PeopleSoft Maximum Availability Architecture

With Case Study on Oracle Private Cloud Appliance and Exadata Database Machine

March 10, 2021, Version 1.1

Public

1 Table of contents

2 Executive Overview

Oracle Maximum Availability Architecture (MAA) is Oracle's best practices 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) at http://www.oracle.com/goto/maa.

PeopleSoft Maximum Availability Architecture is a best practice blueprint for achieving the optimal PeopleSoft high availability deployment using Oracle high availability technologies and recommendations.

In this paper we describe:

- The PeopleSoft MAA architecture along with installation, configuration and operational best practices.
- How PeopleSoft MAA was implemented on Oracle Private Cloud Appliance (PCA) and Exadata.
- Our tests to validate our MAA best practices.

When PeopleSoft was configured with our MAA best practices on PCA and Exadata, we demonstrated that there was minimal user impact during typical failure scenarios. In the event of a total site failure the disaster recovery site could be brought online in as little as 10 minutes.

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 Review, TCO/ROI Analysis, Proof-of-Concepts, Customized Demonstrations and Workshops to support a dynamic community of VAD/VAR, ISV vendors and System Integrators.

If you are considering High Availability, Disaster Recovery and Datacenter Consolidation using Oracle's Maximum Availability Architecture (MAA), 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 SMEs 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.

3 Introduction

In this paper we provide:

- An introduction to Oracle Engineered Systems
- The PeopleSoft Maximum Availability Architecture – a high-level description of the architecture and key technology components
- PeopleSoft MAA Case Study on Private Cloud Appliance and Exadata – how the MAA architecture was established on our target systems
- Operational Procedures – how to operate the system in the event of planned and unplanned outages
- Outage testing and results – our tests to validate our MAA best practices for PeopleSoft.
- Appendixes – details of how PeopleSoft MAA was implemented on the Private Cloud Appliance and Exadata environment

3.1 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.

3.1.1 Oracle Private Cloud Appliance (PCA)

Oracle Private Cloud Appliance and Oracle Private Cloud at Customer are on-premises cloud native converged infrastructure that allows 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 together with Oracle Exadata provides a powerful, single-vendor, application and database platforms 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 together with Oracle Cloud Infrastructure provides customers with a complete solution to securely maintain workloads on both private and public clouds.

3.1.2 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 database workloads. Exadata is a pre-configured, pre-tuned, and pre-tested integrated system of servers, networking and storage all optimized around the Oracle Database. 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.

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 servers 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.

4 PeopleSoft Maximum Availability Architecture Overview

PeopleSoft Maximum Availability Architecture (MAA) is a PeopleSoft high availability architecture layered on top of the Oracle Database and Oracle Fusion Middleware Maximum Availability Architectures, including a secondary site to provide business continuity in the event of a primary site failure.

In this section we will first present the Oracle Database and Oracle Fusion Middleware Maximum Availability Architectures, then we will describe how to provide high availability for the PeopleSoft application on top of that foundation, resulting in a full PeopleSoft MAA implementation.

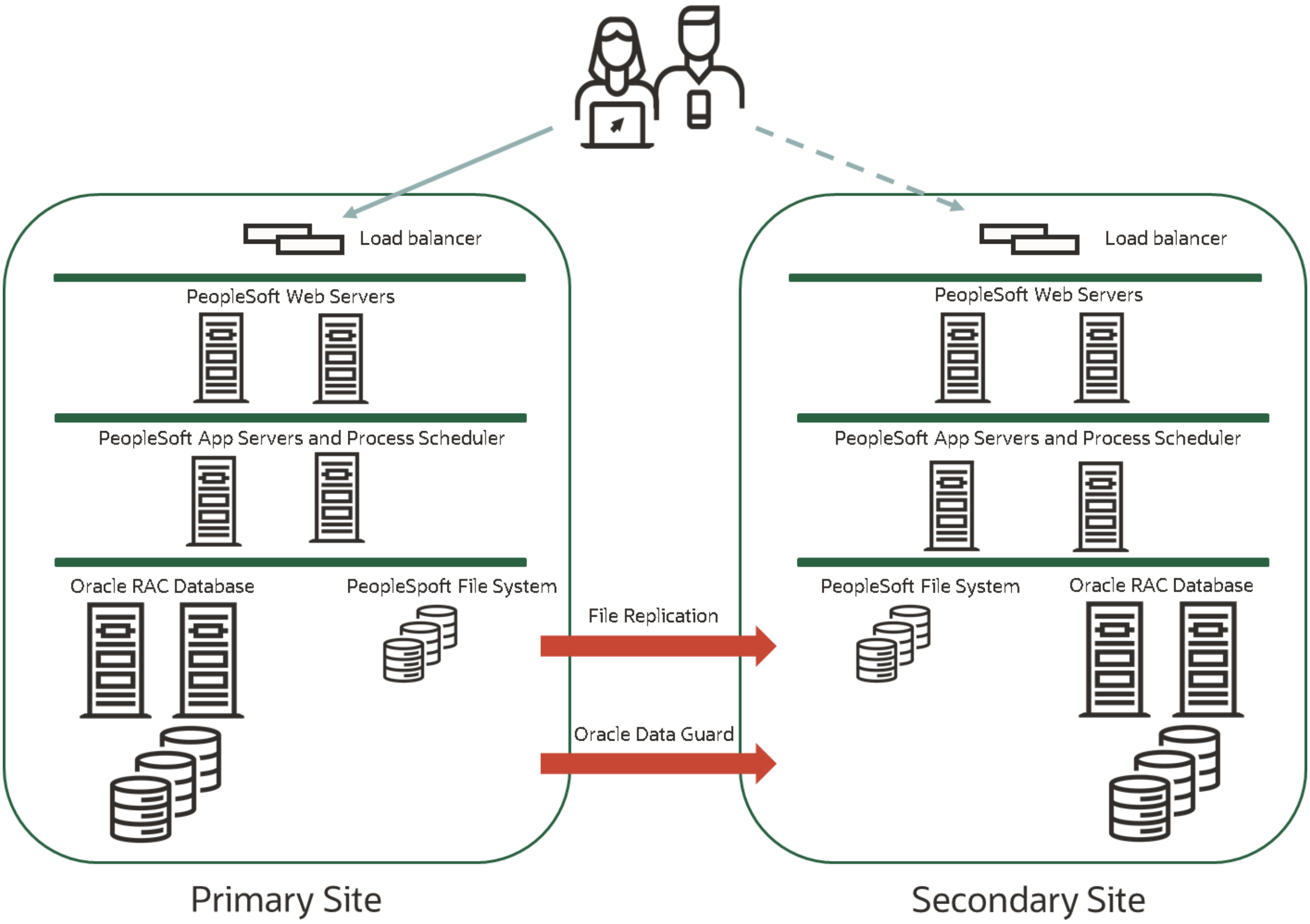

Figure 1. PeopleSoft Maximum Availability Architecture

In figure 1 above, we illustrate a full-stack MAA architecture to include both primary and secondary sites. The secondary site is a symmetric replica of the primary. Each site consists of the following:

- HTTP/HTTPS load balancer for web-based application services
- 2 PeopleSoft web servers composing the PeopleSoft Internet Architecture (PIA)
- 2 servers that hosts both the PeopleSoft Application Server domains and Process scheduler domains
- A shared file system for PeopleSoft application installation and report repository
- A Real Application Cluster (RAC) database with two database servers and shared storage

The shared file system used by the application tier is replicated to the secondary site. Data Guard replicates the primary database to the standby database at the secondary site.

4.1 Oracle Database Maximum Availability Architecture

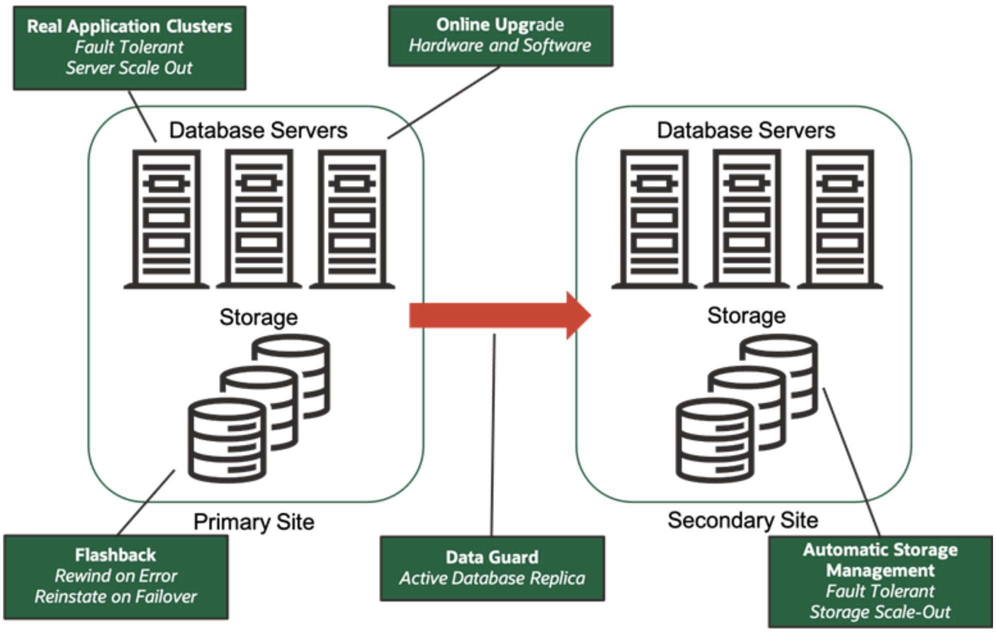

Figure 2. Oracle Database Maximum Availability Architecture

To achieve maximum PeopleSoft application availability, Oracle recommends deploying PeopleSoft on an Oracle Database MAA foundation (shown in figure 2 above) that includes the following technologies:

- Oracle Multitenant
- Oracle Real Application Clusters and Oracle Clusterware
- Oracle Data Guard
- Oracle Flashback
- Oracle Automatic Storage Management
- Oracle Recovery Manager and Oracle Secure Backup
- Oracle Online Upgrade

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

4.1.1 Oracle Multitenant

Oracle Multitenant was introduced in Oracle RDBMS 12c which enables the segmentation of data into separate self-contained pluggable databases. These pluggable databases also provide database virtualization hosted within a single container database. A container database hosting one or more pluggable database all share the same infrastructure resources such background processes, memory, CPU, IO, networking, etc. The following diagram illustrates a simplified form of a single container database (CDB) with three pluggable databases (PDBs).

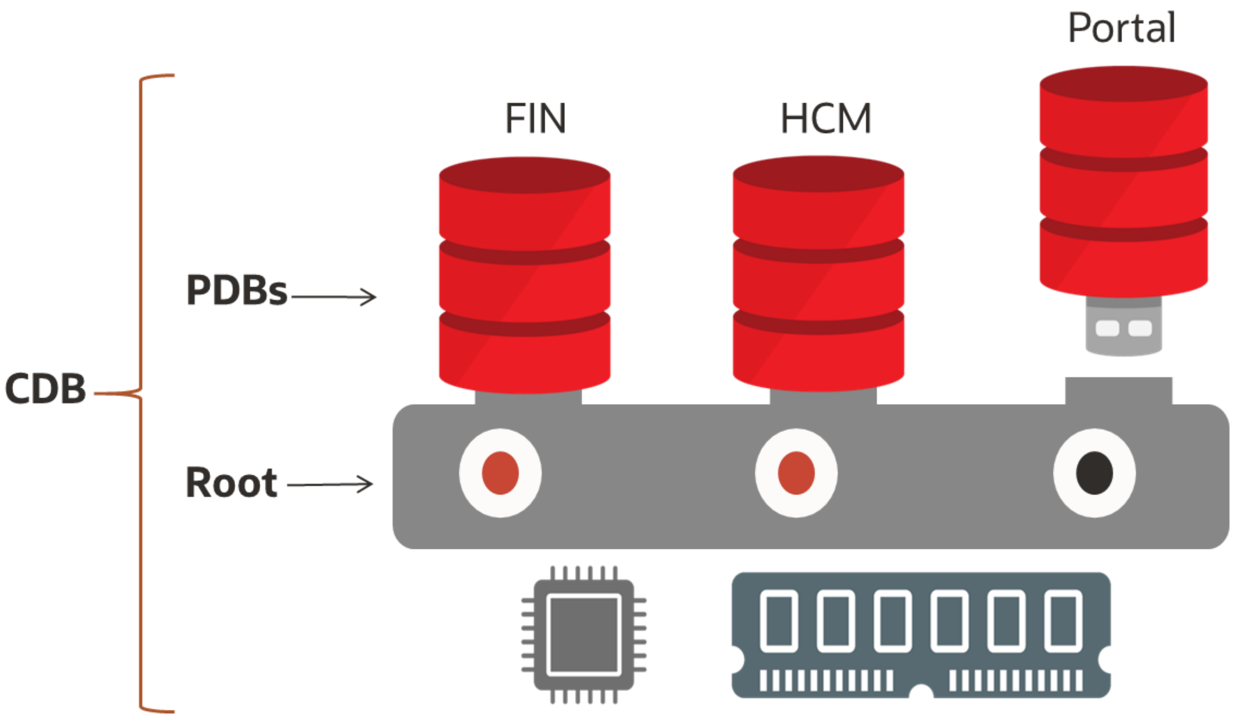

Figure 3: Container database with three pluggable databases

Figure 3 above illustrates the following:

- A single container database (CDB)
- Three pluggable databases (PDB) named: FIN, HCM and Portal
- 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 advantages of Oracle Multitenant are:

- Self-contained application PDB
- Application runs unchanged
- Portability via plug and unplug
- PDB upgrade via plug and unplug
- Common operations at CDB level such as patching and backups
- Granular operations at PDB levels such as backup, restore, flashback
- PDBs can be opened or closed on a per RAC instance bases
- Multiple PDBs hosted on a single CDB provides for higher consolidation while keeping data securely segregated.

PeopleSoft schemas and data are deployed in PDBs allowing multiple PeopleSoft applications to share the same infrastructure and resources.

4.1.2 Oracle Real Application Clusters and Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) allows the Oracle Database to run any packaged or custom application unchanged across a set of clustered nodes. This capability provides the highest levels of availability and the most flexible scalability. If a clustered node fails, the Oracle Database will continue running on the surviving nodes. When more processing power is needed, another node can be added without interrupting user access to data. See also: *Real Application Clusters Administration and Deployment Guide*.

Oracle Clusterware is a cluster manager that is designed specifically for the Oracle Database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources (such as database instances and listeners). If a failure occurs, Oracle Clusterware will automatically attempt to restart the failed resource. During outages, Oracle Clusterware relocates the processing performed by the inoperative resource to a backup resource. For example, if a node fails, Oracle Clusterware will relocate database services being used by the application onto a surviving node in the cluster. See also: *Clusterware Administration and Deployment Guide*.

4.1.3 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, user errors, and data 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 primary 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.

PeopleSoft supports both physical and logical standby databases. See also: *Data Guard Concepts and Administration*.

- A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. A physical standby database is kept synchronized with the primary database, using Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.
- A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database though SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database.

It is possible (and ideal) to deploy a local standby database at the primary site as well as a remote standby at a secondary site. This offers the advantage that a failover to the local standby can be performed while the PeopleSoft Servers continue running, almost transparently to the end users. It also offers the ability to perform an online database upgrade without the need to switch to another site. We would recommend that a local and remote standby be deployed for maximum availability. For further details please see the Oracle MAA white paper "Database Rolling Upgrades Made Easy by Using a Data Guard Physical Standby Database."

4.1.3.1 Physical Standby Features

Oracle Active Data Guard is a physical standby database that receives and applies redo while it is open for read-only access, and so it may be used for other purposes as well as disaster recovery.

With a single command, a physical standby database can be converted into a Snapshot Standby and become an independent database open read-write, ideal for QA and other testing. The Snapshot Standby continues to receive and archive redo data from the primary database while it is open read-write, thus protecting primary data at all times. When testing is complete, a single command will convert the snapshot back into a standby database, and automatically resynchronize it with the primary.

A physical standby database can be used as a "transient logical standby" for rolling database upgrades using the SQL Apply process and return to its function as a physical standby database once the upgrade is complete.

4.1.3.2 Logical Standby Features

A logical standby database can be used for disaster recovery and reporting requirements and can also be used to upgrade the database software and apply patch sets while the application is online and with almost no downtime.

4.1.4 Oracle Flashback

Oracle Flashback quickly rewinds an Oracle database, table or transaction to a previous point in time, to correct any problems caused by logical data corruption or user error. It is like a 'rewind button' for your database. Oracle Flashback is also used to rapidly return a failed primary database to standby operation after a Data Guard failover, thus eliminating the need to recopy or re-instantiate the entire database from a backup. See "Oracle Flashback Technology" in *Oracle Database Development Guide* for more information.

4.1.5 Oracle Automatic Storage Management

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

- Significantly less work to provision database storage
- Higher levels of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, ASM spreads 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.

4.1.6 Oracle Recovery Manager and Oracle Secure Backup

Oracle Recovery Manager (RMAN) is an Oracle Database utility that can back up, restore, and recover database files. It is a feature of Oracle Database and does not require 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 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, Oracle Secure Backup provides a complete tape backup solution for your IT environment. Oracle Secure Backup is tightly integrated with RMAN to provide the media management layer for RMAN.

4.1.7 PeopleSoft Database Configuration Best Practices

We recommend that PeopleSoft database is configured with the following best practices:

4.1.7.1 Add the PeopleSoft Database to Cluster Ready Services

If not done already, add the PeopleSoft database and Oracle RAC instances to Cluster Ready Services (CRS). The following serve as examples:

```
srvctl add database -db CDBHCM_osc1a -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_3 -
spfile +DATAC1/CDBHCM_OSC1A/spfilecdbhcm.ora -diskgroup "DATAC1,RECOC1” -pwfile
+DATAC1/CDBHCM_OSC1A/PASSWORD/pwcdbhcm

srvctl add instance -db CDBHCM_osc1a -instance CDBHCM1 -node exa14db01
srvctl add instance -db CDBHCM_osc1a -instance CDBHCM2 -node exa14db02
```

4.1.7.2 Create FAN Enabled Role Based Database Services

PeopleSoft supports Fast Application Notification (FAN). When an Oracle RAC database instance fails, the recovering instance sends an INSTANCE DOWN event to all clients that were connected to the failed instance. The clients then break their current TCP connections and perform Transparent Application Failover (TAF) and reconnect to the same database service on surviving instances.

Role-based services are created and used by the PeopleSoft application to connect to the database. These role-based services are started based on the database role. See the table below.

