pca1vm55, pca1vm56, applmgr	<p>Start the EBS application services. Start with the primary application server first as this will start the WLS administration server.</p> <pre>\$ adstrta1.sh</pre> <p>Check for errors.</p>
7	At the new primary, on the primary application tier server pca1vm55, applmgr	Perform an adop validation, which will check both RUN and PATCH file systems on all nodes.

```

$ adop -validate
...
Validating credentials.

Initializing.
  Run Edition context :
/u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/app1/admin/VISPRD_ebsapp1.xml
  Patch edition context:
/u02/app/ebs/visprd/fs1/inst/apps/VISPRD_ebsapp1/app1/admin/VISPRD_ebsapp1.xml
Node registry is valid.

=====
ADOP (C.Delta.12)
Node: ebsapp1
Command: validate
Log: /u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/adopConsole.log
=====

Checking for existing patching cycle.
  No existing patching cycle exists

Verifying SSH connection to all nodes.
  Log: /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/logs/app1/rgf/TXK/verifyssh.log
  Output: /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/logs/app1/rgf/TXK/out.xml
  Remote execution is operational.

Running adop validations on Admin node: [ebsapp1].
  Log:
ebsapp1:/u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/ebsapp1
  Output:
/u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/remote_execution_result_level1.xml
      txkADOPEvalSrvStatus.pl returned SUCCESS

Running adop validations on node(s): [ebsapp2].
  Log: ebsapp2:
/u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/ebsapp2
  Output:
/u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/remote_execution_result_level2.xml
      txkADOPEvalSrvStatus.pl returned SUCCESS

adop exiting with status = 0 (Success)

```

8	Web, OAM	Log into the EBS application 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 (now primary) database to the old primary. Instead, we 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 re-instantiate 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:

- Shut down all application services.
- Unmount and drop the file system snapshot clone.
- Using Data Guard Broker, convert the SNAPSHOT STANDBY database back to a PHYSICAL STANDBY database. All outstanding archive logs at the secondary site will then be applied to the physical standby database to complete recovery and minimize data loss.

Perform the failover per the following steps.

