PeopleSoft Maximum Availability Architecture

A Case Study in Oracle Cloud Infrastructure

October, 2023, Version [1.0]

List of figures

List of images

List of tables

1 Introduction

Oracle's PeopleSoft application is a comprehensive suite of integrated business applications providing human capital management, financial management, procurement and supplier management, project portfolio management, asset lifecycle management, order and inventory management, and campus solutions. These mission critical functions require protection, to ensure they are available at all times. Whether PeopleSoft is deployed in Oracle Cloud Infrastructure (OCI) or on-premises, planning and implementing strategies for high availability and disaster recovery are paramount.

In this paper, we apply the principles of Oracle's Maximum Availability Architecture (MAA) to PeopleSoft Release 9.2. Oracle MAA is Oracle's best practice blueprint for implementing Oracle high availability technologies and recommendations. The goal of MAA is to achieve the optimal high availability architecture at the lowest cost and complexity.

This paper provides:

- A high-level introduction to Oracle Cloud Infrastructure (OCI)
- The PeopleSoft MAA architecture
- Installation, configuration, and operational best practices for an end-to-end MAA implementation of PeopleSoft on OCI
- A case study showing migration of a PeopleSoft implementation from on-premises hardware into an MAA deployment on Exadata Database on Dedicated infrastructure (ExaDB-D) within OCI

2 Oracle Cloud Infrastructure and Exadata Cloud Service

OCI provides a broad range of cloud service offerings hosted in Oracle's data centers in various regions around the globe. With OCI, it is simple to provision, operate, and manage all the necessary componentry for deploying large-scale enterprise applications with high volume workload.

2.1 Oracle Database in the Cloud

The Exadata Database on Dedicated Infrastructure (ExaDB-D) provides:

- The choice of Exadata generation and shape (base, quarter, half, or full) or elastic shapes. See Scaling Resources within an Exadata Infrastructure Instance.
- All the high-performance technologies and innovations that the selected generation of Exadata brings, including:
  - MAA best practices integrated into the Exadata infrastructure
  - High redundancy database storage using Automatic Storage Management (ASM)
  - Pre-configured OS for the database servers, optimized for running Oracle Clusterware and Real Application Cluster (RAC) databases
- Virtualization of the environment

It is amazingly simple to deploy an Exadata Database on Dedicated Infrastructure environment using the OCI console. There are only two steps:

1. Provision the Exadata infrastructure
2. Provision the VM clusters on top of that infrastructure

Once these two steps are complete – which should take just a few hours, depending on the shape – you can start using your new ExaDB-D.

This paper will not provide an in-depth discussion of OCI and all its services but will discuss OCI resources and components as they relate to deploying PeopleSoft. If you are new to Oracle Cloud, we highly recommend that you read the Welcome to Oracle Cloud Infrastructure documentation. There is a variety of tutorials to help you understand the concepts that will be used throughout this technical brief.

2.2 Tooling for PeopleSoft on Oracle Cloud Infrastructure

PeopleSoft on OCI operates the same as it does on premises. However, the PeopleSoft Cloud Manager enhances administrators' experience in OCI by providing an additional layer of application-aware management, migration, and life-cycle management tools.

The PeopleSoft Cloud Manager can be used to migrate, then manage, an environment into OCI, or can be configured for use after the environment is deployed. If your environment is already deployed in OCI, you can import its metadata into the PeopleSoft Cloud Manager. For further details, refer to: PeopleSoft Cloud Manager for Oracle.

For an overview of migrating PeopleSoft to Oracle Cloud Infrastructure, we recommend reading Considerations for Migrating PeopleSoft to Oracle Cloud Infrastructure.

2.3 Tooling for Migrating to the Cloud

One of Oracle's cloud tools for migrating databases into Oracle Exadata Database Service on Dedicated Infrastructure (ExaDB-D) is Zero Downtime Migration (ZDM). ZDM was designed and built to help automate and perform the heavy lifting when migrating databases of virtually any size into Oracle Cloud. It was also designed to utilize Oracle

Database technology and MAA best practices to minimize database and application downtime during the migration process. ZDM provides a range of options, accommodating migrating from on-premises, from other cloud providers, from Gen 1 Exadata Cloud at Customer Service (ExaCC), from Oracle Exadata Database Service on Cloud@Customer (ExaDB-C@C), and from Oracle Cloud Classic.

To use ZDM in our project, we followed the steps described in Move to Oracle Cloud Using Zero Downtime Migration.

2.4 Extending for Maximum Availability

Your PeopleSoft implementation is critical to the operation of your business and warrants protection against disasters. For maximum availability, we recommend that you have a redundant image of your production environment in a separate location, ready to take over if disaster strikes your primary data center.

This is where Oracle Cloud shows its power. In non-cloud deployments, you would need to have a second data center sufficiently far away from the primary site to avoid impact from regional outages. Oracle Cloud provides these features for setting up a secondary environment for disaster recovery:

- Data centers already deployed around the globe
- Rapid provisioning of resources using Terraform
- Automation tooling for instantiating and configuring Oracle Data Guard

These features substantially reduce the time and effort required to deploy a fully functional disaster recovery site. In just a few days, a full stack PeopleSoft for DR can be deployed.

3 PeopleSoft Maximum Availability Architecture Overview

PeopleSoft Maximum Availability Architecture 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.

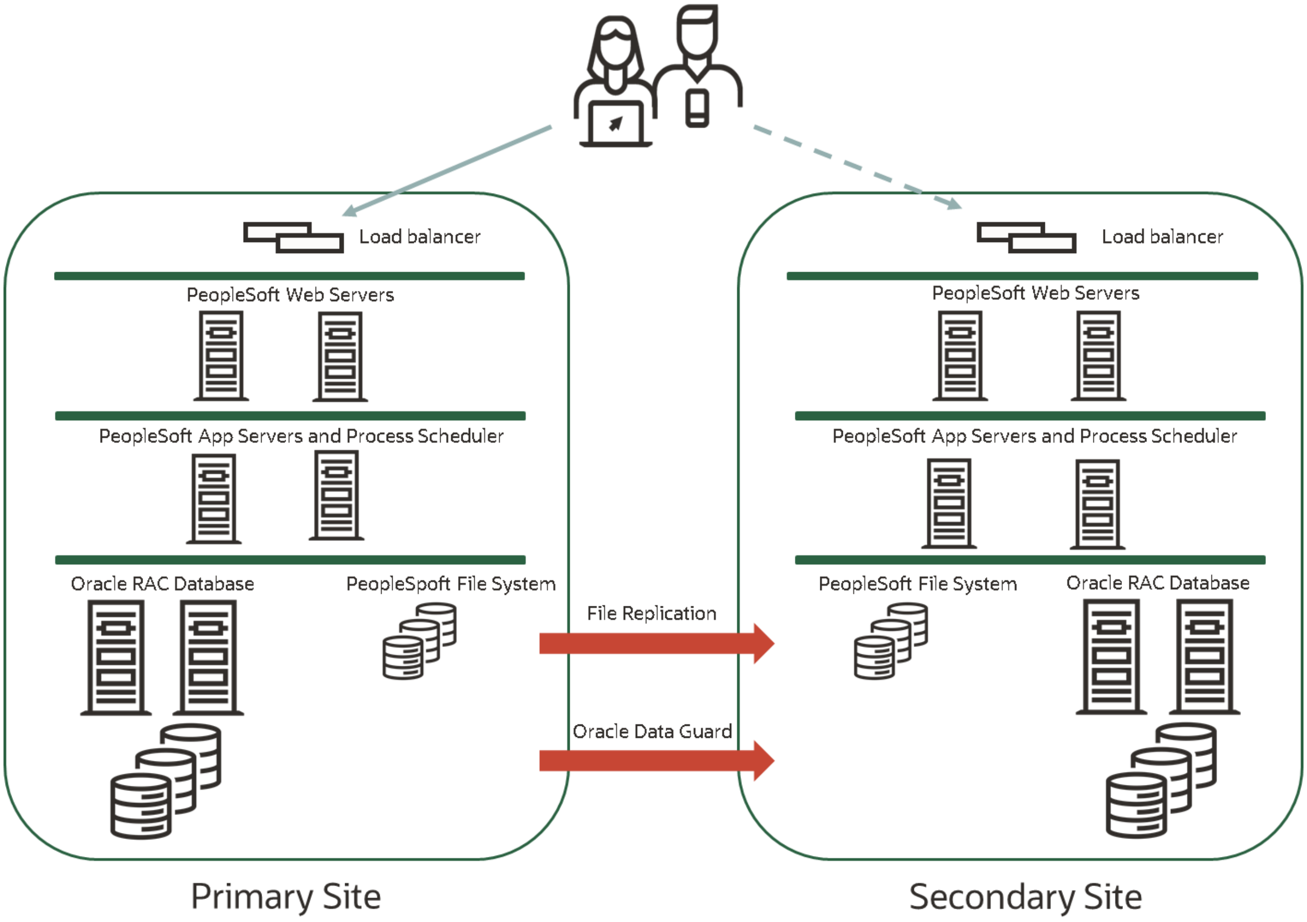

Figure 1: PeopleSoft Maximum Availability Architecture

Figure 1 above shows a full-stack MAA architecture, including both primary and secondary sites. The secondary site is a replica of the primary. Each site consists of the following:

- An HTTPS load balancer for web-based application services
- Two servers that host the PeopleSoft Pure Internet Architecture (PIA) domain
- Two servers that host both the PeopleSoft Application Server and the Process Scheduler domains
- A shared file system for PeopleSoft application software and report repository
- An Oracle Real Application Clusters (RAC) database, with two database servers and shared storage
- Oracle Active Data Guard, which allows routing of “mostly read operations” to the standby database while keeping the standby database current with the primary

Both the application tier shared file system and the database are replicated to the secondary site – the application tier using rsync, and the database tier using Oracle Data Guard.

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

3.1 Oracle Database Maximum Availability Architecture

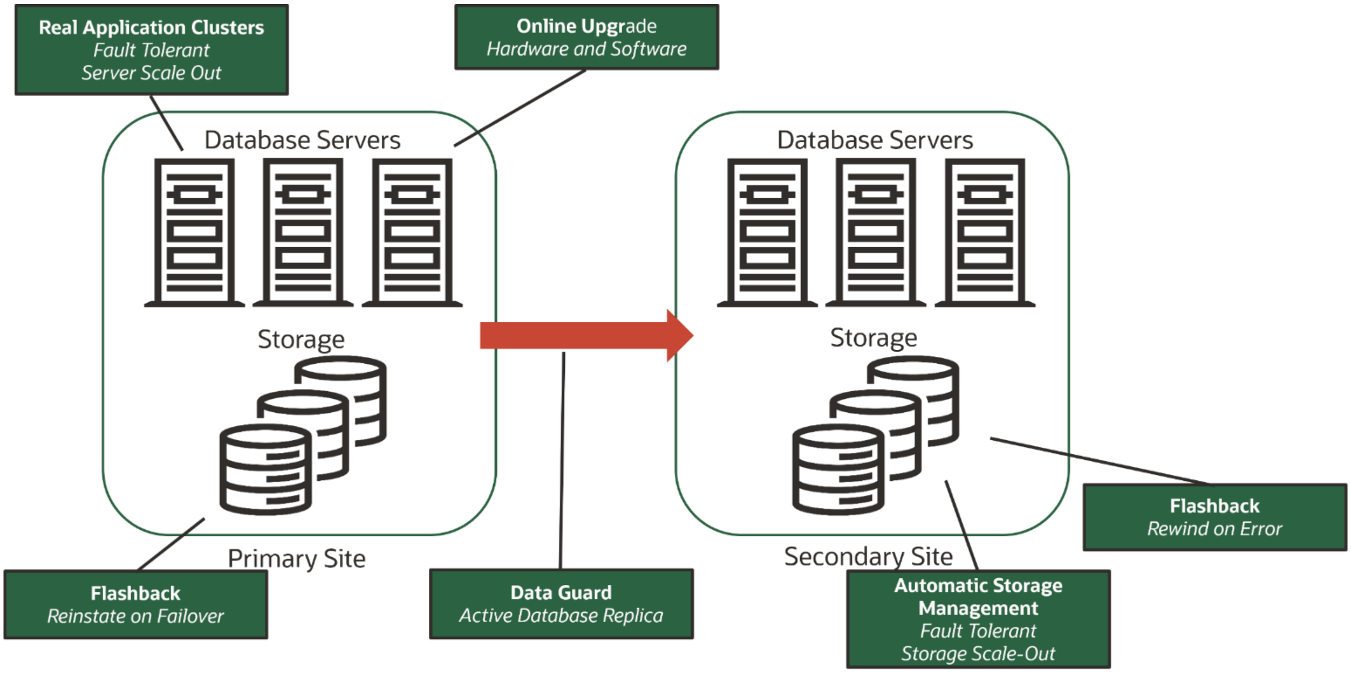

Figure 2: Oracle Database Maximum Availability Architecture

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 Real Application Clusters and Oracle Clusterware
- Oracle Active Data Guard
- Oracle Flashback Database
- Oracle Automatic Storage Management, with high redundancy disk groups
- Oracle Recovery Manager and Oracle Secure Backup

The PeopleSoft deployment as described in this case study utilizes all the above technologies. See: *Oracle Database High Availability Overview* for a thorough introduction to Oracle Database high availability products, features, and best practices. For more information specifically about implementing Active Data Guard with PeopleSoft, see Implementing Oracle Active Data Guard.

3.2 PeopleSoft Maximum Availability Architecture

The following diagram illustrates a simple PeopleSoft high availability architecture, for the primary site:

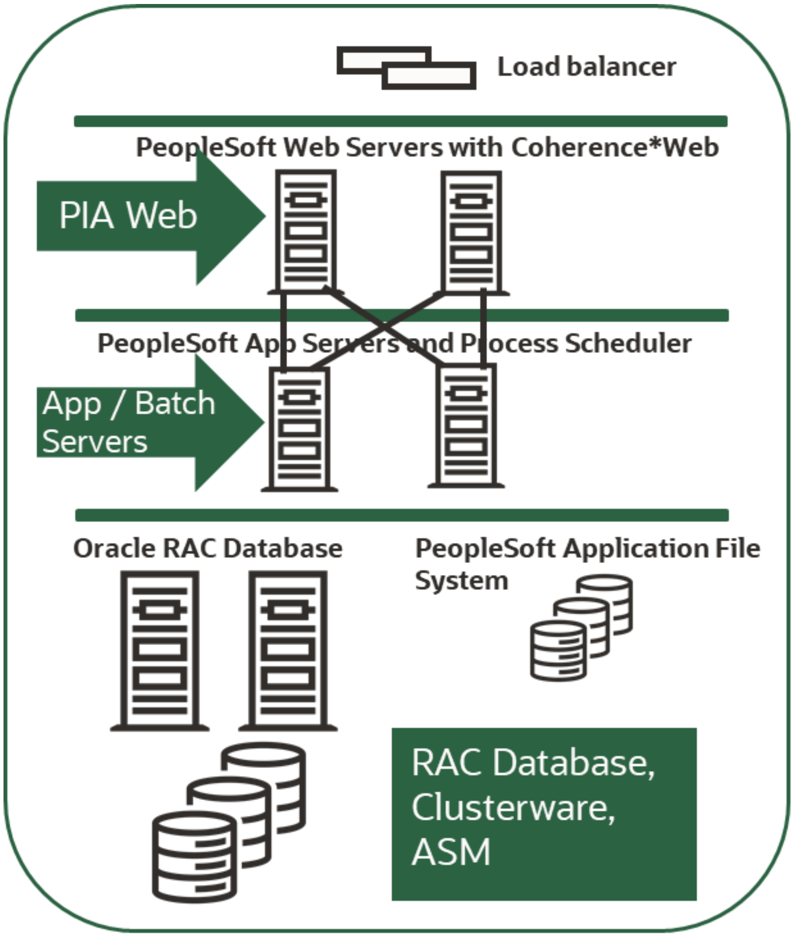
Load balancer
PeopleSoft Web Servers with Coherence*Web
PIA Web
PeopleSoft App Servers and Process Scheduler
App / Batch Servers
Oracle RAC Database
PeopleSoft Application File System
RAC Database, Clusterware, ASM

Figure 3: PeopleSoft High Availability Architecture

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 each group of servers have adequate capacity to run peak load even when one server is unavailable.

This architecture is duplicated at and synchronized to a second site to provide MAA for PeopleSoft.

3.2.1 PeopleSoft Application File System Layout Basics

This case study was done using PeopleSoft HRMS 9.2 U033 and PeopleTools 8.57.11. Before describing the deployment options, we define a few environment variables:

- PS_HOME: the file system location where the PeopleTools software is installed.
- PS_APP_HOME: the file system location where the PeopleSoft application (HRMS, FIN, EPM, etc.) is installed, a variable available since PeopleTools 8.52.
- PS_CFG_HOME: the file system location holding configuration and log files for the application and web server domains.
- PS_CUST_HOME: the file system location for custom code and files for adding customized functionality for PeopleSoft.

- COBDIR: the file system location where the MicroFocus Server Express COBOL compiler and run-time libraries are stored. It is required that MicroFocus Server Express be installed on each server that will run COBOL programs. This component cannot be shared due to license key restrictions.

There is also a parameter in the WebLogic configuration, ReportRepositoryPath, that holds the file system directory for report output.

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

- **Local Homes**: installing all the PeopleSoft software and required infrastructure components such as Oracle JDK/JRE, Oracle WebLogic Server, Oracle Tuxedo, PeopleTools, and the PeopleSoft applications on each server that will host PeopleSoft.
- **Shared Homes**: Installing all the PeopleSoft software and required infrastructure components such as Oracle JDK/JRE, Oracle WebLogic Server, Oracle Tuxedo, PeopleTools, and the PeopleSoft applications on a shared file system. With this option, the PS_HOME and PS_APP_HOME environment variables on all nodes point to a single shared file system directory location.

Whether shared or local homes are used, the homes, COBDIR, and report repository should be installed on file systems that offer a form of redundancy that protects against disk failure.

We recommend that you deploy PeopleSoft using the shared home paradigm onto the OCI File Storage Service (FSS), a fault-tolerant shared file system.

3.2.2 PeopleSoft Web and Application Tier HA

Figure 3 above shows the simplest high availability configuration for a PeopleSoft deployment. It has the following:

- A load balancer is used to distribute web traffic across the web servers. Dual switches are configured for redundancy.

- Two or more PeopleSoft Pure Internet Architecture (PIA) web servers are configured for workload distribution and redundancy. Each PIA web server connects to two or more application servers (described below). Should an application server become unavailable, its requests will be routed to an alternate application server.

Note that, in a standard configuration, PIA web server sessions accumulate state and are “sticky” to a specific server for the duration of their session. Left as-is, if a web server fails, users on that server will be automatically directed to a surviving web server but will have to re-authenticate and will lose their session state.

- A Coherence*Web cache server cluster is configured, to preserve session state if a PIA web server fails.

We placed a Coherence*Web cache server on each PIA web host to form a Coherence*Web cache cluster. With this configuration, users impacted by a failed web server and routed to another do not have to re-authenticate or lose work.

NOTE: Coherence*Web is a separately licensed component and is not included as part of a base PeopleSoft installation.

- Two or more PeopleSoft application servers are configured on separate hosts (VMs) to provide redundancy should an application server becomes unavailable. It is at this layer that the bulk of the business logic is executed. As there is no session state at this level, loss of an application server does not result in a need for user rework. A delay may be observed as the PIA web server routes the request to a remaining application server.

- Two or more PeopleSoft process schedulers are configured, shown here sharing the physical hardware used by the application servers. The first process scheduler that starts is designated “master”; the rest are slaves. The master assigns jobs to the slave process schedulers. If the master fails, one of the slaves will take over the role of master. The process schedulers connect to the database using role-based database services.

3.2.3 PeopleSoft Application Tier MAA

Maintaining an active replica of your mission-critical production system at a secondary site will significantly reduce downtime should the primary site become unavailable. This secondary site can also be used for standby-first platform maintenance, offloading queries via Active Data Guard, production level application patch validation, and other project work.

We will take the PeopleSoft highly available solution described in the previous section to a Maximum Availability Architecture by duplicating the local implementation in another data center, physically separated from the primary data center so that no local or regional disaster could be expected to impact both installations. This is shown in the following diagram.

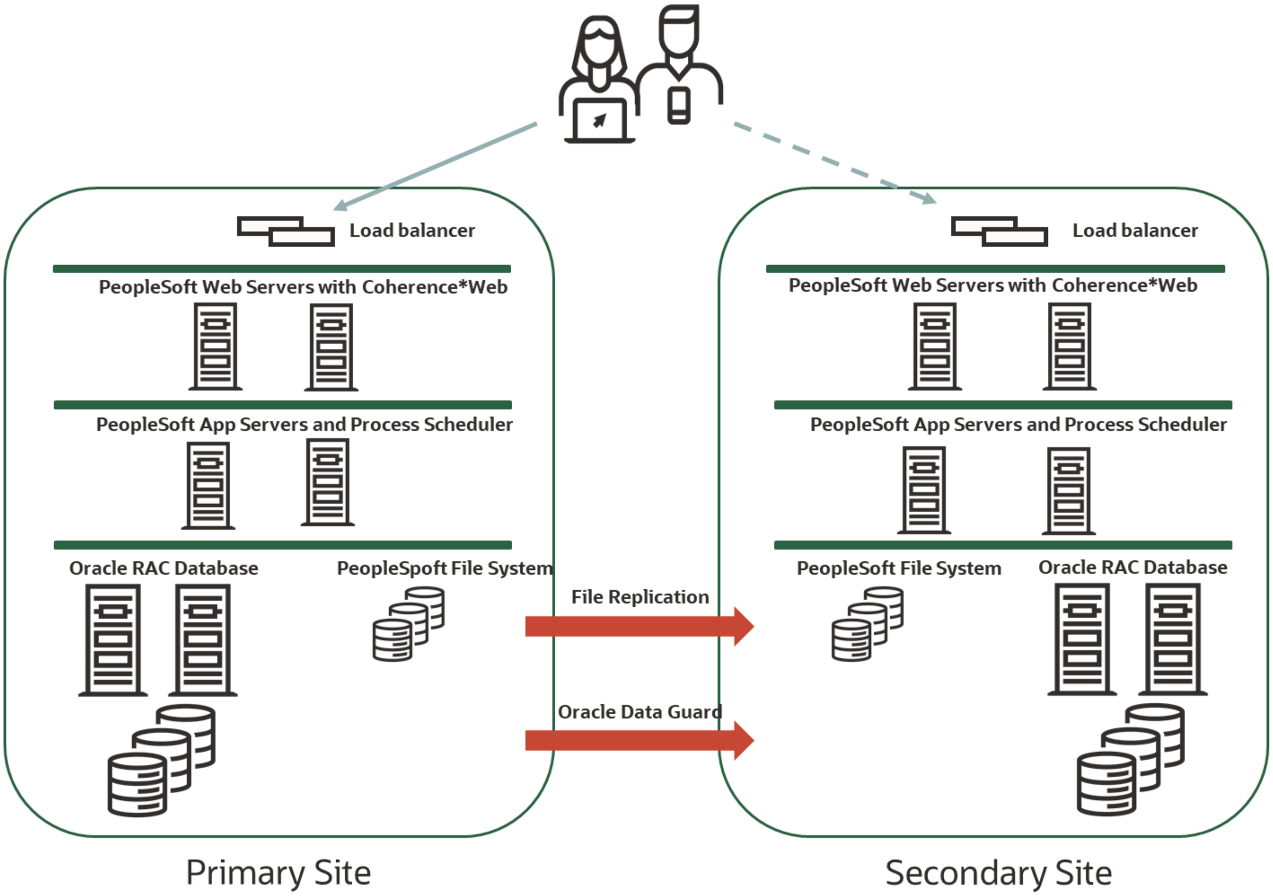
Load balancer
PeopleSoft Web Servers with Coherence*Web
PeopleSoft App Servers and Process Scheduler
Oracle RAC Database
PeopleSpoft File System
File Replication
Oracle Data Guard
Load balancer
PeopleSoft Web Servers with Coherence*Web
PeopleSoft App Servers and Process Scheduler
PeopleSoft File System
Oracle RAC Database
Primary Site
Secondary Site

Figure 4: PeopleSoft Maximum Availability Architecture

The data at the second site is kept in synch with the primary via appropriate replication mechanisms.

- For the database itself, Oracle's Active Data Guard ensures the standby database is kept in sync and transactionally consistent.
- For file system output generated during the operation of the application, rsync will be used to frequently replicate the output to another region. There will be a small gap to resolve by identifying missing file system components and determining the action to take for each.

3.3 Planned and Unplanned Outage Solutions

This section summarizes the outages that may occur in a PeopleSoft environment, and the Oracle solutions that are 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 any individual component.

3.3.1 Unplanned Outage Solutions

Table 1 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.

We recommend you test the basic scenarios below to ensure they are configured correctly in your environment, and to be confident you are ready to act if an emergency occurs.

OUTAGE TYPE	ORACLE SOLUTION	BENEFITS	RECOVERY TIME
Load balancer	Software load balancer, configuration replicated locally	Connections seamlessly migrate to surviving load balancer	No downtime.
PeopleSoft PIA Web Server node or component failure	Redundant Web Servers without Coherence*Web cache server cluster	Connections are redistributed to surviving nodes. Surviving nodes continue processing.	No downtime. Re-authentication and re-submission of work may be required.
	Redundant Web Servers with Coherence*Web cache server cluster	Connections are redistributed to surviving nodes, preserving session state. Surviving nodes continue processing.	No downtime and no re-authentication or re-submission of work
PeopleSoft Application Domain Server node or component failure	Redundant application domain servers PIA servers configured with active connections load balanced across application servers, resubmits the work to a surviving app server	Connections are redistributed to surviving nodes. Surviving nodes pick up the requests, no loss of context	No downtime
Database server or instance failure	Oracle RAC, Application Continuity, FAN events	Automatic recovery of work on failed instance – sessions transparently fail over, updates are resubmitted automatically	Seconds to minutes
Site failure	Data Guard, rsync	Full site failover with minimal to no loss of data	< 10 minutes after the decision is made, for database role transition, file system mount, 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
	Region-local Oracle object storage	Cloud-managed database recovery and disk-based backups	Minutes to hours
	Data Guard, rsync	Full site failover with minimal to no loss of data	< 10 minutes after the decision is made, for database role transition, file system mount, and PeopleSoft application startup
Human error	Data Guard with Flashback Database	Research on copy (standby)	Hours (research through data fix)
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
	Oracle Data Guard	Automatic validation and re-transmission of corrupted redo blocks	No downtime, transparent to application
	Oracle Data Guard Broker	Fast failover to an local standby database, or full site failover to DR site	Local standby: < 5 minutes after the decision is made, for database role transition, file system mount, and PeopleSoft application startup. Full site failover: < 10 minutes after the decision is made, for database role transition, file system mount, and PeopleSoft

			application startup

Table 1: Unplanned outages

Note that it may be possible to recover quickly from a fault at the primary site and resume operations there, which may be less disruptive to the overall operation than switching to the secondary site. Thus, in the table above, we mentioned making a decision to do the failover and the time it is expected to take to perform a scripted transition once the decision is made. If you decide to not require a human decision before a failover to a DR site, you will configure Fast-Start Failover in the database.

If Fast-Start Failover is configured and if the standby database apply lag is within the fast start failover lag limit, then the time to bring up the DR site will only add the fast-start failover timeout threshold to the overall time to transition to the standby.

Whether the action is taken automatically or not, the failover process should be fully scripted to ensure swift and accurate execution.

3.3.2 Planned Maintenance Solutions

Table 2 summarizes the planned maintenance activities that typically occur in a PeopleSoft environment, and the recommended technology solutions to keep downtime to a minimum.

MAINTENANCE ACTIVITY	SOLUTION	PEOPLESOFT OUTAGE
Mid-Tier operating system or hardware upgrade	Load balancing, redundant services across Web and Tuxedo application servers	No downtime, assuming Coherence*Web is running
PeopleSoft (application and PeopleTools)	PeopleSoft out-of-place patching	Minutes (no schema changes) to hours (schema changes required)
PeopleSoft application configuration change	PeopleSoft application rolling restart	No downtime
PeopleSoft upgrades	PeopleSoft out-of-place upgrades	Hours to days (schema changes will be required; time depends on database size)[1]
Database tier operating system patching or hardware maintenance	Oracle RAC rolling, Standby-First	No downtime
Oracle Database Release Update patching	Oracle RAC rolling, Standby-First	No downtime
Oracle Database upgrades	Data Guard transient logical rolling upgrade See: Reducing PeopleSoft	Seconds to minutes

[1] In practice, there are 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.

	Downtime Using a Local Standby Database.	
Oracle Grid and Clusterware upgrade and patches	Oracle RAC rolling, Standby-First	No downtime

Table 2: Planned downtime

4 PeopleSoft HA on Oracle Cloud Infrastructure

This section presents the decisions to make and information to gather to implement a highly available PeopleSoft deployment in Oracle's cloud infrastructure. The implementation would then be duplicated at a second site to provide maximum availability. MAA will be covered in our case study, later in this paper.

While the discussions below will explain several concepts for OCI and ExaDB-D, it is out of scope to provide a detailed discussion of either. Readers should be familiar with the following terms as they apply to OCI and ExaDB-D:

- OCI Tenancies
- Identity Management Cloud Service Providers
- OCI Users and Groups
- Compartments
- Compute Instances
- Database services: Database Cloud Service (DBCS) and Exadata Database on Dedicated Infrastructure (ExaDB-D)
- Object Storage
- Availability Domains
- Virtual Cloud Networks (VCN)
- Classless Inter-Domain Routing (CIDR) addresses
- Subnets
- Security Lists
- Regions

We assume you have an active Oracle Cloud account and access to your OCI tenancy.