SERVICE NAME	DATABASE ROLE	PURPOSE
HR92U033 _ONLINE	PRIMARY	• Online HRMS service
HR92U033 _BATCH	PRIMARY	Batch processing service
PSQUERY	PHYSICAL STANDBY	Offload queries to Oracle Active Data Guard physical standby database

The following are examples of creating FAN-enabled role-based database services:

```
srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service HR92U033_BATCH -preferred
"CDBHCM1,CDBHCM2"  -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO
-failoverretry 10 -failoverdelay 3

srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service HR92U033_ONLINE -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO -
failoverretry 10 -failoverdelay 3
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" -failovermethod BASIC  -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

Using the example for the service HR92U033_ONLINE:

PARAMETER	VALUE	DESCRIPTION
-db	CDBHCM_osc1a	• The database unique name
-pdb	HR92U03	The name of the PDB that this service is for
-service	HR92U033_ONLINE	The database service name
-preference	“CDBHCM1,CDBHCM2”	The preferred instances the service should be started on
-failovermethod	BASIC	TAF failover method
-failovertype	AUTO	Application continuity session failover type: SELECT, TRANSACTIN or AUTO
-notification	TRUE	High Availability FAN events
-role	PRIMARY	The database role in which the service should be started
-failoverretry	10	The number of connection retries before the connection fails
-failoverdelay	3	Time interval in seconds between each connection retry attempt

4.1.7.3 Configure HugePages (Linux Database Server Only)

PeopleSoft will typically run with many database connections and a large SGA; therefore, configuring HugePages for the PeopleSoft 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, which is critically important with a large SGA or when there are high numbers of concurrent database connections. HugePages can only be used for SGA memory space so do not configure more than is required.

My Oracle Support note 361468.1, “HugePages on Oracle Linux 64-bit” describes how to configure HugePages. Automatic Shared Memory Management (ASMM) can be used with HugePages and so use the SGA_MAX_SIZE parameter to set the SGA size for each instance.

Automatic Memory Manager (AMM) cannot be used in conjunction with huge pages and so the MEMORY_TARGET and MEMORY_MAX_TARGET parameters should be unset for each database instance. See My Oracle Support note 749851.1 “HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux” for details.

Set the parameter USE_LARGE_PAGES=’only’ for each instance so that the instance will only start if sufficient HugePages are available. See My Oracle Support note 1392497.1 “USE_LARGE_PAGES To Enable HugePages” for details.

It may be necessary to reboot the database server to bring the new HugePages system configuration into effect. Check to make sure that you have sufficient HugePages by starting all the database instances at the same time.

Starting with Oracle Database 11*g* Release 2 (11.2.0.2), a message is logged to the database alert log when HugePages are being used, for example:

```
****************** Huge Pages Information *****************
Huge Pages memory pool detected (total: 18482 free: 17994)
DFLT Huge Pages allocation successful (allocated: 4609)
***********************************************************
```

In this case, 4609 HugePages were used.

4.1.7.4 Handle Database Password Expiration

The default behavior of Oracle Database has changed in release 11*g* such that database user passwords will expire after 180 days. Processes should be put in place to refresh passwords regularly or expiration should be extended or disabled. PeopleSoft application availability will be impacted if passwords are allowed to expire. Password expiration for the default user profile can be disabled with the following command:

```
alter profile default limit password_life_time unlimited;
```

If passwords are not managed, a non-fatal error message ORA-28002, will be emitted PASSWORD_GRACE_TIME days prior to when the password will expire. The default is 7 days in the DEFAULT profile. The PeopleSoft application server however, sees this as a fatal error and sessions will begin to fail. The application server will fail to start but the password has not yet expired. The error message will show up in the APPSRV.LOG file. This will most likely impact all database schemas that PeopleSoft uses: PS, PEOPLE and the owning schema of the PeopleSoft application objects created at install time. If you encounter this error, you will need to know or have access to the passwords for all of the schema users and reset them as follows (do not change the passwords):

```
alter user <user name> identified by <use the same password>;
```

For example:

```
alter user PEOPLE identified by <use the same password>;
```

Please refer to “Configuring Password Protection” in the *Oracle Database Security Guide.*

4.1.7.5 Configure Dead Connection Detection

When a PeopleSoft Server node fails suddenly there may not be time for the operating system to reset the TCP connections and as a result the connections on the database server will remain open. To clean up the “dead” connections it is recommended that Dead Connection Detection is configured. See My Oracle Support note 151972.1 “Dead Connection Detection (DCD) Explained” for details.

Making these configuration changes may have adverse effect on network utilization and so all changes should be tested and monitored carefully.

4.1.7.6 Reduce Timeout on Oracle RAC Node Failure (Exadata Only)

On Exadata it is possible to failover more quickly in the event of an Oracle RAC node failure by reducing the misscount parameter. The parameter defines how long to wait after a node becomes unresponsive before evicting the node from the cluster. The parameter should not be set to less than 30 (30 seconds). To update the CSS misscount setting, log in as the root user on one of the database servers and run the command:

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

Later releases of Exadata software now implement CSS misscount to 30 seconds by default.

4.2 PeopleSoft High Availability Architecture

The following diagram illustrates a PeopleSoft high availability architecture

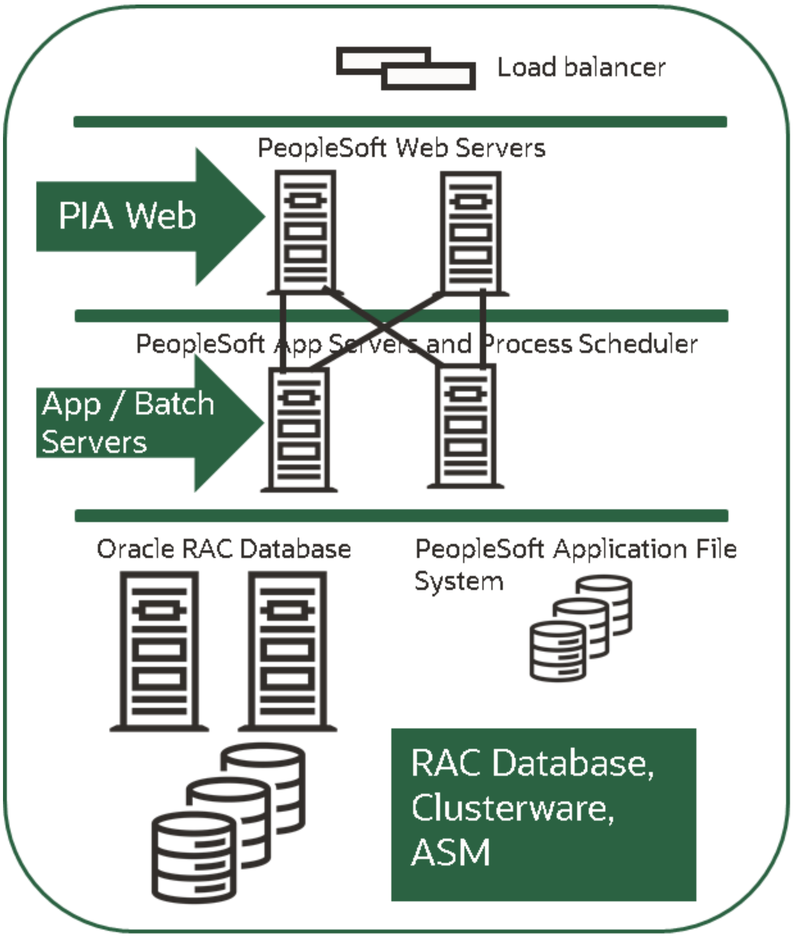
Load balancer
PeopleSoft Web Servers
PIA Web
PeopleSoft App Servers and Process Scheduler
App / Batch Servers
Oracle RAC Database
PeopleSoft Application File System
RAC Database, Clusterware, ASM

Figure 5. PeopleSoft High Availability Architecture

In this section and as illustrated in Figure 5 above, we discuss the high availability deployment of the PeopleSoft application that is layered on top of the Oracle Database MAA foundation.

4.2.1 PeopleSoft Application Software High Availability Deployment

PeopleSoft application components can each be deployed in a highly available manner. We recommend more than one instance of each component be deployed at each site, on separate physical servers so a server outage does not affect availability. We recommend the servers have adequate capacity to run peak load even when one server is down.

- A load balancer is used to balance web traffic across the web servers. The hardware load balancer has dual switches for redundancy.
- Two or more PeopleSoft Pure Internet Architecture (PIA) web servers for workload distribution and redundancy. Web server sessions accumulate state; thus, their routings are “sticky” for a session – once a user is routed to a web server, all future requests for that session are routed to the same web server. If the web server fails the user will be routed to a new web server but will have to re-authenticate and re-start their uncommitted work.

  Note: Web server resiliency can be achieved by implementing a Coherence cluster where session state can be preserved. Our case study did not implement t a Coherence cluster.
- As depicted in the above diagram (Figure 5), each PIA web server connects to and are load balanced across a pair of application domain servers. Should an application domain server become lost, its requests will be routed to its alternate application domain server. A delay can be observed if the node hosting an application

domain becomes unavailable and the PIA web server establishes its routes the request to the remaining application server.

- While the above illustration shows each PIA web server connecting to a pair of application domain servers, they can connect to any number of active application domain servers.
- Any number of application domain servers can be configured to service various requests. It is at this layer where the bulk of the business logic is executed. As there is no session state at this level, loss of an application domain server does not result in a need for user rework. The application domain servers connect to the database using role-based database services.
- A pair of PeopleSoft Batch Process Schedulers can be configured as master, with one being active and one idle. Any number of "slave" Batch Process Schedulers can be configured. If the active master Batch Process Scheduler goes down, the idle master takes over the task of assigning jobs to the slave process schedulers. If both go down, the slave process schedulers become stand-alone, doing the work already assigned to them, but not assuming the role of master. The process schedulers connect to the database using role-based database services. In Figure 5 above, the Batch Scheduler domains are hosted on the same servers as the application servers.

4.2.2 PeopleSoft Application File System Layout and Deployment

PeopleSoft HRMS 9.2 U033 and PeopleTools 8.57.11 is implemented in the case study described in the next section. The PeopleSoft applications and infrastructure software components can be deployed in two different ways that affect how the system will be managed. Before describing the deployment options, a few terms are defined here:

- PS_HOME: An environment variable that defines the file system location in which the PeopleTools software is installed.
- PS_APP_HOME: An environment variable that defines the file system location in which the PeopleSoft application (HRMS, FIN, EPM, etc.) is installed. It is common to have installed the application into the PS_HOME (PeopleTools) location as it was not until PeopleTools 8.52 and later when the application could be installed into a separate location.
- PS_CFG_HOME: An environment variable that defines the file system location for the application server domains, web server domains, their respective configuration files, and log files.
- PS_CUST_HOME: An environment variable that defines the file system location for custom code and files for adding additional customized functionality for PeopleSoft.
- COBDIR: An environment variable that defines the file system location in which the Micro Focus Server Express COBOL compiler and run-time libraries are stored. It is required that Micro Focus Server Express be installed on all servers that will run COBOL programs. This component cannot be shared due to license key restrictions.

The PeopleSoft software can be installed in one of the following ways:

Local Homes: A deployment paradigm where all of the PeopleSoft software and its required infrastructure components such as Oracle JDK/JRE, Oracle WebLogic Server, Oracle Tuxedo, PeopleTools and the PeopleSoft applications are installed on each server that will host PeopleSoft.

Shared Homes: A deployment paradigm in which all of the PeopleSoft software and its required infrastructure components such as Oracle JDK/JRE, Oracle WebLogic Server, Oracle Tuxedo, PeopleTools and the PeopleSoft applications are installed in a single shared file system location that all nodes in the deployment can access. In this deployment option, the PS_HOME environment variable on all nodes point to the same file system directory location. Alternatively, you can install the PeopleSoft application in a separate shared location and set the PS_APP_HOME environment variable on all nodes to that shared file system location.

The following table outlines the advantages and disadvantages of shared or local homes.

TYPE	ADVANTAGES	DISADVANTAGES

Local Homes	Isolate maintenance to a single or group of PeopleSoft server nodes. Isolate diagnostics to a single or group of nodes which might require patching without affecting other nodes or requiring out of place patching.	Requires PeopleSoft software to be installed on every node which prolongs the install time. Requires patching and software upgrades to be performed on each individual nodes prolonging maintenance time. While some of the patch installs can be performed in parallel, there are components that require user interaction to respond to prompts preventing the entire process from being performed in parallel.
Shared Homes	Installation of PeopleSoft software is performed once on one node; all other nodes share the installation reducing install time. Significantly reduces setup and deployment time of the secondary DR site if the shared homes are replicated. No installation (except for COBOL) is required at the secondary site. Out-of-place patching and upgrades staged and performed once on a single node; all other nodes are restarted reducing maintenance time.*[1]	In-place patching and upgrades can impact all nodes. Loss of the shared storage will result in an application outage.

As discussed in the next section, it is recommended that you deploy PeopleSoft using the shared home paradigm.

In addition to the software installation shared homes, PeopleSoft generates reports and writes them to the PeopleSoft Report Repository. This is a file system directory location where the batch jobs and the application engine place output logs and reports. The report repository is also shared across all application and web servers at the primary site.

4.2.3 Replication of PeopleSoft Shared Home and Report Repository

Data loss is not limited to only the database. Most applications have inbound and outbound data feeds, reports, etc. PeopleSoft HRMS Payroll generates pay slips in the form of files that are sent to, for example, Automated Data Processing (ADP) for electronic check processing and deposits. Should a site failure occur, if the files were not replicated, a loss of inbound and outbound files will occur and a gap in which files were sent (or not sent) must be resolved. Oracle ZFS Storage Appliance within the Private Cloud Appliance provides a replication service that can be used to minimize loss of the files needed by the application tier as discussed later in this paper.

4.2.4 PeopleSoft Application and Web Tier Configuration Best Practices

[1] Upgrades may require the application server domains and web server domains to be re-deployed on each node. Even so, this takes less time than performing software installs on each node.

We recommend the following for configuring the PeopleSoft application:

4.2.4.1 Database Connection for Application Domain Servers

The application domain servers should connect to the database using a TNS connect string that contains address of both the primary and standby database and connect using role-based database services described above. Here are examples:

```
# HR Online users
HR92U033 =
 (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)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
 )

# Batch scheduler
HRBATCH =
 (DESCRIPTION_LIST =
   (LOAD_BALANCE=off)(FAILOVER=on)
   (DESCRIPTION =
       (ADDRESS_LIST =
       (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan1)(PORT = 1521))
       )
       (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_BATCH)
       )
    )
   (DESCRIPTION =
       (ADDRESS_LIST =
       (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
       (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_BATCH)
       )
    )
 )


# Active Data Guard
PSFTADG2 =
 (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)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = PSQUERY)
       )
    )
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = PSQUERY)
       )
    )
 )
```

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
- Allows the application to failover to the standby (or local standby) automatically

4.2.4.2 Configure the PS.PSDBOWNER Table

PeopleSoft treats the TNS connect string alias as the actual name of the database regardless of what the database DB_NAME or DB_UNIQUE_NAME is set to. The PeopleTools layer checks to see if the database it is connecting to matches the "DBName" of its server domain configuration. The DBName must be set to the TNS connect string alias name. The TNS connect string alias name must be inserted into the PS.PSDBOWNER table along with the name of the PeopleSoft schema name that owns all of the PeopleSoft objects. Without these entries, the application domain server will not start.

The rows in the PS.PSDBOWNER table will be similar to the following:

```
DBNAME            OWNERID
--------------    ----------------
HR92U033          EMDBO
HRBATCH           EMDBO
PSFTADG2          PSFTADG2
```

Further discussion on this topic can be found in Appendix section 10.2.5.

4.2.4.3 Ensure All Log Files Are Not Written to a Shared PS_HOME

Make sure that the application domain servers and the PIA web servers write to PS_CFG_HOME instead of PS_HOME. This ensures that no errors or loss of log files occur should the shared PS_HOME become a read-only file system after a switchover or role reversal.

4.2.4.4 Configure PeopleSoft Report Repository

Use a shared file system for the PeopleSoft report repository. Place the log_output directory in the shared report repository. Also, ensure that the PIA web servers can access the same shared file system containing the report repository.

4.3 PeopleSoft MAA Site State Model and State Transitions

Figure 6 below shows the states that a deployment goes through as it progresses from the initial single site implementation through the setup, testing, and an eventual dual site MAA deployment. The systems have a specific configuration in each state and there is a set of documented steps to move from one state to the next.

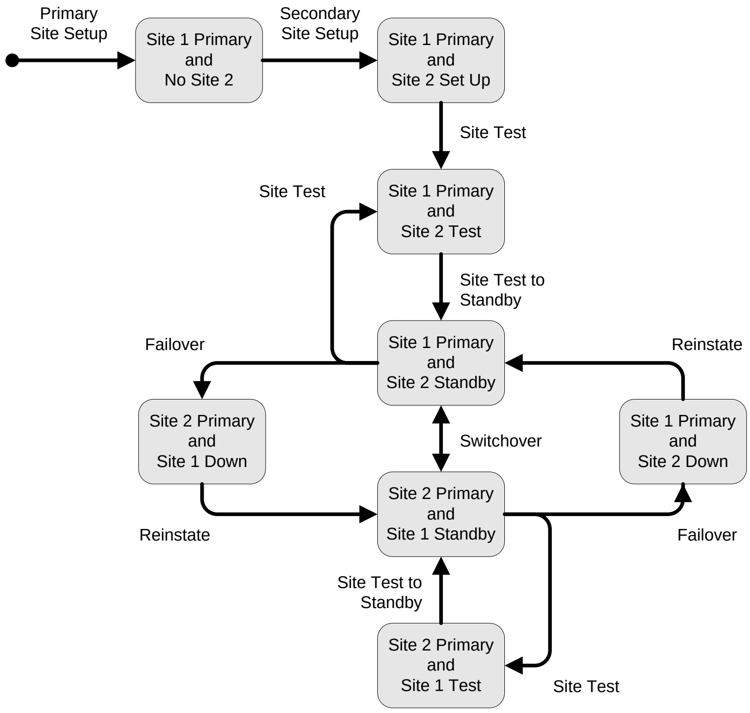
Primary Site Setup
Site 1 Primary and No Site 2
Secondary Site Setup
Site 1 Primary and Site 2 Set Up
Site Test
Site Test
Site 1 Primary and Site 2 Test
Site Test to Standby
Failover
Site 1 Primary and Site 2 Standby
Reinstate
Site 2 Primary and Site 1 Down
Switchover
Site 1 Primary and Site 2 Down
Site 2 Primary and Site 1 Standby
Reinstate
Failover
Site Test to Standby
Site 2 Primary and Site 1 Test
Site Test

Figure 6. PeopleSoft MAA Site State Model and State Transitions

A summary description of the state transitions is provided in the following table.

Table 1. Description of State Transitions

TRANSITION	DESCRIPTION
Primary Site Setup	Install and configure the primary site.
Secondary Site Setup	Establish the secondary site.
Site Test	Prepare the standby site for a site test.
Site Test to Standby	Convert the site performing a site test back to standby mode.
Switchover	Switch the roles so that the current standby becomes the primary and the current primary becomes the standby.
Failover	Switch the current standby to primary mode. The current primary is assumed to be down or unavailable.
Reinstate Standby	Reinstate the old primary as a standby after failover.

The following table summarizes how the system databases and file systems are configured in each state.

SITE STATE	PEOPLESOFT DATABASE - DATA GUARD	PEOPLESOFT SHARED HOMES AND REPORT REPOSITORY - REPLICATION
Site 1 Primary and No Site 2	Not configured	Not configured
Site 1 Primary and Site 2 Set Up	Site 1 primary and site 2 physical standby. Snapshot standby, (Oracle Active Data Guard) during setup.	Site 1 primary with continuous replication to site 2. Site 2 snapshot during setup.
Site 1 Primary and Site 2 Test	Site 1 primary and site 2 snapshot standby.	Site 1 primary with continuous replication to site 2. Site 2 snapshot created for test.
Site 1 Primary and Site 2 Standby	Site 1 primary and site 2 physical standby (Oracle Active Data Guard).	Site 1 primary with continuous replication to site 2.
Site 2 Primary and Site 1 Down	Site 2 primary through failover, and site 1 down.	Site 2 primary established from replica, and site 1 down.
Site 2 Primary and Site 1 Standby	Site 2 primary and site 1 physical standby (Oracle Active Data Guard).	Site 2 primary and continuous replication to site 1.

Site 1Primary and Site 2 Down	Site 1 primary through failover and site 2 down.	Site 1 primary established from replica, and site 2 down.
Site 2 Primary and Site 1 Test	Site 2 primary and site 1 snapshot standby.	Site 2 primary with continuous replication to site 1. Site 1 snapshot created for test.

PeopleSoft PeopleTools versions 8.52 and later can optionally support Oracle Active Data Guard if the PeopleSoft application domain is configured with a “secondary” PeopleSoft access ID. The above table denotes this with “(Oracle Active Data Guard).”

4.4 Planned and Unplanned Outage Solutions

This section summarizes the outages that may occur in a PeopleSoft environment and the Oracle solution that is used to minimize application downtime. In all cases, we are focused on PeopleSoft Application downtime as perceived by the end user, not the downtime of the individual component.

4.4.1 Unplanned Outage Solutions

Table 2 describes the unplanned outages that may be caused by system or human failures in a PeopleSoft environment and the technology solutions that would be used to recover and keep downtime to a minimum.

Table 2. Unplanned Outage Solutions

OUTAGE TYPE	ORACLE SOLUTION	BENEFITS	RECOVERY TIME
PeopleSoft PIA Web Server Node or Component Failure	Load Balancing	Surviving nodes pick up the slack	Affected users re-authenticate and resubmit work
	Redundant Web Servers	Surviving nodes continue processing	No downtime
PeopleSoft Application Domain Server Node or Component Failure	PIA servers configured with active connections load balanced across application servers Redundant application domain servers	Surviving nodes pick up the slack	No downtime
Database Node or Instance Failure	Oracle RAC	Automatic recovery of failed instance, FAN events, transparent application and service failover	Users transparently fail over Updates may need to be re-submitted
Site Failure	Data Guard	Fast Start Failover	< 5 minutes for database role transition and

			PeopleSoft application startup
Storage Failure	ASM	Mirroring and automatic rebalance	No downtime
	RMAN with flash recovery area	Fully managed database recovery and disk based backups	Minutes to hours
	Data Guard	Fast Start Failover	< 5 minutes
Human Error	Oracle Flashback	Database and fine grained rewind capability	Minutes
	Log Miner	Log analysis	Minutes to hours
Data Corruption	RMAN with fast recovery area	Online block media recovery and managed disk-based backups	Minutes to hours
	Oracle Active Data Guard	Automatically detects and repairs corrupted blocks using the physical standby database	No downtime, transparent to application
	Data Guard	Automatic validation of redo blocks before they are applied, fast failover to an uncorrupted standby database	Seconds to 5 minutes

The recommended solution for site failure is Data Guard configured with Data Guard Broker to simplify the process of switchover or failover. In the event that the primary site is lost, it is best to assess the nature of the failure and if the primary can be recovered quickly. As there will most likely be other system integrations into PeopleSoft, a failover decision should be left to decision makers for your business. However, the failover process itself should be fully automated once it has been decided to perform a failover.

If configured with Fast-Start Failover, so long as the standby database apply lag is within the fast start failover Lag limit, then the time to bring up the DR site will depend on the fast start failover timeout threshold, the time to fail over the standby to a primary role, and the time to start all of the PeopleSoft application and PIA web servers. PeopleSoft application servers, batch scheduler and PIA we servers can all be started in parallel to reduce service restoration time.

For storage failure, if on-disk backups are kept on a separate storage array such as an Oracle ZFS Storage Appliance, it is possible to have the restore take minutes to complete.

4.4.2 Planned Maintenance Solutions

Table 3 summarizes the planned maintenance activities that typically occur in a PeopleSoft environment and the recommended technology solutions to keep downtime to a minimum.

Table 3. Planned Outage Solutions

MAINTENANCE ACTIVITY	SOLUTION	PEOPLESOFT OUTAGE
Mid-Tier operating system or hardware upgrade	Hardware Load balancing, Redundant services across Web and Tuxedo Application Servers	No downtime

PeopleSoft application patching (application tier only)	PeopleSoft out-of-place patching	Minutes to hours
PeopleSoft application configuration change	PeopleSoft application rolling restart	Minutes
PeopleSoft upgrades	PeopleSoft out-of-place upgrades	Hours to days (depending on database size)[2]
Database tier operating system or hardware upgrade	Oracle RAC	No downtime
Oracle Database interim patching	Oracle RAC rolling apply, Standby-First	No downtime
Oracle Database online patching	Online patching	No downtime
Oracle Grid and Clusterware upgrade and patches	Rolling apply / upgrade	No downtime
Database storage migration	Oracle ASM	No downtime
Migrating to Oracle ASM or migrating a single-instance database to Oracle RAC	Data Guard	Seconds to minutes
Patch set and database upgrades	Release 11.2 or greater: Data Guard transient logical rolling upgrade	Seconds to minutes

[2] In practice, there are a number of ways to mitigate the impact of extended upgrade downtime, for example, by providing a read-only replica. Oracle Consulting Services can help you plan and execute the upgrade.

5 PeopleSoft MAA Case Study on Private Cloud Appliance and Exadata

In this section we describe how the PeopleSoft Maximum Availability Architecture described in the earlier chapters was deployed on a system consisting of Oracle Private Cloud Appliance (PCA) and Exadata machines. Oracle Site Guard was also installed and configured to manage the site transitions. A high-level view of the configured system is pictured in Figure 7 below:

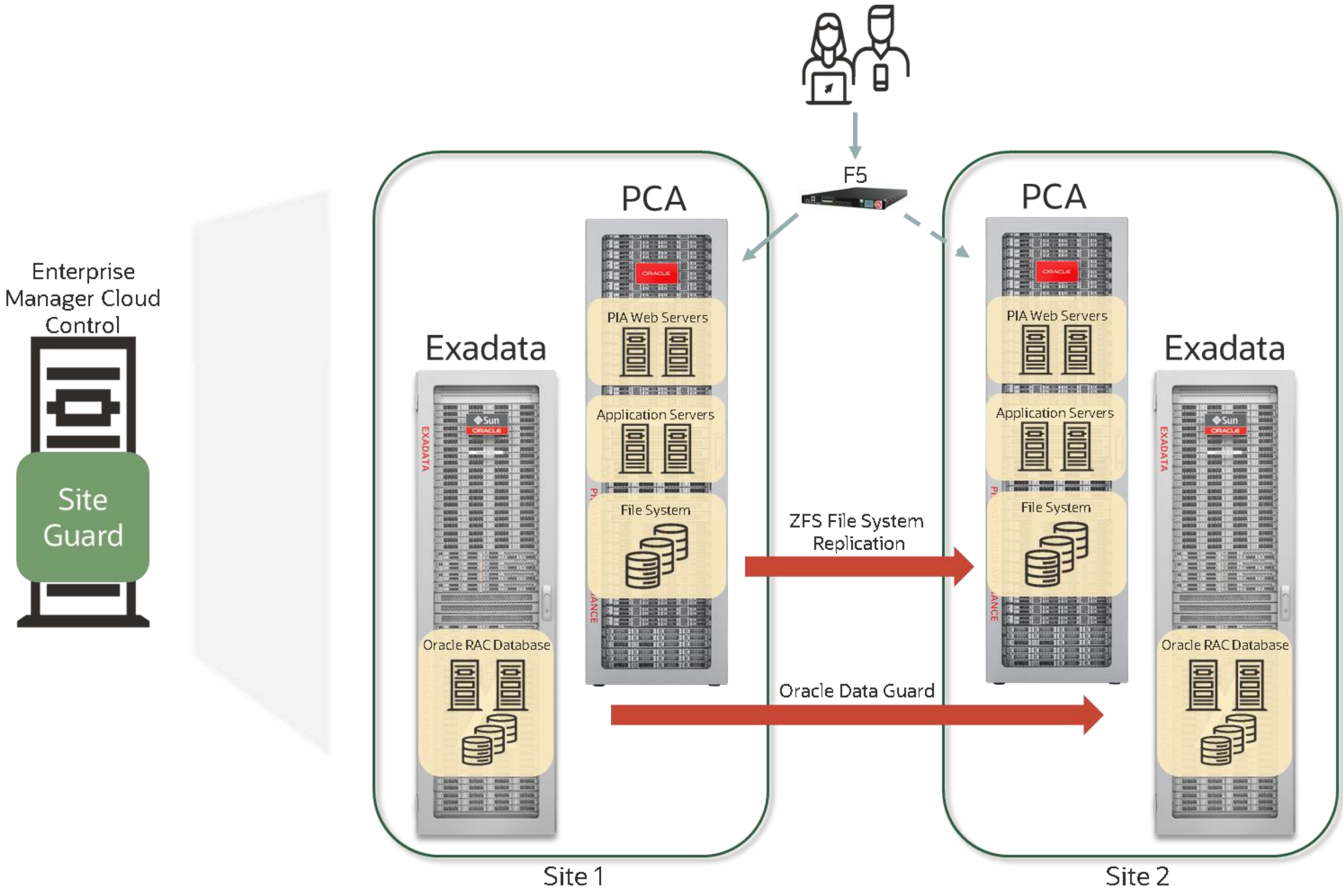

Figure 7. PeopleSoft full MAA implementation

Figure 7 illustrates a primary and a secondary site. Each site is configured such that each can assume the primary or standby role whether planned or unplanned. At each site, the web, application, and batch servers and the middle tier file system reside in the Oracle PCA machine; the database servers and storage reside in the Oracle Exadata Database Machine. A separate server is used to host Oracle Enterprise Manager with the Site Guard plugin that automates the full stack switchover or failover.

5.1 Systems

The following systems were used for the case study:

SYSTEM	SITE 1	SITE 2
F5	F5 BIG-IP 4200	
PCA	X8-2	X5
Exadata	X7-2 Quarter Rack	X7-2 Quarter Rack

5.2 Software

The following software was used for the case study:

SOFTWARE	VERSION
PCA	2.4.3
PeopleSoft HCM Application	9.2 U033
PeopleTools	8.57.11
Micro Focus COBOL	Micro Focus COBOL Server Express 5.1 WP14
Exadata Database	19.8
Exadata Grid Infrastructure	19.8
Database Client	19.3
Exadata	20.1.1
F5 BIG-IP Local Traffic Manager	15.1.0 Build 0.0.31

5.3 F5 Networks BIG-IP Local Traffic Manager

F5 BIG-IP hardware load balancers at each site are used for distributing traffic across PeopleSoft PIA web servers. The BIG-IP Local Traffic Manager (LTM) was configured at each site. The health of each web server within the BIG-IP server pool is monitored at the TCP layer as well as at the application layer. The application level monitor is a user-defined monitor to determine the health of the PeopleSoft web servers. The combination of the built-in TCP monitor provided by F5 Networks and the user-defined health monitor minimizes the impact on users should a web server node fail or if the web server is stalled as discussed later.

5.4 Database Setup

Whether created afresh or migrated to the new infrastructure, the PeopleSoft database should be configured following the PeopleBooks documentation as well as implementing the Oracle Exadata Database Machine best practices. For this case study, the PeopleSoft database for HRMS 9.2 U033 was copied from a test system onto Oracle Exadata Database Machine using the Recovery Manager (RMAN). Once there, Oracle Exadata Database Machine best practices for OLTP applications were implemented. For full details of these best practices, please see My Oracle Support note 1067527.1.

The standard Exadata configuration was deployed at both the primary and secondary sites. This includes the following:

- ASM disk groups (DATA and RECO) with HIHG redundancy
- PeopleSoft database configured with Oracle RAC across both nodes of the X7-2 quarter rack Exadata Database Machine
- Database services registered in the Oracle Cluster Ready Services
- For connection load balancing, the pre-configured SCAN listener is used

5.4.1 Database Features for the DR Site

The traditional role of a DR site has typically been passive – hardware and software that sits idle until it is needed. While a DR site serves to protect against data loss due to a site failure, it is a large investment for this one critical benefit. Oracle technology has evolved to leverage that investment of the DR site to further increase return on investment (ROI) and to utilize what would otherwise be idle hardware. The past several years has provided the following to achieve this benefit:

- **Oracle Active Data Guard** allows for off-loading queries for reporting, decision support, OLAP and ad-hoc queries to the physical standby database leaving more capacity for the production OLTP system. PeopleSoft, as of PeopleTools 8.52, supports Oracle Active Data Guard. If the application domain is configured to support Oracle Active Data Guard, then some Application Engine (AE) and PSQUERY reports can be configured to run against an Oracle Active Data Guard standby database.
- Oracle Active Data Guard is discussed in this paper as being implemented at the DR site. If the DR site is located at a substantial distance away from the primary (on two separate continents with several thousand miles in between), then using Oracle Active Data Guard at such a DR site might not be a viable option given the high network latency. Implementing a local standby database for Oracle Active Data Guard and a remote standby is an ideal solution. This provides a local Oracle Active Data Guard standby, a local standby database for switchover for maintenance plus a remote standby database for site failover.
- PeopleSoft supports both Data Guard logical and physical standby databases. Oracle GoldenGate one-way replication is also supported. For the purposes of this discussion, we will focus on Data Guard physical standby database.
- Database Snapshot Standby allows for the physical standby to be opened read-write without compromising data loss service level agreements (SLAs). This allows the entire application technology stack to be started for testing and maintenance. Once the testing is completed, the database is returned to its role as a physical standby and all outstanding changes from the production database are applied.
- Once the secondary site was established, the standby database was instantiated using the RMAN RESTORE FROM SERVICE feature, Data Guard was configured with redo shipping from the primary and redo apply at the standby. Data Guard Broker was used to configure and enable Data Guard.

5.4.2 Database Service Design

The database services used by the PeopleSoft application server are

DATABASE SERVICE NAME	DATABASE ROLE	PURPOSE
HR92U033_BATCH	PRIMARY	Production process scheduler
HR92U033_ONLINE	PRIMARY	Production application server
PSQUERY	Oracle Active Data	Offload read-only queries

	Guard Standby	

These services are configured in the database with Fast Application Notification (FAN). If one of the Oracle RAC database instances should become unavailable, the surviving RAC instance sends all previously connected clients an out-of-band FAN event causing the clients to drop their current TCP connections and reconnect via transparent application failover (TAF) to the services running on the surviving RAC instance, expediting the failover process. This same functionality exists for a failover to a physical standby database. The application domain servers do not need to be restarted after a failover.

In addition to these services being FAN enabled, they are also defined as role based services depending upon the database role (PRIMARY, PHYSICAL STANDBY, SNAPSHOT STANDBY) that determines if a service will start (or not start).

5.5 Delegation of Roles and Administration

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

5.5.1 Administrative Roles on Oracle Exadata Database Machine

On Oracle Exadata Database Machine, the Oracle Grid Infrastructure (Oracle Clusterware and Automatic Storage Manager – ASM) manages all cluster and storage services. Although not required, it is recommended that you install the Oracle Grid Infrastructure using a separate and dedicated OS user for example, “grid”. Application databases should be installed into their own OS user account so that the grid infrastructure is managed separately from that of the PeopleSoft application database.

The following table illustrates how this was configured in this case study.

OS USER	OS GROUPS	ROLE
grid	oinstall, dba	Clusterware and ASM administrator (Grid)
oracle	oinstall, dba_psft, dba	PeopleSoft database administrator

Notice that the oinstall and dba OS groups are common between the “grid” OS user account and that of “oracle”. This allows the PeopleSoft database being managed by the oracle user to access the ASM services.

5.5.2 Administrative Roles on Oracle Private Cloud Appliance

On Oracle PCA, PeopleSoft PeopleTools and HRMS have been installed into the OS user account oracle_psft with group oinstall. This installation includes all of the infrastructure needed for PeopleSoft shown in the following table.

OS USER	OS GROUPS	ROLE
oracle_psft	oinstall	PeopleSoft Application and PIA administration

Micro Focus Server Express COBOL compiler and runtime environment should be installed on local storage on each node as root and must have its license manager configured and running on each node that will run COBOL code.

5.6 PeopleSoft Application and Web Tier Setup

In this case study, PeopleSoft was installed using a shared PS_HOME . The application servers and PIA web servers access the same software install locations. These servers also access the PeopleSoft report repository. For this case study, HRMS 9.2 was installed in a separate location from PS_HOME and pointed to by PS_APP_HOME. The PS_APP_HOME is also shared across application and web server tiers.

5.7 Oracle Appliance ZFS Storage and Shared File System

All of the shared and local file systems are stored on the Sun ZFS Appliance, which comes installed as part of the Oracle Private Cloud Appliance. The file systems are exported from ZFS and are mounted with NFS v3 by all mid tier servers. Specific shared file systems are replicated to the secondary DR site using ZFS replication. The following table shows each file system, mount points and state for each site.

PURPOSE	SHARE TYPE	EXPORTED AS (MOUNTED AS)	SITE STATE	COMMENTS
PeopleSoft Software Install File System – PS_HOME	Local Replicated	Exported as: /export/psoft Mounted as NFS v3 and located at: /u01/app/psft/pt/ps_home8.57.11	Site 1 Primary Site 2 Replica	Replicated to site 2, primary on site 2 post switchover/failover
PeopleSoft HCM 9.2 application install – PS_APP_HOME	Local Replicated	Exported as: /export/psoft Mounted as NFS v3 and located at: /u01/app/psft/pt/hcm_app_home	Site 1 Primary Site 2 Replica	Replicated to site 2, primary on site 2 post switchover/failover
PIA Report Repository	Local Replicated	Exported as: /export/psoft Mounted as NFS v3 and located at: /u01/app/psft/pt/psft_reports	Site 1 Primary Site 2 Replica	Replicated to site 2, primary on site 2 post switchover/failover
Server Specific Configuration File System – PS_CFG_HOME	Local Not Replicated	Mounted as local storage at: /peoplesoft/local/ps_config	Site 1 Primary Site 2 Primary	The application Tuxedo and the PIA web server domains configurations are placed here.
Required COBOLB Server Express	Local Not	Mounted as local storage /opt/MFCobol	Site 1 Primary Site 2	The COBOL compiler and run-time environment

installation-COBDIR	Replicated		Primary	installed on each app server VMs at both sites. Not replicated

6 PeopleSoft Unplanned Outage Behavior

6.1 PeopleSoft Failover Behavior

Table 4 summarizes the PeopleSoft behavior during Oracle RAC or Data Guard failover when client failover is configured. Except for a short pause as the failover occurs, the failure is transparent to the end user in most cases:

TABLE 4. PEOPLESOFT FAILOVER BEHAVIOR DURING ORACLE RAC OR DATA GUARD FAILOVER

PeopleSoft Client Operations	Behavior
End user is inserting, updating, or deleting data and submits or saves the inserts/updates/deletes during or just after the database failure.	The data manipulation language (DML) will fail. Transactions will not get resubmitted. Oracle reconnects and reconstructs the database session on a surviving node and the end user must resubmit the transaction.
End user is paging through queried data (SELECTs) when the database failure occurs.	Oracle reconnects and reconstructs the database session on a surviving node, re-executes the query, repositions the SQL cursor, and returns the next set of rows.
End user is issuing a new query (SELECTs) or switching screens just after the database failure.	Oracle reconnects and reconstructs the database session on a surviving node.

Table 5 summarizes the failover behaviour of the PeopleSoft Batch Process Scheduler, Application Engine (AE) jobs, Structured Query Report (SQR), PeopleSoft Query (PSQuery), XML Publisher (XMLP), and COBOL programs.

TABLE 5. PEOPLESOFT FAILOVER BEHAVIOR DURING CLIENT BATCH OPERATIONS

PeopleSoft Client Batch Operation	Behaviour
Process Scheduler	Oracle reconnects and reconstructs the session on a surviving node. The process scheduler fails over with no administration intervention required.
Application Engine (AE) job submitted just *BEFORE* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node but AE jobs may fail and show up in the PeopleSoft Process Monitor as “No Success”. These jobs must be resubmitted. If the AE job has been implemented to be restartable, then the process scheduler will automatically restart the job.[3]
Application Engine (AE) submitted *during* or just *after* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node, the AE job is then submitted on the

[3] If the AE job was not in an open transaction and the job was performing only `SELECT` statements, then it will fail over and complete successfully.

	surviving RAC node (or the new primary database failover) and completes successfully.
COBOL jobs just *before* primary instance failure	If the COBOL program runs pure queries (`SELECT` statements), then it will fail over to the surviving node and complete successfully. If the COBOL program executes `INSERT`s, `UPDATE`s, and `DELETE`s, it will *not* complete successfully on the surviving node. Manual intervention is required to restart the COBOL jobs.
Crystal and SQR reports	The behavior is the same as COBOL
PSQUERY, Tree Viewer, XMLP Viewer	These PeopleSoft components will fail over and complete successfully.

6.2 PIA Web Server Unplanned Outage

The Pure Internet Architecture (PIA) web servers maintains session state for all active sessions. Failure of a given PIA web server will result in loss of session state and errors for those sessions. Therefore, downtime is not measured here as they will need to log back on and re-authenticate. When front-ended by the F5 load balancer, the number of users who encounter errors depends on the nature of the failure. Here are the following scenarios:

- **Scenario 1:** Active user sessions logged directly on a web server that fails: All such users will fail and must re-login to a separate server or wait until the service is restored on the failed server.
- **Scenario 2:** Active user sessions logged on via F5 load balancer and a PIA web server instance fails, but the hardware remains up: In this case the number of failures depends on the implemented F5 health monitor, its timeout values and the number of active sessions at the time of the failure. In our case, we have implemented a simple HTTP “GET /” with a timeout value of 60 seconds. If this timeout is reached, the load balancer removes the PCA VM hosting the PIA web server from the server pool and returns all outstanding HTTP requests with “reject” errors. During the timeout period, users are blocked until the timeout timer expires. The longer the timeout, the higher the error count will be.
- **Scenario 3:** Users logged on via F5 load balancer and PIA web server node fails: In this scenario, the F5 load balancer internally pings the PCA VMs with ICMP pings to determine the health of the network TCP layer. This ping occurs at a fixed interval of 5 seconds. If the pings fail to return for up to 16 seconds, the node is removed from the server pool, and all outstanding HTTP requests are returned with “reject” errors. This case presents a smaller window of time for blocked sessions from piling up and consequently, fewer session errors will occur.

Testing and care must be taken when implementing a health monitor to check the status of the web servers. Remember that this health monitor is designed for when the PIA web server experiences a failure or have higher response times such as when an application server domain fails, thus, response times can increase while the PIA reconnects to an alternate application domain server.

Setting the timeout value depends on the average length of time it takes to recover the web services, not including node or hardware failures, where ICMP pings would fail and force the web server to be removed from the server pool.

A PIA web instance failure will always result in all active (non-idle) sessions encountering failures no matter what the health monitor timeout is.

6.3 Application Server Unplanned Outage

The PeopleSoft Application Server failures result in some amount of transaction delay depending on the nature of the failure. An abrupt and forced shutdown of the application server (with `shutdown!`) brings down the Oracle Tuxedo application services along with all of the application threads that were running. In doing so, the network sockets used by the connected JOLT sessions are removed causing the PIA web server to detect and react to the failure. Our results show a one second increase in response times at the PIA web server during the one second time period the PIA server took to reconnect or reroute the request to an alternate application server.

A failure of the node hosting one of the application servers (for bare metal) shows a much longer downtime as there are no network events that the PIA web server can react to. The JOLT sessions will hang until the JOLT layer times out at which point the PIA re-establishes its JOLT connections to an alternate application server. No online user encountered errors.

A failure of the PCA VM (virtualized) node hosting an application server will result in a much smaller delay. To cause a failure, the `xm destroy` Xen command was used. When this command was used to destroy a PCA VM running an active application domain server, the PIA JOLT sessions were able to quickly detect and re-establish connections or reroute requests to an alternate application server. As part of the PCA VM tear down, the virtual network interfaces are de-allocated that causes a socket close network event transparent to online users.

6.4 RAC Database Instance Unplanned Outage

The database services that the PeopleSoft application server uses to connect are FAN enabled. A RAC instance failure of the databases causes Clusterware to issue an INSTANCE DOWN event that the application server connection pool receives and then reconnects to one the remaining Oracle RAC instance where the service is running.

All recent generations of Exadata supports a feature known as *Instance Failure Detection* formally known as *Fast Node Death Detection (FNDD)* to evict an Exadata compute node out of the cluster should that node become unavailable, unresponsive or otherwise, considered to be unhealthy. This is achieved by probing the compute node on its private network ports to determine if it is responding or not. This technique will reduce delays and downtime since the eviction will force a compute node reboot and re-join the cluster. The PeopleSoft application servers will react accordingly to FAN events it receives.

All recent versions of the Exadata software now set the Clusterware CRS MISSCOUNT to 30 seconds. This parameter was discussed earlier. In the scenario in which the node is responding to probes on both of its private network ports, yet, critical components of the cluster services are no longer responding, the compute node will be evicted after 30 seconds. Again, the PeopleSoft application servers will react accordingly to FAN events it receives.

7 Site Outage Testing and Results

All servers on the primary site were stopped abruptly and site failover was performed by Site Guard. All application users failed on site outage and the site failover procedure was followed to restore service on the standby site. The failover procedure was performed by Site Guard and the timing for each step are shown in the following table:

OPERATION	TYPICAL ELAPSED TIME
Failover Storage	1m 10s
Mount PeopleSoft File System	18s
Database Failover	2m 55s
Start PeopleSoft App Servers Startup (Post-Scripts)	56s

Start PeopleSoft Process Scheduler (Post-Scripts)	45s
Start PeopleSoft PIA Web Server (Post-Scripts)	25s
TOTAL	6m9s

Table 6: Site Outage Testing Results

Note: The total does not include the elapse time for when the first user can logon to PeopleSoft.

8 Best Practices

8.1 Exadata and Database Best Practices

- Review all of the relevant My Oracle Support notes. Specifically review My Oracle Support note 888828.1. This My Oracle Support note contains links to other My Oracle Support notes that should be reviewed.
- Configure Linux HugePages on each database server node. See My Oracle Support note 361468.1, "HugePages on Oracle Linux 64-bit".
- Set the database parameter USE_LARGE_PAGES='ONLY' to utilize HugePages.
- Run the latest version of Exachk on a regular basis and review its output report. See My Oracle Support note 1070954.1 latest version of Exachk. Correct any critical issues that are identified. Reviewed any items marked as "Failed" and if appropriate, correct those items. See Appendix A for items that can be ignored.
- Set CSS MISSCOUNT to 30 seconds for Oracle RAC clusters. If the application is critical and requires a faster Oracle RAC node failover, then the CSS MISSCOUNT can be set to 30 seconds. It is not recommended to set this parameter lower. See Appendix A for further details.
- Configure database role based services that are FAN enabled.
- Deploy PeopleSoft on an Oracle RAC database configured with role base services for database connections.
- Enable Flashback on both primary and standby databases. This allows for quick re-instantiation of a failed primary database if the database itself was not lost. This feature also allows for error correction without having to perform a restore from backups.
- Use Recovery Manager (RMAN) to back up the PeopleSoft database on a regular backup schedule. For example, perform full level 0 backups weekly and incremental level 1 backups daily.
- Configure Data Guard Broker to manage and monitor the Data Guard configuration.
- Enable Active Data Guard for read-only workloads. PeopleSoft can be configured for Active Data Guard.

8.2 Private Cloud Appliance and PeopleSoft Application Best Practices

- Configure PCA anti-affinity such that no two application servers or PIA web servers reside on the same physical PCA compute node.
- Make sure all PeopleSoft application domains and PIA web server logs are written to the PS_CFG_HOME directory location and not to PS_HOME or PS_APP_HOME which are shared homes.
- Use shared homes to reduce the time needed to deploy the secondary site and for performing out-of-place patching which also should reduce maintenance time.
- Ensure that the application server domains use FAN enabled role-based database services in their TNS connect strings.

- Configure the PIA web servers to connect to and are load balanced across application servers.
- Enable PIA_access.log with extended log format which can be configured using the Oracle WebLogic Server Administration console on the PIA servers.
- Take regular backups of PS_HOME, PS_APP_HOME, each local PS_CFG_HOME, and the report repository.

8.3 High Availability and Disaster Recovery Best Practices

- Deploy a second geographically separate site to serve as the disaster recovery site which can run the PeopleSoft workload.
- Always use multiple PeopleSoft servers for application, batch, and PIA web servers at both the primary and secondary DR site.
- Implement a Data Guard physical standby database for failover in case of a site failure or disaster.
- Use Oracle Active Data Guard with PeopleSoft so that higher leverage of the standby database can be realized by offloading reporting and adhoc queries to the standby database. If the DR site is a substantial distance away from the primary (on two different continents thousands of miles apart) and network latency is high, implement a local standby and configure it for Oracle Active Data Guard along with the remote Data Guard standby.
- Use Data Guard Broker to simplify the Data Guard standby database configuration.
- Use ZFS continuous replication for PeopleSoft PS_HOME, PS_APP_HOME and the report repository.
- ZFS replication should be configured at the project level as a prerequisite for Oracle Site Guard.
- Make sure the surviving nodes can handle the load that was on the failed node.
- Export the PS_HOME, PS_APP_HOME, and the report repository shared file systems at the DR site. It is not necessary to remount the file systems at the DR site when the DR site assumes the primary role. Their state will change from read-only to read-write.
- Set F5 load balancer health monitor timeouts appropriately according to your environment and testing.
- Conduct annual or semi-annual switchover testing between the primary and DR sites to validate processes and procedures.
- Leverage the DR site for testing using database snapshot standby and ZFS cloning of the PeopleSoft report repository.
- Use Oracle Site Guard to automate site switchovers and failovers.
- Always test patches and configuration changes in a test environment before promoting to production. This practice reduces risks by following proper change control and validation processes.

9 Conclusion

Deploying PeopleSoft on both Oracle Exadata Database Machine and Oracle Private Cloud Appliance serves as a common standard platform where unified deployments and management can be leveraged. MAA technologies and best practices are integrated at all layers of the stack. The use of both Data Guard and Oracle Cloud Appliance ZFS storage replication reduces the time to deploy the secondary site for disaster recovery as well as for switchover, while simplifying the requirements for failover. PeopleSoft has adapted database technologies such as FAN, TAF, and SCAN support, and is capable of remaining up for rolling maintenance activity reducing downtime. For performance, Exadata smart logging, smart scans, and write back flash cache provide higher throughput for PeopleSoft batch processes, and online users enjoy a consistent reliable performance.

The PeopleSoft applications deployed on Oracle Exadata Database Machine and Oracle Private Cloud Appliance realize a high level of HA when MAA best practices are followed. Our testing shows that zero users are impacted for Oracle RAC database instance failures and application domain server failures.

With Oracle Private Cloud Appliance and ZFS storage, physical deployment footprint can be reduced and at lower cost. ZFS replication provides the much needed safeguards for the middle tier where critical data feeds (inbound and outbound) can be replicated to a secondary site should a loss of the primary site occur.

10 References

- MAA on Oracle Technology Network (OTN)
  http://www.oracle.com/goto/maa
- Oracle Database High Availability Overview
  https://docs.oracle.com/en/database/oracle/oracle-database/19/haovw/index.html
- Oracle Real Application Clusters Administration and Deployment Guide
  https://docs.oracle.com/en/database/oracle/oracle-database/19/racad/index.html
- Oracle Data Guard Concepts and Administration
  https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/index.html
- Oracle Flashback Technology
  https://docs.oracle.com/en/database/oracle/oracle-database/19/adfns/flashback.html#GUID-03D1CAAE-D940-444A-8771-B1BC636D105D
- Oracle Automatic Storage Management Administrator's Guide
  https://docs.oracle.com/en/database/oracle/oracle-database/19/ostmg/index.html
- Oracle Database Backup and Recovery User's Guide (Oracle Recovery Manager)
  https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/index.html
- Oracle Secure Backup Administrator's Guide
  http://www.oracle.com/pls/topic/lookup?ctx=db112&id=OBADM
- My Oracle Support note 361468.1, “HugePages on Oracle Linux 64-bit”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=361468.1
- My Oracle Support note 749851.1 “HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=749851.1
- My Oracle Support note 1392497.1 “USE_LARGE_PAGES To Enable HugePages In 11.2”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1392497.1
- My Oracle Support note 151972.1 “Dead Connection Detection (DCD) Explained”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=151972.1
- BIG-IP Product Suite
  http://www.f5.com/products/big-ip/
- F5 DevCentral
  http://devcentral.f5.com/
- My Oracle Support note 1334857.1 “32-bit Client Install On Linux x86-64: PRVF-7532 Error Occurs For gcc-4.1.2 i386”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1334857.1
- PeopleSoft Enterprise PeopleTools 8.57 Documentation Library
  http://docs.oracle.com/cd/E25741_01/psft/html/docset.html
- PeopleSoft: Implementing Oracle Active Data Guard
  http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tadm/task_ImplementingOracleActiveDataGuard-3b7d04.html
- Oracle® ZFS Storage Appliance Administration Guide
  https://docs.oracle.com/cd/F13758_01/html/F13769/index.html
- My Oracle Support note 1274318.1 “Oracle Exadata Database Machine Setup/Configuration Best Practices (Doc ID 1274318.1)”
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1274318.1

- PeopleSoft MAA Best Practices
  http://www.oracle.com/technetwork/database/features/availability/maa-peoplesoft-bestpractices-134154.pdf
- PeopleSoft on Exadata
  http://www.oracle.com/technetwork/database/features/availability/maa-wp-peoplesoft-on-exadata-321604.pdf
- Oracle Site guard “Application-Level Disaster Recovery using Site Guard”

  https://www.oracle.com/database/technologies/high-availability/site-guard.html

11 Appendix A: Primary Site Setup

11.1 Primary Site Description

This section describes the steps taken in the project to set up the primary site to include Oracle Exadata Database Machine, database setup, Oracle Private Cloud Appliance, Oracle ZFS Appliance Storage and the PeopleSoft application deployment. The primary site employed Oracle Private Cloud Appliance with PCA VMs configured with anti-affinity.

11.2 Database Server Setup on Oracle Exadata Database Machine

The standard Oracle Exadata Database Machine configuration was deployed on the primary site. As described earlier, the Oracle Exadata Database Machine at the primary site is an X7-2 quarter rack with high capacity disks.

11.2.1 Grid Home and Database Home

The Grid Home was installed following the Exadata installation convention by Exadata OneCommand and is installed on all database nodes in the location:

/u01/app/19.0.0.0/grid

The Grid Home is owned by the oracle OS user and is in the oinstall and dba groups.

The Oracle Database software for the PeopleSoft database is installed into its own ORACLE_HOME location, separate from where the OneCommand installed the DBM starter database ORACLE_HOME. There are two ways to install the database software into a separate ORACLE_HOME:

- Download the Oracle Database 19c (19.0.0.0) software and install it using Oracle Universal Installer and applying the latest 19c Release Update. See My Oracle Support note 888828.1 for details.
- Clone the existing DBM ORACLE_HOME over to the new ORACLE_HOME location for PeopleSoft.

This project chose the second option of cloning from the Exadata DBM starter database ORACLE_HOME as this would give us an 19.0.0.0.0 database software version – 19c Release Update 8.

The Oracle Database software home for the PeopleSoft database was installed in:

/u01/app/oracle/product/19.0.0.0/dbhome_3

It is owned by the oracle user and in the oinstall and dba groups.

To clone from the DBM home, follow these steps:

As oracle on each database node, create the /u01/app/oracle_psft directory and make it owned by oracle_psft:oinstall:

```
mkdir -p /u01/app/oracle/product/19.0.0.0/dbhome_3
```

As the software owner (oracle), create the adump directory:

```
mkdir -p /u01/app/oracle/admin/CDBHCM_osc1a/adump
```

As root on one of the compute nodes, zip up the dbmhome_1 ORACLE_HOME:

```
cd /u01/app/oracle/product/19.0.0.0/dbhome_1

zip -r 19000_dbhome_1.zip *
```

As the software owner (oracle) on each database node, copy the 19000_dbhome_1.zip to the new ORACLE_HOME location under dbhome_3 and unzip it.

```
$ cd /u01/app/oracle_psft/product/19.0.0.0/dbhome_3

$ unzip 19000_dbhome_1.zip
```

Create a small shell script to run the clone.pl procedure. The script should look something like the following example but replace the host names and Oracle home path to match the environment it will run on.

```
echo "Clone started at `date`" | tee -a clone.log
# The 19c version
ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_3
ORACLE_HOME_NAME=dbhome_3_psft
ORACLE_BASE=/u01/app/oracle
THISNODE=exa14db01
C01="CLUSTER_NODES={exa14db01}"
C02="LOCAL_NODE=$THISNODE"

perl $ORACLE_HOME/clone/bin/clone.pl ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_3
ORACLE_HOME_NAME=dbhome_3_psft ORACLE_BASE=/u01/app/oracle OSDBA_GROUP=dba $C01 $C02 -local

echo "Clone ended at `date`" | tee -a clone.log
```

Place the above code into a script say clone.sh, add execute privileges and run it. Do this step on each database node modifying the script accordingly. Note: on each node that you run this script on, set CLUSTER_NODES only to that node's name – the same value of THISNODE.

As root on each database node, run the $ORACLE_HOME/root.sh script.

Set up the environment for each compute node.

```
set -o vi

ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_3; export ORACLE_HOME
OH=/u01/app/oracle/product/19.0.0.0/dbhome_3; export OH
ORACLE_UNQNAME=CDBHCM_osc1a; export ORACLE_UNQNAME
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/u01/app/oracle/product/19.0.0.0/dbhome_3/bin:/u01/app/ora
cle/product/19.0.0.0/dbhome_3/OPatch; export PATH
LD_LIBRARY_PATH=/u01/app/oracle/product/19.0.0.0/dbhome_3/lib; export LD_LIBRARY_PATH
TNS_ADMIN=/u01/app/oracle/product/19.0.0.0/dbhome_3/network/admin;
export TNS_ADMIN ORACLE_HOSTNAME
ORACLE_SID=CDBHCM1; export ORACLE_SID
```

At this point, all nodes should be ready for use.

NOTE: Be sure that whether you create a new database or copy one over, as in the case for this project, that the database parameter AUDIT_FILE_DEST points to the correct location. In our case it points to:

/u01/app/oracle/admin/CDBHCM_osc1a/adump

11.2.2 Linux HugePages Configuration

HugePages were configured on each database node. It is critically important that HugePages are configured when consolidating several application databases including PeopleSoft on Linux platforms. Please see My Oracle Support note 361323.1 for further details on HugePages and how to calculate the proper value for your environment. For our X6-2 database nodes, the HugePages were set in the sysctl.conf:

```
vm.nr_hugepages=41466
```

11.2.3 ASM Disk Groups

The following table describes the ASM storage configuration:

ASM DISK GROUP NAME	REDUNDANCY	TOTAL SIZE (TB)
+DATAC1	HIGH	215.9
+RECOC1	HIGH	1.01
+DBFSC1	HIGH	53.7

11.2.4 Exadata Exachk

Exachk is an extremely valuable utility to run on Oracle Exadata Database Machine to identify potential issues at the OS, database, and storage layers. This utility should be run at regular intervals and after patching and software upgrades are performed. Exachk was run after the database was created to validate the configuration. However, some items that Exachk flags may conflict with PeopleSoft requirements and documentation. These items are:

PROCESSES may be flagged if not set to the Exadata recommended value of 2048. The recommendation is founded on experience where memory resources are exhausted. Sometimes this parameter is set too high such that high CPU run queue size build up impacting performance.

However, for a heavily loaded PeopleSoft deployment with several application server domains, it may be necessary to increase the PROCESS parameter above 2048. For each Oracle RAC database instance, do not exceed the value of 4096 for PROCESSES on each Oracle RAC instance running an X6-2 compute node. See My Oracle Support note Oracle Exadata Initialization Parameters and Diskgroup Attributes Best Practices (Doc ID 2062068.1) for details on this and other database initialization parameters for Exadata.

Hidden parameters may be flagged, because in most cases these parameters are not required for proper operation and functioning of the database. However, some hidden parameters may be required to ensure performance or functional behavior expected by PeopleSoft. These parameters are:

```
_unnest_subquery=false
_ignore_desc_in_index=true
_gby_hash_aggregation_enabled=false
```

The above parameters are documented in the PeopleBooks PeopleTools documentation which can be found at: http://docs.oracle.com/cd/E25741_01/psft/html/docset.html.

11.2.5 Database Initialization Parameters

The following is the full list of database initialization parameters for the primary database:

```
*._file_size_increase_increment=2143289344
*._gby_hash_aggregation_enabled=false
*._ignore_desc_in_index=true
*._unnest_subquery=false
*.audit_file_dest='/u01/app/oracle/admin/CDBHCM_osc1a/adump'
*.cluster_database=TRUE
*.cluster_database_instances=2
*.compatible='19.0.0'
*.control_files='+DATAC1/CDBHCM_OSC1A/CONTROLFILE/current.675.1053176587'
*.CONTROL_MANAGEMENT_PACK_ACCESS='DIAGNOSTIC+TUNING'
*.db_block_size=8192
*.db_create_file_dest='+DATAC1'
*.db_create_online_log_dest_1='+DATAC1'
*.db_domain=''
*.db_files=1024
*.db_name='CDBHCM'
*.db_recovery_file_dest='+RECOC1'
*.db_recovery_file_dest_size=10000g
*.db_unique_name='CDBHCM_osc1a'
*.dg_broker_config_file1='+DATAC1/CDBHCM_OSC1A/dr1CDBHCM_osc1a.dat'
*.dg_broker_config_file2='+RECOC1/CDBHCM_OSC1A/dr2CDBHCM_osc1a.dat'
*.dg_broker_start=TRUE
*.diagnostic_dest='/u01/app/oracle'
*.enable_pluggable_database=TRUE
*.fal_server=''
*.filesystemio_options='SETALL'
CDBHCM1.instance_number=1
CDBHCM2.instance_number=2
*.log_archive_config='dg_config=(CDBHCM_osc1a,CDBHCM_OSC1B)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST'
*.log_archive_dest_2='service="cdbhcm_osc1b"','ASYNC NOAFFIRM delay=0 optional
compression=disable max_failure=0 reopen=300 db_unique_name="CDBHCM_OSC1B"
net_timeout=30','valid_for=(online_logfile,all_roles)'
*.log_archive_dest_state_2='ENABLE'
*.nls_length_semantics='CHAR'
*.open_cursors=1000
*.optimizer_dynamic_sampling=2
*.parallel_degree_policy='MANUAL'
*.pga_aggregate_target=5G
*.processes=1000
*.recyclebin='OFF'
*.remote_login_passwordfile='EXCLUSIVE'
*.session_cached_cursors=300
*.SGA_TARGET=24G
*.statistics_level='ALL'
CDBHCM1.thread=1
CDBHCM2.thread=2
CDBHCM1.undo_tablespace='UNDOTS'
CDBHCM2.undo_tablespace='UNDOTS2'
*.use_large_pages='ONLY'
```

11.2.6 PeopleSoft Database Creation

The PeopleSoft HRMS payroll database was copied from the performance test lab. It contains the HRMS 9.2 application installed schemas and HR data for 500,000 employees. The database size is 1.3TB. Once the database was copied over and configured for Oracle RAC, it was then registered with Oracle Cluster Ready Services as follows:

```
srvctl add database -db CDBHCM_osc1a -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_3 -
spfile +DATAC1/CDBHCM_OSC1A/spfilecdbhcm.ora -diskgroup "DATAC1,RECOC1” -pwfile
+DATAC1/CDBHCM_OSC1A/PASSWORD/pwcdbhcm

srvctl add instance -db CDBHCM_osc1a -instance CDBHCM1 -node exa14db01
srvctl add instance -db CDBHCM_osc1a -instance CDBHCM2 -node exa14db02
```

11.2.7 Database Service Setup

The following commands set up the required FAN-enabled role-based database services to be used by the application server domains.