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

- Perform a Data Guard failover (not switchover).
- Perform a role reversal of the ZFS filer at the standby to assume the primary role.
- Perform state cleanup.
- 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 -db EBSCDB_OSC5XL -stopoption abort
```

8.6.1 Perform a Data Guard Failover

STEP	SERVER, USER	ACTION
1	One standby database node, oracle	<p>Perform a failover using Data Guard Broker on one database node at the secondary site.</p> <p>The SHOW CONFIGURATION command issued in Data Guard Broker should show errors indicating that the primary is not available. For example:</p> <pre>DGMGRL> show configuration</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSCDB_OSC5XL - Primary database Error: ORA-12514: TNS:listener does not currently know of service requested in connect descriptor EBSCDB_PHX3XD - Physical standby database Fast-Start Failover: Disabled Configuration Status: ERROR (status updated 0 seconds ago) DGMGRL> failover to 'EBSCDB_PHX3XD'; Performing failover NOW, please wait... Failover succeeded, new primary is "EBSCDB_PHX3XD"</pre> <p>NOTE: In an actual failure event where the primary is lost or experiences a complete failure, you may see other errors instead of the one shown above.</p>

8.6.2 Perform a ZFS Role Reversal

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

ST EP	SERVER, USER	ACTION
1	DR site ZFS, administrator	To perform a ZFS role reversal, follow the steps in the ZFS documentation at: https://docs.oracle.com/cd/F13758_01/html/F13769/gppni.html#scrolltoc If the primary site is not available, you may be required to “sever” the replication relationship.
3	Each DR site application tier server pca3vm55, pca3vm56, root	In /etc/fstab, ensure the mount point can be mounted, then mount the NFS file system. <pre>pca3zfs.cloud.osc.oracle.com:/export/ebs /u02 nfs _netdev,hard,intr,noatime,rsize=32768,wsiz=32768 0 0</pre> <p>\$ mount /u02</p>

8.6.3 Perform State Cleanup

ST EP	SERVER, USER	ACTION
1	Each DR site application tier server pca3vm55, pca3vm56, applmgr	<p>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.</p> <p>Each application tier node has a subdirectory EBS_web_VIS_OHSx, where x corresponds to the application tier node. For example, EBS_web_VISPRD_OHS1 is for the master application tier node, EBS_web_VISPRD_OHS2 for the second application tier node, and so on.</p> <p>NOTE: 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.</p> <p>For example:</p> <pre>\$ cd /u02/app/ebs/visprd/fs2/FMW_Home/webtier/instances/EBS_web_VISPRD_OHS1/config/OPMN/opmn \$ mv states states_backup</pre> <p>For further details on the internal error and the state cache file, see MOS note 2542082.1.</p>
2	DR site application tier servers pca3vm55, pca3vm56, applmgr	<p>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.</p>
3	All NEW primary application tier nodes, applmgr	<p>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. If you have completed a switchover as described above prior to performing a failover, this step should not be necessary.</p> <p>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.</p> <p>On each application tier node, in both the RUN and the PATCH file systems:</p> <p>Go to the directory that would be pointed to by \$INST_TOP:</p> <pre><your base directory>/<run file system>/inst/admin/scripts</pre> <p>For example, on our system (fs2 is the run file system):</p> <pre>/u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/admin/scripts</pre> <p>Execute AutoConfig from this location:</p> <pre>./adautocfg.sh</pre> <p>Source the environment.</p>

		As the replication has been reversed, the updated node_info.txt file will be replicated to the new standby (old primary) 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 pca3vm55, pca3vm56, 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 application tier servers pca3vm55, pca3vm56, applmgr	Start the EBS application services on the master application tier node, then on all slave nodes. On each node: <code>\$ adstrta1.sh</code> If there are errors, check the startup logs for issues.
2	DR site primary application tier server pca3vm55, applmgr	To validate that the RUN and PATCH file systems on all nodes check out, perform an adop validate.

		<pre> \$ adop -validate ... Validating credentials. Initializing. Run Edition context : /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/app1/admin/VISPRD_ebsapp1.xml Patch edition context: /u02/app/ebs/visprd/fs1/inst/apps/VISPRD_ebsapp1/app1/admin/VISPRD_ebsapp1.xml Node registry is valid. ===== ADOP (C.Delta.12) Node: ebsapp1 Command: validate Log: /u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/adopConsole.log ===== Checking for existing patching cycle. No existing patching cycle exists Verifying SSH connection to all nodes. Log: /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/logs/app1/rgf/TXK/verifyssh.log Output: /u02/app/ebs/visprd/fs2/inst/apps/VISPRD_ebsapp1/logs/app1/rgf/TXK/out.xml Remote execution is operational. Running adop validations on Admin node: [ebsapp1]. Log: ebsapp1:/u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/ebsapp1 Output: /u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/remote_execution_result_level1.xml txkADOPevalSrvStatus.pl returned SUCCESS Running adop validations on node(s): [ebsapp2]. Log: ebsapp2: /u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/ebsapp2 Output: /u02/app/ebs/visprd/fs_ne/EBSapps/log/adop/101/20201212_171415/validate/remote_execution_result_level2.xml txkADOPevalSrvStatus.pl returned SUCCESS adop exiting with status = 0 (Success) </pre>
3	Browser logged on to application (OAM) at DR site	<p>If all the services started, log into the EBS application to check that all administrative tasks and functions are running properly.</p> <p>Your new primary site should be up and ready for production workloads.</p>

8.6.5 Reinstate Old Primary as New Standby

When the original primary site is available (assuming that no infrastructure was destroyed and the database is still intact), the database can be reinstated as the new standby. Flashback Database must have been enabled on this database prior to when the database was no longer available. If Flashback Database was not enabled, then use the steps above under Instantiating the Physical Standby Using RMAN RESTORE FROM SERVICE to instantiate the standby database at the old primary site.

In the following steps, we show how Data Guard Broker can be used to reinstate the old primary database as the new standby.

STEP	SERVER, USER	ACTION
1	One database	Start the old primary database using srvctl and monitor the alert.log files. Data Guard will notice that a failover has occurred and that this database is no longer the primary, and prevents it from opening. It will be in a mounted state.

	node on OLD primary, oracle	<pre>\$ srvctl start database -db ebscdb_osc5x1</pre>