The layout provided in this case study is simply an example. We cannot stress enough that attention to planning for your PeopleSoft deployment into Oracle Cloud, then testing your design and implementation decisions, are key to a successful OCI PeopleSoft MAA implementation. You may wish to try out a few deployment scenarios to see how they work so you can both familiarize yourself with how OCI works and learn if the scenarios you envision meet your requirements.

4.1 Availability domains, regions

Oracle's public cloud is delivered by networks of globally distributed cloud *regions* that provide secure, high-performance, local environments organized into separate, secure cloud realms. Organizations can build, run, and move all workloads and cloud applications on OCI while complying with regional data regulations. At the time of this publication, Oracle Cloud offers services from 41 public cloud regions in 22 countries.

An OCI region has one or more *Availability Domains* (ADs). An AD is one or more data centers. ADs are isolated from each other, fault tolerant, and do not share infrastructure such as power or cooling or the internal AD network. Because ADs do not share core infrastructure, a failure at one AD within a region is unlikely to impact the availability of other ADs within the same region.

ADs within the same region are connected to each other with a low latency, high bandwidth network. Traffic between ADs is encrypted. This configuration makes it possible to deploy a high-performing disaster recovery configuration that meets local regulatory requirements.

Note that when an AD is provisioned to a tenancy, a logical name is provided, such as PHX-AD-1. This name does not mean the assigned AD is the first data center provided to the region. Specifying a different AD in the same region for a disaster recovery configuration will create an environment in a different AD / data center, and will return a logical name with a different AD designation, such as PHX-AD-2.

4.2 Designing for a Secure Implementation

The components of your installation – the database, the middle tier software, the application software, the network configuration – should be installed in a manner that ensures the right people have the access required to manage each part, and only those people have that access. This is managed in OCI by defining *compartments* that hold these elements, and by limiting access to those compartments to specific groups, thus specific lists of users.

Compartments are virtual containers that hold resources. Specific administrators are assigned to groups; groups are then given permission to access or manage specific compartments. Thus, it will likely be appropriate for you to have separate groups – and separate compartments – for network, middle/application tier, and database resource management. You will also have separate sets of compartments for development and test resources, to better secure your production environment.

4.2.1 Compartments

The following table shows the compartment layout used in this case study. We have defined specific administrative groups for each type of compartment, to facilitate a robust and secure implementation of segregation of duties.

COMPARTMENT NAME	DESCRIPTION	GROUP
PSFT-APP-COMPARTMENT	Contains all PeopleSoft WebLogic PIA compute instances, Tuxedo application compute instances. and FSS shared file system	Application administrators
PSFT-EXADB-D-COMPARTMENT	Contains the Exadata Database on Dedicated Infrastructure (ExaDB-D) infrastructure and VM cluster for the PeopleSoft database	Database administrators
PSFT-NETWORK-COMPARTMENT	Contains the virtual cloud network (VCN), gateways (service, NAT, internet), dynamic routing gateways (DRG), subnets, security lists and load balancers (LBaaS)	Network administrators
PSFT-BACKUP-COMPARTMENT	Contains the object storage buckets for middle tier file system backups. May contain more.	Application administrators, optionally database administrators
PSFT-CLOUDMANAGER-COMPARTMENT	Contains the compute instance for the PeopleSoft Cloud Manager	Application and database administrators
Optional: PSFT-VMDB-COMPARTMENT	Contains the optional lightweight VMDB cluster, used to host the PeopleSoft database	Database administrators

Table 3: OCI compartments

The compartments in Table 3 that will hold our PeopleSoft deployment are:

- PSFT-APP-COMPARTMENT - On the middle tier:
  - Two Tuxedo application servers run the PeopleSoft HCM 9.2 business logic. These two application servers connect to the database using role-based FAN-enabled database services.
  - Two Pure Internet Architecture (PIA) web servers serve online users. Each PIA server is configured to connect to the Tuxedo application servers in a load-balanced fashion over JOLT. Should an application server fail or become unavailable, the PIA server can transparently connect the remaining application server and carry out its request.

  - A Coherence*Web cache server runs on each VM that hosts a PIA server, forming a cache server cluster for web session state resiliency.
  - Two process schedulers provide batch processing for scalability and resiliency.
  - File Storage Service (FSS) provides a shared file system for the application and PIA servers. These servers mount the shared file system as an NFS v3 file system.
- PSFT-EXADB-D-COMPARTMENT - The Exadata Database on Dedicated Infrastructure (ExaDB-D) provides a quarter rack hosting the Peoplesoft database.
  - The 19c RAC database, running on two domUs, is deployed on the ExaDB-D quarter rack. A dedicated pluggable database (PDB) hosts all the PeopleSoft schemas.
  - Note: Database backups are managed by the ExaDB-D backup automation tooling through a dedicated backup subnet to the region-local object storage.
- PSFT-NETWORK-COMPARTMENT – See Virtual Cloud Network and Network Components for design details for this compartment.

The other three compartments are for infrastructure or testing:

- PSFT-BACKUP-COMPARTMENT – backups of the middle tier (software and configuration).
- PSFT-CLOUDMANAGER-COMPARTMENT – The PeopleSoft Cloud Manager provides an OCI life-cycle and operational management interface for the PeopleSoft application. We configured this to operate in its own compartment.
- PSFT-VMDB-COMPARTMENT – This optional compartment can be used to establish a lower-cost initial familiarity testing environment, functional testing environments, smaller customer production installs.

4.2.2 Groups and Policies

A *policy* allows a *group* of *users* to work in certain ways with specific types of *resources* in a particular *compartment*. In our installation, the main three compartments above were established to contain specific resources with these groups in mind:

apps_admin –Users who can do the following in the PSFT-APP-COMPARTMENT:

- Create, manage, and terminate compute instances
- Create and drop block volumes for compute instances
- Create and drop file systems from the file system service (FSS) cloud service
- Back up and restore file systems from object storage in the PSFT-BACKUP-COMPARTMENT

exa_admin – Users who can do the following in the PSFT-EXA-COMPARTMENT:

- Create, manage, and terminate oracle homes and databases
- Configure Data Guard Association
- Configure and manage automatic backups
- Create user-initiated full backups
- Start / stop VM clusters on Exadata infrastructure
- Provision or terminate VM clusters on Exadata infrastructure
- Apply OS patches to VM clusters (domUs), apply grid and database home patches

network_admin – Users who can do the following in the PSFT-NETWORK-COMPARTMENT

- Provision, manage, and terminate virtual cloud networks (VCN)
- Provision, manage, and terminate gateways (Internet and NAT)
- Provision, manage, and terminate route tables
- Provision, manage, and terminate subnets
- Provision, manage, and terminate security lists
- Provision, manage, and terminate OCI load balancers (LBaaS)
- Provision, manage, and terminate dynamic routing gateways (DRG)
- Establish remote and local peering agreement polices and provision, manage and terminate remote and local peering connections

4.3 Application Tier File System Layout

We have file system data that is shared across middle and application tier servers and configuration data that is specific to each server. We will use the FSS shared file system to host the shared file system data and local storage for the server-specific configuration data.

4.3.1 Shared File Systems

The data that is shared across application and web tiers will be managed according to how frequently it changes, creating one FSS file system for the application and web tier software and another for the report repository, inbound/outbound interface files, and process scheduler job logs. As we will discuss in detail later, this design allows for a more flexible disaster recovery strategy.

- PS_HOME, PS_APP_HOME, and PS_CUST_HOME contain the application and web tier software , and will reside on one FSS file system at each region. The name of this FSS file system is <region>_PSFT_APP_INSTALL, where region is either IAD for Ashburn, or PHX for Phoenix.
- The PeopleSoft report repository, process scheduler job logs, and inbound/outbound interface files will reside in a second FSS file system at each region. The name of this FSS file system is <region>_PSFT_APP_INTERFACE, where region is either IAD for Ashburn, or PHX for Phoenix.

4.3.2 PS_CFG_HOME

The PeopleSoft configuration directory home (PS_CFG_HOME) must be held in a separate location for each node, to keep configuration and infrastructure log files separate, so that:

- Each node can be configured independently of other nodes, and
- Each node can only access its own configuration data.

The basic options for setting up PS_CFG_HOME so that each node has its own domain configuration and infrastructure log files are:

1. Create a directory structure on the local (non-shared) file system of each VM
2. Create a directory structure on a shared file system that includes the host name of each compute instance VM

Our source was configured with option 1, so we configured our OCI target the same. We will describe the process we followed to set these directories up for initial and for final migration later in this paper.

To use option 2, user psadm2 would create a directory structure in FSS for every middle tier compute instance, with separate subdirectories for configuration and for log files. Each compute instance would then be altered to point PS_CFG_HOME to its directory location.

4.4 Virtual Cloud Network (VCN) and Network Components

Oracle VCNs provide a flexible and powerful set of networking features based on *software defined networking* (SDN). SDN decouples network definition and control from the physical components, allowing the creation of data-driven virtual networks that can be deployed anywhere within the OCI infrastructure.

Each tenancy in OCI can have one or more VCNs, allowing customers to segregate environments according to security requirements – e.g., production in one VCN, disaster recovery in another, performance in a third, dev/test in others.

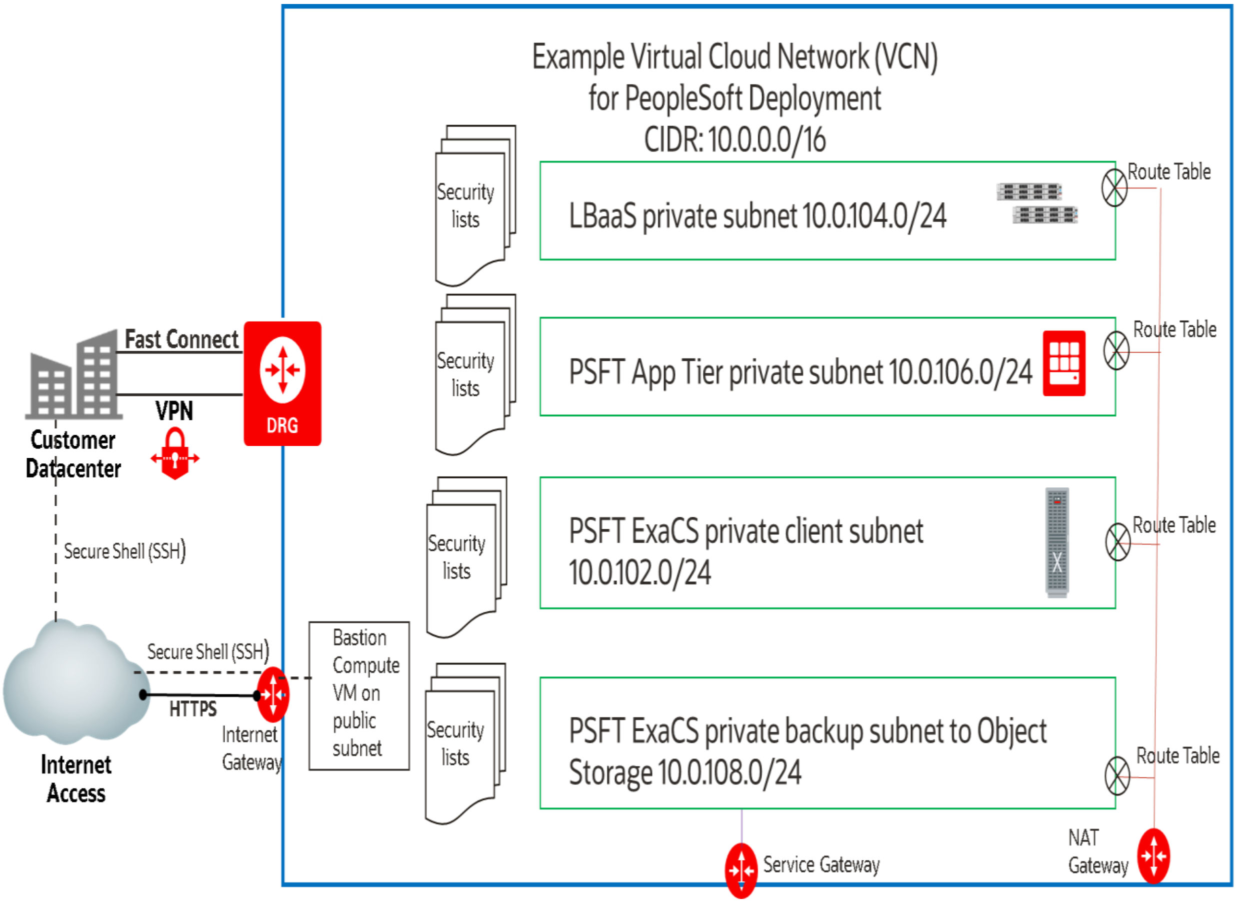
Example Virtual Cloud Network (VCN)
for PeopleSoft Deployment
CIDR: 10.0.0.0/16
Security lists
LBaaS private subnet 10.0.104.0/24
Route Table
Security lists
PSFT App Tier private subnet 10.0.106.0/24
Route Table
Security lists
PSFT ExaCS private client subnet 10.0.102.0/24
Route Table
Security lists
PSFT ExaCS private backup subnet to Object Storage 10.0.108.0/24
Route Table
Fast Connect
VPN
DRG
Customer Datacenter
Secure Shell (SSH)
Secure Shell (SSH)
HTTPS
Internet Gateway
Internet Access
Bastion Compute VM on public subnet
Service Gateway
NAT Gateway

Figure 5: PeopleSoft Virtual Cloud Network topology

Figure 5: PeopleSoft Virtual Cloud Network topology illustrates the network topology used in this case study, showing a basic production installation architected for high availability. The network topology has these characteristics:

- The VCN CIDR is 10.0.0.0/16. A network this size can support up to 65,534 IP addresses. It is divided into smaller subnets.
- VCNs are confined to a specific OCI region but can span availability domains within that region.
- Except for the bastion host subnet, all subnets are private.
- Each private subnet supports a specific network zone – e.g., the PSFT ExaDB-D Client subnet (10.0.102.0/24) only allows database client traffic. The ExaDB-D infrastructure is allocated IP addresses from this subnet.
- The PSFT App Tier private subnet (10.0.106.0/24) supports all web servers, application servers, and shared file system storage.

- The LBaaS load balancer subnet (10.0.104.0/24) only allows network traffic specific for the load balancer – e.g., ports 80 and 443 – to traverse this subnet.
- The ExaDB-D private backup subnet (10.0.108.0/24) isolates backup traffic from the database client network. The backups are routed via the service gateway to the region-local object storage.
- Each private subnet has its own route tables, to route network traffic to other networks.
- All private subnets share the NAT gateway, to allow specific network traffic into or out of each private subnet.
- Each subnet (public or private) has one or more security lists that allow specific network traffic from other subnets into (ingress) or out of (egress) the subnet.
- Customers access their OCI environments either through FastConnect or IPSec VPN via the dynamic route gateway (DRG).
- Access to the OCI VCN from the public internet is via a bastion host on a highly restricted public subnet. SSH access uses password-less RSA private keys that are not shared or passed to any of the hosts within the VCN.

This configuration will be duplicated at the disaster recovery (DR) site, described later in this document. The DR site will use a different VCN CIDR, since you cannot have the same CIDR or overlapping IP addresses within a tenancy.

4.4.1 Gateways

Gateways are network components that provide a pathway to other entities outside your VCN. The VCN described above shows each of these gateway types in use:

- Network Address Translation (NAT) Gateway: Passes network traffic to and from private subnets. Only one NAT gateway is allowed per VCN.
- Internet Gateway: Passes network traffic to and from the public internet to a public subnet
- Service Gateway: Passes traffic to and from region-local object storage service, yum repositories, and other services
- Dynamic Routing Gateway (DRG): Passes network traffic to and from various entities such as:
  - Other VCNs in different regions or OCI tenancies
  - FastConnect from customer's on-premises sites
  - IPSeC VPN from customer's on-premises sites
  - Other cloud providers

4.4.2 Subnets

Subnets divide the VCN into smaller IPv4 CIDR blocks that have specific requirements:

- Private versus public access
- Specific security requirements, documented in security lists
- Routing rules that limit the kinds of traffic allowed in and out

These requirements drive the decision to configure a subnet for each deployment tier:

- Database
- Middle / Application
- Load balancer
- DMZ (for public Internet access, not shown in diagram)
- Optional: Object storage, for database backups

OCI subnets can be created as regional or AD-specific. Because PeopleSoft is a very “chatty” application, requiring very low latency between the middle tier compute instances and the ExaDB-D hosting the database, we must configure AD- specific subnets. This keeps all the subnets within the same data center.

The following table lists the subnets we defined for each region, where “rgn” will be the region designation.

SUBNET NAME	CIDR	SUBNET ACCESS	SINGLE AD?	ROUTE TABLE	SECURITY LISTS
rgn-exadb-private-subnet	10.0.101.0/24	Private	Yes	db-private-RT	db-private-seclist
rgn-exadb-private-backup-subnet	10.0.108.0/24	Private	Yes	db-private-RT	exadb-backup-seclist
rgn-app-private-subnet	10.0.103.0/24	Private	Yes	app-private-RT	app-private-seclist
rgn-app-LBaaS-private-subnet	10.0.105.0/24	Private	Yes	app-lbaas-private-RT	app-LBaaS-seclist

Table 4: Subnet list

NOTE: Subnets within a VCN must not have overlapping IP addresses. There are many tools that can be found on the internet, such as ipcalc, that can be used to calculate subnets.

4.4.3 Route Tables

Route tables are used to route network traffic from a given subnet, to specific destination targets. Targets are typically gateways such as Internet, NAT, DRG, and service gateways. Each route table has one or more route rules that define where traffic can be routed.

Route tables can be shared by several subnets, but because subnets often have different routing and security requirements, we recommend you create route tables for each subnet. Doing this provides more flexibility and reduces security risks. Our exception in this environment: in the table below, the db-private-RT route table is shared by exadb_private_subnet-ad2 and exadb-backup-private-subnet-ad2 since both subnets will always have the same routing requirements.

The route tables we need in our environment are:

ROUTE TABLE NAME	ASSOCIATED SUBNETS
db-private-RT	exadb_private_subnet-ad2 exadb-backup-private-subnet-ad2
app-private-RT	ebs-app-private-subnet-ad2
app-lbaas-private-RT	app-LBaaS2-private-subnet-ad2

Table 5: List of route tables

Each route table will have route rules that define a network destination, the type of gateway to be used, and the specific target.

ROUTE TABLE NAME	DESTINATION	TARGET TYPE	TARGET
db-private-RT	0.0.0.0/0 All IAD Services in Oracle Service Network	NAT Gateway Service Gateway	maa-ngw Maa-lad-sgw
app-private-RT	0.0.0.0/0	NAT Gateway	maa-ngw

app-lbaas-private_RT	0.0.0.0/0	NAT Gateway	maa-ngw

Table 6: Route Rules for each route table

For now, the app-private-RT and app-lbaas-private-RT route tables have identical route rules. Rules will be added to the app-private-RT route table when we discuss DR later in this document.

4.4.4 Security Lists

We have made several references to security lists in the previous steps. Security lists are similar in principle to firewalls or iptables, which govern what network traffic is or is not allowed to traverse the network. Security lists are composed of rules and are attached to subnets, in essence providing each subnet its own firewall. Security lists have two types of rules:

- Ingress rules: What network traffic is allowed into the subnet
- Egress rules: What network traffic is allowed out of the subnet

OCI security uses the "least privileged access" paradigm, where the two types of rules specify only what network traffic is allowed. If there is no rule for a particular CIDR, IP protocol, and port, then access is denied. When the VCN is created, a highly restrictive default route table and a default security list are created. If a subnet is created and no security list is specified, the default security list is applied to that subnet.

Subnets can have one or more separate security lists, where each security list may have specific CIDR, IP protocol, and port requirements. For example, if SSH access to the ExaDB-D VM cluster nodes and compute instances will be through a bastion host, a separate security list is required to allow access (ingress) to port 22 from the subnet that the bastion host resides on.

In this case study we created a 1:1 correlation between subnets and security lists.

4.4.4.1 db-private-seclist

The following two tables show the ingress rules and egress rules for db-private-seclist, attached to the exadb-private-subnet-ad2 and <backup>.

RULE #	STATELESS	SOURCE CIDR	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE
1	NO	10.0.102.0/24	TCP	ALL	ALL
2	NO	10.0.103.0/24	TCP	ALL	1521

Table 7: Ingress rules for db-private-seclist

The ingress rules explained:

- Rule 1: Allows all TCP traffic to any port only on the CIDR block 10.0.101.0/24. Since this is the ExaDB-D client subnet, any of the ExaDB-D VM cluster nodes can access any other node from any port to any port, but only on this subnet.
- Rule 2: Allows network traffic from the CIDR block 10.0.103.0/24 (app-private-subnet-ad2), but only on destination port 1521, the TNS listener port.

RULE #	STATELESS	DESTINATION CIDR	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE
1	NO	0.0.0.0/0	TCP	ALL	ALL
2	NO	0.0.0.0/0	ICMP	ALL	ALL

Table 8: Egress rules for db-private-seclist

The egress rules explained:

- Rule 1: Allows any outbound TCP traffic to anywhere (0.0.0.0/0) and to any port. This is for allowing outbound TCP traffic anywhere within the VCN. Outbound traffic to the public internet is blocked, as there are no routes to the public internet.
- Rule 2: Similar to egress rule 1 but for ICMP pings.

4.4.4.2 app-private-seclist

The security list app-private-seclist, attached to the app-private-subnet-ad2 subnet, will have more rules than the database tier requires, as there are more ports and protocols required for the PeopleSoft middle tier and application servers. The following tables show the ingress and egress rules of the app-private-seclist security list.

RULE #	STATELESS	SOURCE CIDR	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE	COMMENTS
1	NO	10.0.103.0/24	TCP	ALL	22	SSH on port 22
2	NO	10.0.103.0/24	TCP	ALL	9000-9100	JOLT ports for PeopleSoft
3	NO	10.0.103.0/24	TCP	ALL	8088-8089	Coherence*Web cache server ports
4	NO	10.0.103.0/24	UDP	ALL	7574	Coherence*Web cluster port
5	NO	10.0.103.0/24	IMCP	(n/a)	7	Coherence*Web ICMP port
6	NO	10.0.103.0/24	TCP	ALL	111	FSS NFS port
7	NO	10.0.103.0/24	UDP	ALL	111	FSS NFS port
8	NO	10.0.103.0/24	TCP	ALL	2048-2050	FSS NFS mount ports
9	NO	10.0.104.0/24	TCP	ALL	80,443,8080	PIA web access from the load balancer subnet

Table 9: Ingress rules for the app-private-seclist security list

RULE #	STATELESS	DESTINATION	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE	COMMENTS
1	NO	0.0.0.0/0	TCP	ALL	ALL	Allows outbound TCP traffic to any location
2	NO	0.0.0.0/0	ICMP	(n/a)	ALL	Allows outbound PING traffic to any location
3	NO	0.0.0.0/0	UDP	ALL	ALL	Allows outbound UDP traffic to any location

Table 10: Egress rules for the app-private-seclist security list

4.4.5 Load Balancer as a Service (LBaaS)

OCI Load Balancer as a Service (LBaaS) is provisioned as a virtual resource just like any other service. LBaaS will distribute traffic across multiple web servers. We recommend LBaaS be provisioned onto its own subnet (private or public). It can have one or more front-end listeners, each with different purposes. One listener may be handling requests for a production application while another may be handling requests for a test system or a different application.

You have a number of decisions to make when deciding how you will configure LBaaS for your production environment:

1. LBaaS shape
   OCI offers several choices for the shape, or capacity, of your LBaaS. Select a minimum and a maximum number of megabits per second, where the maximum is greater than your environment requires at peak load.

2. LBaaS display name
   You can specify a display name or let one be generated. We recommend specifying a meaningful name describing its purpose.

3. Backends and backend sets
   In the parlance of OCI LBaaS, the load balancer distributes requests across *backends*. For PeopleSoft, the PIA web servers will be the target backends.

   A collection of one or more backends is a *backend set*. You can have one or more backend sets. This might be done to manage tiers of functionality delivered to different user communities – users of different production applications as well as users testing different instantiations of test environments.

   You will need to configure these items for your backend set:

   - Traffic distribution policy
     The traffic distribution policy will determine how new session requests will be distributed across all available backends belonging to the backend set. The choices are IP hash, least connections, and weighted round robin. We will use *weighted round robin*.

   - Session persistence
     Session persistence in the load balancer will enforce session "stickiness" to a given backend sever. PeopleSoft requires the option "Enable application cookie persistence" in the load balancer. You will need the cookie name from the PIA servers' weblogic.xml to complete this text box.

   - SSL enablement
     Allows you to require SSL for access from your load balancer to the VMs in your backend sets. Most companies choose to terminate SSL at the load balancer. If you choose to enable SSL on the backend servers, you will need to follow the appropriate PeopleSoft documentation for doing so, and you will need to use the same SSL CA signed certificate(s) or self-signed certificate(s) for both the front-end listener on the load balancer and each back-end web server

   - Health check
     The health check is required, and will be used to make sure traffic is only routed to a healthy backend server.

ATTRIBUTE	VALUE
Protocol	HTTP or HTTPS Since SSL is terminated at the load balancer, we will use HTTP.
Port	8080 HTTP port for all PIA web servers.
Interval in milliseconds	10000 Number of milliseconds between each check. 10000ms = 10 seconds
Timeout milliseconds	3000 Number of milliseconds that the check will wait before timing out. 3000 = 3 seconds.

Number of retries	3 Number of attempts to get a response from a given backend server.
Status code	200 The expected successful status code for HTTP GET calls.
URL path (URI)	/ Starting path, normally the root path of /.
Response Body RegEx	.* Regular expression that allows any response returned from the HTML page to be acceptable.

Table 11: Load balancer health check

4. Host Names
OCI LBaaS provides a facility to create one or more virtual host names that can be associated with one or more LBaaS listeners. If you choose to terminate SSL at the load balancer as we did, we recommend you create your listener's virtual hostnames with the same name you used to create the SSL signed certificate. This makes it easier to configure the listener correctly.

5. Certificate Name
It is *highly recommended* that SSL be configured, and for it to terminate on the load balancer listener. This requires an SSL bundle to be created and uploaded to the load balancer. You will give this SSL bundle a certificate name when you upload it.

The LBaaS listener can accept the following types of SSL certificates:

- Vendor signed CA certificates such as Verisign or GoDaddy
- Self-signed certificates using open-source tools such as OpenSSL to generate the x509 private key, the CA request, and the self-signed certificate. These should only be used for testing purposes. They are not trusted by web browsers, and will require the user to accept the certificate when initiating their session.

For further details on SSL certificates for LBaaS, please refer to: SSL Certificate Management in the Oracle Cloud Infrastructure documentation.

6. Load Balancer Listener

This table holds information you will need to create the load balancer listener:

LBAAS LISTENER FIELD	VALUE
Display name	A user chosen listener name that will show up in the OCI console.
Protocol	HTTPS Typically, this should be HTTPS for SSL based connections. HTTP is also allowed.
Port	443 If protocol is set to HTTPS, then the default value for port is 443.
Use SSL checkbox	Checked Next to the port value is a checkbox labelled "Use SSL". If port is set to 443, this checkbox will have a check mark.
Certificate Resource	Load Balancer Managed Certificates There are two options: "Load Balancer Managed Certificate" and "Certificate Service Managed Certificate". In this project, since a certificate will be uploaded, the "Load Balancer Managed

	Certificates" will be selected.
Certificate Name	PSFTHCM_SSL_LBaaS_CERT (for this project) This is the name you provided when the certificate bundle was uploaded.
Hostnames	psfthcm (for this project) If you created any virtual hostnames within the load balancer hostname section, add one or more if they are to be associated with this listener.
Backend Set	psfthcm_LBaaS_Prod_BS (for this project) This is a drop-down combo field providing a list of one or more backend sets. Select the appropriate backend set.
Idle Timeout in Seconds	60 This is the default for HTTPS.

5 Provisioning the Primary Infrastructure

Once the compartments, access requirements, and network topology are defined, these resources can be provisioned. We provisioned our resources using the OCI console UI. This paper will not go into details for provisioning all the resources but will provide examples. Note that the OCI console flows may change or differ from the steps we provide below.

In our configuration, the primary region for the PeopleSoft deployment is Ashburn, seen within the console as *us-ashburn-1* with abbreviation *IAD*. The complete PeopleSoft stack is deployed within a single AD, in our case AD-2, which holds:

- the quarter rack ExaDB-D,
- the application mid-tier servers, and
- the File Storage Service, which serves as a shared file system mounted by all application tier servers.

Provisioning all components within the same availability domain keeps network latency between components to a minimum.

Note: Oracle Cloud regions have abbreviations based on the international airport's *call sign* located in that area. For instance, the call sign for the Washington Dulles Airport in Ashburn, Virginia is IAD. For Phoenix, the call sign for the Sky Harbor International Airport is PHX. These abbreviations will be used throughout the remainder of this document.

This is a typical order of provisioning resources for the primary environment:

1. Users and groups
2. Compartments
3. Policies
4. Virtual Cloud Network (VCN)
5. Within the VCN:
    a. Internet gateway
    b. NAT gateway
    c. Service gateway
    d. Route tables for each subnet
    e. Subnets for each zone or tier
    f. Security lists for each subnet
6. Compute instances for application mid tiers
7. File Storage Service (FSS)
8. Load balancer (LBaaS)

Not all the resources need to be provisioned for every environment – for instance, a unit test environment could be much simpler. However, you must have groups and users, a compartment, the VCN, and at least one subnet.

Note: The examples in this section show how various components are configured, but do not provide specific screenshots from the OCI console UI. View current OCI documentation for specific provisioning tasks.