```
srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service HR92U033_BATCH -preferred
"CDBHCM1,CDBHCM2"  -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO
-failoverretry 10 -failoverdelay 3

srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service HR92U033_ONLINE -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO -
failoverretry 10 -failoverdelay 3
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -db CDBHCM_osc1a -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" -failovermethod BASIC  -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

In the above, we specify the -pdb command line option to indicate that the service should start when the PDB HR92U033 is opened. On the physical standby, the PDB would be open READ ONLY.

11.2.8 Cluster Ready Service Configuration

As of Oracle 12c, 12.1.0.2 and Exadata X5 generation, a feature called Fast Node Death Detection (FNDD) was introduced to remove (evict) a node that was deemed to be not responding. This detection is achieved by the Infiniband (or RDMA over Converted Ethernet -- RoCE) switch detecting that both HCA ports of an Exadata compute node were no longer responding for approximately 500 milliseconds. Once this condition exists, CRS evicts the node by a forced reboot. This allows the cluster to recover more quickly and the evicted node can re-join the cluster in a proper and healthy state.

The cluster synchronization service (CSS) is the process and mechanism that determines how nodes within an Oracle RAC cluster are synchronized. If a node enters a state in which its HCA ports still respond yet is otherwise in a hung state, CSS will evict the node and initiate cluster re-configuration. Its timeout timer set by MISSCOUNT is set to 30 seconds on Oracle Exadata Database Machine by default. It is NOT recommended that you set the CSS timeout timer to a value less than 30 seconds.

On an Oracle RAC node failure, cluster reconfiguration will take place after 30 seconds and FAN-enabled clients will receive an “INSTANCE DOWN” FAN event to expedite reconnecting to the surviving Oracle RAC instances.

To check and if necessary, change the CSS timeout timer if not set to 30 seconds, do the following steps on only one database logged on as root. It is assumed that GRID_HOME is set to your Grid Infrastructure location, typically /u01/app/19.0.0.0/grid.

Get the current CSS timeout timer setting.

```
$GRID_HOME/bin/crsctl get css miscount
```

Set the timeout timer to 30 seconds if it is not already in step 1 above.

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

The CSS miscount can be set while the cluster is up.

Verify the new setting.

```
$GRID_HOME/bin/crsctl get css miscount
```

11.3 Oracle Private Cloud Appliance Setup for PeopleSoft Applications

The primary site used the Private Cloud Appliance virtual machines for compute nodes to host the PeopleSoft application components and PIA web servers. Two VMs were used for the application and process scheduler domain servers and two were used for the PeopleSoft Pure Internet Architecture (PIA) web servers.

The nodes are outlined in the table below:

NODE NAME	PURPOSE
pca3vm51	Application and process scheduler domain server 1
pca3vm52	Application and process scheduler domain server 2
pca3vm53	PIA web server 1
pca3vm54	PIA web server 2

11.3.1 Application Host Setup

Each of the above hosts were set up identically. The only difference was which components were started on which hosts according to the table above.

Environment configuration is identical for all four hosts:

```
export BASE_DIR=/u01/app/psft
export PS_HOME=$BASE_DIR/pt/ps_home8.57.11
export PS_CFG_HOME=/peoplesoft/local/ps_config
export PS_APP_HOME=$BASE_DIR/pt/hcm_app_home
export PS_FILEDIR=$PS_HOME/file
export ORACLE_HOSTNAME=pca3vm51
export ORACLE_BASE=/u01/app/psft
export ORACLE_HOME=/u01/app/psft/pt/oracle-client/19.3.0.0
export COBDIR=/opt/MFCobol
export CLASSPATH=$CLASSPATH:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export TNS_ADMIN=$ORACLE_HOME/network/admin
export JAVA_HOME=/u01/app/psft/pt/jdk1.8.0_221
export TUXDIR=$BASE_DIR/pt/bea/tuxedo/tuxedo12.2.2.0.0
export NLSPATH=$TUXDIR/locale/C
export LD_LIBRARY_PATH=$TUXDIR/lib:$PS_HOME/bin:$ORACLE_HOME/lib:$COBDIR/lib:$LD_LIBRARY_PATH
export LIBPATH=$COBDIR/lib
export SHLIB_PATH=$SHLIB_PATH:$COBDIR/lib
export PATH=$ORACLE_HOME/bin:$TUXDIR/bin:$PS_HOME/jre/bin:$PS_HOME/bin:$JAVA_HOME/bin:$PATH
export PATH=$PATH:$COBDIR/bin
export PS_SERVER_CFG=$PS_HOME/appserv/prcs/HR92U033/psprcs.cfg
export WLS_HOME=$BASE_DIR/pt/bea/wlserver

cd $PS_HOME
. ./psconfig.sh
cd
```

11.3.2 Oracle PCA ZFS Storage Appliance and Shared File System Creation for PeopleSoft Applications

All of the shared and local file systems are stored on the Sun ZFS 7320 Appliance which comes installed as part of the Oracle Private Cloud Appliance. The file systems are exported from ZFS and are mounted with NFS v3 by all VM mid tier servers. Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS (MOUNTED AS)	SITE STATE	MOUNT OPTIONS
PeopleSoft shared homes and Report Repository File System	All PeopleSoft app and Web Servers	Exported as: /export/psoft Mounted on all PCA VMs as: /u01	Site 1 Primary Site 2 Read-Only	nfs,_netdev,hard,intr,noatime,rsize=32768,wsize=32768 0 0

11.4 PeopleSoft Application Software Installation

For this project, we cloned (copied) an existing PeopleSoft installation on a previous system over to the ZFS project MAA_PS mounted by all PCA VMs.

PeopleSoft on Oracle Private Cloud Appliance can be deployed in one of two ways:

- PeopleSoft and all required software installed and run locally on each PCA VM.
- Using shared PS_HOME, PS_APP_HOME and ORACLE_HOME, and shared infrastructure that is required, but server specific configurations (PS_CFG_HOME) and MicroFocus Server Express COBOL are installed locally.

For minimizing maintenance downtime, the second option was implemented for this case study. Once the ZFS project MAA_PS and its share /export/posft was created, the following was performed:

As root on all PeopleSoft PCA VMs the /i01 mount point directory was created:

```
# mkdir -p /u01
```

As root, the following file system entries to /etc/fstab was added on all PeopleSoft PCA VMs:

```
pca1zfs.cloud.osc.oracle.com:/export/psoft /u01   nfs
_netdev,hard,intr,noatime,rsize=32768,wsize=32768 0 0
```

On all PeopleSoft PCA VMs Mount the /u01 file system:

```
# mount /u01
```

As root on one of the PeopleSoft PCA VMs, create the subdirectories and set the ownership:

```
# mkdir -p /u01/app/psft
# cd /u01/app
# chown oracle_psft:oinstall psft
```

As root on one of the PeopleSoft PCA VMs, we cloned (copied) in the PeopleSoft installation under /u01/app/psft.

NOTE: This paper does not provide details for installing all of the necessary software components required for PeopleSoft. It also does not provide details on the PeopleTools Windows utilities such as Application Designer or Change Assistant.

The required software components for PeopleSoft HRMS in this project are:

- Oracle WebLogic Server 12.2.1.3.0
- Oracle Java JDK/JRE 1.8.0_221
- Oracle Tuxedo 12.2.2.0.0
- Oracle Database 19c Release Update 8 (19.3.0.0.0) client software (64 bit)
- PeopleSoft PeopleTools 8.57.11
- PeopleSoft HRMS 9.2 U033
- Micro Focus COBOL Server Express 5.1 WP14

Except for Micro Focus COBOL, all of the above software components were installed in the shared directory structure:

/u01/app/psft/pt

The Micro Focus COBOL compiler and run-time environment should be installed on a local file system. It is often installed in the directory /usr/local/microfocus or /opt/microfocus mand owned by root. It must have the Micro Focus License Manager (mflman) configured and running regardless of where it is installed.

11.4.1 Install Micro Focus Server Express COBOL

On each application server node that will run COBOL programs, install the Micro Focus Server Express 5.1 WP14 COBOL compiler, runtime libraries and license manager facility following the instructions in the accompanying README. This must be installed as root. In our case study, it is installed in the /opt/MFCobol directory. It should also be configured so that the license manager is started after each node reboot. Ensure that the COBDIR environment variable is set to the install directory location.

11.4.2 PS_CFG_HOME Specific Directories for Multi-Node Deployments

When configuring PeopleTools, the PS_CFG_HOME should be configured to point to a directory location specific to the host running PeopleSoft application server or the PIA web server. Host and domain specific configuration files are stored in this location. Application and web server domain log files are also stored in this location. The location for PS_CFG_HOME can point to a local file system or on a shared NFS mounted file system but it should contain only domain configurations and log files specific for the given node. In this project, the directory structure on each node have the same directory structure layout. On all application and Web nodes, the PS_CFG_HOME was set to: /peoplesoft/local/ps_config. This is not a shared location rather, each PCA VM had its own local storage that this directory structure was created on. The benefits for this configuration are:

- Allows for each node to be configured independent of other nodes
- Allows the ability to test the configuration at the DR site when the physical standby database is in the SNAPSHOT STANDBY role

For existing PeopleSoft implementation and assuming that PS_HOME is pointing to: /u01/app/psft/pt/ps_home8.57.11 and PS_CFG_HOME is pointing to $PS_HOME/ps_config, here are a set of steps to move the application server and PIA domains to a directory structure local to each node:

Create the local directories on each application and Web PIA PCA server.

As root:

```
mkdir -p /peoplesoft/local/ps_config

chown -R oracle_psft:oinstall /peoplesoft
```

Only for the PeopleSoft application server (APPSRV), copy all content in $PS_HOME/ps_config to the local directory on each PCA VM which host the application server:

```
cp –r $PS_HOME/ps_config/* /peoplesoft/local/ps_config/.
```

It is optional to remove the ps_config directory from PS_HOME. Make sure that PS_CFG_HOME points to this new location.

11.4.3 PeopleSoft Application Server Domain Database Connection

To configure the application server domain, two items need to be configured.

- The tnsnames.ora file for connecting to the database
- The application server domain

In this case study, the tnsnames.ora file contains two specific connect string alias entries: one for the application server and one for the process scheduler. Each connect string has the following attributes:

- Multiple DESCRIPTION each with their own address using the SCAN listener at each site
- Each description has their own connect and transport timeout parameters
- Each description connects to a FAN enabled role-based database service

This type of connect string alias allows for the application server or process scheduler to transparently connect to the database where the services are available. During a switchover or failover, no modification to the tnsnames.ora connect strings are required. The TNS_ADMIN environment variable points to the same directory location at both primary and standby sites.

The following provides an example of a working tnsnames.ora used in our project.

The environment variable TNS_ADMIN points to:

/u01/app/psft/pt/oracle-client/19.3.0.0.0/network/admin

Both sites share the same tnsnames.ora file which has the following entries:

```
#HCM Online users
HR92U033 =
 (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)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
 )

# Batch scheduler
HRBATCH =
 (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)(PORT = 1521))
       )
       (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_BATCH)
       )
    )
   (DESCRIPTION =
       (CONNECT_TIMEOUT=5(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
       (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
       (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_BATCH)
       )
    )
 )
```

Validate that your connect string alias works by using SQL*Plus to ensure proper connections:

```
sqlplus “/@HR92U033 as sysdba”
```

PeopleTools treats the above two TNS connect strings as actual database names and refers to them as the DBName when configured in the PSADMIN utility for the application server and the process scheduler. For security purposes, when PeopleTools connects to the database and queries the PS.PSDBOWNER table to validate that the database it is attempting to connect to is the correct database. The table has two columns: DBNAME and OWNER. The DBNAME must match the TNS connect string alias, and the OWNER is the owning schema of the application tables. The entries in our PSDBOWNER table are:

```
DBNAME   OWNERID
-------- --------
HR92U033 EMDBO
HRBATCH  EMDBO
```

Make sure that the TNS connect alias names are in the PSDBOWNER table in the DBNAME column as shown above.

To insert the rows into the PSDBOWNER table. Since the HCM application is running in a PDB named HR92U033, we need to set the session to this PDB:

Log onto the database with SQL*Plus as SYS:

```
sqlplus “/ as sysdba”

SQL> alter session set container = HR92U033;
```

Insert the rows:

```
INSERT INTO PS.DBOWNER VALUES (‘HR92U033’,’EMDBO’);
INSERT INTO PS.DBOWNER VALUES (‘HRBATCH’, ‘EMDBO’);
COMMIT;
```

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

To remedy this situation, use the SQL*Net TCP.EXCLUDED_NODES and TCP.VALIDNODE_CHECKING parameters in the sqlnet.ora for the SCAN and Grid listeners on each DB node to only block the application tier VMs from accessing the database at that specific site. Names of any remote database servers that are used for Data Guard must NOT be in TCP.EXCLUDED_NODES list. This also requires restarting all SCAN listeners and Grid listeners.

11.4.4 Configuring the Application Domain

On every node that runs the PeopleSoft application server, you must configure the Oracle Tuxedo domain server. In this project, we use the same domain configuration on all application server VMs. The domain name is HR92U033. In the $PS_CFG_HOME/appserv/HR91FP3 directory you will find the psappsrv.cfg file. Make any configuration changes using the $PS_HOME/bin/psadmin utility before re-deploying the domain HR92U033 on each node. The steps to deploy the HR92U033 domain are:

Run $PS_HOME/bin/psadmin utility.

```
cd $PS_HOME/bin
./psadmin
```

This starts the psadmin utility.

Select option 1: **Application Domain**.

Select option 1: Administer a Domain.

Select the domain name, in our case HR92U033.

Select option 4: Configure this domain.

Enter Y to shut down the domain.

Review the configuration and ensure that your TNS connect string aliases are correct for DBName. In our case it is HR92U033.

Select option 14: Load domain as shown.

The domain configuration is loaded and all of the required domain files for Oracle Tuxedo are regenerated. It should now be possible to start the application server domain from the PSADMIN utility.

11.4.5 Configuring the Process Scheduler

The PeopleSoft process scheduler (or batch processor) must also be configured.

The steps for configuring the process scheduler are similar to those for the application server domain.

Run $PS_HOME/bin/psadmin utility.

```
cd $PS_HOME/bin
./psadmin
```

This starts the psadmin utility.

Select option 2: **Process Scheduler**.

Select option 1: **Administer a Domain**.

Select the domain name. In our case it is HR92U033.

Select option 4: **Configure this domain**.

Enter Y to shut down the domain.

Review the configuration and ensure that all of your TNS connect string aliases are correct for DBName. In our case it is HRBATCH.

Select option 6: Load domain as shown.

The domain configuration is loaded and all of the required domain files for Oracle Tuxedo are regenerated. It should now be possible to start the process scheduler from the PSADMIN utility.

11.4.6 PeopleSoft PIA Web Server Domain Configuration

The Pure Internet Application (PIA) domain can simply be configured using the PIA setup.sh script on each of the PCA VMs hosting the PIA web server. Perform the following steps to configure the PIA web server as oracle_psft on each VM hosting the PIA web server:

Source the PeopleSoft environment and ensure the PS_CFG_HOME is defined. For this project it is defined to: /peoplesoft/local/ps_config.

Change the working directory to $PS_HOME/setup/ PsMpPIAInstall

```
$ cd $PS_HOME/setup/PsMpPIAInstall
```

Copy the resp_file.txt to a file you will edit i.e., pca_pia_resp.txt

```
$ cp resp_file.txt pca_pia_resp.txt
```

Edit the pca_pia_resp.txt file and modify according to your environment. Below is a listing of this file with comments we have added:

```
# Name of the PIA domain
DOMAIN_NAME=HR92U033   ←- The domain name can be the same on all nodes
# Web server type. Possible values are "weblogic", "websphere"
SERVER_TYPE=weblogic
# WebLogic home, the location where Oracle WebLogic is installed (for WebLogic deployment
only)
BEA_HOME=/u01/app/psft/pt/bea
# admin console user id/password for securing WebLogic/WebSphere admin console credential
USER_ID=system
USER_PWD=<your choice of password>
USER_PWD_RETYPE=<Re-type password from above>
# Install action to specify the core task that installer should perform.
# For creating new PIA domain - CREATE_NEW_DOMAIN.
# For redeploying PIA - REDEPLOY_PSAPP.
# For recreating PIA domain - REBUILD_DOMAIN.
# For installing additional PSFT site - ADD_SITE
# For installing Extensions - ADD_PSAPP_EXT
INSTALL_ACTION=CREATE_NEW_DOMAIN ← Use CREATE_NEW_DOMAIN
# Domain type to specify whether to create new domain or modify existing domain. Possible
values are "NEW_DOMAIN", "EXISTING_DOMAIN".
DOMAIN_TYPE=NEW_DOMAIN
# Install type to specify whether the installation is a single server,  multi server
deployment or ditributed weblogic server .
#Possible values are "SINGLE_SERVER_INSTALLATION", "MULTI_SERVER_INSTALLATION" and
"DISTRIBUTED_SERVER_INSTALLATION"
INSTALL_TYPE=SINGLE_SERVER_INSTALLATION
# WebSite Name
WEBSITE_NAME=ps ← For our project we chose “ps”.
# AppServer Name
APPSERVER_NAME=pca3vm51.cloud.osc.oracle.com ← The application domain server
# Appserver JSL Port
JSL_PORT=9000  ← This is the default port, you can cboose a different port
# HTTP Port
HTTP_PORT=8080 ← PIA front-end port to access PeopleSoft application
# HTTPS Port
HTTPS_PORT=8443 ← PIA front-end SSL port if SSL is enabled on the web server
# Authentication Domain (optional)
AUTH_DOMAIN=cloud.osc.oracle.com ← Change this to match the domain for your environment.
# Web Profile Name Possible Values are "DEV","TEST","PROD","KIOSK"
WEB_PROF_NAME=PROD
# Web Profile password for User "PTWEBSERVER"
WEB_PROF_PWD=PTWEBSERVER
WEB_PROF_PWD_RETYPE=PTWEBSERVER
# Integration Gateway user profile.
IGW_USERID=administrator
IGW_PWD=password
IGW_PWD_RETYPE=password
# AppServer connection user profile
APPSRVR_CONN_PWD=PS
APPSRVR_CONN_PWD_RETYPE=PS
# Directory path for reports
REPORTS_DIR=/u01/app/psft/pt/psft_reports/out
```

Copy this file to $PS_CFG_HOME:

```
$ cp pca_pia_rsp.txt $PS_CFG_HOME/.
```

Run the setup.sh script to configure the PIA using the below command:

```
$ ./setup.sh -i silent -DRES_FILE_PATH=$PS_CFG_HOME/pca_pia_resp.txt
```

Enable load balancing and failover for the PIA web server to the application domain servers. Edit the configuration.properties file located in:

```
$PS_CFG_HOME/webserv/HR92U033/applications/peoplesoft/PORTAL.war/WEB-INF/psftdocs/ps
```

Modify the line:

```
psserver=pca3vm51.cloud.osc.oracle.com:9000
```

and add the second application domain server as follows:

```
psserver=pca3vm51.cloud.osc.oracle.com:9000,pca3vm52.cloud.osc.oracle.com:9000
```

The PIA configuration on the current VM is complete. Repeat the above steps for each PCA VM that will host the PIA web server.