2	One database node on NEW primary, oracle	<p>On one of the database nodes at the new primary, log into Data Guard Broker and execute “show configuration”. You will see that the standby needs to be reinstated:</p> <pre>DGMGRL> show configuration</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSCDB_PHX3XD - Primary database EBSCDB_OSC5XL - Physical standby database (disabled) ORA-16661: the standby database needs to be reinstated</pre> <p>Fast-Start Failover: Disabled</p> <pre>Configuration Status: SUCCESS (status updated 36 seconds ago)</pre>
3	One database node on NEW primary, oracle	<p>Reinstate the standby database:</p> <pre>DGMGRL> reinstate database 'EBSCDB_OSC5XL'; Reinstating database "EBSCDB_OSC5XL", please wait... Reinstatement of database "EBSCDB_OSC5XL" succeeded</pre>
4	Any database node at either primary or standby, oracle	<p>During the initial few minutes after Data Guard Broker completes the reinstatement process, you may see errors. After a few minutes, you should then be able to see the Data Guard configuration:</p> <pre>DGMGRL> show configuration lag;</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSCDB_PHX3XD - Primary database EBSCDB_OSC5XL - Physical standby database Transport Lag: 0 seconds (computed 2 seconds ago) Apply Lag: 0 seconds (computed 2 seconds ago)</pre> <p>Fast-Start Failover: Disabled</p> <pre>Configuration Status: SUCCESS (status updated 58 seconds ago)</pre>

8.7 Site Guard

All the steps described above to perform full site switchover and failover can be orchestrated and fully automated with Site Guard. Site Guard is an Enterprise Manager plug-in and operates under the Enterprise Manager framework. As such, Site Guard can be deployed into an existing Enterprise Management installation.

Site Guard already has the capability to manage databases using Data Guard and ZFS storage role reversals. The steps above for implementing Data Guard Broker and ZFS replication role transitions can be automated by Site Guard out of the box. The only required prerequisite to use Site Guard is to provide scripts that stop, start, configure or otherwise manage the application tier. We have supplied the scripts used for EBS R12.2 in [Appendix F: Scripts Used for Site Guard](#). Our project implemented Site Guard with these scripts.

Once implemented, the entire full-stack switchover or failover can be accomplished with a push of a button. Site Guard uses “plans” to implement the various switchover and failover scenarios. These plans allow some of the steps to be executed in parallel, such as shutting down the application servers on EBS application tier nodes, running autoconfig in parallel, etc.

For more information about Site Guard, see <https://www.oracle.com/database/technologies/high-availability/site-guard.html>

9 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 (Release Updates), 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:

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

9.1.1 Prepare for Database Patching

1. Download the quarterly database 19c Release Update (RU) and any critical database patches. See [Exadata Database Machine and Exadata Storage Server Supported Versions \(Doc ID 888828.1\)](#) for Exadata Database Machines; [Contents of each SuperCluster Quarterly Full Stack Download Patch \(QFSDP\) \(Doc ID 2056975.1\)](#).
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 release update 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. Some patch READMEs may state that the patch is "non-Data Guard Standby-First". If such a patch is already installed at the primary database, you may proceed as if the patch is "Standby-First".
5. Back up or zip up the primary and standby database Oracle homes to be patched before starting.

9.1.2 Patch One Standby Database Instance Home

At the RAC standby database, pick one instance (and oracle home) to be patched. Once patched, this will be the "gold image" to be used to patch all other oracle homes. We leave the remaining instances running at the standby 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 release update to the database homes. Patching one oracle home will create a “gold image” to be used for the rest of the environment.

STEP	SERVER, USER	ACTION
1	One standby database node – exabadm01, oracle	Shut down the standby database instance to be patched first of the RAC standby database. Note that redo will still be received and apply while the first instance is being patched. <pre>srvctl stop instance -db EBSCDB_PHX3XD -instance EBSCDB1 -o immediate</pre>
2	One standby database node – exabadm01, oracle	Follow the instructions in MOS Doc 1392527.1 to apply the release update. First, identify the list of patches that conflict with the release update you are applying. Some patches may require using opatch to be manually rolled back. The opatch tool can automatically roll back other patches. Follow the instructions in the release update README for opatch command examples to checking for patch conflicts, and applying the release update to the database home. For example, to check for patch conflicts with the 19.9 release update, the opatch command on our system is: <pre>\$ORACLE_HOME/OPatch/opatch prereq CheckConflictAgainstOHWithDetail -phBaseDir /u01/app/oracle/patchdepot/19c_RU_199000_GI/31750108/31771877</pre> When rolling back patches, use the -local command line parameter with opatch. For example: <pre>opatch rollback -id 29867728 -local</pre>
3	One standby database node – exabadm01, root	Once all required patches have been rolled back and opatch reports no patch conflicts, apply the quarterly release update (DB RU) to the oracle home. Our example is based on the 19.9 quarterly Release Update. The “opatchauto” is used from the database home and must be run as root. In our system, we execute the following command: <pre># export PATH=\$PATH:/u01/app/oracle/product/19.0.0.0/dbhome_2/OPatch</pre> <pre># opatchauto apply /u01/app/oracle/patchdepot/19c_RU_199000_GI/31750108 -oh /u01/app/oracle/product/19.0.0.0/dbhome_2</pre> DO NOT run datapatch yet, this step will be done later.
4	One standby database node – exabadm01, oracle	Apply the corresponding OJVM patch for the Release Update to the same database oracle home. Follow the instructions in the OJVM README to apply the patch but do NOT run datapatch. This will be done in a later step. To apply the OJVM patch to the oracle home, move to the location where the OJVM patch is located (unzip the patch if necessary) and ensure the ORACLE_HOME environment variable points to the EBS database oracle home to be patched: <pre>\$ cd /u01/app/oracle/patchdepot/19c_RU_199000_plus_EBS_patches/31668882</pre> <pre>\$ opatch apply -local</pre> Now, check the patches that have been applied thus far: <pre>\$ opatch lspatches</pre> <pre>31668882;OJVM RELEASE UPDATE: 19.9.0.0.201020 (31668882)</pre> <pre>31772784;OCW RELEASE UPDATE 19.9.0.0.0 (31772784)</pre> <pre>31771877;Database Release Update : 19.9.0.0.201020 (31771877)</pre> <pre>30432118;MERGE REQUEST ON TOP OF 19.0.0.0.0 FOR BUGS 28852325 29997937</pre> <pre>29997959;DSTV34 UPDATE - TZDATA2019B - NEED OJVM FIX</pre>