5.1 Create Users, Groups, Compartments, Policies

At this point, you have designed the compartments that will hold the major components of your implementation, and what privileges different kinds of users should be granted to manage and secure those components. Based on this design, use the OCI console to create:

- A group for each specific set of privileges that need to be granted
- A compartment for each resource to be provisioned and/or for the logical collection of resources to be managed the same way
- The policies detailing which actions members of each group can take against elements in each compartment
- The users who are members of each group.

5.2 Provisioning OCI Network Components

With the basic user/group/compartment structure in place, we can create our primary VCN. Once the VCN is created, other network components can be created.

5.2.1 STEP 1: Create the Primary Network

Start by creating the VCN. It defines the size of the network (the total number of IP addresses) based on the CIDR block specified at creation time.

From the OCI console main menu, under Networking, select Virtual Cloud Network. This will take you to the page for creating and listing VCNs. Choose Create, then provide the following:

- Name: The name of the VCN. We named ours: cloudmaa-vcn.
- Compartment: Select which compartment the VCN should be created in. In our case, psft-network-compartment.
- IPv4 CIDR block: This is where you define the size of the network. We used 10.0.0.0/16, which will allow up to 65,534 IPv4 addresses.
- Enable IPv6 CIDR block (checkbox): This is optional, and if checked will provide a default /56 IPv6 CIDR block. We left this unchecked.
- DNS Resolution (checkbox) – Use DNS hostnames in this VCN. This is checked by default, which we accepted.
- DNS Label: You can specify your own DNS label or let the system generate a label based on the VCN name you provided.

5.2.2 STEP 2: Create Gateways

All implementations will use private subnets and will require a NAT gateway. If your network will have traffic going to and from the public internet, then you will also need to create an internet gateway. If you need to reach region-local services such as object storage for backups, the YUM repository, etc., then you will need to create a service gateway.

For the initial configuration of our primary site, we created a NAT gateway, an Internet gateway, and a Service gateway.

5.2.2.1 To create a NAT gateway, from the main menu of the OCI console:

1. Select Networking, then Virtual Cloud Networks.
2. Click on the VCN that was created in Step 1 above. This will bring up the VCN detail page.
3. Click on the link “NAT Gateways”.
4. Click on the button “Create NAT Gateway”.

5. A new dialog window is displayed
    a. Provide a name for this NAT gateway. We named ours maa-ngwy
    b. Specify which compartment the NAT gateway should be created in using the combo-box selection, e.g, psft-network-compartment
    c. Choose the type of public IP address you want – Ephemeral or Reserved – for the NAT gateway, using the radio button. In either case, OCI will generate an IP address and assign it to this NAT gateway.
    d. Click the “Create NAT Gateway” button.

The NAT gateway is created.

5.2.2.2 To create an Internet gateway:

1. Clink on the link “Internet Gateways” from the Resources menu on the left of your screen.
2. Click on the “Create Internet Gateway” button.
3. A new dialog window is displayed.
    a. Provide a name for this Internet gateway. We named ours maa-igwy
    b. Specify which compartment the Internet gateway should be created in using the combo-box selection, e.g., psft-network-compartment
    c. Click the “Create Internet Gateway” button

The internet gateway is created.

5.2.2.3 To create a Service gateway:

1. Clink on the link “Service Gateways” from the Resources menu on the left of your screen.
2. Click on the “Create Service Gateway” button
3. A new dialog window is displayed
    a. Provide a name for this service gateway. These gateways are specific to a region. We named ours maa-iad-sgw, where "iad" designates the region hosting the gateway.
    b. Specify which compartment the service gateway should be created in using the combo-box selection, e.g., psft-network-compartment
    c. In the Services combo-box, select “All IAD Services in Oracle Services Network”. This allows access to both Object Storage Services and the region-local yum repository.
    d. Click the “Create Service Gateway” button.

The service gateway is created.

5.2.3 STEP 3: Create Route Tables

While it is easier to design your route tables and route rules after defining your subnets, it’s simpler to create the route tables and rules before creating the subnets.

Here, we give an example of creating the route table exadb-private-RT, including its local route rules, using the OCI console. From the main menu:

1. Select Networking, then Virtual Cloud Networks.
2. Click on the VCN that was created in Step 1 above. This will bring up the VCN detail page.
3. Clink on the link “Route Tables”
4. Click on the “Create Route Table” button

5. A new dialog window is displayed
    a. Provide a name for this route table. We named ours db-private-RT.
    b. Specify which compartment the route table should be created in using the combo-box selection – e.g., psft-network-compartment
    c. Click on the "+ Add another Route Rule" button. Additional fields will be displayed.
    d. For target type, use the combo-box to select NAT Gateway
    e. For destination CIDR block, we entered 0.0.0.0/0 to allow traffic to go anywhere within the VCN through the NAT gateway,
    f. For compartment, use the combo-box to select the compartment that the NAT gateway resides in. In our case: psft-network-compartment
    g. For Target NAT Gateway, using the combo-box, select your NAT Gateway. Ours is: maa-ngw
    h. (Optional) Enter a description.
    i. Click the "+ Add another Route Rule" button. Additional fields will be displayed
    j. For target type, use the combo-box to select Service Gateway
    k. For Destination Service, use the combo-box to select: All IAD Services in Oracle Service Network
    l. For compartment, use the combo-box to select the compartment that the service gateway resides in. In our case: psft-network-compartment
    m. For Target Service Gateway, use the combo-box to select your service gateway. Ours is maa-iad-sgw.
    n. Click the Create Route Table button

The OCI console will display a list of route tables.

Follow similar steps to create all your route tables and route rules.

5.2.4 STEP 4: Create Subnets

Here, we show the steps for creating the exadb-private-subnet-ad2 subnet, used for the ExaDB-D client network.

From the main menu:

1. Select Networking, then Virtual Cloud Networks.
2. Click on the VCN that was created in Step 1 above. This will bring up the VCN detail page.
3. Clink on the link "Subnets"
4. Click on the "Create Subnet" button
5. A new dialog window is displayed
    a. Provide a name for this subnet. We named ours exadb-private-subnet-ad2.
    b. Specify which compartment the subnet should be created in, using the combo-box selection – e.g., psft-network-compartment
    c. For Subnet Type, select either Regional or Availability Domain-specific (radio button option). We chose Availability Domain-specific.
    d. For CIDR Block, select the CIDR for the subnet you are configuring. We chose 10.0.101.0/24
    e. For Route Table, from the combo-box, select the route table to be used for this subnet. For this case, db-private-RT.

f. For Subnet Access, choose if the subnet is to be Private or Public from the radio option. For security, we chose Private for all subnets except the Bastion Host.

g. DNS Resolution checkbox by default is checked. If you have a different DNS resolution, uncheck this checkbox.

h. The DNS label can be system-generated based on the subnet name, or you can specify a specific label. We used the system-generated label.

i. DHCP (Dynamic Host Configuration Protocol) Options can be used to specify a number of different options. Since we assigned IP addresses, we left the default DHCP options.

j. Security Lists can be specified now or added later, after the subnet has been created. We entered them later.

k. Click the "Create Subnet" button.

Once the subnet has been created, it will show up in the table that lists the subnets.

Note:

- All subnets reside in the psft-network-compartment.
- We append "ad2" to the end of the subnet name to designate which availability domain the subnet resides in.
- To create the subnet dedicated to database backups to object storage, follow the steps above but choose a different CIDR block in 5-d. We used 10.0.108.0/24 for our case study. Use the same route table (db-private-RT) since it has the route rule to the service gateway.

5.2.5 Step 5: Create Security Lists

The OCI console provides an easy way to create security lists. These are the steps we followed to create the db-private-seclist from the OCI console main menu:

1. Select Networking, then Virtual Cloud Networks.
2. Click on the VCN that was created in Step 1 above. This will bring up the VCN detail page.
3. Clink on the "Security Lists" link
4. Click on the "Create Security List" button
5. A new dialog window is displayed

   a. Provide a name for the security list. We used db-private-seclist

   b. Specify which compartment the subnet should be created in using the combo-box selection – in our case, psft-network-compartment

6. Under Allow Rule for Ingress, click the "+ Another Ingress Rule" button to enter the first rule. New fields will appear.

   a. Leave the Stateless checkbox unchecked, as all ingress rules in our implementation will be stateful.

   b. For Source Type, select CIDR (default) from the combo-box

   c. For Source CIDR, enter the CIDR block for this rule. We entered 10.0.101.0/24

   d. For IP Protocol, select TCP from the combo-box

   e. For Source Port Range, either enter "All" (the default) or a specific port number or port range. We entered: All.

   f. For Destination Port Range, either enter "All" (the default) or a specific port number or port range. We entered: All.

   g. Optional: Enter a description if desired.

7. Click on "+ Another Ingress Rule" button to enter the second rule. New fields will appear.
   a. Leave the Stateless checkbox unchecked, as all ingress rules will be stateful.
   b. For Source Type, select CIDR (default) from the combo-box
   c. For Source CIDR, enter the CIDR block for this rule. We entered the CIDR of the app-private-subnet-ad2: 10.0.103.0/24
   d. For IP Protocol, select TCP from the combo-box
   e. For Source Port Range, either enter "All" (the default) or a specific port number or port range. We entered: All.
   f. For Destination Port Range, either enter "All" (the default) or a specific port number or port range. We entered:1521.
   g. Optional: Enter a description if desired.
8. Under Allow Rules for Egress, click the "+ Another Egress Rule" button. New fields will appear.
   a. Leave the Stateless checkbox unchecked, as all egress rules will be stateful.
   b. For Destination Type, select CIDR (default) from the combo-box
   c. For Destination CIDR, enter the CIDR block for this rule. We entered the CIDR 0.0.0.0/0. As noted earlier, since there are no routes to the public internet, this simply allows all outbound traffic within our VCN.
   d. For IP Protocol, select TCP from the combo-box
   e. For Source Port Range, either enter "All" (the default) or a specific port number or port range. We entered: All.
   f. For Destination Port Range, either enter "All" (the default) or a specific port number or port range. We entered: All.
   g. Optional: Enter a description if desired.
9. Click on the "Create Security List" button.

In the above steps for creating the db-private-seclist, two ingress rules and one egress rule were created. A second egress rule can be added to allow ICMP (ping) by

- Repeating steps 8-a through 8-c
- Changing the IP protocol to ICMP in the combo box in step 8-d
- Repeating steps 8-e through 8-G

Now that we have the security list db-private-seclist, it needs to be attached to the exadb-private-subnet-ad2 subnet. To do so, from the OCI console main menu:

1. Select Networking then Virtual Cloud Networks.
2. Click on the VCN that was created in Step 1 above, this will bring up the VCN detail page.
3. Clink on the "Subnets" link.
4. In the table listing the subnets, click on the exadb-private-subnet-ad2 link.
5. Click the "Add Security List" button. A new dialog window appears.
6. For Security List Compartment, select the compartment that the security list resides in. In our example: psft-network-compartment.
7. For Security list, select the security list from the combo-box. In our case, db-private-seclist.
8. Click the "Add Security List" button.

Once the security list is added, it takes effect immediately and replaces the default security list. Any changes made to a security list also take effect immediately.

5.3 Provisioning Exadata Database on Dedicated Infrastructure

Provisioning an Exadata Database on Dedicated Infrastructure is done in two steps:

1. Provision your target ExaDB-D infrastructure:
   Select the Exadata model and shape, and specify the availability domain. Complete and submit the provisioning request and wait until the infrastructure provisioning has completed. Note: You can scale the compute and storage capacity up after provisioning, if needed.
2. Provision the VM cluster:
   Once the Exadata infrastructure has been provisioned, you will provision the VM cluster onto the infrastructure. Select the Grid Infrastructure version, starter database version, OCPU count for the cluster, and ASM disk group storage properties. If you plan to store your backups on the region-local object storage, then you should not select local storage for backups. When de-selecting local backups, the ExaDB-D dialog will present additional fields for specifying the backup subnet and the compartment that subnet resides in.`

Refer to Creating an Exadata Cloud Infrastructure Instance for current steps for these tasks.

We provide some guidance and notes about our experiences using the tooling in the rest of this chapter.

5.3.1 Provision Target ExaDB-D Infrastructure

Use the OCI console to describe your target environment. Our Exadata model and shape were provided by an ExaDB-D X6-2 quarter rack with two compute nodes (domUs) and three storage cells. Our availability domain was AD-2, as noted in our design work.

Since we were using a bastion host as a "jump box" to get to compute or VM cluster domUs, we had to provide the bastion opc user's public key at the time of provisioning the ExaDB-D VM cluster. The provisioning process added the bastion opc user's public key to the opc user for each domU.

5.3.2 Provision the VM Cluster

Once the ExaDB-D was provisioned we could provision the Exadata VM cluster onto the infrastructure using the OCI console. We created our VM cluster based on the following:

FIELD NAME	VALUE
Exadata VM Cluster Name	IAD-Exa-VMCluster-1
Compartment	psft_exa_compartment
Host name prefix	iadexadb
Subnet for ExaDB-D client network	exadb_private_subnet-ad2
Subnet for ExaDB-D backups	exadb-backup_private_subnet-ad2
OCPU count	22
Grid Infrastructure version	19c RU 19 (19.19.0.0.0
Database version	19c RU 19 (19.19.0.0.0
Local storage for backup	No – Backups will be stored on region-local object storage
SPARSE ASM Disk Group	No for production, potentially yes for test databases

Table 12: Configuring Exadata VM Cluster

Within just a few hours, the Exadata VM Cluster was completely up, running, and accessible. The following components were fully configured.

- Two domU compute VM nodes
- Clusterware / Grid Infrastructure
- SCAN name with three IP addresses on the client subnet
- SCAN and grid VIPs with their respective listeners
- High redundancy ASM disk groups:

DISKGROUP NAME	REDUNDANCY	TOTAL SIZE MB	USEABLE MB
DATAC1	HIGH	161,206,272	48,055,638
RECOC1	HIGH	53,747,712	16,376,564

Table 13: ASM Disk Group layout

Note: You will notice other small disk groups, which were created automatically to support ACFS.

5.4 Provisioning Compute Instances

The compute instances are your application and middle tier servers. They will be used for PeopleSoft application and PIA web servers.

When provisioning compute instances, select the shape that best supports your workload. OCI provides several shapes to choose from as well as a choice between Intel or AMD based processors. Both Enterprise Unbreakable Linux and Microsoft Windows are supported. When provisioning the application tier compute nodes, specify the compartment (psft-app-compartment) to hold the compute instance resources and specify the subnet for the application tiers (app-private-subnet). The application servers will host:

- Tuxedo application server domain
- Tuxedo batch process server domain
- MicroFocus COBOL compiler and run-time facility

The PIA web servers can also be provisioned and placed into the same compartment and use the same subnet as the application servers. They will host:

- WebLogic Web servers to host the Peoplesoft PIA servers
- Coherence*Web cache servers (optional)

You can find more about provisioning compute instances in Working with Instances from Oracle Cloud Infrastructure Documentation.

We provisioned four compute instances for the PeopleSoft application and web tiers – two to host the application server and process scheduler, and two to host the PIA web server and Coherence*Web. The table below provides the characteristics of these compute instances.

HOST NAME	SHAPE TYPE	OCPU	MEMORY (GB)	BLOCK STORAGE SIZE (GB)	TIER	SUBNET	COMPONENTS
iad-psft-hcm-app01	VM.Standard2.4	4	60	128	Application	app-private-subnet-ad2	Tuxedo: application server, Process scheduler
iad-psft-	VM.Stand	4	60	128	Application	app-private-subnet-ad2	Tuxedo: application server,

hcm-app02	ard2.4						Process scheduler
iad-psft-hcm-web01	VM.Standard2.2	2	30	128	Web	app-private-subnet-ad2	WebLogic: Pure Internet Application server, Coherence*Web
iad-psft-hcm-web02	VM.Standard2.2	2	30	128	Web	app-private-subnet-ad2	WebLogic: Pure Internet Application Server, Coherence*Web

Table 14: Application and Web OCI Compute Instances at primary site

5.5 Provisioning File Storage Service

File Storage Service (FSS) will provide the shared file systems for all application and PIA servers. These servers will use NFS to mount the shared file systems. Provisioning FSS is quite simple from the OCI console. Ensure that FSS is in the same availability domain as the application and PIA servers.

Use the OCI console to provision each FSS file system:

1. From the main menu, select Storage, then File Systems under File Storage
2. Change the compartment to where you want the file system to be placed: psft-app-compartment
3. Click Create File System
4. Select (checkmark) File System for NFS
5. Under File System Information, click Edit Details
   a. Change the default name to a name of your choosing e.g., IAD_PSFT_APP_INSTALL or IAD_PSFT_APP_INTERFACE
   b. Change the availability domain to the availability domain on which the compute instances were provisioned e.g., US-ASHBURN-AD2
   c. Select the compartment where the file system should be placed: psft-app-compartment
   d. Select which encryption option you desire. We used the default: Oracle Managed Keys.
6. Under Export Information, click Edit Details
   a. Provide an export path: e.g., /export/psftapp or /export/psftinterface
   b. If required, check the checkbox for secure exports. See the information icon next to this option for more details.
7. Under Mount Target Information, click Edit Details
   a. Choose either “Select an existing mount target” or “Create a new mount target” radio button.
   b. Click on Enable Compartment Selection link to allow selecting the compartment that the VCN and subnets reside in.
   c. From Create in the Compartment drop-down combo box, select the compartment that the mount target will either be created in or already exists in.
   d. From the Virtual Cloud Network drop-down combo box, select the compartment that the VCN resides in
   e. If creating a new mount target, provide a name of your choosing.

  f. If using an existing mount target, select the compartment that the mount target was provisioned onto from the Subnet dropdown combo box.

8. Click Create

OCI will provide you with the required security ingress and egress rules to add to the appropriate security lists as well as the commands you need to issue on each application and PIA server. To obtain this information, after provisioning the file system:

1. Log in to the OCI console
2. From the main menu, select Storage then File Systems
3. Make sure you select the compartment that contains the file system
4. Select the name of the file system you provisioned
5. Click on the Export Target link
6. Click on “Mount Commands”. A window will display, providing the ingress and egress rules and the commands used to mount the file system. Use the “Copy” button to copy the mount commands for use in Configure OCI File Storage Service (FSS) for Shared Homes. Highlight and copy the ingress and egress rules for use in the next step.
7. Edit the security list associated with the subnet that will be used to mount FSS to add the ingress and egress rules.

6 Provisioning the Secondary Infrastructure

We will use different tooling to quickly build out the secondary site, extracting information from the OCI primary site just built out:

- The tool Terraform will greatly simplify the task of provisioning the network, by duplicating our rich network topology at the secondary site.
- The OCI console will provision the rest of the infrastructure.

6.1 Secondary OCI Region Subscription

You will need to subscribe to a second region geographically separate from your target OCI primary region. This secondary region should support similar infrastructure resources as the primary region – e.g., ExaDB-D of the same or similar shape and number, compute instances of similar shapes and numbers, File Storage Service (FSS) on both sides, etc.

To create your DR replica:

1. Log in to the OCI console for your tenancy.
2. Click on the main menu.
3. Click on Governance and Administration.
4. Under Account Management, click on Region Management.
5. You will see a list of all available regions. Regions that the tenancy is currently subscribed to will have a subscribed status of "Subscribed". Other regions will offer a Subscribe button.
6. Click on the Subscribe button for the region you chose as your secondary site. We used US West (Phoenix), region identifier: us-phoenix-1, for this project.

To switch between regions, use the region combo box on the top banner of the OCI console.

6.2 Secondary Region Network Resource Provisioning with Terraform

OCI provides a rapid method for provisioning resources called Terraform. You can use Terraform to duplicate all your OCI resources or a subset. We used Terraform to duplicate our network definition onto our secondary site, simplifying the task and eliminating a significant potential for errors.

This method has the following tasks:

- Execute a Terraform "export" to discover all or selected resources at the primary region within the tenancy
- Edit the Terraform files (.tf)
- Validate the Terraform "plan" against the secondary site region and resolve any errors
- Execute the Terraform "apply" to provision the resources at the secondary region site

Once there is a valid Terraform plan, Terraform's "apply" function will provision all resources defined in the .tf files, substantially reducing provisioning time.

Terraform can be executed using the Terraform command line interface or the OCI console's Terraform interface. Both approaches provide the same functionality. For installation and configuration of the command line interface version of Terraform, see: Terraform Provider.

NOTE: This technical brief will not provide a comprehensive discussion on using Terraform but will provide examples for guidance. See Appendix A: Working with Terraform for an example of discovering a network configuration on one environment and recreating it on another.

6.3 Completing the Secondary Region Deployment

Once the network is set up, the other components can be provisioned.

We used the OCI console to provision our compute instances, the File Storage Service (FSS), and ExaDB-D. Please see the relevant sections of Provisioning the Primary Infrastructure in this document for further guidance, as the steps are basically the same as for the OCI primary.

6.4 Cross-Region Remote VCN Peering

Remote VCN peering is the process of connecting two VCNs in different regions (but within the same **tenancy**). The peering allows the VCNs' resources to communicate securely using private IP addresses without routing the traffic over the internet or through your on-premises network.

The requirements for establishing remote VCN peering are:

- A Dynamic Routing Gateway (DRG) must be provisioned at each region.
- A Remote Peering Connection (RPC) must be attached to each DRG, to define the pairing between the VCNs at the two regions.
- An *explicit agreement* must be implemented as an IAM policy for each VCN agreeing to the peering relationship.
- Route table rules must be added at each VCN to route traffic. The DRG has a route table specific for remote VCN peering that can be updated.
- Security list ingress and egress rules must be added to subnets that will be allowed to have traffic between regions.

To implement remote VCN peering, follow the documentation Peering VCNs in different regions through a DRG.

When establishing remote VCN peering, update route tables at both regions to allow traffic to traverse. The following tables provide examples from our case study. The rows containing the target type of “Dynamic Routing Gateway” represent the rules that route traffic through that region’s DRG to the DRG at the other region.

Updated route tables in the Ashburn region:

db-private-RT

DESTINATION	TARGET TYPE	TARGET
0.0.0.0/0	NAT Gateway	maa-ngw
10.10.102.0/24	Dynamic Routing Gateway	cloudmaa-vcn-DRG
All IAD Services in Oracle Service Network	Service Gateway	Maa-Iad-sgw

app-private-RT

DESTINATION	TARGET TYPE	TARGET
0.0.0.0/0	NAT Gateway	maa-ngw
10.10.106.0/24	Dynamic Routing Gateway	cloudmaa-vcn-DRG

Updated route tables in the Phoenix region:

db-private-RT

DESTINATION	TARGET TYPE	TARGET
0.0.0.0/0	NAT Gateway	maa-ngw
10.0.101.0/24	Dynamic Routing Gateway	maacloud2-vcn-DRG
All PHX Services in Oracle Service Network	Service Gateway	Maa-phx-sgw

app-private-RT

DESTINATION	TARGET TYPE	TARGET
0.0.0.0/0	NAT Gateway	maa-ngw
10.0.103.0/24	Dynamic Routing Gateway	maacloud2-vcn-DRG

7 Database Migration

With the OCI infrastructure in place, we can instantiate the database and application. We start with the database.

The scenario presented here requires three databases:

- The original on-premises database (CDBHCM_sca6dp)
- The physical standby just created, destined to be the primary (CDBHCM_iad1dx, in the Ashburn region)
- A second physical standby, destined to be our cloud-based disaster recovery database (CDBHCM_phx5s, in the Phoenix region)

We will set up two standby databases in OCI so that we are fully protected when we switch production operations to Oracle's Cloud. The steps we followed to establish our databases in OCI are described here:

Set Up the Future Primary Database

Set Up the Future Secondary Database

Oracle Data Guard will keep these databases current with changes made in the on-premises production database until we are comfortable switching operations fully to OCI. At that point, Data Guard will keep the on-premises database and the OCI-based secondary database in sync with the new OCI-based primary. When ready, the on-premises database can be dropped from the configuration.

Constraints for this setup:

- All database homes must be on the same release update and patch level
- All databases must be registered into Oracle Clusterware
- All databases must be able to connect directly to each of the other databases involved in the Data Guard configuration, including the on-prem database

The software used in this project is listed in the table below. Both primary and standby sites used the same versions.

SOFTEWARE COMPONENT	VERSION
ExaDB-D domU OS version	22.1.10.0.0.230422
Oracle Grid Infrastructure	19.19.0.0.0
Oracle Database	19.19.0.0.0

Table 15: Software versions

7.1 Set Up the Future Primary Database

This section will describe how the PeopleSoft database was migrated to and configured on the future primary ExaDB-D. Covered in this section are:

- ZDM prerequisites and placeholder database creation
- ZDM configuration
- ZDM configuration test
- PeopleSoft database migration

7.1.1 ZDM Prerequisites

Move to Oracle Cloud Using Zero Downtime Migration documentation describes all prerequisites for Zero Downtime Migration. The most critical for a successful migration:

1. ZDM Host Server - A dedicated host or VM should be provisioned to host the ZDM installation, which includes a small Oracle Clusterware footprint, a MySQL database, and Fleet Patching and Provisioning (FPP). This server should have the latest Oracle OL7 Linux image installed. This VM shape can be small – 2 cores with 16GB of physical RAM are sufficient. This ZDM server is used for orchestrating all the database migration tasks at both source and target systems.
2. Network Connectivity – The type of network connectivity you have from on premises to resources in OCI will determine your ZDM migration method and data transfer options. ZDM allows for different network connectivity topologies including direct connections through FastConnect or IPSec VPN, use of SSH tunnels, proxy servers, and bastion hosts. It is *extremely* important to understand how your on-premises systems will access OCI resources and whether OCI resources need to access specific on-premises systems – and if so, by what network path. Note:
    a. The ZDM host sever must be able to access both on-premises source and OCI target systems
    b. For ONLINE migration methods using Oracle Data Guard, the source and target systems must be able to access each other.
3. Transparent Data Encryption (TDE) – OCI requires all databases be encrypted. If it is not possible to encrypt the data itself before the database is transferred into OCI, you can create a TDE keystore wallet at the source and the ZDM migration process will encrypt the data files at the target. A TDE wallet is required at the source for database versions 12.2 and higher, but this method can also be used for earlier database releases. “Setting Up the Transparent Data Encryption Keystore” in the ZDM documentation Move to Oracle Cloud Using Zero Downtime Migration has the steps for setting the TDE keystore.
4. Placeholder Database – you must create a placeholder database on the target OCI ExaDB-D prior to the ZDM migration. Its data structures will be removed by ZDM as part of the migration process, with the source database’s structures restored in its place. Its metadata will remain in place. Use the OCI console to create it, with these constraints:
    a. The database home must be at the same software version, release, and patch level as the primary
    b. The DB_NAME must be the same as on the primary database
    c. The DB_UNIQUE_NAME can be left blank or specified but it must be different from the primary
    d. The SYS password must be the same as on the primary as we are using Data Guard
    e. Do not create a PDB in this CDB.
    f. Do not configure automatic backups when provisioning this database
5. SSH access – ZDM requires SSH access to both the source and target systems. For the target, you will use the opc cloud user and passwordless SSH keys. For an on-premises source you will use the root user. You can configure passwordless SSH keys and use them without passphrases, or you can use the root user and password. Follow the ZDM documentation to set up SSH access and ensure the ZDM host server can access the source and target systems.

7.1.2 Configure ZDM for the Database Migration

Once your prerequisites are addressed and ZDM is installed, you can create a response file to configure your database migration. Copy the response file template found at $ZDM_HOME/rhp/zdm/template/zdm_template.rsp to your working directory on the ZDM host server, then edit it for your database migration. There are several parameters available to control the migration. We set our migration up to configure Data Guard and Data Guard Broker, and to minimize downtime:

ZDM PARAMETER	VALUE	COMMENTS
TGT_DB_UNIQUE_NAME	CDBHCM_iad1dx	Specifies the db_unique_name of the placeholder database.
MIGRATION_METHOD	ONLINE_PHYSICAL	The migration method used by ZDM that does not require the primary database to be down.
DATA_TRANSFER_MEDIUM	OSS	ZDM will use the cloud Object Storage Service to stage the backup of the database to then restore from it. Other transfer methods can be used such as DIRECT which can make use of RMAN RESTORE FROM SERVICE without having to stage the database in object storage. For DIRECT, other ZDM parameters are required, see ZDM documentation.
PLATFORM_TYPE	ExaCS[2]	The target system for the migration is Exadata Cloud Service.
TGT_RETAIN_DB_UNIQUE_NAME	TRUE	For Data Guard to ship logs back to the source, the target (TGT) database DB_UNIQUE_NAME is retained during the migration process.
TGT_SKIP_DATAPATCH	TRUE	Skip running datapatch on the target database.
SHUTDOWN_SRC	FALSE	Do not shut down the source database once migration is complete.
SRC_RMAN_CHANNELS	10	RMAN will allocate 10 channels on the source database for parallel backup of the database.
TGT_RMAN_CHANNELS	10	RMAN will allocate 10 channels on the target database for parallel restore of the database.
ZDM_USE_DG_BROKER	TRUE	ZDM will configure Data Guard Broker as part of the migration process.
HOST	https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/maacloud	Object storage service endpoint URL. Needed for OSS data transfer medium.
OPC_CONTAINER	ZDM_Backup	Object storage bucket name. Needed for OSS data transfer medium.

Table 16: ZDM Parameters for database migration

We were able to accept the default values for remaining parameters. Study the ZDM documentation to determine the parameters and settings that will be appropriate for your scenario.

7.1.3 Test the Configuration and ZDM Parameter File

To test your preparation steps and configuration file, run ZDM in evaluation mode. The “-eval” command line option instructs ZDM to perform prechecks only for all its migration process phases, then to stop. No changes are made to the systems. ZDM prechecks are performed on both source and target databases and, if DATA_TRANSFER_MEDIUM is set to OSS, on object storage.