11.4.7 Application and PIA Web Server Scripts

Simple startup and shutdown scripts were created to make it easy to start and stop each PeopleSoft component. These scripts can be used by Oracle Site Guard when implementing Oracle Site Guard to automate switchover or failover.

Application Domain Server:

startAPP.sh:

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Starting Apps Server for domain: $domain on host: $HOSTNAME ----"
${PS_HOME}/appserv/psadmin -c boot -d $domain
```

stopAPP.sh:

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Stopping Apps Server for domain: $domain on host: $HOSTNAME ----"

#Note the shutdown! Is a forced shutdown.
${PS_HOME}/appserv/psadmin -c shutdown! -d $domain
```

startPS.sh:

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Starting Process Scheduler for domain: $domain on host: $HOSTNAME ----"
${PS_HOME}/appserv/psadmin -p start -d $domain
```

stopPS.sh:

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Stopping Process Scheduler for domain: $domain on host: $HOSTNAME ----"
${PS_HOME}/appserv/psadmin -p kill -d ${domain}
```

PIA Web Server

startWS.sh:

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Starting WLS Server for domain: $domain on host: $HOSTNAME ----"

${PS_CFG_HOME}/webserv/${domain}/bin/startPIA.sh
```

stopWS.sh :

```
#!/bin/sh

source ~/psft.env

export domain=HR92U033
export HOSTNAME=`hostname`

date
echo "------ Stopping WLS Server for domain: $domain on host: $HOSTNAME ----"

${PS_CFG_HOME}/webserv/${domain}/bin/stopPIA.sh
```

11.5 F5 BIG-IP Load Balancer

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

We used two f5 BigIP Load Balancer Switches, one for the primary and one at the standby site for testing the snapshot standby, each configured with Local Traffic Manager (LTM), to direct traffic appropriately across the environments. Here are the steps we followed to configure the f5 BigIP load balancer:

1. From your network administrator, obtain the appropriate IP addresses for the load balancer front-end that application users will use. You will also need the netmask as well as 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. Ensure that the IP address is 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.
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, select Self IPs and then click Create. Enter a name for the self-IP address. We chose to use the host.domain that was added into DNS from step 1 above such as peoplesoft.osc.oracle.com. Enter the IP address and netmask. Select the appropriate VLAN / Tunnel configured on your f5. Ours is set to “client”. 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.
5. Create a separate PeopleSoft-specific health monitor to be used by the pool being defined in the next step. The monitor settings we used in this project for interval and timeout are provided in the example below. The monitor details that can be used for PeopleSoft are:

   ```
   Name:  PeopleSoft_Web
   Type: HTTP
   Parent Monitor:  http
   Interval:  10 seconds
   Timeout: 60 seconds
   Send String:  “GET / HTTP/1.0\r\n\r\n”
   Receive String: “200 OK”
   ```

   The above function monitors pca3vm53 and pca3vm54, which are in the F5 local traffic server pool. There are some considerations to keep in mind:

   - Reducing the timeout time can result in false service down when the web server is stalled or has high response times due to other component failure and recovery.
   - Setting the timeout too high will result in a higher number of user failures if the web server is down due to an abrupt shutdown or crash. In this case, existing user sessions will block and new sessions can be routed to the impacted web server only to eventually receive errors when the timeout period is reached.

   One other important consideration is the capacity of each web server and application server to take on load due to a failure of either component. Ensure that testing and production analyses are conducted to understand the required capacity requirements (CPU, memory) needed for each of the components to assume the additional load.
6. On the primary f5 load balancer, create a pool and enter a name of your choosing in which the production application tier nodes will be configured. We chose to name ours “psft_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, 8080.
8. Add the health monitor to this pool.
9. On the primary f5 load balancer, create a Virtual Server that defines the front-end access. Ours is called “psft_prod”.

10. For the virtual server, we set the source IP address to 0.0.0.0/0, which allows any remote system to access this virtual server. Refer to your system network security guide for an appropriate range.
11. Set the 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 the DNS.
12. Set the Service Port to 8080 or the appropriate port for your environment. 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.
13. Under the Resources tab, add the default pool created in step 6. In our case, we added “psft_site_pca3” as the default pool.

If both the primary and DR site load balancers are using Local Traffic Manager (LTM) instead of Global Traffic Manager (GTM), then repeat the above steps at the standby site with settings appropriate to that site.

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

12 Appendix B: Secondary (DR) Site Setup

This section describes the secondary site setup for DR. This site employs Oracle Private Cloud Appliance similar to the primary site.

12.1 Database Server Setup on Oracle Exadata

An Oracle Exadata Database Machine quarter rack was provisioned at the disaster recovery site. The two Exadata compute nodes were set up identical to that of the primary Oracle Exadata Database Machine to include HugePages, Grid Home, the database home for PeopleSoft, and database services. Exachk was also run to validate and correct any items not configured correctly with the exceptions noted in Appendix A.

For the database software setup, please refer to Appendix A, section 11.2.1: Grid Home and Database Home Setup. Follow the steps for cloning database ORACLE HOMEs in that section.

12.1.1 ASM Disk Groups

The following table describes the ASM storage configuration at the disaster recovery site:

ASM DISK GROUP NAME	REDUNDANCY	TOTAL SIZE (TB)
+DATAC2	HIGH	214.99
+RECOC2	HIGH	53.75
+DBFSC2	HIGH	1.03

12.1.2 Primary Database Preparation

This section provides the steps required to prepare the primary database for Data Guard configuration. Please do not skip any of these steps unless it is not applicable to your environment such as whether or not your database has implemented Transparent Data Encryption (TDE).

12.1.2.1 Create Standby Redo Logs (SRLs)

Standby redo logs (SRLs) are required by Data Guard and are written to on the standby database and are archived at the standby just like online redo logs (ORLs) are archived. They are not used when the database is in the primary role. However, creating standby redo logs now saves time and effort as they will be carried across when the physical standby is instantiated. Follow these steps to create standby redo logs on the primary database.

1. Determine the number of redo threads on the primary database

To determine the number of threads on the primary database, issue the following query on the primary database:

```
SQL> select thread# from v$thread;

   THREAD#
----------
         1
         2
```

We have two threads.

2. Determine the required size for the standby redo logs

To determine the size of the online redo logs, issue the following query on the primary database:

```
SQL>  select thread#,group#,bytes
  2  from v$log
  3* order by 1,2;


   THREAD#     GROUP#      BYTES
---------- ---------- ----------
         1          1 4294967296
         1          2 4294967296
         1          3 4294967296
         1          4 4294967296
         2          5 4294967296
         2          6 4294967296
         2          7 4294967296
         2          8 4294967296
```

The size of the online redo logs is 4294967296 bytes.

3. Determine the number of standby redo logs

From step 2 above, there are two threads each with 4 redo log groups. The number of standby redo logs is then:

SRLs = (# of redo groups per thread * # of threads) + # of threads

SRLs = (4 * 2) + 2 = 10

Assumption: Each thread has the same number of redo log groups. We then need 10 standby redo logs.

4. Create the standby redo logs on the primary database.

On the primary database, create the standby redo logs using the following example:

```
alter database add standby logfile thread 1
group 11 size 4294967296,
group 12 size 4294967296,
group 13 size 4294967296,
group 14 size 4294967296,
group 15 size 4294967296;

alter database add standby logfile thread 2
group 16 size 4294967296,
group 17 size 4294967296,
group 18 size 4294967296,
group 19 size 4294967296,
group 20 size 4294967296;
```

12.1.2.2 Copy the Password file from the Primary to the Standby

If the password file is located within an ASM disk group, then it first must be copied out to a file system location such as $ORACLE_HOME/dbs. If the password file is not in an ASM group, skip to step 2 below.

1. Copy the password file from ASM

If the Grid Infrastructure is owned and managed by a separate OS user such as “grid”, then issue the following commands as the grid user. In the below example, the disk group we are copying the password from is DATAC1.

```
$ asmcmd -p --privilege sysdba
ASMCMD [+] > pwcopy -dbuniquename CDBHCM_osc1a  +DATAC1/CDBHCM_OSC1A/PASSWORD/pwcdbhcm
/u01/app/oracle/product/19.0.0.0/dbhome_3/dbs/orapw<ORACLE_SID>
```

2. Copy the password file to the standby oracle home.

Copy the password file over to one of the standby database nodes and place it in $ORACLE_HOME/dbs. Make sure its name is orapw<ORACLE_SID>. For instance, if the standby DB node you copied the password to has ORACLE_SID set to CDBHCM1, then the password file name would be orapwCDBHCM1.

This will be the node you run the RMAN commands to copy the primary database over.

12.1.2.3 Copy TDE wallets to the Standby

If the database has Transparent Data Encryption (TDE) enabled, you must copy over the TDE wallets used for the keystore.

1. Determine TDE wallet location and copy them to the standby

To determine 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_ADMN 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/CDBHCM/wallet_root/tde)))
```

In the above directory location, copy the wallet files to all standby DB nodes. If your environment uses a shared file system such as ACFS, DBFS or an NFS mount point that all DB nodes use, then you can use this location to place the wallets. Maintain the permissions and ensure the owner and group are set:

```
-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
```

If you do not have AUTO_LOGIN configured in the keystore then the cwallet files may not be present.

2. Copy the sqlnet.ora file from the primary to the standby DB nodes

Copy the sqlnet.ora file on the primary to each DB node placing it in $TNS_ADMIN, then update the sqlnet.ora file in $TNS_ADMIN with the correct location for where the wallets are placed. Use the example in step 1.

12.1.2.4 Copy PFILE from Primary to the Standby

1. Create the PFILE

On the primary database, create a PFILE from the SPFILE which will be edited and used on the standby. Log onto SQL*Plus as SYS and issue the following command:

SQL> create pfile='CDBHCM_pfile.ora' from spfile;

Copy the PFILE to the standby

Copy the pfile created in step 1 above to the DB node where you placed the password file and place it in $ORACLE_HOME/dbs.

2. Edit the PFILE

Edit the PFILE from step 2 above (on the standby) and change the following parameters to reflect what should be set on the standby. For example, db_create_file_dest, db_create_log_dest_1 on the primary was set to +DATAC1. On the standby for this project, it should be set to +DATAC2. Make sure the following parameters are set per below:

- Make sure db_name is the same as the that of the primary
- db_create_file_dest
- db_create_log_dest_1
- db_file_recovery_dest
- db_file_recovery_dest_size
- db_unique_name
- audit_dump_dest
- diagnostic_dest
- Make sure the parameter control_files is NOT set

Here is an example PFILE for this case study:

```
*._gby_hash_aggregation_enabled=FALSE ← Required by PeopleSoft
*._optimizer_skip_scan_enabled=FALSE  ← Required by PeopleSoft
*._unnest_subquery=FALSE              ← Required by PeopleSoft
*.cluster_database=TRUE
*.cluster_database_instances=2
*.compatible='19.0.0'
*.CONTROL_MANAGEMENT_PACK_ACCESS='DIAGNOSTIC+TUNING'
*.db_block_size=8192
*.db_create_file_dest='+DATAC2' ← Disk group for the standby
*.db_create_online_log_dest_1='+DATAC2'  ← Disk group for the standby
*.db_domain=''
*.db_files=1024
*.db_name='CDBHCM'  ← Must be the same as the primary.
*.db_recovery_file_dest='+RECOC2'
*.db_recovery_file_dest_size=10000g
*.db_unique_name='CDBHCM_osc1b' ← Must be different from the primary
*.diagnostic_dest='/u01/app/oracle'
*.enable_pluggable_database=TRUE
*.filesystemio_options='SETALL'
CDBHCM1.instance_number=1
CDBHCM2.instance_number=2
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST'
*.nls_length_semantics='CHAR'
*.open_cursors=1000
*.optimizer_dynamic_sampling=2
*.parallel_degree_policy='MANUAL'
*.pga_aggregate_target=5G
*.processes=1000
*.recyclebin='OFF'
*.remote_login_passwordfile='EXCLUSIVE'
*.remote_listener='exa14-scan2:1521' ← Must point to the correct SCAN listener on the
standby
*.session_cached_cursors=300
*.SGA_TARGET=24G
*.statistics_level='ALL'
CDBHCM1.thread=1
CDBHCM2.thread=2
*.undo_tablespace='UNDOTS'
CDBHCM1.undo_tablespace='UNDOTS'
CDBHCM2.undo_tablespace='UNDOTS2'
*.use_large_pages='ONLY'
```

Note that the control_files parameter is not set.

12.1.2.5 Add TNS Connect String Alias

On all DB nodes at both the primary and standby sites, add TNS connect strings for both primary and standby databases into $TNS_ADMIN/tnsnames.ora. Here is an example:

```
CDBHCM_osc1a =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDBHCM_osc1a)
    )
  )

CDBHCM_osc1b =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDBHCM_osc1b)
    )
  )
```

12.1.2.6 Startup the Standby Instance

With the PFILE saved in $ORACLE_HOME/dbs on the same DB node that the password file was placed, you can startup the instance on this node with NOMOUNT:

```
$ cd $ORACLE_HOME/dbs
Sqlplus / as sysdba
SQL> startup NOMOUNT pfile=’CDBHCM_pfile.ora’
```

If the instance fails to start, check the following:

- The adump directory was not created and/or the audit_dump_dest is not pointing to the correct location
- You have insufficient huge pages to support the size of the SGA. Either increase hugepages or reduce the size of the SGA.
- The diagnostic_dest parameter may not be set correctly

Address any errors that prevents starting the instance in NOMOUNT before proceeding.

12.1.2.7 Test Connections

Testing the connections between both the primary and standby will help prevent other issues as you configure Data Guard.

From the example TNS connect string alias above, attempt to connect from the standby DB node where the instance is running to the primary:

```
sqlplus sys/<sys password>@CDBHCM_osc1a as sysdba
```

From the primary DB node to the standby DB node:

```
sqlplus sys/<sys password>@CDBHCM_osc1b as sysdba
```

NOTE: When connecting to the standby instance, you will get an error indicating that sessions are blocked. This is expected, it indicates that the instance is up in NOMOUNT.

DO NOT PROCEED if you receive other errors and fail to successfully connect to either database instances. Address these errors before proceeding.

12.1.3 Standby Database Instantiation

The standby database was copied from the primary Oracle Exadata Database Machine to the standby Oracle Exadata Database Machine using the Recovery Manager (RMAN) RESTORE FROM SERVICE feature. To use this process, ensure that all steps in section 12.1.2 above have been completed successfully.

As of Oracle 12c (12.1.0.2), a feature was introduced to make 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:

- Copies the control file
- Copies the database: CDB and all PDBs
- Places the PDBs into the correct location within ASM using their GUIDs
- Creates the temp tablespace temp files

Make sure you have set the following parameters per section 12.1.2.4:

- 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.

The db_unique_name parameter is used by RMAN to create the appropriate directory structures within ASM. If necessary, please review section 12.1.2.4 and if any changes are made, shutdown and restart the standby instance NOMOUNT as described above.

To instantiate the physical standby database:

Create an RMAN shell script:

The RMAN script is quite simple, see below and modify according to your environment. Change the TNS connect string alias name for your environment and save the script say to rman_restore_from_service.sh

```
# 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 password> nocatalog  <<EOF! | tee -a
rman_restore_db_from_service.log
set echo on

restore standby controlfile from service 'CDBHCM_osc1a';
alter database mount;

CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;

restore database from service 'CDBHCM_osc1a' section size 64G;

EOF!
```

Notes:

- If your primary database uses device type of SBT_TAPE for performing backups to either tape backups or Oracle Cloud object storage, you must use the CONFIGURE DEFAULT DEVICE TYPE TO DISK; command as shown above otherwise the restore will fail. This is a known issue and will be addressed in later releases.
- The PARALLELISM option should be set according to your environment.

Run the RMAN script

Simply run the above script:

```
./rman_restore_from_service.sh
```

Monitor the output as the restore progresses. If the restore fails, investigate the errors and correct accordingly. Some errors may be caused by:

- 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
- Connection failures

Create an SPFILE on the standby

Log into SQL*Plus as SYS and show the contro_files parameter:

```
sqlplus / as sysdba
SQL> show parameter control_files
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
control_files                        string      +DATAC2/CDBHCM_OSC1B/CONTROLFI
LE/current.276.1054210597
```

Add the parameter control_files and the location to the PFILE used to start up the instance:

```
*.control_files=’+DATAC2/CDBHCM_OSC1B/CONTROLFILE/current.276.1054210597’
```

Create the SPFILE as follows:

```
SQL> create spfile=’+DATAC2/CDBHCM_OSC1B/spfilecdbhcm_osc1b.ora’ from
pfile=’CDBHCM_pfile.ora’;
```

Create an init<ORACLE_SID> file in the $ORACLE_HOME/dbs directory with the following line:

```
spfile=’+DATAC2/CDBHCM_OSC1B/spfilecdbhcm_osc1b.ora’
```

Copy the init<ORACLE_SID>.ora file to the other node(s) in the RAC cluster using the correct ORACLE_SID for the other nodes. For example, on node 1, the file would be named init CDBHCM1.ora, on node 2, initCDBHCM2.ora and so on.

Restart MOUNT the standby instance, do not attempt to open the database.

```
SQL>  shutdown
SQL> startup MOUNT
```

Clear all online redo and standby logs

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 exists. 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. You may use the following script:

```
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
```

12.1.4 Register the Standby Database with Cluster Ready Services

In this section, we add the database to Oracle Cluster Ready Services. The next section will discuss Data Guard configuration.

All of the following steps are completed on the standby:

On one of the standby DB nodes as oracle, add the database and its RAC instances to Oracle Cluster Ready Services (CRS):

```
srvctl add database -db CDBHCM_osc1b -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_3 -
spfile +DATAC2/CDBHCM_OSC1B/spfilecdbhcm_osc1b.ora -diskgroup "DATAC2,RECOC2"

srvctl add instance -db CDBHCM_osc1b -instance CDBHCM1 -node exa14db03
srvctl add instance -db CDBHCM_osc1b -instance CDBHCM2 -node exa14db04
```

Make sure CRS can stop and restart the database on both nodes:

```
srvctl stop database –db CDBHCM_osc1b –o immediate
srvctl start database -db CDBHCM_osc1b -o mount
```

All instances should be started with the database mounted.