5	One standby database node – exabadm01, oracle	If there are any additional merge patches (MLRs) that need to be applied such as those for Exadata, apply them now per their README instructions. This step does NOT include those patches listed in MOS Doc 1392527.1.
6	One standby database node – exabadm01, oracle	Following MOS Doc 1392527.1, now apply the required one-off patches in the list shown in the table for the Release Update and the EBS version you are on. We recommend that you download the consolidated patch ZIP file that contains the patches you will need to apply. Ensure your environment is set and that the ORACLE_HOME environment variable points to the EBS database oracle home to be patched. Apply each patch ensuring you use the -local command line parameter for opatch. DO NOT run datapatch per the README instructions for any of these patches as this will be done in a later step. For example, to apply patch 28698087 on top of the 19.9.0.0 Release Update for example, move to the directory location of the patch, and apply it per the README instructions:

		<pre>\$ cd /u01/app/oracle/patchdepot/19c_RU_199000_plus_EBS_patches/etcc-bundle/LINUX_X86-64/database/19.9.0.0.201020DBRU/28698087</pre> <pre>\$ opatch apply -local</pre> <p>NOTE: When applying these patches, some patches may already be present and opatch will indicate so. Move on to the next patch.</p>
7	One standby database node – exabadm01, oracle	<p>Once all patches have been applied, run the opatch lspatches command to view the list of patches as follows:</p> <pre>\$ opatch lspatches</pre> <pre>31859285;MERGE ON DATABASE RU 19.9.0.0.0 OF 29867728 30222669 30972817 30998035</pre> <pre>31404014;XMLSEQUENCE /.. ISSUES</pre> <pre>31142749;QUERY ON ALL_ARGUMENTS SLOW IN 19.6 PDB</pre> <pre>30518349;STRESS FA PRC ORA-01000 SELECT DOCID FROM DR#IX_TS0481\$U , DR#IX_TS0481\$K...</pre> <pre>30336383;19C UPGRADE XOQU121.SQL - MISSING - ON IF SQLCODE STATEMENT</pre> <pre>28698087;SUPPORT CASE SENSITIVE PDB NAME</pre> <pre>32187518;MERGE ON DATABASE RU 19.9.0.0.0 OF 31408636 31991705 32017301</pre> <pre>31668882;OJVM RELEASE UPDATE: 19.9.0.0.201020 (31668882)</pre> <pre>31772784;OCW RELEASE UPDATE 19.9.0.0.0 (31772784)</pre> <pre>31771877;Database Release Update : 19.9.0.0.201020 (31771877)</pre> <pre>30432118;MERGE REQUEST ON TOP OF 19.0.0.0.0 FOR BUGS 28852325 29997937</pre> <pre>29997959;DSTV34 UPDATE - TZDATA2019B - NEED OJVM FIX</pre> <p>OPatch succeeded.</p>
8	One standby database node – exabadm01, oracle	<p>As instructed by MOS Doc 1392527.1, run the EBS Technology Code Checker (ETCC) tool ONLY on the now patched oracle home. DO NOT start the database instance. The ETCC tool will issue a warning that it cannot connect to the database but it will check the oracle home binaries for the patches. If all patches are present, it will print:</p> <p>All the required one-off bugfixes are present in Database ORACLE_HOME.</p> <p>This is your Gold Image oracle home that you will use to patch all other oracle homes for this EBS RAC database. You will not need to run opatch on any of the remaining oracle homes at either the standby or the primary.</p> <p>To run the ETCC tool, ensure you source your environment using the <PDB>_<logicalhost_name>.env. Then run checkDBpatch.sh.</p> <p>In our example:</p> <pre>\$ cd \$ORACLE_HOME</pre> <pre>\$. ./VISPRD_ebsdb1.env</pre> <pre>\$ cd appsutil/etcc</pre> <pre>\$./checkDBpatch.sh</pre>

9.1.3 Convert to a Snapshot Standby

Convert the patched database home to an Oracle Snapshot Standby, then check it using ETCC.

STEP	SERVER, USER	ACTION
1	One standby database node – exabadm01, oracle	<p>Using SRVCTL, shut down the standby database completely then startup MOUNT only the patched instance:</p> <pre>\$ srvctl stop database -db EBSCDB_PHX3XD -o immediate</pre> <pre>\$ srvctl start instance -db EBSCDB_PHX3XD -instance EBSCDB1 -startoption mount</pre> <p>Note that redo shipping from the primary does not occur between the times the database is shut down and the snapshot standby instance is started.</p>
2	Any database, primary node or standby – exabadm01, oracle	<p>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.</p>

		<pre>DGMGRL> show configuration lag;</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSADB_OSC5XL - Primary database EBSADB_PHX3XD - Physical standby database Transport Lag: 0 seconds (computed 11 seconds ago) Apply Lag: 0 seconds (computed 11 seconds ago)</pre> <p>Fast-Start Failover: Disabled</p> <p>Configuration Status: SUCCESS (status updated 61 seconds ago)</p>
3	Any primary node or exabadm01, oracle	<p>If all is OK within Data Guard Broker, convert the physical standby to a snapshot standby:</p> <pre>DGMGRL> convert database 'EBSADB_PHX3XD' to snapshot standby; Converting database "EBSADB_PHX3XD" to a Snapshot Standby database, please wait... Database "EBSADB_PHX3XD" converted successfully DGMGRL> show configuration</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSADB_OSC5XL - Primary database EBSADB_PHX3XD - Snapshot standby database</pre> <p>Fast-Start Failover: Disabled</p> <p>Configuration Status: SUCCESS (status updated 104 seconds ago)</p> <p>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.</p>
4	Both Standby Oracle RAC nodes – exabadm01, exabadm02, Oracle	<p>If after converting to a snapshot standby the second un-patched instance starts, shut it down.</p> <pre>\$ srvctl status database -db EBSADB_PHX3XD</pre> <p>If the second instance is running:</p> <pre>\$ srvctl stop instance -db EBSADB_PHX3XD -instance EBSADB2 -stopoption immediate -f</pre>

9.1.4 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	<p>Using the script <code>rsync_EBS_OH.sh</code> from Appendix E, run <code>rsync</code> as root, without the <code>--no dry-run</code> option, to ensure that the excluded directories are not included and that all other directories will be updated. Do not specify the <code>--with_ebs_config</code> option as you do not want to copy over the <code>appsutil</code>, <code>db</code>s or the <code>network/admin</code> directories.</p> <p>For example, from exabadm01:</p>

		<pre># ./rsync_EBS_OH.sh --source_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_host exabadm02 --os_user root</pre> <p>Using the following parameters:</p> <pre>source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target_hosts = exabadm02 os_user = root dry_run = true with_ebs_config = false</pre> <p>About to execute the following rsync command:</p> <pre>rsync -avzh --progress --exclude=appsutil --exclude=network/admin --exclude=dbs --dry-run /u01/app/oracle/product/19.0.0.0/dbhome_2/ root@exabadm02:/u01/app/oracle/product/19.0.0.0/dbhome_2/</pre> <p>OK to continue? [Y y N n] :</p>
2	One standby database node – exabadm01, root	<p>If the dry run version of the command was OK, then run the rsync command with the <code>--no-dry-run</code> option.</p> <p>For example:</p> <pre># ./rsync_EBS_OH.sh --source_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_host exabadm02 --os_user root --no_dry_run</pre> <p>Using the following parameters:</p> <pre>source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2 target_hosts = exabadm02 os_user = root dry_run = false with_ebs_config = false</pre> <p>About to execute the following rsync command:</p> <pre>rsync -avzh --progress --exclude=appsutil --exclude=network/admin --exclude=dbs /u01/app/oracle/product/19.0.0.0/dbhome_2/ root@exabadm02:/u01/app/oracle/product/19.0.0.0/dbhome_2/</pre> <p>OK to continue? [Y y N n] :</p>