This was our command to do prechecks of the migration process:

```
$ZDM_HOME/bin/zdmcli migrate database \
 -sourcedb CDBHCM_sca6dp \
 -sourcenode scaqan10dv0505.mycompany.com \
 -srcauth zdmauth \
```

[2] ZDM verbiage continues to reference ExaCS at the time of publication.

```
 -srcarg1 user:opc \
 -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk \
 -srcarg3 sudo_location:/usr/bin/sudo \
 -targetnode iadexadb-bw5wn1.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com \
 -backupuser <oci user name> \
 -rsp /home/zdmuser/zdm_CDBHCM_migration.rsp \
 -tgtauth zdmauth \
 -tgtarg1 user:opc \
 -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk \
 -tgtarg3 sudo_location:/usr/bin/sudo \
 -eval
```

All ZDM jobs are performed through a job scheduling mechanism and are executed asynchronously.

When a ZDM command is issued, ZDM will return the job ID, which you can use to check the job status. This command will show the status of job ID 5:

```
$ $ZDM_HOME/bin/zdmcli query job -jobid 5
```

The output will indicate which task is running, which tasks are pending, and whether the prechecks have succeeded or failed. As you query the job status, you can see the progression until the job has executed all required tasks.

The final output from our execution of our ZDM precheck was:

```
iad-zdm. ebsexadbprivate.ebscloudmaavcn.oraclevcn.com: Audit ID: 50
Job ID: 5
User: zdmuser
Client: iad-zdm
Job Type: "EVAL"

Scheduled job command: "zdmcli migrate database -sourcedb CDBHCM_sca6dp -sourcenode
scaqan10dv0505.mycompany.com -srcauth zdmauth -srcarg1 user:opc -srcarg2
identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -
targetnode iadexadb-bw5wn1.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com -backupuser <oci user
name> -rsp /home/zdmuser/zdm_CDBHCM_migration.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2
identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -
eval"

Scheduled job execution start time: 2022-07-26T20:26:01Z. Equivalent local time: 2022-07-26
20:26:01

Current status: SUCCEEDED

Result file path: "/u01/app/zdmbase/chkbase/scheduled/job-5-2022-07-26-20:26:21.log"
Metrics file path: "/u01/app/zdmbase/chkbase/scheduled/job-5-2022-07-26-20:26:21.json"
Job execution start time: 2022-07-26 20:26:21
Job execution end time: 2022-07-26 20:30:37
Job execution elapsed time: 4 minutes 16 seconds
ZDM_GET_SRC_INFO ........... PRECHECK_PASSED
ZDM_GET_TGT_INFO ........... PRECHECK_PASSED
ZDM_PRECHECKS_SRC .......... PRECHECK_PASSED
ZDM_PRECHECKS_TGT .......... PRECHECK_PASSED
ZDM_SETUP_SRC .............. PRECHECK_PASSED
ZDM_SETUP_TGT .............. PRECHECK_PASSED
```

```
ZDM_PREUSERACTIONS ......... PRECHECK_PASSED
ZDM_PREUSERACTIONS_TGT ..... PRECHECK_PASSED
ZDM_OBC_INST_SRC ........... PRECHECK_PASSED
ZDM_OBC_INST_TGT ........... PRECHECK_PASSED
ZDM_VALIDATE_SRC ........... PRECHECK_PASSED
ZDM_VALIDATE_TGT ........... PRECHECK_PASSED
ZDM_POSTUSERACTIONS ........ PRECHECK_PASSED
ZDM_POSTUSERACTIONS_TGT .... PRECHECK_PASSED
ZDM_CLEANUP_SRC ............ PRECHECK_PASSED
ZDM_CLEANUP_TGT ............ PRECHECK_PASSED
```

Before doing an actual ZDM migration, ensure that the EVALuation mode returns PRECHECK_PASSED for all precheck tasks. If any task is marked PRECHECK_FAILED, then consult the “Result” log file for errors and correct them. Run zdmcli with -eval as many times as necessary for all prechecks to pass.

7.1.4 Migrate the PeopleSoft Database

In this section, we use ZDM to migrate the database. By default, ZDM will migrate the database then switch over to it.

IMPORTANT: We DO NOT want ZDM to perform the switchover, so will use the **-stopafter** clause to stop execution after the phase **ZDM_CONFIGURE_DG_SRC** is complete.

The zdmcli command line differs only in the last line, where we specify -stopafter instead of -eval:

```
$ZDM_HOME/bin/zdmcli migrate database \
 -sourcedb CDBHCM_sca6dp \
 -sourcenode scaqan10dv0505.mycompany.com \
 -srcauth zdmauth \
 -srcarg1 user:opc \
 -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk \
 -srcarg3 sudo_location:/usr/bin/sudo \
 -targetnode iadexadb-bw5wn1.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com \
 -backupuser <oci user name> \
 -rsp /home/zdmuser/zdm_CDBHCM_migration.rsp \
 -tgtauth zdmauth \
 -tgtarg1 user:opc \
 -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk \
 -tgtarg3 sudo_location:/usr/bin/sudo \
 -stopafter ZDM_CONFIGURE_DG_SRC
```

As before, the zdmcli command will return a job ID that is used to monitor status. This example shows the final result for job ID 6:

```
$ $ZDM_HOME/bin/zdmcli query job -jobid 6
```

The final output after completing the phase ZDM_CONFIGURE_DB_SRC was:

```
iad-zdm. ebsexadbprivate.ebscloudmaavcn.oraclevcn.com: Audit ID: 74
Job ID: 6
User: zdmuser
Client: iad-zdm
Job Type: "MIGRATE"
Scheduled job command: "zdmcli migrate database -sourcedb CDBHCM_sca6dp -sourcenode
scaqan10dv0505.mycompany.com -srcauth zdmauth -srcarg1 user:opc -srcarg2
```

```
identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -
targetnode iadexadb-bw5wn1.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com -backupuser <oci user
name> -rsp /home/zdmuser/zdm_CDBHCM_migration.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2
identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -
pauseafter ZDM_CONFIGURE_DG_SRC"
Scheduled job execution start time: 2022-07-26T20:35:24Z. Equivalent local time: 2022-07-26
20:35:24
Current status: PAUSED
Current Phase: "ZDM_CONFIGURE_DG_SRC"
Result file path: "/u01/app/zdmbase/chkbase/scheduled/job-6-2022-07-26-20:35:51.log"
Metrics file path: "/u01/app/zdmbase/chkbase/scheduled/job-6-2022-07-26-20:35:51.json"
Job execution start time: 2022-07-26 20:35:51
Job execution end time: 2022-07-26 21:37:05
Job execution elapsed time: 1 hours 1 minutes 14 seconds
ZDM_GET_SRC_INFO ............... COMPLETED
ZDM_GET_TGT_INFO ............... COMPLETED
ZDM_PRECHECKS_SRC .............. COMPLETED
ZDM_PRECHECKS_TGT .............. COMPLETED
ZDM_SETUP_SRC .................. COMPLETED
ZDM_SETUP_TGT .................. COMPLETED
ZDM_PREUSERACTIONS ............. COMPLETED
ZDM_PREUSERACTIONS_TGT ......... COMPLETED
ZDM_OBC_INST_SRC ............... COMPLETED
ZDM_OBC_INST_TGT ............... COMPLETED
ZDM_VALIDATE_SRC ............... COMPLETED
ZDM_VALIDATE_TGT ............... COMPLETED
ZDM_BACKUP_FULL_SRC ............ COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC ..... COMPLETED
ZDM_DISCOVER_SRC ............... COMPLETED
ZDM_COPYFILES .................. COMPLETED
ZDM_PREPARE_TGT ................ COMPLETED
ZDM_SETUP_TDE_TGT .............. COMPLETED
ZDM_CLONE_TGT .................. COMPLETED
ZDM_FINALIZE_TGT ............... COMPLETED
ZDM_CONFIGURE_DG_SRC ........... COMPLETED
Stop After Phase: "ZDM_CONFIGURE_DG_SRC"
```

When this command has completed the ZDM_CONFIGURE_DG_SRC step, ZDM has copied the source database into OCI, set it up as a standby of the source, configured Data Guard Broker, and started redo apply. The new OCI standby is being synchronized with the source primary. ZDM also completed these tasks:

- Registered the migrated database into Clusterware
- Updated the OCI control plane metadata with updated information including any PDBs that are within the standby database
- Encrypted the data files of the standby database using TDE, as noted in ZDM Prerequisites. Note: the WALLET_TYPE in the view V$ENCRYPTION_WALLET will be set to AUTOLOGIN

7.1.5 Define Role-Based Databases Services for Future Primary

Add role-based database services that the PeopleSoft application will use when the OCI database is filling the PRIMARY role, for both online users and the process scheduler:

```
srvctl add service -db CDBHCM_iad1dx -pdb HR92U033 -service HR92U033_BATCH -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY,SNAPSHOT_STANDBY -failovermethod BASIC -
failovertype AUTO -failoverretry 10 -failoverdelay 3

srvctl add service -db CDBHCM_iad1dx -pdb HR92U033 -service HR92U033_ONLINE -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY,SNAPSHOT_STANDBY -failovermethod BASIC -
failovertype AUTO -failoverretry 10 -failoverdelay 3
```

7.2 Set Up the Future Secondary Database

After the first physical standby is established in OCI, we create a second one in another region. This second database is destined to be the database in our cloud-based disaster recovery environment. We will use Data Guard's *cascade standby* functionality – where this second standby receives its redo from the first standby, not directly from the on-premises primary – to reduce network traffic from the on-premises host site and to establish what will ultimately be the main redo propagation route.

At this time, there are constraints preventing us from using OCI tooling to establish and fully manage our future disaster recovery database:

- Data Guard Association cloud service cannot currently register an existing standby database relationship, therefore will not be able to manage our standby database configuration. Therefore, for example, Disaster Recovery Cloud Service (DRCS) cannot be used.
- Cloud-managed backups, whether user-initiated or automatic – can only be taken when one of the OCI databases is in the primary role.

Since both standby databases are established with an OCI-based placeholder database, the OCI control plane can manage patching and other lifecycle activity for each of them.

Covered in this section are:

- ➢ Create Placeholder Database
- ➢ Prepare for Database Restore
- ➢ Restore Database to Cascade Standby
- ➢ Configure Data Guard Broker for Cascade Standby

7.2.1 Create Placeholder Database

First, use the OCI console to create a new placeholder database in a different region (recommended) or in a different availability domain in the same region. Select Exadata On Oracle Plublic Cloud. Choose the ExaDB-D service that you want to deploy the database on. Follow these constraints:

- The database home must be at the same software version, release, and patch level as the source
- The DB_NAME must be the same as on the primary and the first standby database
- The DB_UNIQUE_NAME can be left blank or specified but must be different from both the on-premises primary and first physical standby database

- Do not configure automatic backups when provisioning this database
- Do not specify a PDB name when provisioning this database

Second - once the placeholder database is created – follow these steps to remove the datafiles, log files, controlfiles, and password file, leaving the software and directory structure in place. DO NOT delete the placeholder database using tooling (e.g. OCI or dbaascli).

1. Capture the cascade standby configuration data. Log in as the oracle OS user, source the environment for this database, and execute:

   ```
   $ srvctl config database -db <DB_UNIQUE_NAME>
   ```

   Save this configuration data - we will use it in several steps below.

2. Shut the placeholder database down:

   ```
   $ srvctl stop database -db <cascade standby placeholder database> -stopoption immediate
   ```

3. Log in as the grid OS user. Using asmcmd, empty the files in the directories under +DATAC1/<DB_UNIQUE_NAME>:
   a. DATAFILE
   b. ONLINELOG
   c. All <PDB GUID>/DATAFILE
   d. All control files under +DATAC1/ <DB_UNIQUE_NAME>/CONTROLFILE
   e. The password file as specified in the configuration data captured in step 1

4. Under +RECOC1/ <DB_UNIQUE_NAME>, remove the files in the directories ARCHIVELOG, AUTOBACKUP and FLASHBACKLOG.

7.2.2 Prepare for Database Restore

Now that the new oracle home is in place, we will configure it in preparation for the restore of the database. These tasks must be done:

- Adjust the tnsnames.ora file on each environment to be aware of each of the other databases. Verify communications between environments.
- Copy the password file from the first standby database
- Copy the TDE wallet from the first standby database
- Adjust the database parameters for the cascade standby database

7.2.2.1 Configure TNS for Cascade Standby

Data Guard Broker must be able to communicate with each database in the configuration no matter which instance it is connected to. ZDM did this configuration for the initial standby relationship. We need to add the cascade standby database into the configuration:

- The TNS connect string for the cascade standby database must be added to the tnsnames.ora files used by all RAC instances of the on-premises primary and the first standby databases, and
- the TNS connect strings for the on-premises primary and the first OCI standby databases must be added to the tnsnames.ora files used by all RAC instances of the cascade standby database:

These TNS entries must each use SCAN IP addresses, not the SCAN name. This is an example of a compliant TNS entry created by ZDM for our first standby database:

```
CDBHCM_iad1dx =
          (DESCRIPTION =
             (ADDRESS = (PROTOCOL = TCP) (HOST = <SCAN IPv4 address  1>) (PORT = 1521))
             (ADDRESS = (PROTOCOL = TCP) (HOST = <SCAN IPv4 address  2>) (PORT = 1521))
             (ADDRESS = (PROTOCOL = TCP) (HOST = <SCAN IPv4 address  3>)) (PORT = 1521))
            (CONNECT_DATA =
              (SERVER = DEDICATED)
              (SERVICE_NAME = CDBHCM_iad1dx)
              (FAILOVER_MODE =
                  (TYPE = select)
                  (METHOD = basic)
              )
              (UR=A)
             )
          )
```

To accomplish this task, you will be logging in to each database server as the oracle OS user, sourcing your environment, then changing directory to $TNS_ADMIN.

1. For each RAC instance of both the on-premises primary and the first OCI standby, edit the tnsnames.ora file and add the cascade standby database TNS connect string.

2. For each RAC instance of the OCI cascade standby, edit the tnsnames.ora file and add TNS connect strings for both the on-premises primary and the first OCI standby databases.

3. Test that you can ping the first standby database from the cascade standby, using the tnsping utility with the added connect string alias:

   ```
   $ tnsping CDBHCM_iad1dx
   ```

   This should return “OK” with latency time in milliseconds. If “OK” is not returned, check for errors and address accordingly.

4. Now, test the connection from each of the database servers that will host the cascade standby database to the first standby database (CDBHCM_iad1dx) using SQL*Plus. You will need the SYS password for the primary.

   ```
   $ sqlplus sys/<password>@CDBHCM_iad1dx as sysdba
   ```

   If the above returns with errors, correct the errors accordingly until you can connect successfully.

7.2.2.2 Copy Password File

Copy the password file from the first standby database.

1. Log in to one of the servers hosting your first standby database (CDBHCM_iad1dx) as the oracle OS user

2. Use srvctl to determine where the password file for this database is located, then copy it to /tmp:

   ```
   $ srvctl config database -db <first standby db name>
   ```

3. Look for the line that says, “Password file:” and record its location (ASM path) as this will be needed for the next step.

4. Become the grid OS user and use asmcmd to copy the password file to /tmp:

   ```
   $ asmcmd -p
   asmcmd> cd +DATAC1/<path from step 3>
   asmcmd> cp <password file name> /tmp/<password file name>
   ```

5. Transfer the password file to a temporary location on one of the cascade standby database servers using scp or by whatever means you use to transfer files within OCI.

6. Log in to the cascade standby database server on which the password file was placed, as the grid OS user. Copy the password file into ASM, using the location specified in the cascade standby configuration data above.

```
$ asmcmd -p --privilege sysdba
asmcmd> pwcopy –dbuniquename <cascade standby db unique name> /tmp/<password file name>
<cascade standby password file ASM path> -f
```

Example:

```
asmcmd> pwcopy –dbuniquename CDBHCM_phx5s   /tmp/<password file name>
+DATAC1/CDBHCM_phx5s/PASSWORD/orapwCDBHCM_phx5s -f
```

7. Ensure all TNS connect strings have been configured correctly by validating that each database can connect to all other databases. Fix any connection errors if any for the following connection attempts fails. Do not proceed until all connection attempts succeed.
   a. From the on-premises (primary) database:

   ```
   $ sqlplus sys/<password>@<first standby db> as sysdba
   $ sqlplus sys/<password>@<cascade standby db> as sysdba
   ```

   b. From the first physical standby:

   ```
   $ sqlplus sys/<password>@<on-prem primary> as sysdba
   $ sqlplus sys/<password>@<cascade standby db> as sysdba
   ```

   c. From the cascade physical standby:

   ```
   $ sqlplus sys/<password>@<on-prem primary> as sysdba
   $ sqlplus sys/<password>@<first standby db> as sysdba
   ```

7.2.2.3 Copy TDE Wallet

Copy the TDE wallet from the first standby database. On ExaDB-D, the location that cloud tooling uses to store the TDE wallets is on ACFS, which all database servers in the cluster share.

1. Log in to one of the database servers hosing the first standby database (CDBHCM_iad1dx) as the oracle OS user and change directory to the wallet root location. To determine the wallet root location:

```
$ sqlplus / as sysdba
SQL> show wallet_root
$ cd <wallet root location from “show wallet_root” above>
```

2. Under the directory given in step 1 – typically /var/opt/oracle/dbaas_acf/<DB_NAME>/wallet_root there is a directory named “tde”. Zip up the tde directory:

```
$ zip -r CDBHDM_tde_wallet.zip  tde
```

3. Transfer this ZIP file to one of the database servers that will host the cascade database (CDBHCM_phx5s) to a temporary location (e.g., /tmp) using scp or by whatever means you use to transfer files within OCI.

4. Log in to the database servers that will host the cascade database (CDBHCM_phx5s) and on which the zip file was place, as the oracle OS user and change directory to the wallet root location. The location should be the same as in step 3-a above since the DB_NAME is the same (CDBHCM):

```
$ cd /var/opt/oracle/dbaas_acf/<DB_NAME>/wallet_root
```

5. Move the existing tde directory to different name:

```
$ mv tde tde_<date>
```

6. Move the ZIP file containing the tde wallet created in step 2 (CDBHDM_tde_wallet.zip) to /var/opt/oracle/dbaas_acf/<DB_NAME>/wallet_root

```
Unzip the CDBHDM_tde_wallet.zip:
$ unzip CDBHDM_tde_wallet.zip
```

   This will create a new tde subdirectory with the wallet files from the first physical standby database.

7.2.2.4 Adjust Database Parameters for Cascade Standby

Finalize the configuration of the cascade standby database:

1. Log in to one of the database servers hosting the first standby database (CDBHCM_iad1dx) as the oracle OS user and source the environment

```
$ . ./CDBHCM.env
```

2. Create a pfile from the first standby database to be used as a reference for adjusting the parameters on the cascade standby database:

```
$ cd $ORACLE_HOME/dbs
$ sqlplus / as sysdba
SQL> create pfile=’tmp_CDBHCM_iad1dx_init.ora’ from spfile;
```

3. Now log in to one of the database servers that will host the cascade standby database (CDBHCM_phx5s) and source the environment:

```
$ . ./CDBHCM.env
```

4. Startup NOMOUNT only one instance:

```
$ sqlplus / as sysdba
SQL> startup nomount
```

5. Make the following adjustments to the database parameters for the cascade database, referencing the list of database parameters from step 2 above:

```
SQL> alter system set control_files=’’ sid=’*’ scope=spfile;
SQL> alter system set undo_tablespace=’<Refer to the parameter list from step 2>’
sid=’<ORACLE_SID for instance 1>’ scope=spfile;
SQL> alter system set undo_tablespace=’<Refer to the parameter list from step 2>’
sid=’<ORACLE_SID for instance 2>’ scope=spfile;
SQL> alter system set undo_tablespace=’<Refer to the parameter list from step 2>’
sid=’<ORACLE_SID for instance N>’ scope=spfile;
SQL> alter system set sga_target=’<Refer to the parameter list from step 2>’ sid=’*’
scope=spfile;
SQL> alter system set log_buffer=’<Refer to the parameter list from step 2>’ sid=’*’
scope=spfile;
```

6. Parameters specific to PeopleSoft:

```
SQL> alter system set “_gby_hash_aggregation_enabled”=false sid=’*’ scope=spfile;
SQL> alter system set “_ignore_desc_in_index”=true sid=’*’ scope=spfile;
SQL> alter system set “_unnest_subquery”=true sid=’*’ scope=spfile;
SQL> alter system set nls_length_semantics='CHAR' sid=’*’ scope=spfile;
```

NOTE:

DO NOT change the DB_NAME parameter.

DO NOT change the DB_UNIQUE_NAME parameter.

DO NOT change the WALLET_ROOT parameter.

7. Shut down and restart NOMOUNT the instance for the changes to take affect:

```
$ sqlplus / as sysdba
SQL> shutdown immediate
SQL> startup nomount
```

7.2.3 Restore Database to Cascade Standby

Restore the database onto the cascade standby footprint from the first physical standby database. We will use the RMAN command RESTORE FROM SERVICE to restore the control file and data files.

1. Continuing from the previous section, if the instance for the cascade standby is not started, start it in NOMOUNT:

```
$ sqlplus / as sysdba
$ SQL> startup nomount
```

2. Use RMAN to restore the control file and data files from the first standby to the cascade standby:

```
$ rman target / nocatalog
RMAN> restore standby controlfile from service ‘<first standby db name>’;
RMAN> alter database mount;
RMAN> restore database from service ‘<first standby db name>’ section size 8G;
RMAN> shutdown immediate;
RMAN> exit
```

Note: It may be necessary to adjust the number of RMAN channels for “device type disk” so as not to saturate the network. If a change is required, do so before executing the command “restore database from service”. This can be done with the following command, replacing N as appropriate:

```
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM <N>;
```

3. Restart all instances and mount the cascade standby database using srvctl:

```
$ srvctl start database -db <cascade standby db unique name> -startoption mount
```

4. Create and clear all online and standby log files using the following script.

```
$ sqlplus “/ as sysdba”
SQL> set pagesize 0 feedback off linesize 120 trimspool on
SQL> spool /tmp/clearlogs.sql
SQL> select distinct 'alter database clear logfile group '||group#||';' from v$logfile;
SQL> spool off
```

Inspect the generated clearlogs.sql script before executing it. It will cause the database instance to create and clear both online and standby logs files for all threads. Then execute the script:

```
SQL> @/tmp/clearlogs.sql
```

7.2.4 Configure Data Guard Broker for Cascade Standby

Data Guard Broker was configured between the on-premises primary and the first OCI standby database by ZDM. In this section, we will add the cascade standby to the configuration.

1. On the database server hosting the cascade standby, configure Data Guard Broker. Log in to one of the database servers hosting the cascade database as the oracle OS user and source the environment. Using SQL*Plus, configure Data Guard Broker:

```
$ sqlplus / as sysdba
SQL>  alter system set dg_broker_config_file1=’+DTAC1/<cascade standby db>/DG/dr1
<cascade standby db>.dat’ sid=’*’ scope=both;
SQL> alter system set dg_broker_config_file1=’+RECOC1/<cascade standby db>/DG/dr2
<cascade standby db>.dat’ sid=’*’ scope=both;
SQL> alter system set dg_broker_start=TRUE sid=’*’ scope=both;
```

2. Log in to either the primary or the first physical standby database, and source the environment. Add the new cascade standby to the existing Data Guard Broker configuration:

```
$ dgmgrl
DGMGRL>  connect sys/<password>
DGMGRL> show configuration
DGMGRL> add database '<cascade standby db>’
 as connect identifier is <cascade standby db>;
```

3. Add “redo routes” as follows:

```
DGMGRL> edit database <on-prem db> set property redoroutes='(LOCAL : <first standby db>
ASYNC)';
DGMGRL> edit database <first standby db> set property redoroutes='(LOCAL : <on-prem db>
ASYNC, <cascade standby db> ASYNC)(<on-prem db> : <cascade standby db> ASYNC)(<cascade
standby db> : <on-prem db> ASYNC)';
DGMGRL> edit database <cascade standby db> set property redoroutes='(LOCAL : <first
standby db> ASYNC)';
```

4. Enable the new cascade standby database

```
DGMGRL> enable database <cascade standby db>;
```

5. Once the cascade database is enabled, it will start to receive redo generated by the on-premises primary database via the first standby database. From within Data Guard Broker, show the configuration:

```
DGMGRL> show configuration lag
Configuration - zdm_psfthcm_dg
  Protection Mode: MaxPerformance
  Members:
  CDBHCM_sca6dp  - Primary database
    CDBHCM_iad1dx - Physical standby database
                      Transport Lag:      0 seconds (computed 0 seconds ago)
                      Apply Lag:          0 seconds (computed 1 second ago)
      CDBHCM_phx5s - Physical standby database (receiving current redo)
                        Transport Lag:      1 second (computed 1 second ago)
                        Apply Lag:          2 seconds (computed 1 second ago)
```

```
       Fast-Start Failover:  Disabled

       Configuration Status:
       SUCCESS   (status updated 47 seconds ago)
```

Notes: The cascade standby and the on-premises databases do not communicate directly with each other. When necessary, their redo is shipped via the first on-premises standby database:

- When the on-premises database is primary, redo is sent from the on-premises primary to / through the first standby, then to the cascade standby:
  - On-premises primary → OCI first standby → OCI cascade standby
- When the first standby is in the primary role, redo will be sent from that database directly to both the on-premises and the cascade standby databases:
  - On-premises primary ← OCI primary → OCI cascade standby
- If the cascade standby becomes primary in this configuration, redo will be sent from that database to / through the OCI first standby, then to the on-premises database:
  - On-premises standby ← OCI first standby ← OCI cascade primary

7.2.5 Define Role-Based Databases Services for Future Primary

Add role-based database services that the PeopleSoft application will use when the OCI secondary database is filling the PRIMARY role, for both online users and the process scheduler:

```
srvctl add service -db CDBHCM_phx5s -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_phx5s -pdb HR92U033 -service HR92U033_ONLINE -preferred
"CDBHCM1,CDBHCM2" -notification TRUE -role PRIMARY -failovermethod BASIC -failovertype AUTO -
failoverretry 10 -failoverdelay 3
```

7.3 Enable Flashback Database

It is an MAA best practice to enable Flashback Database for both PeopleSoft databases just configured in OCI. Flashback Database provides the following benefits:

- Reinstate a failed primary database without copying from the new primary or restoring from a backup.
- Flash the physical standby back for investigative work.

There are three steps to perform on each of the physical standby databases to enable Flashback Database:

1. Disable redo apply
2. Enable Flashback Database
3. Re-enable redo apply

7.3.1 Disable Redo Apply

We use the database CDBHCM_iad1dx in the following steps.

1. Log in to one of the ExaDB-D DB domUs hosting the standby database and become oracle,

Source the PeopleSoft database:

```
$ ./CDBHCM.env
```

Log in to the Data Guard Broker command-line interface:

```
dgmgrl
Connect SYS/<password>
```

Issue the following command to disable redo apply:

```
DGMGRL> edit database CDBHCM_iad1dx set state='apply-off';
Succeeded.
```

Exit Data Guard Broker.

7.3.2 Enable Flashback Database

Start a SQL*Plus session:

```
$ sqlplus / as sysdba
SQL>  alter database flashback on;

Database altered.
```

Exit SQL*Plus

7.3.3 Re-Enable Redo Apply

Log in to the Data Guard Broker command-line interface:

```
dgmgrl
Connect SYS/<password>
```

Issue the following command to disable redo apply:

```
DGMGRL> edit database CDBHCM_iad1dx set state='apply-on';
Succeeded.
```

8 Application Migration

With the database migrated and protected, we turn our focus to the application and web tier. We will set the application and web tiers up at both the future primary and the future standby sites.

There are three options for this task:

1. Freshly install and configure the application and web tier code using the PeopleSoft Cloud Manager
2. "Lift and shift" the application and web tier code from on-premises to OCI using the PeopleSoft Cloud Manager, including configuration of the application and web tiers
3. Manually migrate the application and web tiers (code and configuration) to pre-provisioned OCI compute instances

Options 1 and 2 are described briefly in Appendix B – Using PeopleSoft Cloud Manager to Provision Middle Tiers.

Option 3 does not require the PeopleSoft Cloud Manager, and is the process described here. We first provision compute instances in OCI, then we copy the PeopleSoft application and web tiers there from the on-premises environment.

We will split the install into three major phases:

- PeopleSoft Application and Web Tier Setup, where the bulk of the application migration and setup are completed. These steps include:
  - Provision the compute instances
  - Lift and shift the PeopleSoft software
  - Configure PeopleSoft for the new environment
  - Do a full-stack test
- Add Coherence*Web to the configuration:
  - Provision and configure the OCI load balancer
  - Install and configure Coherence*Web
- Housekeeping
  - Convert the database back to physical standby
  - Back up the OCI middle tier

Once the above tasks have been completed for the future primary site, the steps in Provision and Configure Cascade Standby Middle Tiers are used to set the PeopleSoft application and web tiers up at the future standby.

The software used in this project is listed in the table below. The primary and both standby sites used the same versions.

SOFTEWARE COMPONENT	VERSION
Oracle Database Client	19.3.0.0.0 (64bit)
PeopleSoft PeopleTools	8.57.11
PeopleSoft HRMS	9.2 U0033
Oracle Tuxedo	12.2.2.0.0
MicroFocus COBOL Server Express	MicroFocus COBOL Server Express 5.1 WP14

Oracle Java JDK/JRE	1.8.0_221
Oracle WebLogic	12.2.1.3
Oracle Coherence*Web	12.2.1.3

Table 17: Software versions

8.1 PeopleSoft Application and Web Tier Setup

We lay the foundation for our middle tier architecture in this section.

8.1.1 Create OCI Compute Instances

The configuration of our middle tier servers was simple and standard, with only the sizes of the boot, root, and swap file systems needing adjustment. At the time we provisioned ours, the default size of the boot volume was 46.6GB. This default size contains the basic required Linux file systems, including:

- A /boot file system (200MB)
- A root (/) file system (39GB)
- A swap volume (8GB)

For both the application and web tier servers, we needed to increase the boot file system to 128GB, the root file system to 100GB, and the total swap size to 16GB.