12.1.5 Standby Database Service Setup

Add role-based database services to the standby cluster specifying the PDB the services are to serve:

```
srvctl add service -db CDBHCM_osc1b -pdb HR92U033 -service HR92U033_BATCH -preferred
"CDBHCM1,CDBHCM2"  -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO
-failoverretry 10 -failoverdelay 3

srvctl add service -db CDBHCM_osc1b -pdb HR92U033 -service HR92U033_ONLINE -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO -
failoverretry 10 -failoverdelay 3
```

If the primary database becomes the standby, PSQUERY is started for Oracle Active Data Guard query access.

```
srvctl add service -db CDBHCM_osc1b -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" failovermethod BASIC -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

12.2 Data Guard Broker Configuration and Setup

This section provides the steps used to enable Oracle Active Data Guard. The Data Guard Broker is used for much of the Data Guard configuration work. We do not describe the manual configuration of Data Guard.

12.2.1 Data Guard Broker Prerequisite Configuration

The following steps must be executed on both the primary and standby databases.

Configure the Broker database parameters on both primary and standby. Configuring the Data Guard broker need only to be performed on one RAC node at each site. The following example is from one node on the primary using SQL*Plus.

```
alter system set dg_broker_config_file1=’+DATAC1/CDBHCM_OSC1A/ dr1cdbhcm _osc1a.dat sid=’*’
scope=both;

alter system set dg_broker_config_file2=’+RECOC1/CDBHCM_OSC1A/dr2cdbhcm _osc1a.dat’ sid=’*’
scope=both;

alter system set dg_broker_start=true sid=’*’ scope=both;
```

Repeat the above on one RAC database node at the standby.

12.2.2 Create and Enable the Data Guard Broker Configuration

Ensure that the database parameters for Data Guard Broker have been set properly as detailed in the previous step. Before starting the Broker command line interface (CLI), it is recommended that you log on to each node at the primary and standby site as oracle (for this case study) and use “tail –f” on the alert_<SID>.log file located as specified by the database parameter diagnostic_dest. In our case:/u01/app/oracle/diag/rdbms/ cdbhcm_osc1b/CDBHCM1/trace directory for the first Oracle RAC instance.

Run `dgmgrl` on one RAC database node of the primary database and add the primary and standby database using the TNS connect string aliases in the connect identifier, review section12.1.2.5 above:

```
% dgmgrl
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Mon Nov 16 09:13:18 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.
DGMGRL> connect sys/<password>
Connected.

DGMGRL> create configuration cdbhcm_dg
  AS primary database IS 'CDBHCM_OSC1A'
  CONNECT IDENTIFIER IS CDBHCM_osc1a;
```

Now add the standby:

```
DGMGRL> add database 'CDBHCM_OSC1B'
 as connect identifier is CDBHCM_OSC1B;
```

Note that the name in quotes is the db_unique_name of the two databases.

Enable the configuration. When you issue the following command, run tail –f on the alert.log for each database instance at both the primary and standby sites. You will see a lot of activity as the Broker configures each database.

```
DGMGRL> enable configuration;
Configuration enabled.
```

If the command was successful issue the following:

```
DGMGRL> show configuration

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 53 seconds ago)
```

If the result is not SUCCESS and you receive errors, refer to the Data Guard Broker documentation. Most likely, you have connect strings that cannot be resolved. Use the `show database` command within the Broker to provide more details.

Assuming that enabling the configuration was successful, the standby is brought into synch with the primary. It may take some time to bring the standby database into sync (shipping and applying logs to the standby database).

In 19c, Data Guard Broker has a new command to show both the configuration and the lag time of the physical standby as in the following example:

```
DGMGRL> show configuration lag;

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database
                   Transport Lag:      0 seconds (computed 0 seconds ago)
                   Apply Lag:          0 seconds (computed 0 seconds ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 46 seconds ago)
```

If you issue the “show configuration lag” command following an initial Data Guard Broker configuration or just after a switchover, you may see an error indicating that DG Broker is not yet available. Wait for a few minutes and this error will be resolved.

12.2.3 Oracle Active Data Guard Setup

PeopleSoft supports Oracle Active Data Guard. The following steps show how to set up a database service for offloading queries to the Oracle Active Data Guard database.

Disable the managed recovery process (MRP) from the Broker:

```
DGMGRL> edit database 'CDBHCM_OSC1B' set state='APPLY-OFF';
Succeeded.
```

Open the standby database on read-only.

```
SQL> alter database open read only;
Database altered.
```

Re-enable MRP from the Broker.

```
DGMGRL> edit database 'CDBHCM_OSC1B' set state='APPLY-ON';
Succeeded.

DGMGRL> show database 'CDBHCM_OSC1B';

Database - CDBHCM_OSC1B

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 0 seconds ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 69.00 KByte/s
  Real Time Query:    ON
  Instance(s):
    CDBHCM1 (apply instance)
    CDBHCM2

Database Status:
SUCCESS
```

Oracle Active Data Guard is now enabled.

Configure the PSQUERY service.

PSQUERY is already added as a service to both the primary and standby CRS if you have followed the steps for the primary and standby database setup above. It is necessary now to finish the configuration for PSQUERY. To complete the configuration, do the following steps:

On the primary database execute the following PL/SQL package:

```
EXECUTE DBMS_SERVICE.CREATE_SERVICE('PSQUERY', 'PSQUERY', NULL, NULL,TRUE, 'BASIC', 'SELECT',
180, 1, NULL);
```

DBMS_SERVICE.CREATE_SERVICE executed on the primary database replicates the service definition in the redo stream onto the standby. Note that the parameters passed to DBMS_SERVICE.CREATE_SERVICE must match those passed to the `srvctl` command that was performed earlier (see section 10.1.5) on the standby shown here:

```
srvctl add service -db CDBHCM_osc1b -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" failovermethod BASIC -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

On the primary, start and stop the PSQUERY service.

```
srvctl start service –d CDBHCM_osc1a –service PSQUERY

srvctl stop service –d CDBHCM_osc1a –service PSQUERY
```

Start the PSQUERY service on the standby:

```
srvctl start service –d CDBHCM_osc1b –service PSQUERY
```

12.3 Standby Private Cloud Appliance Setup for PeopleSoft Applications

This section describes the standby Private Cloud Appliance VM setup process for the PeopleSoft application. Each PCA VM is configured such that no one VM for either the application server domain or the PIA web tier resides on the same PCA physical compute node.   For this project, each VM was provisioned per the specifications described earlier and are identical to that of those on the primary PCA. Follow these steps to set up each application and PIA VM.

On each PCA VM, add the PeopleSoft OS user and groups, and configure the group membership on each PCA VM. The group ID and user ID must match what is configured on the primary PCA VMs. In this case, the oinstall group is 1001 and the user oracle_psft is 1003.

```
groupadd -g 1001 oinstall
useradd --uid 1003 -g oinstall oracle_psft
usermod -d /home/oracle_psft -g oinstall -s /bin/bash oracle_psft
```

Set up the environment logged on as oracle_psft.

Create a shell file called psft.env (or a file name of your choice) similar to the following modifying according to your environment. The psft.env is similar to those on the primary PCA VMs.

```
export PS_HOME=$BASE_DIR/pt/ps_home8.57.11
export PS_CFG_HOME=/peoplesoft/local/ps_config
export PS_APP_HOME=$BASE_DIR/pt/hcm_app_home
export PS_FILEDIR=$PS_HOME/file
export ORACLE_HOSTNAME=pca1vm51
export ORACLE_BASE=/u01/app/psft
export ORACLE_HOME=/u01/app/psft/pt/oracle-client/19.3.0.0
export COBDIR=/opt/MFCobol
export CLASSPATH=$CLASSPATH:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export TNS_ADMIN=$ORACLE_HOME/network/admin
export JAVA_HOME=/u01/app/psft/pt/jdk1.8.0_221
export TUXDIR=$BASE_DIR/pt/bea/tuxedo/tuxedo12.2.2.0.0
export NLSPATH=$TUXDIR/locale/C
export LD_LIBRARY_PATH=$TUXDIR/lib:$PS_HOME/bin:$ORACLE_HOME/lib:$COBDIR/lib:$LD_LIBRARY_PATH
export LIBPATH=$COBDIR/lib
export SHLIB_PATH=$SHLIB_PATH:$COBDIR/lib
export PATH=$ORACLE_HOME/bin:$TUXDIR/bin:$PS_HOME/jre/bin:$PS_HOME/bin:$JAVA_HOME/bin:$PATH
export PATH=$PATH:$COBDIR/bin
export PS_SERVER_CFG=$PS_HOME/appserv/prcs/DOMAIN_NAME/psprcs.cfg
export WLS_HOME=BASE_DIR/pt/bea/wlserver

cd $PS_HOME
. ./psconfig.sh
cd
```

The Oracle database client that is shipped with PeopleTools 8.57.11 is 12.1.0.2. Later versions can be used as in our example above with 19.3.0.0. Newer versions of PeopleTools will ship with the 19c database client versions.

This shell script should be executed each time you log onto oracle_psft.

At this point, the VM should be ready to install PeopleSoft.

12.4 Configure and Enable ZFS Replication from the Primary Site

The ZFS replication facility can be used to replicate ZFS projects and their shares to a remote ZFS appliance. Once the target ZFS appliance is configured into both primary and secondary sites, the project can be replicated to the new site.

Replication of the MAA_psft projects was established between pca3zfs at the primary site and pca1zfs at the secondary site.

ZFS PROJECT NAME	SHARE NAME
MAA_psoft	psoft

The projects on pca3zfs was configured to replicate to pca1zfs as its target. The MAA_psoft project is configured to replicate continuously. Note that the PeopleSoft Report Repository is using this same share so that any reports that are created or other changes will be replicated.

As described earlier, the MAA_psoft project contains the common infrastructure software, ORACLE_HOME for the database client software, the PeopleSoft PeopleTools install (PS_HOME) and the HRMS 9.2 application in PS_APP_HOME. These are deployed as shared homes at the primary site.

This same project is also where the process scheduler runs jobs that write to the report repository, and where the PIA web servers access these reports.

For further details on ZFS replication, please see:
http://docs.oracle.com/cd/E26765_01/html/E26397/shares__projects__replication.html

12.4.1 Start ZFS Replication

To configure and enable ZFS replication, these steps were followed:

On the ZFS appliance at the primary site, scan04sn, using the BUI, select the first project, **MAA_psoft**.

Click Replication .

Click the plus sign (+) to add a replication action.

Select the target (pca1zfs) from the pull-down list.

It is optional to select **SSL** (not used in this project.)

Select **Scheduled**. Click **Apply** without adding a schedule.

Click the update icon next to **manual**. this should start the first replication snapshot. Depending on the content size of the source share and the bandwidth and latency of the network this can take some time to complete.

12.4.2 Export ZFS Replicas at the Standby Site

On the ZFS appliance at the standby (pca1zfs), click **Projects** on the left side, then click **REPLICA**.

Select the replica: MAA_posft.

To the right on the line with the share **MAA_psoft**, click the **Edit** icon (you must mouse over it to see it.)

Make sure the check box for **Inherit from project**, is checked.

Check the **Export** check box.

Note the export path (it will be the same as that on the primary.)

Click **Apply**.

12.4.3 Mount the ZFS Shares

The MAA_psoft ZFS share contains the shared home deployment of PeopleSoft. All servers will have their PS_HOME access this share. On each PCA VM node do the following:

Create the directory that the psoft ZFS share will be mounted on:

As root:

```
# mkdir -p /u01
```

Add the following file system entries to /etc/fstab:

```
pca1zfs.cloud.osc.oracle.com:/export/psoft /u01   nfs
_netdev,hard,intr,noatime,rsize=32768,wsize=32768 0 0
```

Mount the /u01 file system:

```
# mount /u01
```

12.5 PeopleSoft Application and PIA Web Server Installation on PCA VMs

To install PeopleSoft at the secondary DR site, we use ZFS replication to replicate the PeopleSoft shared homes. There is no software installation required and thus, reduces the time to set up the application at the secondary site.

12.5.1 Create PS_CFG_HOME on Local File System

As is done for the primary PCA VMs, create the subdirectorys for the PS_CFG_HOME on the VM's local file system. Ensure you have sufficient space to support the configuration log files.

On each PCA VM, create the directories. The following example steps shows the procedure on one of the standby PCA VMs, pca1vm51.

Create the PS_CFG_HOME directory on local file system:

As root:

```
# mkdir -p /peoplesoft/local/ps_config

# chown -R oracle_psft:oinstall /peoplesoft
```

For only the application and process scheduler VMs on the secondary site (pca1vm51 and pca1vm52), copy the content of PS_CFG_HOME from the primary site (pca3vm51 and pca3vm52 respectively). The step below is for pca1vm51:

As oracle_psft on pca1vm51:

```
$ cd /peoplesoft/local/ps_config

$ scp –r oracle_psft@pca3vm51:/peoplesoft/local/ps_config/* .
```

If you have set up the file used to configure the environment for the oracle_psft OS user (psft.env), then PS_CFG_HOME should point to /peoplesoft/local/ps_config.

To complete the installation and configuration, we followed the steps in Appendix A section 11.4.2 through 11.4.6 for configuring the application and process scheduler domain server and the PIA web server.

12.5.2 Install Micro Focus COBOL

On the PeopleSoft application and process scheduler VMs, we chose to install the Micro Focus COBOL compiler and license manager in /opt/MFCobol. Each PCA VM has their own copy of Micro Focus COBOL installed locally using the same directory structure. Follow the accompanying instructions in the README to complete the installation. Make sure that the license manager is restarted after system reboot.

12.6 Configure PeopleSoft for Oracle Active Data Guard

As discussed in the main section of the paper, PeopleSoft PeopleTools version 8.52 and higher supports Oracle Active Data Guard. To enable PeopleTools to support Oracle Active Data Guard, the following is required:

- A physical standby database that has Oracle Active Data Guard enabled – described earlier in Appendix B
- A database service that can be started on the Oracle Active Data Guard database instance – described earlier in Appendix B

- A second database schema – ours is called PSFTADG2
- A secondary Access ID created in PeopleSoft
- A database link that ALWAYS points to the primary database service

The procedures for enabling PeopleTools support for Oracle Active Data Guard are documented at:

http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tadm/task_ImplementingOracleActiveDataGuard-3b7d04.html

These procedures should be carried out at both the primary and secondary sites. Note that much of the configuration is performed at the primary site but the application server at the secondary site must still be configured, specifically the application and batch server configuration files.

A few important key items to note:

- The PeopleSoft application server domains at both the primary and secondary sites must be configured to support Oracle Active Data Guard for switchover operations to work properly.
- If the database parameter GLOBAL_NAMES is set to TRUE, then the database link name must match the name of its target database including the DB_DOMAIN if set. The supplied PeopleTools script that creates the database link will fail if GLOBAL_NAMES is set to TRUE.
- Once the PeopleTools application domain server is configured to support Oracle Active Data Guard, both the primary and standby databases and all database services must be up before attempting to start the application server, otherwise, the application server startup will fail.
- Make sure that the database link you create as part of the scripts supplied by PeopleSoft ALWAYS points to the primary database using a service that only runs on the primary. In our case the connect string alias we used is HR92U033_PRIMARY.
- If the standby database is down for maintenance the PSQUERY service will be down. Start the PSQUERY service on the primary until the standby database is brought back up, at which point you can relocate the PSQUERY service back to the standby.

The following example of the TNS connect string alias HR92U033_PRIMARY should be placed into the $TNS_ADMIN/tnsnames.ora on each RAC DB node at both the primary and standby sites:

```
HR92U033_PRIMARY =
(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)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
 )
```

This TNS connect string will only connect to the primary database as that is where the HR92U033_ONLINE service can start. Do not place this TNS connect string alias onto any of the middle tiers. This should only be on each of the primary and standby database nodes so that the connect string used by the database link can be resolved.

The database link creation statement is:

```
CREATE DATABASE LINK PRIMARY CONNECT TO EMDBO IDENTIFIED BY <password> USING
'HR92U033_PRIMARY';
```

PeopleTools requires a second TNS connect alias to the physical standby service. The PSQUERY service was defined earlier in this appendix. This connect string must be placed into all tnsnames.ora accessible by the application domain server and process scheduler, and the alias name must be used for the Standby **DBName** in the PSADMIN utility. The following is the TNS connect string alias called PSFTADG that is used in this project.

```
PSFTADG=
(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)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = PSQUERY)
       )
    )
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = PSQUERY)
       )
    )
 )
```

On the primary database a new row must be inserted into the PS.PSDBOWNER table so that the application server can authenticate with the Oracle Active Data Guard standby database. Do the following on the primary database using SQL*Plus:

```
sqlplus / as sysdba

SQL> ALTER SESSION SET CONTAINER = HR92U033;
SQL> INSERT INTO PS.DBOWNER VALUES (‘PSFTADG’, ‘EMDBO’);

COMMIT;
```

The PDB in this project is named HR92U033. The above INSERT statement is replicated to the standby database via redo transport. The application server configuration is now complete.

12.7 F5 BIG-IP Load Balancer

See Appendix A section 11.5 for a discussion on implementing the F5 load balancer. The same was implemented at the DR site along with the health monitor.

13 Appendix C: Standby Site Test

This section describes how to validate that the standby site is ready to assume the primary role in the event of a disaster. It is important to test the PeopleSoft application at the DR site to validate that all components are working properly. Because the Oracle database supports snapshot standby it is not necessary to shut down the primary to conduct validation testing. While the physical standby is in the snapshot standby role, redo from the primary is still being received at the standby providing data protection. The outstanding redo is applied when the standby resumes the physical standby role. For the application and PIA components we simply need to create a snapshot of the ZFS share and mount it. This allows for any site-specific configuration changes to be made without disrupting the production site.

Before following the procedures below, ensure that the application server domains and the PIA web servers are down at the standby site. The database should be in the physical standby role with Oracle Active Data Guard enabled.

CAUTION: Before you proceed with the standby site test, make sure that no production user or batch process can access the PeopleSoft application at the standby site. Ensure that the URLs (and DNS) do not accidentally route traffic to the standby site or those transactions will be lost when the database is reverted back to a physical standby.

13.1 Physical Standby to Snapshot Standby

Because we have Data Guard Broker configured, a single command will do the job of converting the physical standby to a snapshot standby. As discussed earlier, the snapshot standby allows the database to be opened read-write for testing. A guaranteed restore point is created so that when the testing is complete and the database is reverted back to a physical standby, all changes made during the testing are discarded. Once the database is open as a snapshot standby, the application can start as it normally would.

13.1.1 Convert the Physical Standby to Snapshot Standby

Log on to Data Guard Broker on either the primary or standby database.

```
$ dgmgrl
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.
DGMGRL> connect sys/<password>
Connected.
DGMGRL> show configuration lag
Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database
                   Transport Lag:      0 seconds (computed 0 seconds ago)
                   Apply Lag:          0 seconds (computed 0 seconds ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 27 seconds ago)
```

Convert to snapshot standby.

```
DGMGRL> convert database 'CDBHCM_OSC1B' to snapshot standby;
Converting database "CDBHCM_OSC1B" to a Snapshot Standby database, please wait...
Database "CDBHCM_OSC1B" converted successfully
```

Check the database services.

```
$ srvctl status service -db CDBHCM_osc1b
Service hr92u033_batch is running on instance(s) CDBHCM1,CDBHCM2
Service hr92u033_online is running on instance(s) CDBHCM1,CDBHCM2
Service PSQUERY is not running.
```

In order for the application servers to start, we need to start all services including PSQUERY on the snapshot standby.

```
$ srvctl start service -db CDBHCM_OSC1B -s PSQUERY
$ srvctl status service -db CDBHCM_osc1b
Service hr92u033_batch is running on instance(s) CDBHCM1,CDBHCM2
Service hr92u033_online is running on instance(s) CDBHCM1,CDBHCM2
Service PSQUERY is running on instance(s) CDBHCM1,CDBHCM2
```

13.1.2 Create a ZFS Snapshot of ZFS share

There is one replicas on pca1zfs:

- MAA_psoft exported as /export/pesoft

Using the Oracle Storage Appliance UI, create a snapshot clone off of the replica and mount it. You can provide a different export mount point such as /export/psoft_snapshot.

13.1.3 Mount the Shared Cloned File System

Now that the ZFS clone has been created, on all PCA VMs at the standby, mount the cloned NFS file system.