3	Each target standby database node – exabadm02, oracle	<p>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 <code>-local</code> option.</p> <p>To detach:</p> <pre>\$ cd \$ORACLE_HOME/oui/bin \$./runInstaller -silent -detachhome ORACLE_HOME="/u01/app/oracle/product/19.0.0.0/dbhome_2" -local</pre> <p>To re-attach:</p> <pre>\$./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle/product/19.0.0.0/dbhome_2" ORACLE_HOME_NAME= dbhome_2_ebs CLUSTER_NODES="{exabadm01,exabadm02}" LOCAL_NODE="exabadm02" -local</pre>
4	One standby database node – exabadm02, oracle	<p>Validate that the target node's inventory is good, and that it shows the patches are the same as at the source.</p> <p>For example:</p>

		<pre> \$ opatch lspatches 31859285;MERGE ON DATABASE RU 19.9.0.0.0 OF 29867728 30222669 30972817 30998035 31404014;XMLSEQUENCE /.. ISSUES 31142749;QUERY ON ALL_ARGUMENTS SLOW IN 19.6 PDB 30518349;STRESS FA PRC ORA-01000 SELECT DOCID FROM DR#IX_TS0481\$U , DR#IX_TS0481\$K... 30336383;19C UPGRADE XOQU121.SQL - MISSING - ON IF SQLCODE STATEMENT 28698087;SUPPORT CASE SENSITIVE PDB NAME 32187518;MERGE ON DATABASE RU 19.9.0.0.0 OF 31408636 31991705 32017301 31668882;OJVM RELEASE UPDATE: 19.9.0.0.201020 (31668882) 31772784;OCW RELEASE UPDATE 19.9.0.0.0 (31772784) 31771877;Database Release Update : 19.9.0.0.201020 (31771877) 30432118;MERGE REQUEST ON TOP OF 19.0.0.0.0 FOR BUGS 28852325 29997937 29997959;DSTV34 UPDATE - TZDATA2019B - NEED OJVM FIX OPatch succeeded. </pre>
5	One standby database node – exabadm02, oracle	<p>Run the EBS Technology Code Checker (ETCC) tool on the node to ensure all patches are present. Before running the ETCC checkDBpatch.sh, ensure that you source the environment using the <PDB_name>_<logical_hostname>.sh. For example:</p> <pre> \$ cd \$ORACLE_HOME \$. ./VISPRD_ebsdb2.env \$ cd appsutil/etcc \$./chackDBpatch.sh </pre> <p>Ensure the result shows:</p> <p>All the required one-off bugfixes are present in Database ORACLE_HOME.</p>
6	One standby database node – exabadm02, oracle	<p>Start the second instance:</p> <pre> \$ srvctl start instance -db EBSCDB_PHX3XD -instance EBSCDB2 </pre>
7	All remaining RAC nodes, oracle	Follow the above steps 1 through 6 on all remaining RAC instances for your standby database.

9.1.5 Complete the Database Patching Process

Once all the standby database instances have been patched, complete the database patching process on the snapshot standby.

STEP	SERVER, USER	ACTION
1	Original standby database node – exabadm01, oracle	<p>Run datapatch. This instance should still be up.</p> <pre> \$ cd \$ORACLE_HOME/OPatch \$./datapatch -verbose </pre> <p>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.</p>
2	One standby database node – exabadm01, oracle	<p>Compile any invalid packages.</p> <pre> \$ cd \$ORACLE_HOME/rdbms/admin \$ sqlplus / as sysdba SQL> @utlrp.sql </pre>
3	One standby database node – exabadm01, oracle	<p>Start all remaining Oracle RAC instances if not already done pre the previous section. For example:</p> <pre> \$ srvctl start database -db EBSCDB_PHX3XD </pre>

9.1.6 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 – pca3vm55, pca3vm56, 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 – pca3vm55, pca3vm56, 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 – pca3vm55, pca3vm56, applmgr	Reset the web entry points using a script similar to the one 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 – pca3vm55, pca3vm56, 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 – pca3vm55, pca3vm56, applmgr	Start application services. \$ adstrtal.sh
6	All standby application tier nodes – pca3vm55, pca3vm56, applmgr	Test the application now running against the patched Oracle RAC snapshot standby database. Conduct testing of critical jobs and online workload where possible.

9.1.7 Revert to Full Physical Standby

When testing is finished, revert the standby environment to a full physical standby.

STEP	SERVER, USER	ACTION
1	All standby application tier nodes – pca3vm55, pca3vm56, applmgr	When testing is complete shut down the application services on all standby application tier nodes. \$ adstpa11.sh
2	All standby application tier nodes – pca3vm55,	Umount the snapshot clone. \$ umount /u02

	pca3vm56, root	
3	ZFS appliance, root or administrator	<p>Drop the snapshot clone.</p> <p>To drop the snapshot clone on ZFS, log into the ZFS browser UI as an administrator.</p> <p>Click Shares.</p> <p>Click Projects on the left.</p> <p>Locate the snapshot clone project (ours is called MAA_EBS_SNAPSHOT).</p> <p>Click the trashcan icon to delete the project.</p> <p>Confirm the operation.</p>
4	One standby database node – exabadm01, oracle	<p>Convert the snapshot standby database back to a physical standby using DGMGRL.</p> <pre>DGMGRL> show configuration</pre> <p>Configuration - ebscdb_dg</p> <pre> Protection Mode: MaxPerformance Members: EBSCDB_OSC5XL - Primary database EBSCDB_PHX3XD - Snapshot standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 46 seconds ago) </pre> <pre>DGMGRL> convert database 'EBSCDB_PHX3XD' to physical standby; Converting database "EBSCDB_PHX3XD" to a Physical Standby database, please wait... Operation requires a connection to database "EBSCDB_OSC5XL" Connecting ... Connected to "EBSCDB_osc5x1" Connected as SYSDBA. Oracle Clusterware is restarting database "EBSCDB_PHX3XD" ... Connected to "EBSCDB_phx3xd" Connected to "EBSCDB_phx3xd" Continuing to convert database "EBSCDB_PHX3XD" ... Database "EBSCDB_PHX3XD" converted successfully</pre>

9.1.8 Propagate the Patched Database Homes to the Primary

STEP	SERVER, USER	ACTION
1	Primary application tier, all nodes – pca1vm55,pc a1vm56, applmgr	<p>This is the start of the primary / production outage.</p> <p>Shut all application services and concurrent managers down on all primary application tier nodes.</p> <pre>\$ adstpall.sh</pre>
2	One primary database node – exaadb03, oracle	<p>Shut all primary database instances down.</p> <pre>\$ srvctl stop database -db EBSCDB_osc5x1 -o immediate</pre>
3	One standby database node – exabadm01, root	<p>Using the script <code>rsync_EBS_OH.sh</code> from Appendix E, run <code>rsync</code> from one standby database Oracle home to the Oracle homes on the primary. Because we exclude the three directories--<code>appsutil</code>, <code>network/admin</code>, and <code>dbs</code>--we can use one <code>ORACLE_HOME</code> as the source. The script excludes these directories by default. Do not specify the <code>-with_ebs_config</code> option when running the script. We first ran without the <code>-no_dry-run</code> option as shown in the following example. The script uses <code>rsync</code> to synchronize the <code>ORACLE_HOME</code> from <code>exabadm01</code> to <code>exaadb03</code>:</p>