Increase the size of the boot volume when you create the compute instance with the OCI console. Under the Compute Instance section, click on Create Instance then enter your provisioning settings. Our settings, common across all four compute instances, were:

FIELD NAME	VALUE
Compartment	psft_app_compartment
Availability domain	AD-2
VM type	Intel
Image version	Oracle Enterprise Linux 7.9
Subnet	app_private_subnet
Boot Volume size (customized)	128 GB

Table 18: Common compute instance characteristics

Click on Create once the Create Instance form is filled out. The provisioning process will create the compute instances.

When that is complete, follow the MOS documentation to increase the root partition and root file system sizes, then add an 8GB swap partition.

- Use My Oracle Support document 2445549.1: How to Create a Linux instance with Custom Boot Volume and Extend the Root Partition in OCI to increase the root partition then the root file system size by 61GB, Note that the process OCI follows to provision the larger boot volume is to create a 39GB root partition then attach a *paravirtualized* block volume for the requested increase.
- Use My Oracle Support document 2475325.1: How to Increase Swap Memory on Linux OCI Instances to add an 8GB swap partition, resulting in a total of 16GB swap space.

8.1.2 Set Up OS Group and User

On all Linux middle tier compute instances, create the OS group and user per the steps in the following table.

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS
1	All compute instances, root	Create the oinstall group: `# groupadd -g 1001 oinstall`
2	All compute instances, root	Create the OS user to be used by the PeopleSoft installation. `# useradd --uid 1005 -g oinstall psadm2`
3	All compute instances, root	Modify the user to set up its home directory and default shell. `# usermod -d /home/psadm2 -g oinstall -s /bin/bash psadm2`

Table 19: Create OS group and User

8.1.3 Configure OCI File Storage Service (FSS) for Shared Homes

Mount your FSS file systems on each middle tier server, using the IP address for your file system, captured in Provisioning File Storage Service. We used these steps to mount our FSS file systems:

1. Log in as root, then create the /u01 and /u02 directories

```
# mkdir /u01
# mkdir /u02
```

2. Add entries to /etc/fstab:

```
<FSS-IP-Address>:/export/psftapp  /u01        nfs
rw,rsize=131072,wsize=131072,bg,hard,timeo=600,nfsvers=3 0 0
<FSS-IP-Address>:/export/psftinterface  /u02    nfs
rw,rsize=131072,wsize=131072,bg,hard,timeo=600,nfsvers=3 0 0
```

3. Mount the file systems:

```
# mount /u01
# df -h /u01
Filesystem                    Size  Used Avail Use% Mounted on
10.0.103.224:/export/psftapp  8.0E  0G   8.0E   0% /u01
# mount /u02
# df -h /u02
Filesystem                          Size  Used Avail Use% Mounted on
10.0.103.224:/export/psftinterface  8.0E   11M  8.0E   1% /u02
```

8.1.4 PeopleSoft Software Directory Structure

Except for MicroFocus COBOL, all the software components will be placed in a shared directory on FSS: /u01/app/psft/pt

PeopleSoft requires several environment variables be mapped to this directory structure, as shown in the following table. We have added one environment variable, for custom scripts needed to automate startup, shutdown, and switchover activities.

ENVIRONMENT VARIABLE	SET TO	PURPOSE
BASE_DIR	/u01/app/psft	Used as the base for all PeopleSoft software install
PS_HOME	/u01/app/psft/pt/ps_home8.57.11	Location of PeooleTools
PS_APP_HOME	/u01/app/psft/pt/hcm_app_home	Location of the PeopleSoft HCM application
PS_CUST_HOME	/u01/app/psft/pt/hcm_cust_home	Location of customizations of the PeopleSoft application
PS_CFG_HOME	/peoplesoft/local/ps_config	Location of node specific configuration and log files for PeopleSoft components such as application and process server domains under Tuxedo, and WebLogic log files.
PS_FILEDIR	/u01/app/psft/pt/ps_home8.57.11/file	Location of integration and interface files such as XML or other file types.
ORACLE_HOME	/u01/app/psft/pt/oracle-client/19.3.0.0	Location of the Oracle client software installation
TNS_ADMIN	/u01/app/psft/pt/oracle-client/19.3.0.0/network/admin	Location of the database client tnsnames.ora for connecting to the PeopleSoft database.
JAVA_HOME	/u01/app/psft/pt/jdk1.8.0_221	Location of the Java install and Java run-time environment
TUXDIR	/u01/app/psft/pt/bea/tuxedo/tuxedo12.2.2.0.0	Location of the Oracle Tuxedo install.
WLS_HOME	/u01/app/psft/pt/bea/wlserver	Location of the WebLogic server
COBDIR	/opt/MFCobol	The installation directory for where MicroFocus (or Visual) Cobol is installed.
SCRIPT_DIR	/u01/app/psft/pt/custom_admin_scripts	Location of administrative scripts created to manage the processes in this paper

Table 20: PeopleSoft application server environment variable mapping

8.1.4.1 PeopleSoft Directories for Multi-Node Deployments

When designing our implementation, we decided to create local, non-shared file system directories on each middle tier VM to hold the PeopleSoft configuration and infrastructure log files. In each case, the PS_CFG_HOME was set to the local directory /peoplesoft/local/ps_config.

Everything else is shared and will be copied over from the on-premises system. To do this, we created the head of the installation directory path for the PeopleSoft software install on shared disk, then created the child psft_reports directory. The other child directories will be created when we copy over the contents. We will refresh the contents of all child directories when we do a final switchover to this environment from the on-premises systems.

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS
1	One compute instance, root	Move to the /u01 directory. `# cd /u01`
2	One compute instance, root	Create the directories: `# mkdir -p app/psft/pt`

3	One compute instance, root	Create the directory for the report repository. For this project: `# mkdir -p /u01/app/psft/pt/psft_reports/out`

4	One compute instance, root	Change ownership of the psft directories.
		`# cd app`
		`# chown -R psadm2:oinstall psft`

Table 21: Create directories on shared file system

8.1.5 Host Environment Setup on OCI Compute Instances

Before configuring the application server, process scheduler, and PIA web server domains, you need to create an environment file for the user psadm2. While most of the values will be the same on each environment, the ORACLE_HOSTNAME will need to be set correctly for each server.

Note we have added a directory to hold new administrative scripts required by this project, and have created an export directive for the new directory as well as adding it to our path. We assume these scripts are in place and in the path when we describe actions later in this document.

The following is the environment file (psft.env) from one of our compute instances:

```
export ORACLE_HOSTNAME=iad-psft-hcm-app01
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=$PS_HOME/file
export ORACLE_BASE=/u01/app/psft
export ORACLE_HOME=/u01/app/psft/pt/oracle-client/19.3.0.0
export COBDIR=/opt/MFCobol
export CLASSPATH=$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

# Add directory for custom scripts
export SCRIPT_DIR=$BASE_DIR/pt/custom_admin_scripts
export PATH=$PATH:$SCRIPT_DIR

# You must be in PS_HOME to execute the psconfig.sh script
cd $PS_HOME
./psconfig.sh
cd
```

Place this environment script in psadm2's home directory on each compute instance, making sure the ORACLE_HOSTNAME is set to that compute instance's host name.

Add execute permission to the file. As the psadm2 user:

```
$ chmod u+x psft.env
```

Then source the environment with the following command:

```
$ . ./psft.env
```

Modify psadm2's .bash_profile script to call psft.env, to set up the environment automatically. This best practice is a requirement if you import the middle tiers into the PeopleSoft Cloud Manager.

Now that all OCI infrastructure and setup components are in place, the PeopleSoft application can be migrated to OCI.

8.1.6 Install MicroFocus COBOL

The MicroFocus COBOL (Visual COBOL for later released versions) is a separately licensed software package that contains the COBOL compiler and run-time environment plus the run-time license manager.

All compute instances that host the process scheduler (PRCS) must have the MicroFocus COBOL compiler, run-time environment, and license manager installed. It is best practice to install them on local file system. These components are owned by the root user, and are often installed in either /usr/local/microfocus or /opt/microfocus. The instructions for installing these components are found in the MicroFocus COBOL README file.

During the installation of the License Manager (mflman) and its database, answer 'Y' when prompted "Should the License Manager be started on reboot" to ensure it is running every time the server is booted. The MicroFocus License Manager must be running for PeopleSoft COBOL programs to be compiled, linked, and allowed to run under the process scheduler.

8.2 Copy the Application Software

If your source application / web tiers are running on the Linux operating system with a current version of PeopleTools, you will be able to manually migrate – "lift and shift" – the application and web tiers to the OCI environment. We followed this method, as this project is using PeopleTools 8.57.11 and Oracle Enterprise Linux version 7 (OEL 7) with the latest update.

If your source middle tier nodes are running a non-Linux operating system, you will need to install the PeopleSoft application and web tiers on your OCI compute instances, using the psft-dpk-setup.sh tooling. You will need to install Puppet for orchestration. Please see the documentation PeopleSoft PeopleTools 8.58 Deployment Packages Installation for these instructions.

8.2.1 Manually Lift PeopleSoft Application and Web Tiers

"Lifting" the PeopleSoft application and web tier software from the source system is just a matter of packaging up the PeopleSoft software install on the source system. The on-premises PeopleSoft environment can be up and running during this process. Log in to one of your source application / web tier servers as the application owner (psadm2) and follow these steps.

1. Zip up the following locations:
    a. PS_HOME
    b. PS_APP_HOME
    c. PS_CUST_HOME
    d. JAVA_HOME
    e. the BEA home directory

f. ORACLE_HOME
g. TNS_ADMIN

```
$ zip -r ps_home.zip $PS_HOME
$ zip -r ps_app_home.zip $PS_APP_HOME
$ zip -r ps_cust_home.zip $PS_CUST_HOME
$ zip -r ps_jdk.zip $JAVA_HOME
$ zip -r bea.zip <BEA directory location/bea>
$ zip -r oracle_home.zip $ORACLE_HOME
$ zip -r tns_admin.zip $TNS_ADMIN
```

TIP: If several of these locations are subdirectories under a main directory e.g., /u01/app/psft/pt, you can just zip up the main directory:

```
$ zip -r  pt.zip  /<full-path>/pt
```

If you wish to capture the Tuxedo application and process scheduler domain configurations, also zip up the PS_CFG_HOME/appserv directory and PS_CFG_HOME/peoplesoft.properties file. Do not zip up the PS_CFG_HOME/webserv directory as this will be rebuilt on the OCI compute instances.

2. Upload (copy) all ZIP files to the shared directory created on one of the OCI compute instances. Note you will perform the remote copy on the OCI environment as the opc user. It may be necessary to allow write privileges for the opc user to write to the above directory. This can be removed once the copy is complete.

   From the source or on-premises system, you can use the scp command to copy the ZIP files if you have several:

```
$ scp -I <path to key file> *.zip opc@<IP address to iad-psft-hcm-
app01>:/u01/app/psft/pt/.
```

   If you created one large ZIP file, then your command may be similar to:

```
$ scp -I <path to key file> pt.zip opc@iad-psft-hcm-app01:/u01/app/psft/.
```

3. Change ownership of the ZIP files to psadm2 on the OCI compute instance:

```
$ ssh -I <path to key file> opc@iad-psft-hcm-app01
$ sudo su – root
# cd /u01/app/psft/pt
# chown psadm2:oinstall *.zip
```

8.2.2 Install PeopleSoft Software

Installing the PeopleSoft software is just a matter of unzipping the uploaded ZIP files into the correct directory locations. Since the file system is shared across all middle tiers in OCI, this is done on just one of the PeopleSoft compute instances:

An example of unzipping individual ZIP files:

```
$ ssh -I <path to key file> opc@iad-psft-hcm-app01
$ sudo su – psadm2
$ cd /u01/app/psft/pt
$ unzip ps_home.zip
$ unzip ps_app_home.zip
$ unzip ps_cust_home.zip
$ unzip ps_jdk.zip
$ unzip ps_bea.zip
$ unzip oracle_home.zip
$ unzip tns_admin.zip
```

If all your directories are in one ZIP file, your command would be like this:

```
$ ssh -I <path to key file> opc@iad-psft-hcm-app01
$ sudo su – psadm2
$ cd /u01/app/psft
$ unzip pt.zip
```

8.2.3 Restoring Application / Process Scheduler Domain Configuration

If you captured the application server (APPSRV) and process scheduler (PRCS) domain configurations from the source system under PS_CFG_HOME/appsrv directory, then these configurations can be restored onto those OCI compute instances that will host the application server and process scheduler server domains.

To restore the application and process scheduler domain configuration, the following steps:

1. Copy the ps_cfg_home.zip file to all OCI compute instances that will host the application and process scheduler domains.

2. As the psadm2 user on each OCI compute instance from step 1 above:

   ```
   $ cd $PS_CFG_HOME
   $ unzip ps_cfg_home.zip
   ```

3. Verify that the directory structure looks similar to:

   For the application server domain:

   `$PS_CFG_HOME/appserv/<App server domain name from source system>`

   For the process scheduler domain

   `$PS_CFG_HOME/appserv/<App server domain name from source system>/prcs/<Process server domain name from source system>`

8.2.4 Create Custom Administrative Scripts

At this point you can create and populate your custom administrative script directory, $SCRIPT_DIR, introduced in PeopleSoft Software Directory Structure. See the sample scripts in Appendix C – Basic Tasks and Scripts for examples.

8.3 Configure PeopleSoft

We need to access the database to complete the file system configuration of the middle tier servers at the standby. We follow these steps:

- Convert the standby database to snapshot standby
- Configure the database connections
- Configure the Tuxedo and application server domains
- Configure the process (batch) scheduler domains

- Set up monitoring for the application servers
- Configure the web server domains

8.3.1 Convert First Standby Database on ExaDB-D to Snapshot Standby

We need to bring the application up to complete configuration of the standby environment. To do this, we need to temporarily access the database in read-write mode, which cannot be done while the database is applying redo from the primary. We will convert the OCI database from a physical standby to a snapshot standby. When in this state, redo from the primary database will be shipped to the standby but will not be applied until the database is reverted to a physical standby.

Follow these steps to convert the database on ExaDB-D to snapshot standby.

1. With SSH, log in to one of the ExaDB-D DB nodes (domUs).
2. Become the oracle user

   ```
   $ sudo su – oracle
   ```

3. Source the standby database environment. This environment was created when the placeholder database was created in preparation for using ZDM.

   ```
   $ . ./CDBHCM.env
   ```

4. Start Data Guard Broker and enter the SYS password.

   ```
   $ dgmgrl
   DGMGRL for Linux: Release 19.0.0.0.0 - Production on Tue Nov 15 18:38:11 2022
   Version 19.14.0.0.0

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

   Welcome to DGMGRL, type "help" for information.
   DGMGRL> connect sys/<password>
   Connected to " CDBHCM_iad1dx"
   Connected as SYSDBA.
   DGMGRL>
   ```

5. Show the Data Guard configuration, including redo and apply lag times.

   ```
   DGMGRL> show configuration lag

   Configuration - ZDM_CDBHCM_iad1dx

     Protection Mode: MaxPerformance
     Members:
     CDBHCM_sca6dp   - Primary database
       CDBHCM_iad1dx - Physical standby database
                        Transport Lag:      0 seconds (computed 1 second ago)
                        Apply Lag:          0 seconds (computed 1 second ago)
       CDBHCM_phx5s - Physical standby database (receiving current redo)
                         Transport Lag:      1 second (computed 1 second ago)
                         Apply Lag:          2 seconds (computed 1 second ago)


   Fast-Start Failover:  Disabled
   ```

```
Configuration Status:
SUCCESS   (status updated 43 seconds ago)
```

6. If there are no errors and the above show “SUCCESS”, convert the standby database to a snapshot standby.

```
DGMGRL> convert database CDBHCM_iad1dx to snapshot standby
Converting database "CDBHCM_iad1dx" to a Snapshot Standby database, please wait...
Database "CDBHCM_iad1dx" converted successfully
```

7. Show the configuration once more to see that the physical standby is now a snapshot standby. You may see warnings indicating that the transport lag has exceeded thresholds. This warning will go away after several minutes.

```
DGMGRL> show configuration lag

Configuration - ZDM_ CDBHCM_iad1dx

  Protection Mode: MaxPerformance
  Members:
   CDBHCM_sca6dp - Primary database
   CDBHCM_iad1dx - Snapshot standby database
                     Transport Lag:      0 seconds (computed 2 seconds ago)
                     Apply Lag:          10 minutes 14 seconds (computed 2 seconds ago)
    CDBHCM_phx5s - Physical standby database (receiving current redo)
                      Transport Lag:      1 second (computed 1 second ago)
                      Apply Lag:          2 seconds (computed 1 second ago)


Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 60 seconds ago)
```

8. Check to see that the services required for PeopleSoft have been started. For this project, the services are: HR92U033_BATCH and HR92U033_ONLINE. ZDM will have migrated these services from the source (on-prem) database and registered them into Clusterware.

```
$ srvctl status service -db CDBHCM_iad1dx -s HR92U033_BATCH
Service HR92U033_BATCH is running on instance(s) CDBHCM1,CDBHCM2
$ srvctl status service -db CDBHCM_iad1dx -s HR92U033_ONLINE
Service HR92U033_ONLINE is running on instance(s) CDBHCM1,CDBHCM2
```

9. If the required services have not been started, start them per the below example:

```
$ srvctl start service -db CDBHCM_iad1dx -s HR92U033_BATCH
$ srvctl start service -db CDBHCM_iad1dx -s HR92U033_ONLINE
```

The snapshot standby should now be ready for configuring the application and process scheduler.

8.3.2 PeopleSoft Database Connection

Edit the tnsnames.ora file, changing the HOST value to the cluster SCAN name in the database connect strings for both the application server and process scheduler domains.

Note that PeopleTools uses TNS connect aliases as DBNames. Make sure the TNS alias names match the DBNAME column in your PS.PSDBOWNER table. . The entries in our PSDBOWNER table are:

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

Our tnsnames.ora file has the following entries:

```
# Application server
HR92U033 =
    (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
           (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = iadexadb-bw5wn-
scan.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )

# Process scheduler
HRBATCH =
   (DESCRIPTION =
       (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS_LIST =
       (LOAD_BALANCE=on)
           (ADDRESS = (PROTOCOL = TCP)(HOST = iadexadb-bw5wn-
scan.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com)(PORT = 1521))
       )
       (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_BATCH)
       )
    )
```

To test this configuration, first make sure the database services migrated by ZDM – in our configuration, HR92U033_ONLINE and HR92U033_BATCH – are up and running. As the psadm2 OS user, we tested the first TNS connect string alias with SQL*Plus:

```
$ sqlplus PS/<password>@HR92U033
SQL*Plus: Release 19.0.0.0.0 - Production on Sat Nov 20 00:50:53 2021
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle.  All rights reserved.
Last Successful login time: Fri Nov 19 2021 02:06:04 +00:00

Connected to:
Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production
Version 19.19.0.0.0

SQL>
```

Test the HRBATCH connect string alias in a similar fashion, to ensure it successfully connects to the database. Perform these tests on all compute instances that will host the application and process scheduler.

8.3.3 PeopleSoft Application and Process Scheduler Domains

In our implementation, two of the four compute instances will each host both the application servers and the process scheduler: iad-psft-hcm-app01 and iad-psft-hcm-app02. The application servers and the process scheduler each run within their own Oracle Tuxedo domain.

The psadmin utility is used to configure the Tuxedo domain, including the application server.

Find the psappsrv.cfg file in the $PS_CFG_HOME/appserv/<Application server domain> directory. Make any required configuration changes to this file, using the psadmin utility ($PS_HOME/bin/psadmin), before deploying the domain on each node.

In our case, the application server domain is HR92U033. The steps we used to deploy the HR92U033 domain are provided here:

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS

1	iad-psft-hcm-app01,
	iad-psft-hcm-app02,
	psadm2
		Start the psadmin utility:
		`cd $PS_HOME/bin`
		`./psadmin`
		When the psadmin utility starts, it will show the directory paths for PS_HOME, PS_CFG_HOME and PS_APP_HOME. Make sure they are correct. In our environment:
		`PS_CFG_HOME    /peoplesoft/local/ps_config`
		`PS_HOME        /u01/app/psft/pt/ps_home8.57.11`
		`PS_APP_HOME    /u01/app/psft/pt/hcm_app_home`

2	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	To configure the application server domain, Select option 1: **Application Domain**
3	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 1: Administer a Domain. Select the domain name, in our case **HR92U033**.
4	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 4: Configure this domain. Enter Y to shut the domain down.
5	iad-psft-hcm-app01, iad-psft-hcm-app02,	There are several settings presented in this display. Review the configuration options: ● Ensure your TNS connect string aliases are correct for DBName. In our case it is

	psadm2	**HR92U033**. ● Make any adjustments needed for ports used by the application domain server. ● Take note in particular of the JOLT port range – the ports used by the PIA web server to connect to the application server. You will use these values in a later step. The default is 9000 – 9010.
6	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 14: Load domain as shown. The domain configuration will be loaded and all the required domain files for Oracle Tuxedo will be regenerated.
7	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Under Domain Administration, option 1: *Boot this domain* will start this domain. Monitor the startup process. If errors are reported, check the log files in. $PS_CFG_HOME/appserv/<domain name>/LOGS

Table 22: Application server domain configuration

The PeopleSoft process scheduler (or batch server) is configured in much the same way as the application server, plus the configuration of the process scheduler logs. As user psadm2, we will create one directory for each process scheduler node:

/u02/app/psft/ps/log_output/node1/HR92U033 (for node: iad-psft-hcm-app01)

/u02/app/psft/ps/log_output/node2/HR92U033 (for node: iad-psft-hcm-app02)

We can do this from any one of the middle tier compute instances.

With the log output directories in place, we did the following to configure the process scheduler:

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS

1	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Start the psadmin utility:
		`cd $PS_HOME/bin`
		`./psadmin`
		When the psadmin utility starts, it will show the directory paths for PS_HOME, PS_CFG_HOME and PS_APP_HOME. Make sure they are correct. In our environment:
		`PS_CFG_HOME            /peoplesoft/local/ps_config`
		`PS_HOME                /u01/app/psft/pt/ps_home8.57.11`
		`PS_APP_HOME            /u01/app/psft/pt/hcm_app_home`

2	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 2: **Process Scheduler**.
3	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 1: Administer a Domain. Select the domain name, in our case **HR92U033**.
4	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 4: Configure this domain. Enter Y to shut the domain down.
5	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	There are several settings presented in this display. Review the configuration options and ensure that your TNS connect string aliases are correct for DBName. In our case it is **HRBATCH**.
6	iad-psft-hcm-app01, iad-psft-hcm-app02,	Select option 16: Log/Output and provide the directory paths you just created. On our system: /u02/app/psft/ps/log_output/node1/HR92U033 (for node: iad-psft-hcm-app01)

	psadm2	/u02/app/psft/ps/log_output/node2/HR92U033 (for node: iad-psft-hcm-app02)
7	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Select option 6: Load domain as shown. The domain configuration will be loaded and all the required domain files for Oracle Tuxedo regenerated.
8	iad-psft-hcm-app01, iad-psft-hcm-app02, psadm2	Under Domain Administration, option 1: *Boot this domain* will start the domain. Monitor the startup process. If errors are reported, check the log files in. $PS_CFG_HOME/appserv/prcs/<domain name>/LOGS

Table 23: Process scheduler domain configuration

Our final step is to open the JOLT port on all application server and process scheduler compute instances. This is accomplished with the firewall-cmd, which is run as root. We issued these commands – modify them for your environment:

```
# firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source address=10.0.103.0/24 port
port=9000-9100 protocol=tcp accept' –permanent
# firewall-cmd --reload
```

This simple query can be used to monitor the application and process scheduler database connections as these Tuxedo domains start. Log in to one of the ExaDB-D RAC instances as the Oracle OS user to run the query:

```
$ sqlplus / as sysdba
SQL> col service_name format a20
SQL> select a.inst_id,a.instance_name,b.service_name, count(*)
2> from gv$instance a, gv$session b
3> where a.inst_id = b.inst_id
4> and service_name not like 'SYS%'
5> group by a.inst_id,a.instance_name,b.service_name
6> order by 1;
```

This is the output from our system:

```
   INST_ID INSTANCE_NAME    SERVICE_NAME           COUNT(*)
---------- ---------------- -------------------- ----------
         1 CDBHCM1          HR92U033_BATCH                8
         1 CDBHCM1          HR92U033_ONLINE              54
         2 CDBHCM2          HR92U033_BATCH                7
         2 CDBHCM2          HR92U033_ONLINE              48
```

The above output shows the number of connections to each service on each RAC instance. The processes should be scattered across the database servers.

8.3.4 PeopleSoft PIA Web Server Domain

We now turn our attention to the PIA web servers.

We will use the PIA setup.sh script to configure the PIA web server domain.

Perform the steps in the table below on every compute instance VM that will host a PIA web server:

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS
1	iad-psft-hcm-web01, iad-psft-hcm-web02, psadm2	As psadm2, source the environment: `$ . ./psft.env` Ensure the PS_CFG_HOME is defined. For this project it is set to: /peoplesoft/local/ps_config. If it is not, be sure your .bash_profile calls the psft.env file..

2	iad-psft-hcm-web01,
	iad-psft-hcm-web02,
	psadm2
		Copy the template response file to the PS_CFG_HOME directory for customization.
		`$ cd $PS_HOME/setup/PsMpPIAInstall`
		`$ cp resp_file.txt $PS_CFG_HOME/iad_oci_pia_resp.txt`

3	iad-psft-hcm-web01,
	iad-psft-hcm-web02,
	psadm2
		Edit the iad_oci_pia_resp.txt file for your environment. We highlighted the items that needed to be changed in our file below:
		`# 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=welcome1`
		`USER_PWD_RETYPE=welcome1`
		`# 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 distributed webLogic server .`
		`#Possible values are "SINGLE_SERVER_INSTALLATION", "MULTI_SERVER_INSTALLATION" and "DISTRIBUTED_SERVER_INSTALLATION"`
		`INSTALL_TYPE=SINGLE_SERVER_INSTALLATION`
		`# WebSite Name` **← THIS MUST BE THE SAME FOR ALL PeopleSoft WEB SERVERS**
		`WEBSITE_NAME=ps` **← For our project we chose “ps”.**
		
		`# AppServer Name`
		`APPSERVER_NAME=iad-psft-hcm-app01` **← 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=appprivatesu.ebscloudmaavcn.oraclevcn.com` **← Change this to match the network 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=/u02/app/psft/ps/report_repository ← **Report repository location**

4	iad-psft-hcm-web01, iad-psft-hcm-web02, psadm2	Run the setup.sh script to configure the PIA, pointing to your new response file. We used this command: `$ ./setup.sh -i silent -DRES_FILE_PATH=$PS_CFG_HOME/iad_oci_pia_resp.txt`

5	iad-psft-hcm-web01,
	iad-psft-hcm-web02,
	psadm2
		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 defining the psserver to add the second application domain server. For example, our original entry was:
		`psserver=iad-psft-hcm-app01.appprivatesu.ebscloudmaavcn.oraclevcn.com:9000`
		We edited it in this manner:
		`psserver=iad-psft-hcm-app01.appprivatesu.ebscloudmaavcn.oraclevcn.com:9000,iad-psft-hcm-app02.appprivatesu.ebscloudmaavcn.oraclevcn.com:9000`

Table 24: PIA web server configuration

8.4 PeopleSoft Full Stack Test in OCI

It's best to conduct a full stack test of PeopleSoft at this point – before configuring Coherence*Web – to ensure all components are working. Doing so validates that all core components are functioning properly. A methodical test procedure will help isolate any issues to a specific component within the Peoplesoft stack or the OCI infrastructure. If the full stack test is skipped, then diagnosing and debugging issues when adding a load balancer and Coherence*Web cache server will be more difficult.

We created startup and shutdown scripts for our PeopleSoft environment. There are simple examples in the appendix. We will use the scripts in the next step, where we perform tests to make sure the basic configuration is complete.

We first tested each application server domain individually with one web server domain, then we started all application server and web server domains and connected through each web server domain individually.

STEP #	NODE, USER	
1	iad-psft-hcm-web01, psadm2	Start one PIA web server domain on a web server compute instance. Make sure it starts successfully, and it is the only one running.
2	iad-psft-hcm-appXX, psadm2	Start one application server domain on one application server compute instance. Make sure it starts successfully, and it is the only one running.
3	iad-psft-hcm-web01, user desktop, PeopleSoft application Administrator	Using a web browser, log in to the PeopleSoft application through the PIA web server in this case: iad-psft-hcm-web01. Once logged in, navigate around the application to ensure the application is functioning properly. Our URL to login (non-SSL) http://<web server address>,<network domain>:8080/psp/ps/?cmd=login&languageCd=ENG

		The above URL should redirect the browser to the application login. See the Note below if you are using a bastion host in your network topology.
4	iad-psft-hcm-appXX,psadm2	If you have not tested all your application server domains, shut down the one running now and start one not yet tested. Execute step 2 above. Else proceed to step 5.
5	All application domain severs, all PIA web servers, psadm2	Now, start all application server domains and all PIA web server domains.
6	Each PIA web server, user desktop, PeopleSoft application Administrator	Using a web browser, log in to the PeopleSoft application, adjusting your URL to connect via each PIA web server and again navigate around the application.

Table 25: PeopleSoft Full Stack Test

Note: For steps 3 and 6 above, if the PIA web servers are running on compute instances that were provisioned onto a private subnet, then it will be necessary to access the application either through the FastConnect or IPsec VPN network you use to access OCI. If using a bastion host, each user desktop will need the following::

1. Edit the local hosts file and add each PIA web server's host name. On Windows, the file to edit is C:\Windows\System32\drivers\etc\hosts. Open this file with Notepad as Administrator.
2. Add an entry similar to this, adjusted for your environment:

   ```
   127.0.0.1    localhost iad-psft-hcm-web01.<Private-subnet-domain>
   ```

   Save the file.