As root, on each PCA VM, create the same mount point directory if not already created:

```
# mkdir /u01
```

On each PCA VM as root, add the ZFS export and mount point into /etc/fstab

```
pca1zfs.cloud.osc.oracle.com:/export/psoft_shapshot /u01   nfs
_netdev,hard,intr,noatime,rsize=32768,wsize=32768 0 0
```

On each PCA VM as root, mount the file system:

```
# mount /u01
```

13.2 Start the Application Domain Server and Process Scheduler

We use the startAPP.sh, startPS.sh and startWS.sh scripts described earlier. If not already done, copy over these scripts from the primary application server VMs and the PIA web server VMs to the VMs on the standby PCA.

On the application server VMs, as oracle_psft, start the application server and process scheduler:

```
$ ./startApp.sh
$ ./startPS.sh
```

On the PIA web server VMs as oracle_psft, start the PIA web server:

```
$ ./startWS.sh
```

13.3 Application Testing

At this point you can test the PeopleSoft application. Perform workload testing and run actual batch jobs to ensure everything is working properly. Make sure that all of the configurations are correct and all of the application components function properly.

13.4 Shutdown Application Servers and PIA Web Servers

When testing is complete, shut down the PIA web server and the application domain servers. The stopAPP.sh, stopPS.sh, and stopWS.sh scripts will bring down all components.

13.5 Convert the Snapshot Standby Back to Physical Standby

Use Data Guard Broker to convert the snapshot standby back to a physical standby database. The physical standby will resume its role as an Oracle Active Data Guard standby.

```
$ dgmgrl
DGMGRL> connect sys/<password>
Connected.
DGMGRL> show configuration lag

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Snapshot standby database
                   Transport Lag:      0 seconds (computed 1 second ago)
                   Apply Lag:          41 minutes 27 seconds (computed 1 second ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 61 seconds ago)
DGMGRL> convert database 'CDBHCM_OSC1B' to PHYSICAL STANDBY;
Converting database "CDBHCM_OSC1B" to a Physical Standby database, please wait...
Operation requires a connection to database "CDBHCM_OSC1A"
Connecting ...
Connected to "CDBHCM_osc1b"
Connected as SYSDBA.
Oracle Clusterware is restarting database "CDBHCM_OSC1A" ...
Connected to "CDBHCM_osc1b"
Connected to "CDBHCM_osc1b"
Continuing to convert database "CDBHCM_OSC1B" ...
Database "CDBHCM_OSC1B" converted successfully
DGMGRL> show configuration lag

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database
                   Transport Lag:      0 seconds (computed 0 seconds ago)
                   Apply Lag:          0 seconds (computed 0 seconds ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 50 seconds ago)
```

13.6 Unmount the ZFS Snapshot on All Servers

On all PCA VMs, unmount the ZFS snapshot as follows:

```
# umount /u01
```

It is optional to discard the clone however, it is likely that you will want to create a new clone for future testing.

14 Appendix D: Site Switchover

Site switchover is when the primary and standby reverse roles. Site switchover is always a planned event. This capability is very valuable for allowing business to continue at the secondary site while maintenance is performed at the primary site.

The PeopleSoft site switchover can be performed by Oracle Site Guard or manually. The process for using Oracle Site Guard is documented in the technical brief entitled “Application-Level Disaster Recovery using Site Guard” which can be found on the Oracle Site Guard site. The manual steps for performing a switchover (or switchback) are:

1. Drain the process scheduler queues or place some jobs on hold.

It is important that no process scheduler jobs are running when attempting to switchover to the secondary DR site.

2. Shut down the PeopleSoft application and process scheduler domains and PIA web servers.

Use stopAPPS.sh, stopPS.sh, and stopWS.sh scripts described earlier to shut down the application.

3. Perform a database switchover with Data Guard Broker.

```
$ dgmgrl
DGMGRL> connect sys/<password>
Connected.
DGMGRL> show configuration lag

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database
                   Transport Lag:      0 seconds (computed 1 second ago)
                   Apply Lag:          0 seconds (computed 1 second ago)

Fast-Start Failover:  Disabled

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

It is a good practice to validate that the standby database is ready:

```
DGMGRL> validate database 'CDBHCM_OSC1B'

  Database Role:     Physical standby database
  Primary Database:  CDBHCM_OSC1A

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

  Managed by Clusterware:
    CDBHCM_OSC1A:  YES
    CDBHCM_OSC1B:  YES
```

4. Now, do the switchover:

```
DGMGRL> switchover to 'CDBHCM_OSC1B'
Performing switchover NOW, please wait...
Operation requires a connection to database "CDBHCM_OSC1B"
Connecting ...
Connected to "CDBHCM_osc1b"
Connected as SYSDBA.
New primary database "CDBHCM_OSC1B" is opening...
Oracle Clusterware is restarting database "CDBHCM_OSC1A" ...
Connected to "CDBHCM_osc1a"
Connected to "CDBHCM_osc1a"
Switchover succeeded, new primary is "CDBHCM_OSC1B"

DGMGRL> show configuration lag

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1B - Primary database
    CDBHCM_OSC1A - Physical standby database
                   Transport Lag:      0 seconds (computed 0 seconds ago)
                   Apply Lag:          0 seconds (computed 0 seconds ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 44 seconds ago)
```

5. Perform a role reversal of the PCA ZFS share. For details on role reversal for the Oracle Cloud Appliance ZFS storage, see the Oracle Cloud Appliance documentation: Disaster Recovery with Remote Replication at: https://docs.oracle.com/cd/F13758_01/html/F13769/grfok.html#scrolltoc
6. At the new primary (pca1) site, once the ZFS role reversal has been completed, mount the /u01 file system as root on all PeopleSoft application tier PCA VMs

   # mount /u01
7. Start the PeopleSoft application and PIA web servers on the new primary (pca1 site) using the startAPP.sh, startPS.sh, and startWS.sh scripts.
8. If required, perform a DNS push to propagate name resolution to the new primary. If you have an F5 BIG-IP load balancer this may be handled for you with Global Traffic Manager.

To switch back where pca3 resumes its original primary role and pca1 is the standby, follow the above steps but for switching the database over to CDBHCM_OSC1A and the replication of the shares from pca3 to pca1.

15 Appendix E: Site Failover

For a site failover scenario, we assume that the primary site is unavailable and completely inaccessible. In this scenario the standby assumes the primary role.

The PeopleSoft site failover can be performed by Oracle Site Guard or manually. The process for using Oracle Site Guard is documented in the technical brief entitled "Application-Level Disaster Recovery using Site Guard".

The following are manual steps to perform a site failover.

1. Push new DNS entries for the new primary for client user access. If an F5 BIG-IP load balancer is used, this may be achieved with Global Traffic Manager inside an enterprise network cloud. For customer-facing services and B2B interfaces this may require the network administrators to push routing rules to the various point of presence (POP) to affect the change globally.
2. In Data Guard Broker issue a failover command to cause the standby database to become the new primary. Because the Data Guard protection mode is Maximum Performance there may be a small amount of data loss. The failover will take seconds to just a few minutes. Log into Data Guard Broker on the standby. The `show configuration` command will show errors. Perform the failover command as shown below.

```
DGMGRL> show configuration

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1B - Primary database
    Error: ORA-12514: TNS:listener does not currently know of service requested in connect
descriptor

    CDBHCM_OSC1A - Physical standby database

Fast-Start Failover:  Disabled

Configuration Status:
ERROR   (status updated 0 seconds ago)
```

Note: In the above show configuration output, we simulated a database failure by issuing a SHUTDOWN ABORT on all primary database instances. This will generate the ORA-12154 error as shown above. If the site was unreachable due to a power or network failure, or the entire site was lost, there would be other errors.

3. Now, we perform the failover running Data Guard Broker at the standby site on one database node:

```
DGMGRL> failover to 'CDBHCM_OSC1A';
Performing failover NOW, please wait...
Failover succeeded, new primary is "CDBHCM_OSC1A"
```

The standby database is now the primary.

4. Force the ZFS storage the new primary site to now assume the primary role. This may require severing the replication relationship. For details, please refer to the Oracle Cloud Appliance ZFS storage documentation at: https://docs.oracle.com/cd/F13758_01/html/F13769/grfok.html#scrolltoc.
5. On the standby PCA VMs (now, the new primary) start the application using startAPP.sh, startPS.sh, and startWS.sh scripts.

NOTE: Be sure that the PSQUERY database service is started on the new primary. In this project the PSQUERY service was defined for both PRIMARY and PHYSICAL_STANDBY database roles. Once the failover completed, the PSQUERY service was started automatically. If this service is not started, the PeopleSoft application domain servers will fail to start.

16 Appendix F: Site Reinstantiate

Recovery of the primary site after a failure depends on the nature of the failure. For a complete catastrophic lost or severe damage to the primary data center facility, this requires rebuilding and/or repairing that data center or relocating to new facilities. New systems must be provisioned and configured. Oracle Exadata Database Machine comes pre-configured to reduce a significant amount of the setup time. In addition, Oracle Cloud Infrastructure (OCI) can be used to quickly provision all of the infrastructure needed to support the new standby site. The OCI has Exadata Cloud Service (ExaCS) or Exadata Cloud at Customer (ExaCC) whichever is preferred.

If the primary site was lost due to a prolonged network or power outage, but the facility remained intact, once the infrastructure services have been restored it is then just a matter of synchronizing the old primary site and bringing it online as a standby. This is the scenario for this case study.

ZFS replication will need to be resumed. See the Oracle Cloud Appliance ZFS storage documentation at: https://docs.oracle.com/cd/F13758_01/html/F13769/grfok.html#scrolltoc for further details on restoring the replication process.

16.1 Reinstate the PeopleSoft Database

To reinstate the old primary database the following steps were performed.

1. Bring up the Grid Infrastructure and databases.

Once the power and/or network have been restored, bring up the cluster if it is not already up. Ensure that the ASM instances are up. If the systems are powered up the database nodes will start Cluster Ready Services (CRS) then starts and form the cluster. In turn, the cluster will start up ASM, the listeners, VIPs, and databases.

Because our database is under Data Guard control, once it starts, Data Guard notices that it is up and then takes control and keeps it from opening as a primary. At this point we can use Data Guard Broker to reinstate the database as the new standby. If flashback was not enabled, then a full restore from a backup (or from the primary using RMAN RESTORE FROM SERVICE) would be necessary to reinstantiate the standby database.

If the database on the old primary (pca1) has not yet been started, start it with the following command:

```
srvctl start database -db CDBHCM_OSC1B
```

Let it start. You will see errors similar to the following:

```
Data Guard: version check completed
Data Guard determines a failover has occurred - this is no longer a primary database
ORA-16649 signalled during: ALTER DATABASE OPEN
```

2. On one of the new primary Exadata database nodes, run DGMGRL and display the configuration:

```
DGMGRL> show configuration lag

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database (disabled)
      ORA-16661: the standby database needs to be reinstated
                   Transport Lag:      (unknown)
                   Apply Lag:          (unknown)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 59 seconds ago)
```

3. From the Broker, reinstate CDBHCM_OSC1B as the new standby:

```
DGMGRL> reinstate database 'CDBHCM_OSC1B'
Reinstating database "CDBHCM_OSC1B", please wait...
Reinstatement of database "CDBHCM_OSC1B" succeeded
```

Now, check the Data Guard Broker configuration:

```
DGMGRL> show configuration lag;

Configuration - cdbhcm_dg

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_OSC1A - Primary database
    CDBHCM_OSC1B - Physical standby database
                   Transport Lag:      0 seconds (computed 0 seconds ago)
                   Apply Lag:          0 seconds (computed 0 seconds ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 55 seconds ago)
```

4. Migrate the PSQUERY database service to the standby.

The PSQUERY database service is started when the database is in the role of a physical standby. On the standby compute nodes, the PSQUERY should be up. If not, start the PSQUERY service:

```
srvctl start service –db CDBHCM_OSC1B –service PSQUERY
```

Once it is started, you can shut down the PSQUERY on the primary and users running reports and queries against that service will migrate back to the Oracle Active Data Guard standby database.

```
srvctl stop service –db CDBHCM_OSC1A –s PSQUERY
```

17 Appendix G: PeopleSoft Planned Maintenance

Both Oracle Exadata Database Machine and Oracle Private Cloud Appliance hardware infrastructure have provisions for reducing downtime for a variety of maintenance activities such as patching and upgrades. For Oracle Exadata Database Machine, the following are activities that can be performed with no down time for any running application:

- Rolling storage cell patching or upgrades
- Cell disk replacement or entire cell repair[4]

The following maintenance activities can be performed with no down time for PeopleSoft:

- Rolling database node OS kernel patching or upgrades
- Rolling grid infrastructure patching or upgrades
- Oracle RAC rolling database bundle patch release update patch application

[4] Subject to the available amount of required mirror free to restore redundancy of ASM disk groups. Please see My Oracle Support note 1551288.1

- Out-of-Place patching of PS_HOME or PS_APP_HOME[5]

Oracle database version 19c now has support for PLANNED MAINTENANCE in which connections are “drained” off a database instance that will undergo maintenance. When there is to be a planned maintenance on a given database, it is performed in a RAC rolling – instance by instance fashion. This has been available for several database releases and for many years. What is new is the ability to have active connections on the instance about to undergo maintenance (patching) move from that instance to other instances where the service is still available during the drain timeout period. This feature is enabled by specifying a drain timeout period when the service is created, modified or at the time the service is to be shutdown using srvctl command. There are two flavors of this feature beginning in 19c:

- Use of Fast Application Notification (FAN) events
- Use of connection test (non-FAN events)

Use of FAN events allows applications that are FAN aware such as OCI clients, Universal Connection Pool (UCP), ojdbc drivers that support FAN, ODP.NET to benefit automatically without any user intervention.

Use of connection tests (non-FAN events) is another mechanism in which the application issues a simple SQL statement to test the connection such as “SELECT X FROM DUAL”, “BEGIN NULL; END;” or any other simple SQL statement to test if the connection is valid. In addition, the Java jdbc call to isValid() can also be used. When the drain timeout period begins for a given instance, when a connection test is issued, the database closes the connection causing the application to reconnect to another instance where the service is available. For more information, please see the Oracle documentation: Real Application Clusters Administrator’s Guide, Chapter 6.

PeopleSoft is FAN event aware in all recent PeopleTools version. Therefore, PeopleSoft sessions will benefit from the use of FAN events.

17.1 Case Study Example of Planned Maintenance

The following is a specific example workflow of applying a database release update (RU) for Oracle Database 19c while PeopleSoft HCM is running with online users.

The RU application workflow is performed as follows:

- Apply a database release update (RU) to the standby database first (Standby-First)
- Apply a database release update (RU) to the primary database using Oracle RAC rolling upgrade

17.2 Standby First Maintenance

Because PeopleSoft is configured with Oracle Active Data Guard we need to apply the RU on the standby database first in an Oracle RAC rolling manner. The database unique name for the standby is CDBHCM_OSC1B.

The steps to apply the patch are:

Stop and disable the PSQUERY service on the first RAC instance CDBHCM1. Notice the drain_timeout of 300 seconds (5 minutes). During this time, any active sessions on this instance will be moved to CDBHCM2 where the PSQUERY service is still running. The -verbose will present output to the console showing the number of sessions to be drained.

NOTE: Any *inactive sessions* connected to the PSQUERY service on instance CDBHCM1 may remain and not move for the duration of the drain timeout period (300 seconds) and will be disconnected when the service is shutdown. These sessions may not reconnect to other instances unless there is a demand for new connections. This is expected behaviour.

[5] Out-of-place patching depends on the nature of the patch. If the patch being applied to the PeopleSoft stack is compatible with the existing code, the patch can be applied to a cloned PS_HOME and servers can be shut down and restarted in a rolling manner on the new PS_HOME.

```
srvctl stop service -db CDBHCM_OSC1B -service PSQUERY -instance CDBHCM1 -drain_timeout 300 -
verbose

srvctl disable service -db CDBHCM_OSC1B -service PSQUERY -instance CDBHCM1
```

Once the service is stopped, stop the instance CDBHCM1.

```
srvctl stop instance -db CDBHCM_OSC1B -instance CDBHCM1 -stopoption immediate

srvctl start instance -db CDBHCM_OSC1B -instance CDBHCM1
```

Check that the connections went to the second instance. We know that the server name connections are from the scan04 Oracle Exadata Database Machine.

```
sqlplus / as sysdba
  SQL> select inst_id,count(*)
  from gv$session
  where program like '%pca1%'
  group by inst_id;

   INST_ID   COUNT(*)
---------- ----------
         1          0
         2          15
```

Instance 1 has no connections.

Conduct the maintenance, apply the RU per the instructions. Check for any errors in the patch log files and correct any issues as necessary.

Re-enable the PSQUERY service on the CDBHCM1 instance and start the service. Note that the instance may already be up.

```
srvctl enable service -db CDBHCM_OSC1B -service PSQUERY -instance CDBHCM1

srvctl start service -db CDBHCM_OSC1B -service PSQUERY -instance CDBHCM1
```

Repeat the above steps for the second instance, CDBHCM2, replacing CDBHCM1with CDBHCM2.

Test and validate the patch by running the standby for a period of time (up to 7 days if necessary). Some of the PeopleSoft Application Engine processes, Query Viewer, XML Plublisher, and Tree Viewer will use the Oracle Active Data Guard database with this release update in place.

As this is a Standby-First patching scenario, you can convert the physical standby database to a SNAPSHOT STANDBY, run datapatch and start the PeopleSoft application. This allows testing of the full application at the standby site. Refer back to section 13: Appendix C Standby Site Test above for further details. Once the database is converted back to a physical standby, any changes made by datapatch and the application will be undone.

NOTE: For the physical standby the datapatch script has not yet been run. This will be done later.

17.3 Primary RAC Rolling Maintenance

On the primary database stop the HR92U033_ONLINE and HR92U033_BATCH services specifying a drain_timeout, and disable them, stop the instance for each node we patch, then re-enable and restart them, one node at a time, one node after another.

Stop and disable the HR92U033_ONLINE and HR92U033_BATCH services specifying a drain timeout:

```
srvctl stop service -db CDBHCM_OSC1A -service HR92U033_ONLINE -instance CDBHCM1 -
drain_timeout 300 -verbose

srvctl stop service -db CDBHCM_OSC1A -service HR92U033_BATCH -instance CDBHCM1 -drain_timeout
300 -verbose
```

Once both services have been stopped following the drain timeout, then disable the services:

```
srvctl disable service -db CDBHCM_OSC1A -service HR92U033_ONLINE -instance CDBHCM1

srvctl disable service -db CDBHCM_OSC1A -service HR92U033_BATCH -instance CDBHCM1
```

Shut down and restart the CDBHCM1 instance.

```
srvctl stop instance -db CDBHCM_OSC1A -instance CDBHCM1 -stopoption immediate

srvctl start instance -db CDBHCM_OSC1A -instance CDBHCM1
```

Check that the connections went to the second instance.

```
sqlplus / as sysdba
  SQL> select inst_id,count(*)
  from gv$session
  where program like '%pca3%'
  group by inst_id;
   INST_ID   COUNT(*)
---------- ----------
         1          0
         2          67
```

Instance 1 has no connections.

Conduct maintenance and apply the RU per the instructions.

Check for errors in the patch log files and correct any issues as necessary.

Re-enable and restart the HR92U033_ONLINE and HR92U033_BATCH services.

```
srvctl enable service -db CDBHCM_OSC1A -service HR92U033_ONLINE -instance CDBHCM1

srvctl enable service -db CDBHCM_OSC1A -service HR92U033_BATCH -instance CDBHCM1

srvctl start service -db CDBHCM_OSC1A -service HR92U033_ONLINE -instance CDBHCM1

srvctl start service -db CDBHCM_OSC1A -service HR92U033_BATCH -instance CDBHCM1
```

Repeat the above steps for the second instance, CDBHCM2, replacing CDBHCM1with CDBHCM2.

Once the RU binaries have been applied for the CDBHCM2 instance, run the datapatch:

```
cd $ORACLE_HOME/OPatch

./datapatch -verbose
```

Check for any errors. If there are no errors, the RU patch process is complete.

Connect with us

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

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

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.