```
# ./rsync_EBS_OH.sh --source_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_host exaadb03 --os_user root
```

Using the following parameters:

```
source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
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/product/19.0.0.0/dbhome_2/ root@exaadb03:/u01/app/oracle/product/19.0.0.0/dbhome_2/
```

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
```

```
# ./rsync_EBS_OH.sh --source_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_oh /u01/app/oracle/product/19.0.0.0/dbhome_2 --target_host exaabd04 --os_user root
```

Using the following parameters:

```
source ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target ORACLE_HOME = /u01/app/oracle/product/19.0.0.0/dbhome_2
target_hosts      = exaabd04
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/product/19.0.0.0/dbhome_2/ root@exaabd04:/u01/app/oracle/product/19.0.0.0/dbhome_2/
```

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

If the output is as expected, 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 8 minutes in our environment. Your results may vary depending on network bandwidth and latency.

Repeat this action for all database homes on the primary.

4

Each primary database node - exaadb03, exaadb04, oracle

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/product/19.0.0.0/dbhome_2" -local
```

To re-attach for exaadb03:

```
$ ./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle/product/19.0.0.0/dbhome_2" ORACLE_HOME_NAME=dbhome_2_ebs CLUSTER_NODES="{exaadb03,exaadb04}" LOCAL_NODE="exaadb03" -local
```

To re-attach for exaabd04:

```
$ ./runInstaller -silent -attachHome ORACLE_HOME="/u01/app/oracle/product/19.0.0.0/dbhome_2" ORACLE_HOME_NAME=dbhome_2_ebs CLUSTER_NODES="{exaadb03,exaadb04}" LOCAL_NODE="exaabd04" -local
```

5	Each primary database node – exaadb03, exaadb04, oracle	<p>Run the OPatch lspatches command on each node to validate that the Oracle home inventories are good and they have the new patches.</p> <pre>\$ opatch lspatches 31859285;MERGE ON DATABASE RU 19.9.0.0.0 OF 29867728 30222669 30972817 30998035 31404014;XMLSEQUENCE /.. ISSUES 31142749;QUERY ON ALL_ARGUMENTS SLOW IN 19.6 PDB 30518349;STRESS FA PRC ORA-01000 SELECT DOCID FROM DR#IX_TS0481\$U , DR#IX_TS0481\$K... 30336383;19C UPGRADE XOQU121.SQL - MISSING - ON IF SQLCODE STATEMENT 28698087;SUPPORT CASE SENSITIVE PDB NAME 32187518;MERGE ON DATABASE RU 19.9.0.0.0 OF 31408636 31991705 32017301 31668882;OJVM RELEASE UPDATE: 19.9.0.0.201020 (31668882) 31772784;OCW RELEASE UPDATE 19.9.0.0.0 (31772784) 31771877;Database Release Update : 19.9.0.0.201020 (31771877) 30432118;MERGE REQUEST ON TOP OF 19.0.0.0.0 FOR BUGS 28852325 29997937 29997959;DSTV34 UPDATE - TZDATA2019B - NEED OJVM FIX</pre> <p>OPatch succeeded.</p> <p>Both primary Oracle RAC nodes are patched and show the same output.</p>
6	One primary database node – exaadb03, oracle	<p>Using SRVCTL, start all instances and run datapatch.</p> <pre>\$ srvctl start database -db EBSCDB_osc5x1 \$ cd \$ORACLE_HOME/OPatch \$./datapatch -verbose</pre> <p>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.</p>
7	All primary database nodes – exaadb03, exaadb04, oracle	<p>After all patches have been applied and datapatch successfully completes, run the EBS Technology Code Checker (ETCC) tool to ensure all required patches are present. Make sure you source the environment with the <PDB_name>_<logical_hostname>.env file for each DB node. For exaadb03:</p> <pre>\$ cd \$ORACLE_HOME \$. ./VISPRD_ebsdb1.env \$ cd appsutil/etcc \$./checkDBpatch.sh</pre> <p>You should see the following on each node after running checkDBpatch.sh:</p> <p>All the required one-off bugfixes are present in Database ORACLE_HOME.</p>
8	One primary database node – exaadb03, oracle	<p>Compile any invalid packages.</p> <pre>\$ cd \$ORACLE_HOME/rdbms/admin \$ sqlplus / as sysdba SQL> @utlrp.sql</pre>
9	One primary database node – exaadb03, oracle	<p>Check the Data Guard Broker status with SHOW CONFIGURATION.</p> <pre>DGMGRL> show configuration lag</pre> <p>Configuration - ebscdb_dg</p> <pre>Protection Mode: MaxPerformance Members: EBSCDB_OSC5XL - Primary database EBSCDB_PHX3XD - Physical standby database Transport Lag: 0 seconds (computed 4 seconds ago) Apply Lag: 0 seconds (computed 4 seconds ago)</pre> <p>Fast-Start Failover: Disabled</p> <p>Configuration Status: SUCCESS (status updated 50 seconds ago)</p>
10	Primary application tier, all nodes – pca1vm55,	<p>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.</p>

	pca1vm56, applmgr	
15	Primary application tier, all nodes – pca1vm55, pca1vm56, applmgr	Start all application services at the primary. \$ adstrta1.sh This ends the outage at the primary site.

10 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.

Database MAA:

[Oracle Database High Availability Overview and Best Practices](#)

[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

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 Oracle Database 19c \(Doc ID 2608030.1\)](#)

[Oracle Patch Assurance - Data Guard Standby-First Patch Apply \(Doc ID 1265700.1\)](#)

Oracle Real Application Clusters (RAC):

[Real Application Clusters Administration and Deployment Guide](#)

[Clusterware Administration and Deployment Guide](#)

[Using Oracle 19c RAC Multitenant \(Single PDB\) with Oracle E-Business Suite Release 12.2 \(Doc ID 2530665.1\)](#)

[Oracle RAC Rolling Install Process for the “Oracle JavaVM Component Database PSU” \(OJVM PSU\) Patches \(Doc ID 2217053.1\)](#)

Oracle E-Business Suite MAA:

[Business Continuity for Oracle E-Business Suite Release 12.2 on Oracle Database 19c Using Logical Host Names \(Doc ID 2617788.1\)](#)

[Oracle E-Business Suite on Exadata Database Machine \(White Paper\)](#)

[Oracle Site Guard](#)

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 \(PIC\) \(Doc ID 1304305.1\)](#)

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 \(Doc ID 1967205.1\)](#)
[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 \(Doc ID 1375769.1\)](#)

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 \(Doc ID 1383621.1\)](#)

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 \(Doc ID 2174221.1\)](#)

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 Exadata Database Machine Performance Best Practices \(Doc ID 1274475.1\)](#)
[Configuring Exadata I/O Resource Manager for Common Scenarios \(Doc ID 1363188.1\)](#)
[Oracle Exadata Database Machine Setup/Configuration Best Practices \(Doc ID 1274318.1\)](#)
[Oracle SuperCluster Supported Software Versions – All Hardware Types \(Doc ID 1567979.1\)](#)

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 \(Doc ID 1392497.1\)](#)

11 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 required OS packages for Oracle E-Business Suite and to 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. Make sure you download the correct public yum for your Linux version.

Log on as root to the application server

Move to the yum configuration directory:

```
$ cd /etc/yum.repos.d
```

Download the public yum configuration file:

```
$ wget http://public-yum.oracle.com/public-yum-ol7.repo
```

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

```
[ol7_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
```

```
[ol7_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
```

```
[ol7_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
```

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 \(Doc ID 1330701.1\)](#).

12 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, and 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 aliases 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:

- Log onto each Oracle RAC database tier node as the owner of the EBS Oracle home (i.e., oracle_ebs).
- Create a custom.txt file in a directory of your choice, and add any custom connect strings.
- Copy the ebs_scan_db.sh script provided below to the same directory as your custom.txt script.
- 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.
- Copy the edited scripts to all of the Oracle RAC nodes.
- Run the ebs_scan_db.sh script on each node.
./ebs_scan_db.sh
- 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:

- Log onto the application tier node and source the RUN file system.
- 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.
- Copy the ebs_scan_app.sh script provided below to an application tier node into the same directory as your custom.txt script.
- 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.
- 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.
- Run the ebs_scan_app.sh script on each node.

./ebs_scan_app.sh

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

12.1 The custom.txt File

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

```
# For the local EBS listener if not using the Grid listener:
```

```
VISPRD2_LOCAL=  
    (DESCRIPTION=  
        (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient02-vip.mycompany.com)(PORT=1523))  
        (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient02-vip.mycompany.com)(PORT=1521))  
    )  
  
VISPRD1_LOCAL=  
    (DESCRIPTION=  
        (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient01-vip.mycompany.com)(PORT=1523))  
        (ADDRESS=(PROTOCOL=tcp)(HOST=exabclient01-vip.mycompany.com)(PORT=1521))  
    )
```

12.2 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\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

```

12.3 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

```

13 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 DNS, a known entity on the network such as a load balancer is used in order to enable web access. Several MOS Docs for EBS 12.2 describe setting the web entry point. The following is a script for implementation.


```
# 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 8000
```

```
$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 8000
```

Usage note:

The above script is not configured for SSL. To configure the above script for SSL terminated at the load balancer, do the following:

- Edit the context file and remove the '#' in the context variable: s_enable_sslterminator.
- In the context file, change the context variable s_webentryurlprotocol from http to https in the URL.
- In the above script on each application tier node, set s_login_page and s_external_url from http to https in the URL.
- In the above script on each application tier node, set s_active_webport to 443.
- Run the above script on each node.
- Run autoconfig on each node.
- Restart application services.

14 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://pca1vm55.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 0
- 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.

14.1 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://pca1vm55.mycompany.com:7001/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.

14.2 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=oaocore_server10 -servicetype=oaocore \  
  -managedsrvport=7204 -logfile=$APPLRGF/TXK/addMS_oaocoreserver10.log <<!EOF  
<apps-password>  
<Weblogic-password>  
!EOF
```

```
/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh start  
oaocore_server10 <<!EOF  
<Weblogic-password>  
!EOF
```

```
perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \  
  ebs-create-managedserver -contextfile=$CONTEXT_FILE \  
  -managedsrvname=oaocore_server11 -servicetype=oaocore \  
  -managedsrvport=7205 -logfile=$APPLRGF/TXK/addMS_oaocoreserver11.log <<!EOF  
<apps-password>  
<Weblogic-password>  
!EOF
```

```
/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh start  
oaocore_server11 <<!EOF  
<Weblogic-password>  
!EOF
```

[...]..

```
perl $AD_TOP/patch/115/bin/adProvisionEBS.pl \  
  ebs-create-managedserver -contextfile=$CONTEXT_FILE \  
  -managedsrvname=oaocore_server16 -servicetype=oaocore \  
  -managedsrvport=7210 -logfile=$APPLRGF/TXK/addMS_oaocoreserver16.log <<!EOF  
<apps-password>  
<Weblogic-password>  
!EOF
```

```
/u06/app/ebs/vis/fs2/inst/apps/VIS_ebsapp2/admin/scripts/admanagedsrvctl.sh start  
oaocore_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/TKK/addMS_formserver5.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/TKK/addMS_formserver6.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
```

14.3 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 \
  -configuration=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
```

14.4 Bounce the HTTP Server

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

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

15 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.

15.1 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 specify 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`
 - `db`

15.1.1 Copy a patched ORACLE_HOME

To use the script to copy a patched `ORACLE_HOME`:

Copy the script to a directory of your choice on any one Oracle RAC database nodes.

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  
exadb03 --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          = exadb03  
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@exadb03:/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).

15.1.2 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 exadb01 --  
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          = exadb01  
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@exadb03:/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.

15.2 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
permissions
# will be preserved.
#
# Since this should be run after patching one oracle home, do not start 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 primary and DR site.
# For RAC, all ORACLE_HOMES need to be synced up.
# Exclude specific directories:
#   - $ORACLE_HOME/appsutil
#   - $ORACLE_HOME/network/admin
#   - $ORACLE_HOME/dbs
#
# To override excluding the above directories, use the --with_ebs_config command line
option.
#
# To preserve the output of rsync, the following example to run this script can be executed:
#
#   ./rsync_EBS_OH.sh | tee -a rsync.out
#
#
# Parse the command line arguments.
TEMP=`getopt -o h --long
help,source_oh:,target_oh:,no_dry_run,with_ebs_config,target_host:,os_user: -n 'test.sh' --
"$@"`

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 ${ORACLE_HOME} ]]; then
        echo "source_oh not specified and ORACLE_HOME not defined. Specify source_oh on
command line or set ORACLE_HOME."
        echo "Exiting..."
        exit 1
    else
        TARGET_OH=${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