3. Create an ssh tunnel on your desktop through the bastion host to one of the web servers: Run the following command to start an ssh process that routes all traffic on port 8080 of the localhost IP address through the bastion host to the specified web server: Note: this configuration assumes the private key is loaded using the local ssh agent.

   ```
   $ ssh -4 -fN opc@ashbastion -L 8080:<PIA-web server private IP address>:8080
   ```

4. Once the tunnel is established, enter the following into the browser:

   ```
   http://iad-psft-hcm-web01.<Private-subnet-domain>:8080/psp/ps/?cmd=login&languageCd=ENG
   ```

   If SSL certificates have been installed on the PIA web server, then change the above URL accordingly.

8.5 OCI Load Balancer Provisioning and Configuration

OCI Load Balancer as a Service (LBaaS) is provisioned as a virtual resource just like any other resource. LBaaS will distribute traffic across multiple web servers. We recommend LBaaS be provisioned onto its own subnet (private or public).

8.5.1 Load Balancer Prerequisites for PeopleSoft

Before provisioning the OCI Load Balancer, make sure these configurations are correct:

All PIA web server domains must have the same cookie name and network domain specified in the $PS_CFG_HOME/webserv/<domain>/applications/peoplesoft/PORTAL.war/WEB-INF/weblogic.xml file. For this project, the WebLogic domain is HR92U033, so the location of the file is: $PS_CFG_HOME/webserv/HR92U033/applications/peoplesoft/PORTAL.war/WEB-INF/weblogic.xml.

Within this XML file, we specified our cookie as follows:

```
<cookie-name>iad-hcm-8080-PORTAL-PSJSESSIONID</cookie-name>
<cookie-domain>.appprivatesu.ebscloudmaavcn.oraclevcn.com</cookie-domain>
```

The cookie domain is the network domain name associated with the private subnet app_private_subnet. Make sure all PIA WebLogic server domains have the exact same cookie and network domain names in their respective weblogic.xml file.

Next, create an SSL bundle with the certificate files in PEM format. This bundle will be uploaded during or after the LBaaS creation process. You may upload the certificate bundle by clicking on Certificates link after the load balancer has been created. The certificate bundle must be uploaded prior to associating it with the listener that will be SSL-enabled. Please consult your corporate security team for what is required by your company for your SSL certificates.

8.5.2 Provision OCI Load Balancer

This section will *not* walk you through the provisioning process of the OCI load balancer, because the steps change as OCI is continually enhanced. See the OCI documentation on load balancing.

This is how we configured our LBaaS load balancer:

- LBaaS shape – we used 400 megabits per second for our test environment
- LBaaS display name – IAD_PSFT_LBaaS_PROD
- Backend set:
  - Traffic distribution policy – weighted round robin
  - Session persistence – iad-hcm-8080-PORTAL-PSJSESSIONID.
  - SSL enablement
    If you choose to have the backend servers SSL enabled, check the checkbox for Use SSL and fill in the appropriate information. We did not check this in our test environment.
  - Health check
    Health check must be defined for the backend set. It is applied to all available backend servers to determine their health according to your configuration. The load balancer will not route traffic to an unhealthy backend server. These are the attributes we specified for our health check:

ATTRIBUTE	VALUE
Protocol	HTTP Since SSL is terminated at the load balancer, HTTP is selected.
Port	8080 HTTP port for all PIA web servers.
Interval in milliseconds	10000 Number of milliseconds between each check. 10000ms = 10 seconds
Timeout milliseconds	3000 Number of milliseconds that the check will wait before timing out. 3000 = 3 seconds.
Number of retries	3

	Number of attempts to get a response from a given backend server.
Status code	200 The expected successful status code for HTTP GET calls.
URL path (URI)	/ Starting path, normally the root path of /.
Response Body RegEx	.* Regular expression that allows any response returned from the HTML page to be acceptable.

Table 26: Load balancer health check

8.5.3 Configure PeopleSoft for SSL Termination at Load Balancer

Configure PeopleSoft to use your new SSL-terminated load balancer. This configuration is required so that dynamically generated URL redirects use the https protocol. You will copy your current PIA web profile and adjust it to use your SSL configuration. Follow the steps in the table below:

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS
1	PeopleSoft Admin User, PS	Log in to the PIA Web application as a PeopleSoft administrator such as PS.
2	PeopleSoft Admin User, PS	Copy the current or active web profile: a. Click on the navigation bar icon labeled “NavBar”. b. Click on Navigator c. Click on PeopleTools (may require scrolling through the options) d. Click on Web Profile (may require scrolling through the options) e. Click on Copy Web Profile f. Click Search (not required to enter anything into the search box) g. Click on an active web profile from the list such as PROD h. In the text box labelled “To”, enter a name e.g., PROD_SSL or a name of your choosing. i. Click Save j. Click the Home icon labelled “Home”.
3	PeopleSoft Admin User, PS	Configure the copied web profile: a. Click on the navigation bar icon labelled “NavBar”. b. Click on Navigator (If in the same session from step 2, the navigator remembers where you were.) c. Click Web Profile Configuration d. Click Search (not required to enter anything into the search box) e. Click on an active web profile from the list such as PROD_SSL we created as a copy of PROD. f. Click the Virtual Addressing tab g. In the Default Addressing area, for the Protocol text box, enter “https” (lower case and without the quotes) h. In the text box for port, you can enter the default port number of 443 or a different port you wish to use. Leaving it blank will default to port 443. i. Click Save j. Exit the application.

4	All PIA web servers, psadm2	SSH to each PIA web server
		Modify the PIA configuration.properties file.
		On each PIA web domain server, change the WebProfile setting to the modified web profile.
		$ cd /peoplesoft/local/ps_config/webserv/<PIA domain>/applications/peoplesoft/PORTAL.war/WEB-INF/psftdocs/<site-name>
		For this project:
		$ cd /peoplesoft/local/ps_config/webserv/HR92U033/applications/peoplesoft/PORTAL.war/WEB-INF/psftdocs/ps
		Create a backup of the configuration.properties file:
		$ cp configuration.properties configuration.properties.backup
		Edit the configuration.properties file and set the WebProfile profile as below. Note that we comment out the original setting.
		# WebProfile=PROD ← **Original web profile**
		WebProfile=PROD_SSL ← **New web profile**
		Save the file.

5	All PIA web nodes, psadm2	Restart each PIA web server, using the scripts from PeopleSoft Startup / Shutdown Scripts. We assume these scripts are in the administrator account's PATH.
		$ stopWS.sh
		$ startWS.sh

Table 27: Enabling SSL terminated at the load balancer

Just as with the application server compute instances, traffic from the load balancer must be allowed onto each PIA web server compute instance. You will issue firewall-cmd commands on each PIA web compute instance, as root. The commands for our environment were:

```
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.105.0/24 port port=8080 protocol=tcp accept' –permanent
$ firewall-cmd --reload
```

Check the LBaaS backend servers to ensure they each come up with a status of "OK".   It may take a few minutes for the status to change from a red diamond "Critical" to a yellow triangle "Warning" and then to a green "OK".

At this point, you should be able to log in to the PeopleSoft application through the LBaaS using a URL similar to the following:

https://<load balancer alias name.VCN doman>/ psc/ps//?cmd=login&languageCd=ENG

For this project the URL is:

https://psfthcm.appprivatesu.ebscloudmaavcn.oraclevcn.com/psc/ps//?cmd=login&languageCd=ENG

8.6 PIA Web Resiliency with Coherence*Web

Implementing Coherence*Web for PeopleSoft PIA web servers has the following tasks:

- Coherence*Web configuration
- Coherence*Web network configuration
- PIA web server configuration

8.6.1 Coherence*Web Configuration

Coherence*Web is configured on each PIA web server and runs parallel to but separate from the PIA WebLogic servers. This is called *Out-of-Process Topology* for Coherence*Web. A cluster is formed when two or more Coherence*Web cache servers are started. Once the PIA web servers have been configured to work with Coherence*Web (described in the next section), they too will join this cluster.

With PeopleTools 8.57, Coherence*Web is bundled with the Fusion Middleware installation. In this project, it is in the directory: /u01/app/psft/pt/bea/coherence. This is a shared directory that all PIA web servers can access.

In this section, we will set up the infrastructure and scripting needed to start and manage Coherence*Web. Note that while we will create the script that starts the product, we have a few more tasks to accomplish before we can run it.

The following table provides the steps for configuring Coherence*Web.

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS

1	All PIA web server nodes, psadm2	Make directories for Coherence*Web configuration and log files.
		`$ . ./psft.env`
		`$ mkdir -p $PS_CFG_HOME/coherence/config`
		`$ mkdir -p $PS_CFG_HOME/coherence/log`

2	All PIA web server nodes, psadm2	Edit the psft_env file to add these environment variables: to the psft.env file:.
		`export COHERENCE_HOME=$BASE_DIR/pt/bea/coherence`
		`export COHERENCE_CONFIG=$PS_CFG_HOME/coherence/config`
		`export COHERENCE_LOG=$PS_CFG_HOME/coherence/log`
		
		`Re-source the environment file.`

3	All PIA web server nodes, psadm2, OCI console user	In our project, each PIA web server has a private IP address, referred to as a *well-known-address* (WKA) in the Coherence*Web documentation. You will need these private IP addresses in later steps. To obtain the private IP address of each PIA web server compute instance, log in to the OCI console, click on Compute, then Instances. Select the compartment holding the web servers, then click on each of the PIA web servers listed to see their private IP address. Record the addresses.

4	All PIA web server nodes, psadm2	In the COHERENCE_CONFIG directory, create a Coherence override file. You can choose any name for the file, but it is generally called: tangosol-coherence-override.xml, and is referenced with that name in several Coherence*Web documents. The XML file will have content similar to the following:
		`<?xml version="1.0"?>`
		`<!DOCTYPE coherence SYSTEM "coherence-override.dtd">`
		`<!--`
		`This operational configuration override file is set up for use with Coherence`
		`-->`
		`<coherence>`
		`<cluster-config>`
		`<unicast-listener>`
		`  <well-known-addresses>`
		`   <socket-address id="1">`
		`      <address>10.0.103.85</address>`
		`   </socket-address>`
		`   <socket-address id="2">`
		`      <address>10.0.103.203</address>`
		`  </socket-address>`
		`</well-known-addresses>`
		`<port system-property="tangosol.coherence.localport">8088</port>`
		`<address system-property="coherence.localhost">10.0.103.0/24</address>`
		`</unicast-listener>`
		`</cluster-config>`
		`</coherence>`
		Notes:
		• You will specify the private IP address for every PIA web server. We have two in our configuration.

		• The above file must be placed on ALL PIA web servers and the content must be identical. • The parameter <address system-property="coherence.localhost">10.0.103.0/24</address> specifies the entire private subnet serving the PIA web servers. When the Coherence*Web cache server starters, it will obtain the local host private IP address that it is running on.
5	All PIA web server nodes, psadm2	Create a script that will start the Coherence*Web cache server in the background on each PIA web server. The script we created can be found in startCacheServer.sh. We placed the script in our custom scripts directory, discussed in PeopleSoft Startup / Shutdown Scripts. Note: wWe will only create the script here – we will not execute it until after we configure the Coherence*Web network.

Table 28: Coherence*Web configuration

8.6.2 Coherence*Web Network Configuration

Each PIA web server runs in isolation and does not communicate with the other web servers. To provide cross-server resiliency at this layer, the Coherence*Web servers form a cluster and do require network configuration that allows their cache servers to communicate with each other.

To allow Coherence*Web to form a cache cluster, we will use the OCI console to add the ingress and egress rules in the following two tables to the security list associated with the app_private_seclist. In addition to ports 8088 and 8089, we are using the default ports as specified in the Coherence*Web documentation.

For Ingress rules:

STATELESS	SOURCE CIDR	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE
NO	10.0.103.0/24	TCP	All	7
NO	10.0.103.0/24	UDP	All	7
NO	10.0.103.0/24	TCP	All	7574
NO	10.0.103.0/24	UDP	All	7574
NO	10.0.103.0/24	TCP	All	8088-8089

Table 29: Ingress rules for Coherence*Web

For egress rules:

STATELESS	DESTINATION CIDR	IP PROTOCOL	SOURCE PORT RANGE	DESTINATION PORT RANGE
NO	0.0.0.0/0	TCP	All	All
NO	0.0.0.0/0	UDP	All	All

Table 30: Egress rules for Coherence*Web

The ports in the above tables must also be enabled on each of the PIA web compute instances. To do this, the following commands were executed as root on each compute instance. Modify for your environment:

```
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.103.0/24 port port=7 protocol=tcp accept' –permanent
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.103.0/24 port port=7 protocol=udp accept' –permanent
```

```
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.103.0/24 port port=8088-8089 protocol=tcp accept' –permanent
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.103.0/24 port port=7574 protocol=tcp accept' –permanent
$ firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=10.0.103.0/24 port port=7574 protocol=udp accept' --permanent
$ firewall-cmd –reload
```

These actions explicitly allow traffic from each compute instance to appropriately traverse the network.

It should now be possible to start the Coherence*Web cache servers. This first time, we will start them one at a time and check each one to be sure they are configured correctly.

Use the startCacheServer.sh script found in Coherence*Web Configuration to start a cache server, by issuing the following commands on one compute instance as psadm2. We assume the script directory is in the administrator's PATH.

```
$ startCacheServer.sh
```

When the first cache server is started, the output will look similar to the following:

```
…
2022-08-31 21:15:11.539/0.639 Oracle Coherence 12.2.1.3.0 <Info> (thread=main,
member=n/a): Loaded operational configuration from
"jar:file:/u01/app/psft/pt/bea/coherence/lib/coherence.jar!/tangosol-coherence.xml"
…
2022-08-31 21:15:16.137/5.237 Oracle Coherence GE 12.2.1.3.0 <Info> (thread=main,
member=n/a): Started cluster Name=psadm2's cluster, ClusterPort=7574

WellKnownAddressList(
  10.0.103.203
  10.0.103.85
  )

MasterMemberSet(
  ThisMember=Member(Id=1, Timestamp=2022-08-31 21:15:12.825, Address=10.0.103.85:8088,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web01,process:21537, Role=CoherenceServer)
  OldestMember=Member(Id=1, Timestamp=2022-08-31 21:15:12.825, Address=10.0.103.85:8088,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web01,process:21537, Role=CoherenceServer)
  ActualMemberSet=MemberSet(Size=1
    Member(Id=1, Timestamp=2022-08-31 21:15:12.825, Address=10.0.103.85:8088,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web01,process:21537, Role=CoherenceServer)
    )
  MemberId|ServiceJoined|MemberState|Version
    1|2022-08-31 21:15:12.825|JOINED|12.2.1.3.0
  RecycleMillis=1200000
  RecycleSet=MemberSet(Size=0
```

We can see the configuration is being picked up correctly by seeing the IP addresses listed in the WellKnownAddresslist section.

Since we have only started one cache server, there is only one member – the compute instance on which the cache server was started: iad-psft-hcm-web01.

Use the same script to start the cache server on the second middle tier compute instance. It should join the cluster. its log file should look similar to the following:

```
2022-08-31 21:15:36.513/0.628 Oracle Coherence 12.2.1.3.0 <Info> (thread=main,
member=n/a): Loaded operational configuration from
"jar:file:/u01/app/psft/pt/bea/coherence/lib/coherence.jar!/tangosol-coherence.xml"
…
2022-08-31 21:15:38.307/2.421 Oracle Coherence GE 12.2.1.3.0 <Info> (thread=main,
member=n/a): Started cluster Name=psadm2's cluster, ClusterPort=7574

WellKnownAddressList(
  10.0.103.203
  10.0.103.85
  )

MasterMemberSet(
  ThisMember=Member(Id=2, Timestamp=2022-08-31 21:15:37.981, Address=10.0.103.203:8088,
MachineId=10880, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web02,process:12859, Role=CoherenceServer)
  OldestMember=Member(Id=1, Timestamp=2022-08-31 21:15:12.825, Address=10.0.103.85:8088,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web01,process:21537, Role=CoherenceServer)
  ActualMemberSet=MemberSet(Size=2
    Member(Id=1, Timestamp=2022-08-31 21:15:12.825, Address=10.0.103.85:8088,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web01,process:21537, Role=CoherenceServer)
    Member(Id=2, Timestamp=2022-08-31 21:15:37.981, Address=10.0.103.203:8088,
MachineId=10880, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-
psft-hcm-web02,process:12859, Role=CoherenceServer)
    )
  MemberId|ServiceJoined|MemberState|Version
    1|2022-08-31 21:15:12.825|JOINED|12.2.1.3.0,
    2|2022-08-31 21:15:37.981|JOINED|12.2.1.3.0
  RecycleMillis=1200000
  RecycleSet=MemberSet(Size=0
    )

…
```

The log file snippet above shows two members in the cluster, both with a role of “CoherenceServer”.

Now look at the log file of the cache server that was started first on compute instance iad-psft-hcm-web01. You should see entries similar to the following showing that a connection to iad-psft-hcm-web02 has been established.

```
2022-08-31 21:15:38.179/27.279 Oracle Coherence GE 12.2.1.3.0 <D6> (thread=Cluster,
member=1): TcpRing connected to Member(Id=2, Timestamp=2022-08-31 21:15:37.981,
Address=10.0.103.203:8088, MachineId=10880,
Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-psft-hcm-
web02,process:12859, Role=CoherenceServer)
```

Once you verify the configuration is solid / in the future when starting your middle tiers, you can start all the Coherence*Web cache servers at one time.

Tips:

If on each compute instance you only see one member, the Coherence*Web cache servers cannot "see" each other. This is caused by one or more network ports being blocked. One way to test for ports being blocked is to use the Netcat (nc) utility, which must be installed on each PIA web compute instance. If the tool is not already installed, use yum to install it on each PIA web compute instance as the user root:

```
# yum install -y nc
```

Once installed, you can test ports for both TCP and UDP packets on your PIA web servers. For example, we can test to see if port 7574 is open on iad-psft-web01 by doing the following:

Log in as root on iad-psft-web01. Start a Netcat *listener* that listens for UDP packets on port 7574:

```
# nc -u -l iad-psft-web01 7574
```

The nc command will wait – let it sit there.

In a separate session, log in as root on iad-psft-web02. Start a Netcat UDP *client* that connects to iad-psft-web01 on port 7574:

```
# nc -u iad-psft-web01 7574
```

The nc command will wait. On that terminal, type something like "test" and press Enter. The word "test" should show up on the other terminal. Now type "test" on the server terminal where nc is listening and press Enter. It should show up in the client terminal. If this is successful, then switch roles – run the nc listener on iad-psft-web02 and the client on iad-psft-web01 and repeat the test.

If the above is not successful, then one or more of the following are likely the cause:

- There is no ingress rule allowing UDP protocol for port 7574 in the relevant security list for the PIA web server subnet.
- There is no egress rule allowing UDP protocol for port 7574 in the relevant security list for the PIA web server subnet.
- The firewall-cmd command has not been run on each PIA web server to allow UDP traffic for port 7574.

Coherence*Web provides a datagram test utility to test connections for both TCP and UDP. Refer to Using the Datagram Test Utility for further details.

8.6.3 PIA Web Server Configuration for Coherence*Web

With the Coherence*Web cache servers configured and running on all PIA web server compute instances, the PIA web servers can be configured. Recent versions of PeopleTools configuration files already have most of the required configuration, commented out. To enable access to Coherence*Web cache servers, follow the steps in the table below, adjusting as appropriate for your environment.

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS
1	All PIA web servers, psadm2	Review Load Balancer Prerequisites for PeopleSoft. Ensure that the session cookie and network domain names are the same for all PIA web servers in the weblogic.xml file. The weblogic.xml file will be located in $PS_CFG_HOME/webserv/HR92U033/applications/peoplesoft/PORTAL.war/WEB-INF. For this project, the domain is HR92U033.

<table>
<tr>
<td>2</td>
<td>All PIA web servers, psadm2</td>
<td>
In the same weblogic.xml file, move the “end comment” string from below the <persistent-store-type>coherence-web</persistent-store-type> tag to above it, so it is no longerpart of the comment:

Here is a full listing of the weblogic.xml file:

<pre>
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
<description>PeopleSoft Internet Architecture</description>
  <session-descriptor>
  <id-length>32</id-length>
  <cookie-name>iad-hcm-8080-PORTAL-PSJSESSIONID</cookie-name>
  <cookie-domain>.appprivatesu.ebscloudmaavcn.oraclevcn.com</cookie-domain>
  <monitoring-attribute-name>USERID</monitoring-attribute-name>
  <persistent-store-table>wl_servlet_sessions</persistent-store-table>
  <http-proxy-caching-of-cookies>true</http-proxy-caching-of-cookies>
<!-- Coherence*Web
-->
  <persistent-store-type>coherence-web</persistent-store-type>
  </session-descriptor>
    <container-descriptor>
      <servlet-reload-check-secs>-1</servlet-reload-check-secs>
      <session-monitoring-enabled>true</session-monitoring-enabled>
  </container-descriptor>
  <context-root>/</context-root>
</weblogic-web-app>
</pre>
</td>
</tr>
<tr>
<td>3</td>
<td>All PIA web servers, psadm2</td>
<td>
The web.xml file is in the same directory as the weblogic.xml file. In the web.xml file, uncomment the section starting with “Coherence*Web parameters”. This section contains all the required parameters for Coherence*Web. Here is a listing of this specific section:

<pre>
<!-- Coherence*Web parameters
-->
  <context-param>
    <description>With this set to "true", attributes that are deemed to be
      mutable (detected with a simple check) and which are accessed through a get,
      are deemed to be suspect in that they may have been changed in application code.
      Suspect attributes are treated as changed. Defaults to "false".</description>
    <param-name>coherence-enable-suspect-attributes</param-name>
    <param-value>true</param-value>
  </context-param>
  <context-param>
    <description>This value specifies a class name of the optional
      com.tangosol.coherence.servlet.HttpSessionCollection.AttributeScopeController
      interface implementation to use.</description>
    <param-name>coherence-scopecontroller-class</param-name>
    <param-
value>com.tangosol.coherence.servlet.AbstractHttpSessionCollection$GlobalScopeController</param-
value>
  </context-param>
  <context-param>
    <description>Specifies a cache delegator class that is responsible for manipulating
     (getting, putting, or deleting) data in the distributed cache
    </description>
    <param-name>coherence-cache-delegator-class</param-name>
    <param-value>com.tangosol.coherence.servlet.LocalSessionCacheDelegator</param-value>
  </context-param>
  <context-param>
    <description>This value, if set to true, specifies that non-serializable attributes
      should be preserved as local ones. This parameter requires a load balancer to
      be present to retrieve non-serializable attributes for a session
    </description>
    <param-name>coherence-preserve-attributes</param-name>
    <param-value>true</param-value>
  </context-param>
  <context-param>
    <description>
</pre>
</td>
</tr>
</table>

		`This value specifies a class name of the`
		`com.tangosol.coherence.servlet.HttpSessionCollection`
		`$SessionDistributionController interface implementation`
		`</description>`
		`<param-name>coherence-distributioncontroller-class</param-name>`
		`<param-`
		`value>com.tangosol.coherence.servlet.AbstractHttpSessionCollection$HybridController</param-`
		`value>`
		`</context-param>`

5	All PIA web servers, psadm2	Add the additional parameters to the PIA setEnv.sh located in $PS_CFG_HOME/webserver/HR92U033/bin. There is a comment in this file for Coherence. Here is a snippet for the setEnv.sh file where we have added the PIA web startup parameters:
		`# Coherence*Web related parameters`
		`# COHERENCE_PARAMETERS=-Dtangosol.coherence.session.localstorage=false`
		`# Optionally add -Dtangosol.coherence.override=file:OVERRIDE_FILE_PATH`
		`COHERENCE_PARAMETERS="-`
		`Dtangosol.coherence.override=file:/peoplesoft/local/ps_config/coherence/config/tangosol-`
		`coherence-override.xml -Dtangosol.coherence.session.localstorage=false -`
		`Djava.net.preferIPv4Stack=true"`
		`# Example: JAVA_OPTIONS_LINUX="$JAVA_OPTIONS_LINUX $COHERENCE_PARAMETERS"`
		`# export JAVA_OPTIONS_LINUX`
		`JAVA_OPTIONS_LINUX="$JAVA_OPTIONS_LINUX $COHERENCE_PARAMETERS"`
		`export JAVA_OPTIONS_LINUX`
		`# Refer to Coherence*Web documentation`
		Notes:
		• There are hyphens immediately before each parameter (word wrap may split the display)
		• The COHERENCE_PARAMETERS must be defined before exporting JAVA_OPTION_LINUX
		• The Dtangosol.coherence.override=file:/peoplesoft/local/ps_config/coherence/config/tangosol-coherence-override.xml must be specified so that the PIA WebLogic server can properly register as a Coherence*Web client.

6	All PIA web servers, psadm2	Restart the PIA web server, using the custom scripts you created as described in PeopleSoft Startup / Shutdown Scripts:
		`$ stopWS.sh`
		`$ startWS.sh`

Table 31: PIA Web configuration for Coherence*Web

As each PIA web server is restarted, they will join the Coherence*Web cache server cluster as clients. Each cache server log should show the WebLogic servers establishing a connection and joining the cluster with a role of "WeblogicServer". Here is a snippet from the log:

```
2021-10-28 23:27:01.237/160.748 Oracle Coherence GE 12.2.1.3.0 <D6> (thread=Cluster, member=1):
TcpRing connected to Member(Id=3, Timestamp=2021-10-28 23:27:01.044, Address=10.0.103.85:8089,
MachineId=10879, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-psft-hcm-
web01,process:16067, Role=WeblogicServer)
…
2021-10-28 23:28:14.958/234.469 Oracle Coherence GE 12.2.1.3.0 <D6> (thread=Cluster, member=1):
TcpRing connecting to Member(Id=4, Timestamp=2021-10-28 23:28:14.768, Address=10.0.103.203:8089,
MachineId=10880, Location=site:appprivatesu.ebscloudmaavcn.oraclevcn.com,machine:iad-psft-hcm-
web02,process:14708, Role=WeblogicServer)
```

The cache server logs on each PIA web server should show something like the above. The PeopleSoft deployment now has web server resiliency.

8.6.4 PIA Web Resiliency Testing

To test and validate that Coherence*Web is providing web resiliency, follow these steps:

1. Shut down all but one PIA web server
2. Using a web browser, log in to the PeopleSoft application
3. Navigate around, do some work. Start a transaction, but do not complete it.
4. Start up a second PIA web server and allow it to completely come up. Check the load balancer via the OCI console, to ensure the status of the second PIA web server shows "OK".
5. Shut down the PIA web server that was left up in step 1 above.
6. In your web browser connected to the PeopleSoft application, continue your transaction. You may detect a slight pause, but you should not receive an error, need to log in again, or lose any work.

If an error such as "Unauthorized token" occurs, review the following:

- The session cookie and domain names must be the same for all PIA web server weblogic.xml files.
- The OCI load balancer backend set must be configured with "Enable application cookie persistence" and the cookie name must be specified.
- All Coherence*Web cache servers must be able to see each other and have formed a cluster.
- All PIA web servers must have joined the cache server cluster as WeblogicServer clients.
- Double-check networking configurations - security list ingress/egress rules, firewall-cmd commands, etc.

8.7 Convert Standby Database on ExaDB-D Back to Physical Standby

Now that we have completed all configurations and have a fully running PeopleSoft application, the snapshot standby can be converted back to a physical standby and allowed to be synchronized with production. All outstanding redo will be applied. Follow these steps:

1. On each compute instance of the PIA and Application tier, shut down the Peoplesoft application using the scripts in PeopleSoft Startup / Shutdown Scripts.

2. As the user oracle, in the home directory, source the standby database environment.

```
$ . ./CDBHCM.env
```

3. Start Data Guard Broker and enter the SYS password.

```
$ dgmgrl
```

4. As the oracle user on any of the ExaDB-D domUs, convert the snapshot standby database back to a physical standby using Data Guard Broker:

```
DGMGRL> convert database CDBHCM_iad1dx to physical standby
Converting database "CDBHCM_iad1dx" to a Physical Standby database, please wait...
Database "CDBHCM_iad1dx" converted successfully
```

5. Wait a few minutes to allow the managed recover process (MRP) to start applying redo. Then while still logged into Data Guard Broker, show the configuration:

```
DGMGRL> show configuration lag

Configuration - ZDM_ CDBHCM_iad1dx

  Protection Mode: MaxPerformance
  Members:
   CDBHCM_sca6dp - Primary database
   CDBHCM_iad1dx  - Physical standby database
                     Transport Lag:      0 seconds (computed 2 seconds ago)
                     Apply Lag:          10 minutes 14 seconds (computed 2 seconds ago)
      CDBHCM_phx5s - Physical standby database (receiving current redo)
                     Transport Lag:      1 second (computed 1 second ago)
                     Apply Lag:          2 seconds (computed 1 second ago)

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 60 seconds ago)
```

You may see warnings indicating that the transport lag has exceeded thresholds. This warning will go away after several minutes.

6. Continue to show the configuration until both Transport Lag and Apply Lag reach 0 minutes and as close to 0 seconds. At that point, the physical standby is within the lag thresholds and is caught up with the primary database.

8.8 PeopleSoft Application Tier Backups

OCI does not currently provide automation for backing up the entire VM of compute instances. This section will describe how to back your PeopleSoft application and web tiers up and place them into the OCI object store. We have three basic things to back up:

- The application configuration for at each node at each site, as the configuration is unique for every node
- The PeopleSoft and middle tier software. We back this up at each site, as it will be simpler to restore locally if needed.
- The report repository, which needs to be backed up frequently, as the state of this file system should be kept as close as possible to the state of the database itself.