```

16 Appendix F: Scripts used by Site Guard

The following scripts are used by Site Guard to orchestrate and automate the switchover and failover processes. These are example scripts and must be modified to suite your environment. These scripts must be placed on all application tier nodes at both primary and DR sites. Once each script has been modified for your environment, test each script to ensure they complete successfully before integrating with Site Guard.

EBS_stopServices.sh:

```

#!/bin/sh

EGREP_STRING="u02|FND|LIB"
PROCESS_COUNT=0
PID_LIST=""

# Set the apps env.
if [ -f /u02/app/ebs/visprd/EBSapps.env ]; then
    . /u02/app/ebs/visprd/EBSapps.env run
    $ADMIN_SCRIPTS_HOME/adstpall.sh <<!EOF
apps
<apps password>
<Webloginc Admin console password>
!EOF

else
    echo "EBS environment not available to source. Please check that
/u02/app/ebs/vis/EBSapps.env exists and the file system is mounted."
    exit -1
fi

# Sleep for 30 seconds to allow all processes to terminate.
sleep 30

PROCESS_COUNT=`ps -elf | grep applmgr | egrep "${EGREP_STRING}" | grep -v grep | wc -l`
echo "Number of remaining process : ${PROCESS_COUNT}"

i=1

while [ ${PROCESS_COUNT} -ne 0 ];
do
    sleep 15
    PROCESS_COUNT=`ps -elf | grep applmgr | egrep "${EGREP_STRING}" | grep -v grep | wc -l`
    echo "Number of remaining process : ${PROCESS_COUNT}"
    if [[ $i -gt 4 && ${PROCESS_COUNT} -ne 0 ]]; then
        PID_LIST=`ps -elf | grep applmgr | egrep "${EGREP_STRING}" | grep -v grep | awk '{
print $4 }'`
        echo "Killing processes:"
        echo "${PID_LIST} "
        kill -9 ${PID_LIST}
    fi
    i=`expr $i + 1`
done

EBS_cleanUpStates.sh

```

```

#!/bin/sh

# The EBS_OHS_NODE is set to the node number i.e, 1, 2, 3, etc., to determine which cache
state directory to remove.
# The primary application tier node should be set to 1. The seconde node should be set to 2,
etc.

EBS_OHS_NODE=1

# Set the apps env.
if [ -f /u02/app/ebs/visprd/EBSapps.env ]; then
    . /u02/app/ebs/visprd/EBSapps.env run
else
    echo "EBS environment not available to source. Please check that
/u02/app/ebs/visprd/EBSapps.env exists and the file system is mounted."
    exit -1
fi

# Remove any leftover states either from a failover where the app services crashed or from a
snapshot clone from a running
# EBS services.

count=`ls
${RUN_BASE}/FMW_Home/webtier/instances/EBS_web_VISPRD_OHS${EBS_OHS_NODE}/config/OPMN/opmn/sta
tes | wc -l`

# Do we have content in the states directory? If so, remove the states directory. If not, do
nothing.
if [[ ${count} -ne 0 ]]; then

    # Remove the states directory.
    rm -r
${RUN_BASE}/FMW_Home/webtier/instances/EBS_web_VIS_OHS${EBS_OHS_NODE}/config/OPMN/opmn/states
fi

exit 0

EBS_autoconfig.sh

```

```

#!/bin/sh

# Set the apps env.
if [ -f /u02/app/ebs/visprd/EBSapps.env ]; then
    . /u02/app/ebs/visprd/EBSapps.env run

    perl $AD_TOP/bin/adconfig.pl -contextfile=$CONTEXT_FILE -parallel <<!EOF
apps
<Apps password>
!EOF
    exit 0
else
    echo "EBS environment not available to source. Please check that
/u02/app/ebs/visprd/EBSapps.env exists and the file system is mounted."
    exit -1
fi

```

EBS_startWLSAdminServer.sh:

```

#!/bin/sh

# Set the apps env.
if [ -f /u02/app/ebs/visprd/EBSapps.env ]; then
    . /u02/app/ebs/visprd/EBSapps.env run
    $ADMIN_SCRIPTS_HOME/adadminsrvctl.sh start <<!EOF
<Weblogic Admin oonsole password>
<Apps password>
!EOF
    exit $?
else
    echo "EBS environment not available to source. Please check that
/u06/app/ebs/vis/EBSapps.env exists and the file system is mounted."
    exit -1
fi

```

EBS_startServices.sh:

```
#!/bin/sh
```

```
MT_SCRIPT_DIR=/home/applmgr/EBSRoleChange
```

```
# Set the apps env.
```

```
if [ -f /u02/app/ebs/visprd/EBSapps.env ]; then
```

```
    . /u02/app/ebs/visprd/EBSapps.env run
```

```
    $ADMIN_SCRIPTS_HOME/adstrtal.sh -msimode <<!EOF
```

```
apps
```

```
<Apps password>
```

```
<Weblogic Admin console password>
```

```
!EOF
```

```
    exit 0
```

```
else
```

```
    echo "EBS environment not available to source. Please check that  
/u06/app/ebs/vis/EBSapps.env exists and the file system is mounted."
```

```
    exit -1
```

```
fi
```

Connect with us

Call +1.800.ORACLE1 or visit [oracle.com](https://www.oracle.com). Outside North America, find your local office at: [oracle.com/contact](https://www.oracle.com/contact).

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.