8.8.1 Backup Prerequisites

The OCI user – not the OS user establishing the OCI session – will be performing the backups, using the OCI command line interface (CLI). To make this possible:

- The OCI CLI must be installed on each node at each site
- The OCI user must have access to the region-local object storage.

8.8.1.1 Install OCI CLI

Follow the instructions in the Quickstart documentation to install the OCI CLI, paying close attention to the instructions for the configuration file setup. Once installed and configured, query your tenancy name to ensure that you can access your tenancy and verify the OCI CLI is running properly:

```
$ oci os ns get
```

The output will be in JSON format and will resemble this:

```
{
  "data": "<your OCI tenancy name>"
}
```

If the command fails, address the error before proceeding. Likely causes of an error at this point:

- The OCI python script cannot open the configuration file. This is typically located in $HOME/.oci
- The fingerprint specified in the OCI configuration file does not match that in the private key pem file and the public key file stored in OCI.
- The pem file with the private key cannot be accessed by OCI.
- The OCI user's OCID may be incorrect.

8.8.1.2 Create object storage bucket

Create a storage bucket to hold your backups:

1. Change to the appropriate region
2. Select Storage form the navigation menu
3. Select Buckets. A table of object storage buckets will display.
4. Specify the compartment that will contain the object storage for backups.
5. Select Create Bucket.
6. Answer the questions for the Create Bucket dialog:
    a. Provide a name for the bucket
    b. Choose a bucket tier. The default of Standard is suitable for backups.
    c. Optional: select any of the additional checkbox items you require.
7. Choose the Encryption option. The default is Encrypt using Oracle managed keys.
8. Click Create.

If the bucket is created, the OCI user has appropriate access.

Alternatively, create an object storage bucket using the OCI CLI:

```
$  oci os bucket create -ns <OCI Tenancy name> --name PSFT_APP_TIER_BACKUPS_20230403 --
compartment-id <Compartment OCID> --storage-tier Standard
```

8.8.2 Back Up Application Tier

With OCI in place, you can back your middle tiers up to region-local object storage. There are three basic types of file systems in the middle tiers:

- Shared homes, which hold the application code used by the middle tiers to run the application. These directories change when the application is patched or upgraded.
- Configuration files, which are fairly static but can change occasionally during system operation.
- Report repository, which changes frequently and reflects state also held in the database.

8.8.2.1 Back Up Shared Homes

Since all compute instances share access to a single copy of the application home directories, one backup needs to be taken to protect the resource. This backup should be taken each time the software is updated. It is also best practice to schedule a backup of this resource on a regular basis.

To take a manual backup:

1. Log in to a compute instance hosting the application or web tier as psadm2.
2. Use TAR to back up the pt subdirectory. Our commands were:

   ```
   $ cd /u01/app/psft
   $ time tar -zcvf PSFT_HMC92_APP_20230403.tgz pt | tee -atar_PSFT_HCM92_APP_20230403.log
   ```

3. When the TAR command has completed, upload the tarfile and the log file to object storage:

   ```
   $ time oci os object put -ns <Tenancy name> -bucket-name PSFT_APP_TIER_BACKUPS_20230403 -file
   PSFT_HCM92_APP_20230403.tgz
   $ time oci os object put -ns <Tenancy name> -bucket-name PSFT_APP_TIER_BACKUPS_20230403 -file
   tar_PSFT_HCM92_APP_20230403.log
   ```

4. Log in to the OCI console and verify the backups were successfully uploaded to object storage.

8.8.2.2 Back Up PeopleSoft Configuration Files

As PeopleSoft middle tier configurations are node-specific, you will need to back up each instance's PS_CFG_HOME now, to save the work you've completed installing the application in OCI. You should also schedule a backup of this resource on a regular basis, as you may adjust these configurations during normal operations.

To take a manual backup, on each middle tier compute instance:

1. Log in to the compute instance as user psadm2.
2. Zip up the PS_CFG_HOME:

   ```
   $ zip -r backup_ps_cfg_home_<instance name>_<date>.zip $PS_CFG_HOME
   ```

3. Upload the zip file to object storage:

   ```
   $ oci os object put -ns <Tenancy name> -bucket-name PSFT_APP_TIER_BACKUPS_20230403 -file
   backup_ps_cfg_home_<instance name>_<date>.zip
   ```

When all your middle tier configuration files are backed up, log in to the OCI console and verify the backups were successfully uploaded to object storage.

8.8.2.3 Back Up Report Repository

The report repository changes constantly as the system is in operation. In a later section, we will configure frequent replication of the report repository contents to the DR site so that on switchover or failover the data there will be as current as possible.

We also want to back the report repository contents up once a day.

The report repository is shared by all compute instances.

To take a manual backup, follow these steps, substituting today's date for <YYYYMMDD>:

1. Log in to a compute instance hosting the application or web tier as psadm2.
2. Use TAR to back up the report repository subdirectory. Our commands were:

```
$ cd /u02/app/psft/ps/report_repository
$ time tar -zcvf PSFT_HMC92_REPORTS_BACKUPS_<YYYYMMDD>.tgz pt | tee -a
tar_PSFT_HCM92_REPORTS_BACKUPS_<YYYYMMFF>.log
```

3. When the TAR command has completed, upload the tarfile and the log file to object storage:

```
$ time oci os object put -ns <Tenancy name> -bucket-name PSFT_REPORTS_BACKUPS_<YYYYMMDD> -
file PSFT_HCM92_REPORTS_BACKUPS_<YYYYMMDD>.tgz
$ time oci os object put -ns <Tenancy name> -bucket-name PSFT_REPORTS_BACKUPS_<YYYYMMDD> -
file tar_PSFT_HCM92_REPORTS_BACKUPS_<YYYYMMDD>.log
```

These steps automate backups of the report repository once a day at 02:00 am:

1. Create a script that contains the TAR and OCI cli commands called psft_reports_backup.sh located in your custom script directory:

```
#!/bin/bash

CURRENT_DATE=$( date +"%d-%b-%Y_%T" )

# Create the TAR backup file
cd /u02/app/psft/ps/report_repository
time tar -zcvf PSFT_HMC92_REPORTS_BACKUPS_${CURRENT_DATE}.tgz out | tee -a
tar_PSFT_HCM92_REPORTS_BACKUPS_${CURRENT_DATE}.log

# Upload the files.
time oci os object put -ns <Tenancy name> -bucket-name PSFT_REPORTS_BACKUPS -file
PSFT_HCM92_REPORTS_BACKUPS_${CURRENT_DATE}.tgz

time oci os object put -ns <Tenancy name> -bucket-name PSFT_REPORTS_BACKUPS -file
tar_PSFT_HCM92_REPORTS_BACKUPS_${CURRENT_DATE}.log
```

2. As root, add an entry into /etc/crontab that will run the above script as psadm2 at 02:00am each night:

```
0  2  *  * *   psadm2  <script directory>/psft_reports_backup.sh
```

8.8.3 Restoring Backups from Object Storage

You can use the OCI CLI to restore files from object storage. You would retrieve each file specifically. For example, to retrieve the shared software directories:

```
$ time oci os object get -ns <Tenancy name> --bucket-name PSFT_APP_TIER_BACKUPS_20230403 --file
PSFT_HCM92_APP_20230403.tgz –name PSFT_HCM92_APP_20230403.tgz
```

This command gives you the opportunity to rename your file when it is restored, via the -name parameter. In our example, we kept the same name.

8.9 Provision and Configure Cascade Standby Middle Tiers

In OCI, we use Data Guard to keep the database at the DR site synchronized with production. But what about the application tier? The service level agreement and tolerance for data loss will drive what approach is needed. Assuming that the target for data loss on the middle tier is similar to the database, this project configures the rsync utility to frequently replicate changes to the PeopleSoft report repository and process scheduler job logs to the DR site.

In addition, rsync is used to replicate the PeopleSoft installation directories post application patching.

In Set Up the Future Secondary Database, we set up a second standby database, one that will be our OCI-based DR standby after we switch operations to OCI. For now, it is configured as a cascade physical standby.

The steps for setting up the PeopleSoft application middle tiers at the cascade standby site are nearly identical to those for setting up the middle tiers at the future primary site. The following will provide guidance referencing back to each section:

PeopleSoft Application and Web Tier Setup

- It is recommended (not required) that the same number of middle tiers be provisioned as at the OCI primary.
- In section 8.1.2, it is *required* to use the same OS group and user name, with the same gid and uid respectively, as that used at the OCI primary.
- Follow the steps in sections 8.1.3 through 8.1.6.

Copy the Application Software

- Pick a middle tier from the OCI primary site and one from the DR site. Configure user equivalence between these two middle tier compute instances, using the psadm2 OS user.
- Ensure that the base directory structure is the same, e.g., /u01/app/psft/pt
- Use rsync to copy the entire PeopleSoft installation:

  ```
  $ rsync -avzh --progress /u01/app/psft/pt/  psadm2@<TARGET HOST IP
  Address>:/u01/app/psft/pt/
  ```

Configure PeopleSoft

- Convert the cascade standby database to a snapshot standby by following the steps in section 8.3.1.
- In section 8.3.2 when configuring the database connection TNS connect string, ensure you use the SCAN name that the cascade database is registered on.
- Follow all steps in sections 8.3.3 and 8.3.4 to complete the PeopleSoft application, process scheduler, and PIA web server configurations.

PeopleSoft Full Stack Test in OCI

- Because SSL is already configured, we will wait for our full stack testing until after configuring the load balancer.
- With that configuration in place, do some simple tests of your future DR site as described in Section 8.4.

OCI Load Balancer Provisioning and Configuration

- In section 8.5.1, change the PIA web server domains at the future DR site to have the same cookie name and network domain specified in the weblogic.xml file found here: $PS_CFG_HOME/webserv/<web server domain>/applications/peoplesoft/PORTAL.war/WEB-INF.

  The PSJSESSION cookie name can be different for each site, but must be the same on all PIA web servers at each site. For example, in this project, the PIA web servers deployed in the Ashburn (IAD) region have the following entries:

  ```
    <cookie-name>iad-hcm-8080-PORTAL-PSJSESSIONID</cookie-name>
    <cookie-domain>.appprivatesu.ebscloudmaavcn.oraclevcn.com</cookie-domain>
  ```

  And the PIA web servers deployed in the Phoenix (PHX) have the following entries:

  ```
  <cookie-name>phx-psft-hcm-app01-8080-PORTAL-PSJSESSIONID</cookie-name>

  <cookie-domain>.appprivad1.maacloud2vcn.oraclevcn.com</cookie-domain>
  ```

  The network domain must match that of the subnet of the PIA web servers.

- In section 8.5.2, for the OCI DR site, provision an OCI load balancer (LBaaS) like the one provisioned at the OCI primary. If you created a hostname to be associated with a listener during provisioning, ensure you create an SSL bundle based on that hostname.
- In section 8.5.3, since the cascade standby database is part of the Data Guard configuration, all the web profile information stored in the database will be present at the cascade standby database. Therefore, you only need to complete step 4 to set the web profile for SSL termination at the load balancer and step 5 to restart the PIA web servers.

PIA Web Resilience with Coherence*Web

- In section 8.6.1, you can copy the Coherence*Web configuration to the DR site. Change the "well-known-addresses" to those for the compute instances hosting the PIA web servers you are configuring at the DR site.
- Follow the steps in section 8.6.2 to provide both ingress and egress rules to allow Coherence*Web to traverse the subnet hosting the standby PIA web servers.
- Follow the steps in section 8.6.3 to configure the PIA web servers as clients of the Coherence*We cache cluster.
- Perform both the full stack tests from section 8.4 and the resilience testing from section 8.6.4 to ensure all is functioning properly.

Once the above configuration and testing have been completed, shut the PeopleSoft application server, process scheduler, and PIA web servers down. Convert the cascade standby from a snapshot standby back to a physical standby, as described in section 8.7. You can back the middle tier file systems up at the DR site to the region-local object store, following the steps in section 8.8. Backups of the cascade database cannot be performed until it assumes the primary role.

8.10 PeopleSoft and Oracle Active Data Guard

PeopleSoft PeopleTools versions 8.52 and higher support Oracle Active Data Guard, which enables offloading queries to a physical standby. You need the following to enable PeopleTools to support Oracle Active Data Guard:

- A physical standby database that has Oracle Active Data Guard enabled
- A database service that can be started on the Oracle Active Data Guard database instance
- A secondary Access ID created in PeopleSoft
- An additional database schema that will be associated with the secondary Access ID.
- A database link that uses a service that *only* runs on the primary

The procedure for enabling PeopleTools support for Oracle Active Data Guard is documented in Implementing Oracle Active Data Guard.

If your on-premises PeopleSoft implementation is already configured to use Oracle Active Data Guard for offloading queries, then you must make sure that the configuration for the application server and process scheduler are carried over to the first and second OCI deployments. The database will already have the schema user associated with the PeopleSoft secondary access ID. In our case, this additional schema is called PSFTADG2.

If you choose to newly implement Active Data Guard support for offloading queries, the steps in this chapter complement those described in Implementing Oracle Active Data Guard, and are designed to provide MAA best practices.

Keep the following important items in mind when implementing support for Active Data Guard:

- The PeopleSoft application server domains at both the primary and secondary sites must be configured to support Oracle Active Data Guard for query offload to work after a switchover or failover.

- The PSFTADG2 user will need its own database link to access the standby. When manually creating the new database link, make sure the database link name matches the name of the target database, including the DB_DOMAIN if set.

- IMPORTANT: once the PeopleTools application domain server is configured to support Oracle Active Data Guard, the PeopleSoft application will *not* start if the PSQUERY service is not available. If the standby database is not available, the PSQUERY service must be started on the primary. It can be relocated back to the standby when the standby is again available without restarting the application.

NOTE: If you plan to open the standby database as a snapshot standby for testing, you must *first* relocate the PSQUERY service to the primary.

8.10.1 Configure Primary and Standby Database Servers for Active Data Guard

The primary database server needs to access a service at the standby database via database link. We start by creating that database service and adding the tnsnames.ora entries to be able to resolve the network address of the standby. We will be:

- Adding a role-based database service to the primary and secondary regions

- Creating a tnsnames.ora “include file”, or ifile, to allow reports running on the Active Data Guard database to connect back to the primary, to update run data in the database

- Adding a line to the bottom of your database tnsnames.ora files so the ifile is included in the tnsnames definition

First, add the role-based database service PSQUERY at both the primary and secondary regions, to run only when the database is fulfilling the PHYSICAL_STANDBY role. These are the services we added in our environment:

At the primary:

```
$ srvctl add service -db <primary DB unique name> -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" failovermethod BASIC -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

At the standby:

```
$ srvctl add service -db <standby DB unique name> -pdb HR92U033 -service PSQUERY -preferred
"CDBHCM1,CDBHCM2" failovermethod BASIC -failovertype SELECT -notification TRUE -role
PHYSICAL_STANDBY -failoverretry 10 -failoverdelay 3
```

Second, create an "include file", or ifile, that contains the TNS connect string to be used by reports running at the Active Data Guard standby database to record information about the jobs in the primary database.

Place the ifile in the $TNS_ADMIN directory on each RAC database node at both primary and standby sites.

This is the TNS connect string we placed in our ifile for this purpose. We named the ifile: tns_ps_adg.ora:

```
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 = iadexadb-bw5wn-
scan.exadbprivate.ebscloudmaavcn.oraclevcn.com)(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 = phxexadb-krppw-
scan.dbprivateexa.maacloud2vcn.oraclevcn.com)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = HR92U033_ONLINE)
       )
    )
 )
```

NOTE: Do not place this TNS connect string alias onto any of the middle tiers. This connect string is only used by the database link created in a later step on the database servers.

Third, add an IFILE directive at the bottom of the $TNS_ADMIN/tnsnames.ora file, so the ifile created above is included in the tnsnames.ora definitions: This is the IFILE directive we added to our tnsnames.ora file:

IFILE=<TNS_ADMIN full path>/ tns_ps_adg.ora

8.10.2Configure Primary and Standby Application Servers for Active Data Guard

Configuring the application tier servers requires adding a TNS connect string alias to the new PSQUERY physical standby service. This connect string must be placed in all tnsnames.ora files accessed by the application domain servers and process schedulers, and the alias name must be used for the *StandbyDBName* in the PSADMIN utility.

This is the entry we added to the middle tier tnsnames.ora files, using PSFTADG as the alias:

```
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 = iadexadb-bw5wn-
scan.ebsexadbprivate.ebscloudmaavcn.oraclevcn.com)(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 = phxexadb-krppw-
scan.ebsdbprivateexa.maacloud2vcn.oraclevcn.com)(PORT = 1521))
       )
        (CONNECT_DATA =
           (SERVER = DEDICATED)
           (SERVICE_NAME = PSQUERY)
       )
    )
 )
```

8.10.3Database Updates for Active Data Guard

To complete the configuration, we will update the database contents to hold both a pointer used by the PeopleSoft application at the standby site to locate the primary database, and the database link used for that purpose. The data inserted at the primary will of course be propagated to the standby by Oracle Data Guard.

First insert a new row into the PS.PSDBOWNER table on the primary database , so the application servers can authenticate with the Oracle Active Data Guard standby database. To do this, log in to one of the ExaDB-D database servers as the oracle OS user and source the database environment (CDBHCM.env). Then start a SQL*Plus session and insert the new row.

Here is an example using our PDB name HR92U033:

```
sqlplus / as sysdba

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

COMMIT;
```

Finally, the database link can be created. The document Implementing Oracle Active Data Guard provides steps to create a database link on the primary database. Using SQL*Plus, connect to the second database schema (PSFTADG2) and create the database link. See this example from our implementation:

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

Note this uses the TNS alias 'HR92U033_PRIMARY’, a service that only runs when the database is in the primary role.

8.11 Report Repository and Process Scheduler Job Log Replication

The contents of the report repository and the process scheduler logs are file system artifacts created during normal PeopleSoft application operations. The report repository contents and the logs correspond to work done in the database, and need to be as close to in sync with the database as possible. These file system artifacts need to be replicated to the DR site just as Data Guard replicates the database.

The PeopleSoft report repository is typically a shared file system directory that all process scheduler and PIA web tier servers can access. In this project, it is located on FSS at:

```
/u02/app/psft/ps/report_repository
```

The report repository is defined in the PeopleSoft PIA configuration.properties file:

```
PS_CFG_HOME/webserv/<PIA web domain>/applications/peoplesoft/PORTAL.war/WEB-
INF/psftdocs/ps/configuration.properties
```

The report repository location is set when the PIA web servers are configured, and is specified in this line In the configuration.properties file:

```
ReportRepositoryPath=/u02/app/psft/ps/report_repository
```

The process scheduler logs are specific to the compute instance, as detailed in PeopleSoft Application and Process Scheduler Domains. The directories for the process scheduler logs in this project are:

/u02/app/psft/ps/log_output/node1/HR92U033

/u02/app/psft/ps/log_output/node2/HR92U033

8.11.1 Prerequisites

The scripts use the OCI command line interface, which was installed earlier in this flow.

The scripts also need to log in to the database without exposing any passwords, either in their body or on the command line. For this, a region-local KMS OCI Vault cloud service will be provisioned at each site. It will hold the database administration (SYS) password as a *secret.*

We recommend that your security administrators review the policies, key rotation, and secret bundle functions of the OCI Vault Service. For an overview and details on creating and managing Vaults, see OCI Vaults.

8.11.2 Implementing Replication

Establishing the replication process requires scripts to perform the replication and a cron job that defines the schedule and frequency the scripts run.

We wrote these simple scripts to frequently push these files to the standby:

- rsync_psft.sh
- get_site_role.sh
- enable_psft_rsync.sh
- disable_psft_rsync.sh

See rsync Log and Report Files for the scripts.

The rsync_psft.sh script is executed as a cron job. It will run at both the primary and the secondary site once every five minutes. When it starts, it checks three things then exits without doing any work if any of the following are true:

- Is the rsync script disabled? If so, exit.
- Is the database at this site in primary or standby role? If standby, exit.
- Is there an earlier instantiation of this script still running? If so, exit.

We coded the application startup scripts to also run the enable_psft_rsync.sh script. It is commented out as delivered, as it will fail until the standby is fully configured.

We coded the application shutdown script to call the rsync_psft.sh script one last time after shutting down the application and batch servers, then run the disable_psft_rsync.sh script. These are commented out as delivered, as they will fail until the standby is fully configured.

Because the report repository and process scheduler job logs are implemented as shared file systems, we are running the rsync replication process on only one node at each site. To configure the rsync scripts, follow these steps:

1. Configure user equivalence between a pair of compute instances hosting the process scheduler, one from each site, for the psadm2 OS user. We used the same pair we had configured with user equivalence when setting up the secondary site middle tier servers.
2. Place all three scripts into a shared directory on FSS. Modify the scripts for your environment.
3. Give the scripts execute permission for the psadm2 user:

```
$ chmod u+x rsync_psft.sh
$ chmod u+x get_site_role.sh
$ chmod u+x enable_psft_rsync.sh
$ chmod u+x disable_osft_rsync.sh
```

4. Test the scripts to ensure they all work as expected.
5. Uncomment the execution of the three scripts in the application startup and shutdown scripts.
6. If the application is running, on the primary compute instance, execute the enable_psft.sh

```
$ ./enable_psft_rsync.sh
```

7. If the application is running, on the standby compute instance, execute the disable_psft.sh

```
$ ./disable_psft_rsync.sh
```

8. As root on each compute instance, add an entry into /etc/crontab that will run the rsync_psft.sh script. The entry we used is:

```
*/5 * * * * psadm2 /u01/app/psft/pt/custom_admin_scripts/rsync_psft.sh
```

Monitor the log file at both sites. Each log should be written to every five minutes. The log file at the primary site should show rsync is running. The log file at the standby site should show that rsync is disabled.

9 Switch Over to Cloud

When your functional and performance tests show your new environment is ready, you can schedule and perform switching production operations to OCI. There are three parts to this process:

- Shut down the on-premises production PeopleSoft application
- Perform a Data Guard switchover role change, making the physical standby on the ExaDB-D in OCI the primary and the on-premises database a physical standby
- Start the PeopleSoft applications in OCI at the new primary site

These three points are covered in the first section below, then we take a backup of the primary OCI environment.

9.1 Full Stack Switchover

1. Shut down on-premises production PeopleSoft applications

   On the on-premises systems, shut the PeopleSoft PIA, batch, and application servers down. Let them shut down completely and cleanly.

2. Log in to an OCI database server at the primary site as the oracle user. In the home directory, source the standby database environment:

```
$ . ./CDBHCM.env
```

3. Start Data Guard Broker as SYS:

```
$ dgmgrl
DGMGRL> connect sys/<password>
```

4. Using Data Guard Broker, verify the database is ready for switchover:

```
DGMGRL> validate database CDBHCM_iad1dx
  Database Role:     Physical standby database
  Primary Database:  CDBHCM_sca6dp
  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
  Managed by Clusterware:
    CDBHCM_sca6dp:  YES
    CDBHCM_iad1dx:  YES
```

5. Perform the switchover:

```
DGMGRL> switchover to CDBHCM_iad1dx;
Performing switchover NOW, please wait...
New primary database " CDBHCM_iad1dx" is opening...
Oracle Clusterware is restarting database " CDBHCM_sca6dp" ...
Connected to " CDBHCM_sca6dp"
Connected to " CDBHCM_sca6dp"
Switchover succeeded, new primary is "CDBHCM_iad1dx"
```

   Make sure the role-based database services – in our environment, HR92U033_BATCH and HR92U033_ONLINE – have started. If they have not started, start them manually. In our environment:

```
$ srvctl start service -db CDBHCM_iad1dx -s HR92U033_BATCH
$ srvctl start service -db CDBHCM_iad1dx -s HR92U033_ONLINE
```

Note: You can start the next section – back up the now-primary OCI database – as soon as the database has taken the primary role.

6. Start the PeopleSoft application using the scripts in Application Domain Server and PIA Web Server. Use basic sanity checks to be sure the environment started correctly:

   - Check the logs on the application servers to make sure the PeopleSoft application and process scheduler domains start all Tuxedo processes without errors.
   - When the PIA web servers are started, use the OCI console to check the load balancer to be sure the backend sets are healthy, showing a green "OK" for each compute instance in the backend set.
   - Make sure you can log in to the application as an end user, using a browser.

9.2 Back Up the PeopleSoft OCI Primary Database

We recommend you take a full database backup right away once the switchover is complete and the database in OCI is in the primary role, to establish your baseline database backup in OCI.

To take a full backup now and set up automatic backups:

1. Log in to the OCI console
2. Select the compartment containing the ExaDB-D cluster
3. Click on the ExaDB-D cluster
4. Click on the database from the list of databases
5. Click "Backups" under Resources.
6. On the new page, click on Configure Automatic Backup.
7. A new form will be presented. On this form, check the checkbox labeled *Enable automatic backups.* Additional fields will be displayed. Use them to specify the following:
   - Backup retention period (7 days to 60 days)
   - The day of the week that a full backup is to be taken
   - The two-hour time window (UTC) on the day selected above that the full backup will be taken
   - The two-hour window (UTC) on each day that the incremental backups will be taken
   - If you want a full backup taken right away, select the checkbox labeled: *Take the first backup immediately*. This will start a full backup once the changes made in this form are saved.
   - Click on "Save Changes".

If you indicated the first backup should be taken immediately, OCI will start a full backup when you save your changes.

Backup activity from that point forward will take place during your specified backup window, starting the pattern of taking an incremental backup six days a week and a full backup once a week. Archived redo logs are automatically backed up once every 30 minutes. OCI-based backups will configure RMAN to use the cloud backup module in OCI.

As each backup completes, it will be listed in the table of backups on the database's Resources page.

9.3 Remove On-Premises Database

At this point, you have switched operations to your OCI environment, but you are still keeping your on-premises database in sync using Data Guard. While it is a good practice to do this for a brief period as a fallback, at some point you will stop redo replay to your old on-premises database in anticipation of dropping that environment.

These are the steps we followed in our environment:

1. Log in to an OCI database server at the primary site as the oracle OS user.
2. Source the environment

   ```
   $ ./CDBHCM.env
   ```

3. Log in to Data Guard Broker

   ```
   $ dgmgrl sys/<sys password>
   ```

4. As we will no longer need a cascade standby configuration, we will remove the *redoroutes* for all databases. In our environment, we issued these commands:

   ```
   DGMGRL> EDIT DATABASE CDBHCM_sca6dp reset property RedoRoutes;
   DGMGRL> EDIT DATABASE CDBHCM_iad1dx reset property RedoRoutes;
   DGMGRL> EDIT DATABASE CDBHCM_phx5s reset property RedoRoutes;
   ```

5. Remove the on-premises database CDBHCM_sca6dp from the configuration

   ```
   DGMGRL> REMOVE DATABASE CDBHCM_sca6dp;
   ```

Once the above steps are completed, the current primary will send its redo to the remaining standby database. The on-premises database can now be shut down.

10 Secondary Region Site Testing

OCI provides the ability to start the full PeopleSoft stack at the standby site for various types of testing without impacting the primary region, while the production PeopleSoft application is running. This can be used to test patches and various configuration changes, for a final validation of core changes before promoting the change to production.

An important prerequisite that should be met prior to conducting any DR site testing is ensuring that the URL used for standby testing is different from that of production. There are several ways to achieve this, but one way is to create an additional hostname alias associated with a listener within the load balancer at each region. Load Balancer as a Service (LBaaS) describes the various attributes of the OCI load balancer, including how to create a hostname alias. We recommend that a hostname that is clearly for testing – e.g., *psfthcm-test* – be created and associated with the LBaaS listener. You will also need to upload a signed SSL certificate for the hostname you create and associate with the listener. This should be done at each region, since either site might serve as the standby for the other.

Note: Within the PeopleSoft application, there are different page styles that can be selected to change the theme and banner information. We recommend that you use this feature when testing PeopleSoft at your standby. This will help anyone logging in to the application running at the DR site know that this is not production, and that all their changes will be lost when the site is reverted back to standard standby operation. For more details on changing PeopleSoft themes and downloading themes from Github, please go to: https://github.com/psadmin-io/io-styles .

Once your test URL is ready and you have chosen a “Test” theme, you are ready to enable the PeopleSoft application to run at the standby. These tasks must be completed, all at the secondary site:

1. If Active Data Guard support is configured (see Configure Primary and Standby Database Servers for Active Data Guard), then first relocate the ADG service for PeopleSoft (PSQUERY) from the standby to the primary database.
2. Convert the physical standby database to a snapshot standby database
3. Start database services for PeopleSoft
4. Create an FSS *snapshot* of the shared file systems that hold the PeopleSoft software and the report repository
5. Create an FSS *clone* file system using the FSS snapshot
6. Mount the cloned file system on each of the compute instances hosting the PeopleSoft application
7. Start the PeopleSoft application servers and all PIA web servers, but do not start the process scheduler
8. Adjust the PeopleSoft display theme. Verify the change took place
9. IMPORTANT: As necessary, place process scheduler jobs on hold or cancel them, to prevent inadvertently running jobs that might transmit data externally, such as sending electronic checks to third parties
10. When step 8 is complete, you can start the process scheduler

Site-and node-specific configurations defined in PS_CFG_HOME remain in place and will take effect. No configuration changes are needed.

Note that while your database is in snapshot standby mode, you will be accumulating but not applying redo from your primary database. Your rsync process is also continuing to keep the file system in sync with changes on the primary. This does not compromise the recovery point objective (RPO) of your standby site, should the primary site become unavailable. The recovery time objective (RTO) will be affected, since you will need to convert your snapshot standby database back to a normal standby and catch up on redo apply, as well as bring down the test application services and re-mount the now-production file system. To re-mount the file system, you will need to unmount the snapshot, disable the FSS snapshot, and re-mount the original file system.

The following table outlines the steps for site testing.

STEP #	NODE, USER	INSTRUCTIONS, COMMANDS

1	One ExaDB-D domU hosting standby database, oracle	Convert the physical standby to snapshot standby. Log in to one of the ExaDB-D domU as oracle, source your environment, then execute the following dbaascli command:
		`$ dbaascli dataguard convertStandby --dbname <DB_NAME> --standbyName <standby DB_UNIQUE_NAME> --standbyType snapshot`
		On our system, we issued this command:
		`$ dbaascli dataguard convertStandby --dbname CDBHCM --standbyName CDBHCM_phx5s --standbyType snapshot`

2	One ExaDB-D domU hosting standby database, oracle	If the role-based database services have not started after the database is converted to a snapshot standby, then they can be started with srvctl. In this project the following commands were executed:
		$ srvctl start service -db CDBHCM_phx5s -service HR92U033_ONLINE
		$ srvctl start service -db CDBHCM_phx5s -service HR92U033_BATCH

3	OCI console, administrator	Create a snapshot of the PeopleSoft install and report repository file systems, then create a clone file system on the snapshot. 1. Log in to the OCI console. From the main menu, navigate to Storage, File Systems under File Storage. 2. Change the region to where the standby is: Phoenix 3. Change to the appropriate compartment that contains the FSS file systems for PeopleSoft: psft_app_compartment 4. Click on the file system for PeopleSoft: PHX_PSFT_APP_INSTALL 5. Click on Snapshots 6. Create a new snapshot and give it a meaningful name, e.g., psft_snapshot 7. Once the snapshot is created, click on Clone 8. Specify the new file system name, network compartment, and compartment for the file system itself: a. Provide a meaningful name for the new cloned file system b. Select the appropriate compartment that the VCN and subnets reside in, e.g., network_compartment, then select the appropriate subnet, e.g., app_private_subnet_ad1. c. This new file system should be created in the same compartment that the compute instances reside in, e.g., psft_app_compartment d. Click Create. e. Once the cloned file system has been created, click on Create Export and give it a path such as : /export/psftapp-snapshot f. Either use an existing mount point target or create a new mount point target to mount the cloned file system on. g. Click Create Once the file system is created, OCI will provide a link to show mount commands by opening a window displaying the mount commands, the IP address, and the mount point.
4	OCI console, administrator	Repeat step 3 above for the second PeopleSoft file system: PHX_PSFT_APP_INTERFACE Ensure an export path is provided for example: /export/psftinterface-snapshot.

5	All PeopleSoft application and web compute instances, root	Mount the snapshot-based file system.
		If the PeopleSoft production file system (/u01 in this case) is mounted, it must be unmounted.
		`# umount /u01`
		`# umount /u02`
		Edit /etc/fstab. Comment out – do not remove – the entry for /u01, then add a new entry for the snapshot clone file system. Here is an example:
		`# PeopleSoft FSS mount.`
		`#10.10.106.35:/export/psftapp  /u01        nfs`
		`rw,rsize=131072,wsize=131072,bg,hard,timeo=600,nfsvers=3 0 0`
		`# Snapshot FS mounts`

		`10.10.106.19:/export/psftapp-snapshot  /u01        nfs`
		`rw,rsize=131072,wsize=131072,bg,hard,timeo=600,nfsvers=3 0 0`
		`10.10.106.19:/export/psftinterface-snapshot  /u02        nfs`
		`rw,rsize=131072,wsize=131072,bg,hard,timeo=600,nfsvers=3 0 0`
		Mount the snapshot file system:
		`# mount /u01`
		`# mount /u02`

Table 32: Configuration for site testing

CAUTION: As noted above, the process scheduler should not be started unless jobs that might inadvertently cause business impact are placed on hold, suspended, or cancelled. If this cannot be done, then do not start the process scheduler. For further details, please see the following from the PeopleSoft People Book:

Viewing the Status of Processes

Viewing the Status of Servers

Creating Server Definitions

Once the above steps are complete, the PeopleSoft application servers, process schedulers, Coherence*Web cache servers (if configured), and all PIA web servers can be started for the snapshot test environment using the scripts provided in Application Domain Serverand PIA Web Server.

The PeopleSoft application should be accessible through the OCI load balancer. For this project, the region is Phoenix with the load balancer listener hostname: psfthcm-test along with the VCN domain at the Phoenix region:

https://psfthcm-test.appprivatesu.ebscloudmaa2vcn.oraclevcn.com/psc/ps//?cmd=login&languageCd=ENG

Once all testing activities have been completed, return the secondary site back to its standby state:

1. Shut down all PeopleSoft application servers, the process scheduler, and all PIA web servers
2. Convert the snapshot standby database back to a physical standby
3. Unmount the FSS snapshot clone file system and remount the production file system
4. Optional: Remove the snapshot clone

11 Site Switchover

It is an MAA best practice to perform a full stack site switchover semi-annually, reversing the roles of the primary and secondary sites, to test switchover procedures as well as to catch and correct any unmanaged changes or other issues that might have occurred. You can also switch to the secondary site to continue providing services while the primary site is undergoing major maintenance.

You can use OCI to perform site switchover either step by step / manually or by scripting the steps into a single flow. In either case, you will use a combination of REST APIs for the database tier and scripts for the application and web tiers. This section will provide the manual steps.

We assume that the on-premises database has already been dropped from the Data Guard Broker configuration.

The steps in this section are designated Site 1 and Site 2, where Site 1 is originally the primary and Site 2 is originally the secondary. They switch roles during this exercise.

The high-level tasks for performing a switchover in OCI are:

Site 1:

1. Drain or place on hold batch jobs in the PeopleSoft Process Scheduler ahead of the planned switchover event.
2. Shut all PeopleSoft application servers, the process scheduler, and all PIA web servers down
3. Validate that the PeopleSoft database is ready for switchover
4. Perform Data Guard switchover
5. Perform FSS storage role reversal

Site 2:

6. Validate that role-based database services have been started
7. Start PeopleSoft application servers, the process scheduler, and all PIA web servers
8. Validate the status of backend servers on the new primary region load balance (green “OK”)
9. Validate you can log in to the PeopleSoft PIA

The following table provides the detailed steps for performing a full-stack PeopleSoft switchover. These examples use names from our test environment.

STEP #	SITE, NODE, USER	INSTRUCTIONS, COMMANDS
1	Site 1, each process scheduler server compute instance, psadm2	In preparation for site switchover, it may be necessary to shut down the process scheduler at some point ahead of the scheduled switchover. This will place any recurring and new jobs in “queued” status. When shutting down the process scheduler ahead of the scheduled switchover time, use the individual script stopPS.sh in stopPS.sh.. Do NOT use the wrapper script at this time. Step 4 below will execute the wrapper script as part of the actual switchover process. `$ stopPS.sh`

2	Site 1, one ExaDB-D domU, oracle	Validate that the standby is ready for switchover.
		To do this, log in to one of the primary ExaDB-D domUs hosting a PeopleSoft RAC instance, and become the oracle user. Then:
		Source the environment:
		`$ . ./CDBHCM.env`
		Start the Data Guard command line interface:
		`$ dgmgrl sys/<sys password>`
		`DGMGRL> show configuration lag`

		`Configuration - fsc`
		`  Protection Mode: MaxPerformance`
		`  Members:`
		`  CDBHCM_iad1dx - Primary database`
		`    CDBHCM_phx5s  - 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 35 seconds ago)`
		
		Validate the standby database:
		`DGMGRL> validate database 'CDBHCM_phx5s'`
		
		`  Database Role:     Physical standby database`
		`  Primary Database:  CDBHCM_iad1dx`
		
		`  Ready for Switchover:  Yes`
		`  Ready for Failover:    Yes (Primary Running)`
		
		`  Managed by Clusterware:`
		`    CDBHCM_iad1dx:  YES`
		`    CDBHCM_phx5s :  YES`
		The standby database is ready for switchover.

3	Site 1, PIA web server compute instances, psadm2	Shut the PIA web servers down. Log in to the PIA middle tier servers and become psadm2. Using the wrapper scripts from Wrapper Scripts to shut the PIA web servers and Coherence*Web cache servers down: `$ stopPSFTWEB.sh`
4	Site 1, application/process scheduler server compute instances, psadm2	Shut both the application server and process scheduler down. To do this, log in to the compute instances hosting the PeopleSoft application servers and process scheduler and become psadm2. Using the wrapper script from stopPSFTAPP.sh] `$ stopPSFTAPP.sh` Note that the first instance to run the stopPSFTAPP.sh script will perform one final rsync of the file systems after the rest of the application server and process scheduler domains are down, then will disable rsync. Use the SQL script in PeopleSoft Application and Process Scheduler Domains to monitor database sessions. Once all the stopPS scripts have completed, check the rsync log to verify that a final rsync was performed. Refer to rsync_psft.sh for the script.
5	Site 1, one ExaDB-D domU, oracle	Use the Data Guard Broker command line interface to perform the switchover. $ dgmgrl sys/<sys password> DGMGRL> switchover to CDBHCM_phx5s; Performing switchover NOW, please wait... New primary database " CDBHCM_phx5s" is opening... Oracle Clusterware is restarting database "CDBHCM_iad1dx" ... Connected to " CDBHCM_iad1dx" Connected to " CDBHCM_iad1dx" Switchover succeeded, new primary is " CDBHCM_phx5s"

6	Site 1, one ExaDB-D domU, oracle	Use the Data Guard Broker command line interface to monitor and verify the switchover succeeded.
		`$ dgmgrl sys/<sys password>`
		`DGMGRL> show configuration lag`
		`Configuration - fsc`
		`  Protection Mode: MaxPerformance`

<table>
<tr>
<td></td>
<td></td>
<td>

```
  Members:
  CDBHCM_phx5s  - Primary database
    CDBHCM_iad1dx - Physical standby database
                    Transport Lag:      0 seconds (computed 2 seconds ago)
                    Apply Lag:          0 seconds (computed 2 seconds ago)

Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 22 seconds ago)
```

</td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>Site 1, one ExaDB-D domU, oracle</td>
<td>

If Active Data Guard support is configured (see Configure Primary and Standby Database Servers for Active Data Guard), ensure that the ADG service for PeopleSoft (PSQUERY) has been started on the new standby database post switchover.

```
$ srvctl status service -db CDBHCM_iad1dx -s PSQUERY
Service PSQUERY is running on instance(s) CDBHCM1,CDBHCM2
```

This service should be running on all RAC instances.

NOTE: This service must be started before starting the process scheduler. Otherwise, the process scheduler will fail on startup.

</td>
</tr>
<tr>
<td><strong>8</strong></td>
<td>Site 2, all ExaDB-D domUs, oracle</td>
<td>

Verify that the role-based database services are up on the new primary. In our environment, we issued this command on each domU hosting a PeopleSoft RAC database instance:

```
$ srvctl status service -db CDBHCM_phx5s -s HR92U033_BATCH
Service HR92U033_BATCH is running on instance(s) CDBHCM1,CDBHCM2
$ srvctl status service -db CDBHCM_phx5s -s HR92U033_ONLINE
Service HR92U033_ONLINE is running on instance(s) CDBHCM1,CDBHCM2
```

These services should be running on all RAC nodes.

</td>
</tr>
<tr>
<td><strong>9</strong></td>
<td>Site 2, application/process scheduler server compute instances, psadm2</td>
<td>

Start the application server and process scheduler domains:

Log in to the compute instances hosting the PeopleSoft application servers and process scheduler and become psadm2. Using the scripts from Wrapper Scripts:

```
$ startPSFTAPP.sh
```

We used the query from PeopleSoft Application and Process Scheduler Domains to monitor the startup:

```
col service_name format a20
select a.inst_id,a.instance_name,b.service_name, count(*)
from gv$instance a, gv$session b
where a.inst_id = b.inst_id
and service_name not like 'SYS%'
group by a.inst_id,a.instance_name,b.service_name
order by 1

SQL> /

   INST_ID INSTANCE_NAME    SERVICE_NAME          COUNT(*)
---------- ---------------- ------------------- ----------
         1 CDBHCM1          CDBHCM_phx5s                 2
SQL> /

   INST_ID INSTANCE_NAME    SERVICE_NAME          COUNT(*)
---------- ---------------- ------------------- ----------
         1 CDBHCM1          CDBHCM_phx5s                 2
         1 CDBHCM1          HR92U033_BATCH               8
         1 CDBHCM1          HR92U033_ONLINE             52
         2 CDBHCM2          HR92U033_BATCH               7
         2 CDBHCM2          HR92U033_ONLINE             50
```

</td>
</tr>
<tr>
<td><strong>10</strong></td>
<td>Site 2, all PIA web server compute instances, psadm2</td>
<td>

Start web services.

If Coherence*Web is configured, you will first start the cache cluster on all compute instances that host the PIA web servers, then start the PIA web servers. In this project, one script is used to start both in the proper order.

Log in to the PIA web servers and become psadm2. Using the script from startPSFTAPP.sh, start the web

</td>
</tr>
</table>

		servers: `$ startPSFTWEB.sh`
11	Site 2 Region, OCI console, Tenancy administrator	Check the load balancer: Log in to the OCI console and change the region to your new primary (Phoenix in our case). Then from the main menu, select Networking, then Load Balancer. Select the appropriate compartment. Click Backend Set, then click Backends. Each backend should show "OK". It may take a few minutes after each PIA web server has been started.
12	PeopleSoft PIA web user	Attempt to log in to the PIA web server from a web browser. For this project, the URL is: https://psfthcm.appprivad1.maacloud2vcn.oraclevcn.com/psp/ps/EMPLOYEE/HRMS/?cmd=login

Table 33: PeopleSoft full stack switchover

When the above steps have successfully completed, production will be running at site 2.

12 Site Failover

A site failover is required if your infrastructure has suffered an unplanned event that causes the primary site to be unavailable and completely inaccessible for a duration of time that will negatively impact the business. In this scenario, the standby will assume the primary role.

A primary site can become unavailable for a variety of reasons, including but not limited to:

- Issues that might cause the primary database instances not to start such as failed or extensively corrupted media, or a flawed OS or firmware upgrade
- Infrastructure failure such as a full power or cooling system outage within the OCI region infrastructure
- Complete network failures
- Natural disasters such as earthquakes, fires, and floods.

While the above unplanned events are rare, they can and do occur.

12.1 Perform Site Failover

As a true failover is disruptive and may result in some small loss of data, we will test in a TEST environment. We will initiate the test by forcing an abort of all database RAC instances on the primary. Do not conduct this test on a production environment:

```
$ srvctl stop database -db CDBHCM_phx5s -stopoption abort
```

From this point on, our primary is assumed (simulated) to be completely unavailable. We will make our secondary region our new primary site. The table below shows the steps using our test implementation. All steps in this table are performed at the secondary site (new primary).

<table>
<tr><th>STEP #</th><th>NODE, USER</th><th>INSTRUCTIONS, COMMANDS</th></tr>
<tr><td>1</td><td>One ExaDB-D domU, oracle</td><td>At the secondary site, log in to any one of the ExaDB-D domUs, become the oracle OS user, and invoke the Data Guard Broker command line interface. Notice the error:
<pre>
$ dgmgrl sys/<sys password>
DGMGRL> show configuration lag

Configuration - fsc

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_phx5s - Primary database
    Error: ORA-12514: TNS:listener does not currently know of service requested in connect descriptor
    CDBHCM_iad1dx - Physical standby database
                    Transport Lag: 0 seconds (computed 18 seconds ago)
                    Apply Lag: 0 seconds (computed 18 seconds ago)

Fast-Start Failover: Disabled

Configuration Status:
</pre>
ERROR   (status updated 0 seconds ago)</td></tr>
<tr><td>2</td><td>One ExaDB-D domU at secondary site, oracle</td><td>Perform a failover using the Data Guard Broker command line interface.
DGMGRL> failover to CDBHCM_iad1dx;</td></tr>
<tr><td>3</td><td>One middle tier, failed primary site,</td><td>The application and web tiers may still be functional, but the application and process scheduler processes will begin to fail trying to connect to the failed database. This will cause the rsync script to stop performing rsync.
If the primary middle tiers, including the shared file system (FSS), are still intact, then we need to manually</td></tr>
</table>

	psadm2	perform a “forced” rsync from the failed primary site to the DR site. To do so with the sample script in the scripts directory rsync_psft.sh, as user psadm2: `$ rsync_psft.sh <path to file system/parameter file> -f` Example: $ rsync_psft.sh $SCRIPT_DIR/fs1 -fIf the rsync script is disabled, the -f will prompt to continue with a forced rsync. A forced rsync will not consult the database to determine the site’s role, then will perform the requested rsync. This should only be done when forcing a final refresh during a site failover. Using the FORCE option will be logged. Monitor the rsync process’s log to be sure the process completes successfully.
4	One middle tier, new primary site, psadm2	If the site failure is complete and a final rsync process cannot be run, disable rsync at the new primary by running the disable_psft_rsync.sh script.

5	One ExaDB-D domU, oracle	If Active Data Guard support is configured (see Configure Primary and Standby Database Servers for Active Data Guard), ensure that the ADG service for PeopleSoft (PSQUERY) has been started on the new primary database.
		`$ srvctl status service -db CDBHCM_iad1dx -s PSQUERY`
		`Service PSQUERY is running on instance(s) CDBHCM1,CDBHCM2`
		This service should be running on all RAC instances.
		NOTE: This service must be started before starting the process scheduler. Otherwise, the process scheduler will fail on startup.

Table 34: Database failover

After completing the steps in the above table, follow steps 8 through 11 of Site Switchover to bring up PeopleSoft and complete the failover process. **NOTE**: If you have a complete site failure as described in step 4 in the above table, then after completing step 9 of Site Switchover you will then need to run the disable_psft_rsync.sh script again, as the startPSTAPP.sh script will enable rsync.

In an actual failure event where the primary site is lost or becomes inaccessible, an assessment of the scope and impact will need to be conducted. Here are a few items to consider:

- Possible database data loss
- Missing file system artifacts (reports, logs, inbound and outbound files, etc.)

Depending on the outage, you may or may not be able to recover every transaction committed at the original primary. If possible, ask the users to verify the very last transactions they were working on.

It is likely there will be missing file system artifacts, from output being written or transmitted when access to the original primary ceases. Use the report logging in the database to identify file system artifacts created near the time of the outage, then determine what needs to be done, case by case, for missing or incomplete files.

12.2 Reinstate Failed Primary as New Standby

You will want to protect your new production environment with a standby. Ideally, you will be able to reinstate the failed primary as a new standby. You will reinstate both the database and the file systems.

12.2.1 Reinstate Old Primary DB as Standby

Data Guard will prevent the old primary database from being opened when it is made available again after a primary site failure. Any attempt to start the database normally will fail, with messages written to its alert log indicating reinstatement is required. If Flashback Database was enabled on this database prior to the failure, then the old primary can be reinstated as the new standby.

To reinstate the old primary as a standby of current production:

1. Log in to one of the domUs hosting the old primary database and start the database:

   ```
   $ srvctl start database -db CDBHCM_phx5s
   ```

2. Log in to the Data Guard Broker command line interface at the new primary region and show the configuration. Note the ORA-16661 error we have bolded below:

   ```
   $ dgmgrl sys/<sys password>
   DGMGRL> show configuration
   onfiguration - fsc

     Protection Mode: MaxPerformance
     Members:
     CDBHCM_iad1dx - Primary database
       CDBHCM_phx5s  - Physical standby database (disabled)
         ORA-16661: the standby database needs to be reinstated

   Fast-Start Failover:  Disabled

   Configuration Status:
   SUCCESS   (status updated 12 seconds ago)
   ```

3. Reinstate the standby:

   ```
   DGMGRL> REINSTATE DATABASE CDBHCM_phx5s;
   ```

   Note: The process of recovering the failed database and making a valid standby starts and may take some time to complete.

After we completed the above steps, we used the Data Guard Broker command line interface to check the status of our environment:

```
DGMGRL> show configuration lag

Configuration - fsc

  Protection Mode: MaxPerformance
  Members:
  CDBHCM_iad1dx - Primary database
    CDBHCM_phx5s  - 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 35 seconds ago)
```

The reinstated database is now serving as standby, protecting the primary and available for switchover and failover. If Active Data Guard support is configured (see Configure Primary and Standby Database Servers for Active Data

Guard), then the ADG service for PeopleSoft (PSQUERY) can be relocated from the primary back to the standby database.

12.2.2 Reinstate Old Primary Middle Tiers as Standby

If your old primary middle tier servers remained available while the database failover event occurred, you should have done a final forced rsync from the failed primary site to the standby at the time of the failure, then reversed direction of the rsync processes in the same manner as when you do a switchover. Refer to step 3 in Perform Site Failover for perform a forced rsync.

If the old primary site is completely inaccessible even for rsync, then disable the rsync scripts at the new primary site with disable_psft_rsync.sh for all file systems being replicated. If you can resume activity on the original middle tiers at a later time, re-enable the rsync scripts and let them catch up.

If you need to rebuild your middle tiers, follow the processes described earlier in this paper for that activity.

See rsync Log and Report Files for details on using the rsync scripts.

13 Appendix A: Working with Terraform

Terraform Discovery will discover the definitions of resources within a compartment at the primary region. Terraform Discovery can be accessed from the OCI console. It will create a Terraform *stack*, held in a downloadable ZIP file.

This project has several subnets, each with one or more security lists, with a few complex security lists containing dozens of ingress rules. For this case study, we chose to use Terraform to discover, then replicate, the resources in the network compartment.

Terraform discovered the following network components:

- Virtual Cloud Network (VCN)
- Gateways (Internet, NAT and Service gateways)
- Route tables
- Security lists
- Subnets

To execute this Terraform discovery:

1. Log in to the OCI console
2. Change the region to the primary region
3. Click on main menu
4. Click Development Services
5. Under Resource Manager, click Stacks
6. Click on the Create Stack button
7. Select the "Existing Compartment" radio option with the text that says: "Create a stack that captures resources from the selected compartment (resource discovery)"
8. For "Compartment for Resource Discovery" drop down combo box, select the compartment to discover resources from. You will need to expand the root to get the full list of compartments. We discovered the network compartment
9. For "Region for Resource Discovery", select the OCI primary region
10. For Terraform Provider Services radio button options, select All
11. Provide a name for the ZIP file that will be created, in the Name text box
12. Description is optional
13. Choose the compartment that the stack should be created in
14. Click Next twice to get to the Review page
15. If the information on the Review page is correct, click Create

Once the stack creation job completes, the stack will appear in the compartment selected in step 13. Click on the link for the stack to get the stack details page. On the stack details page, click on the Download link on the Terraform Configuration, to download the stack ZIP file to your local computer. Save the ZIP file to a directory of your choosing and unzip it.

13.1.1 Editing Terraform Files

When you unzip the stack ZIP file, you will find several Terraform files in JSON format, ending with .tf. The contents of the .tf files will depend on what resources were discovered within the compartment where the Terraform discovery was performed.

There will be changes required to the .tf files. For example, “export_” is added to all resource definitions, and must be removed. When working with a network discovery, we must also assign a different and non-overlapping CIDR block, provide a new display name, provide a different DNS label, and provide a different VCN reference.

Most of the resource definitions will be found in the core.tf JSON file.

Note: Before making changes to the .tf files, we recommend you back them up.

The following table shows examples of Terraform definitions from the primary region and the changes needed for the secondary region.

RESOURCE TYPE	PRIMARY REGION DEFINITION (ASHBURN)	MODIFICATIONS FOR SECONDARY REGION (PHOENIX)

Virtual Cloud Network	resource oci_core_vcn export_iad-cloudmaa-vcn {
	  #cidr_block = <<Optional value not found in discovery>>
	  cidr_blocks = [
	    “10.0.0.0/16”,
	  ]
	  compartment_id = var.compartment_ocid
	  defined_tags = {
	  }
	  display_name = “iad-cloudmaa-vcn”
	  dns_label = “iadcloudmaavcn”
	  freeform_tags = {
	  }
	  #is_ipv6enabled = <<Optional value not found in discovery>>
	}
		Required modifications include removing “export_”, assigning a different non-overlapping CIDR, display name and changing the DNS label:
		resource oci_core_vcn phx-cloudmaa-vcn {
		  #cidr_block = <<Optional value not found in discovery>>
		  cidr_blocks = [
		    “10.10.0.0/16”,
		  ]
		  compartment_id = var.compartment_ocid
		  defined_tags = {
		  }
		  display_name = **“phx-cloudmaa-vcn”**
		  dns_label = **“phxcloudmaavcn”**
		  freeform_tags = {
		  }
		  #is_ipv6enabled = <<Optional value not found in discovery>>
		}

NAT Gateway	resource oci_core_nat_gateway export_iadmaa-ngwy {
	  block_traffic = “false”
	  compartment_id = var.compartment_ocid
	  defined_tags = {
	  }
	  display_name = “iadmaa-ngwy”
	  freeform_tags = {
	  }
	  public_ip_id = “ocid1.publicip.oc1.iad.aaaaaaaagwkvnlh6y4irjubj63dm36mdsuig6zbc2oakgmssvifpprvx6kzq”
	  vcn_id = oci_core_vcn.export_iad-cloudmaa-vcn.id
	}
		Modifications include removing “export_”, changing the display name and VCN reference.
		resource oci_core_nat_gateway **phxmaa-ngwy** {
		  block_traffic = "false"
		  compartment_id = var.compartment_ocid
		  defined_tags = {
		  }
		  display_name = **"phxmaa-ngwy"**
		  freeform_tags = {
		  }
		  public_ip_id = "ocid1.publicip.oc1.iad.aaaaaaaagwkvnlh6y4irjubj63dm36mdsuig6zbc2oakgmssvifpprvx6kzq"
		  vcn_id = oci_core_vcn.**phx-cloudmaa-vcn**.id
		}

Route Table	resource oci_core_route_table export_iad-db-private-RT {
	  compartment_id = var.compartment_ocid
	  defined_tags = {
	  }
	  display_name = "iad-db-private-RT"
	  freeform_tags = {
	  }
	  route_rules {
	    #description = <<Optional value not found in discovery>>
	    destination = "0.0.0.0/0"
	    destination_type = "CIDR_BLOCK"
	    network_entity_id = oci_core_nat_gateway.export_iadmaa-ngwy.id
		Modifications include removing “export_”, changing the name of the route table, display name, and VCN reference.
		resource oci_core_route_table **phx-db-private-RT** {
		  compartment_id = var.compartment_ocid
		  defined_tags = {
		  }
		  display_name = **"phx-db-private-RT"**
		  freeform_tags = {
		  }
		  route_rules {
		    #description = <<Optional value not found in discovery>>
		    destination = "0.0.0.0/0"

	}
	 vcn_id = oci_core_vcn.export_iad-cloudmaa-vcn.id
	 }
		destination_type = "CIDR_BLOCK"
		 network_entity_id = oci_core_nat_gateway.phxmaa-ngwy.id
		 }
		 vcn_id = oci_core_vcn.**phx-cloudmaa-vcn**.id
		 }

Security list	resource oci_core_security_list export_iad-db-private-seclist {
	 compartment_id = var.compartment_ocid
	 defined_tags = {
	 }
	 display_name = "iad-db-private-seclist"
	 egress_security_rules {
	 #description = <<Optional value not found in discovery>>
	 destination = "0.0.0.0/0"
	 destination_type = "CIDR_BLOCK"
	 #icmp_options = <<Optional value not found in discovery>>
	 protocol = "6"
	 stateless = "false"
	 #tcp_options = <<Optional value not found in discovery>>
	 #udp_options = <<Optional value not found in discovery>>
	 }
	 egress_security_rules {
	 #description = <<Optional value not found in discovery>>
	 destination = "0.0.0.0/0"
	 destination_type = "CIDR_BLOCK"
	 #icmp_options = <<Optional value not found in discovery>>
	 protocol = "1"
	 stateless = "false"
	 #tcp_options = <<Optional value not found in discovery>>
	 #udp_options = <<Optional value not found in discovery>>
	 }
	 freeform_tags = {
	 }
	 ingress_security_rules {
	 #description = <<Optional value not found in discovery>>
	 #icmp_options = <<Optional value not found in discovery>>
	 protocol = "6"
	 source = "10.0.102.0/24"
	 source_type = "CIDR_BLOCK"
	 stateless = "false"
	 #tcp_options = <<Optional value not found in discovery>>
	 #udp_options = <<Optional value not found in discovery>>
	 }
	 ingress_security_rules {
	 #description = <<Optional value not found in discovery>>
	 #icmp_options = <<Optional value not found in discovery>>
	 protocol = "1"
	 source = "10.0.102.0/24"
	 source_type = "CIDR_BLOCK"
	 stateless = "false"
	 #tcp_options = <<Optional value not
		Modifications include removing "export_", changing name of the security list and its display name, changing the CIDR blocks in each ingress rule that have 10.0.x.y to **10.10**.x.y, and changing the VCN reference. Leave the 0.0.0.0/0 unchanged.
		
		 resource oci_core_security_list **phx-db-private-seclist** {
		 compartment_id = var.compartment_ocid
		 defined_tags = {
		 }
		 display_name = **"phx-db-private-seclist"**
		 egress_security_rules {
		 #description = <<Optional value not found in discovery>>
		 destination = "0.0.0.0/0"
		 destination_type = "CIDR_BLOCK"
		 #icmp_options = <<Optional value not found in discovery>>
		 protocol = "6"
		 stateless = "false"
		 #tcp_options = <<Optional value not found in discovery>>
		 #udp_options = <<Optional value not found in discovery>>
		 }
		 egress_security_rules {
		 #description = <<Optional value not found in discovery>>
		 destination = "0.0.0.0/0"
		 destination_type = "CIDR_BLOCK"
		 #icmp_options = <<Optional value not found in discovery>>
		 protocol = "1"
		 stateless = "false"
		 #tcp_options = <<Optional value not found in discovery>>
		 #udp_options = <<Optional value not found in discovery>>
		 }
		 freeform_tags = {
		 }
		 ingress_security_rules {
		 #description = <<Optional value not found in discovery>>
		 #icmp_options = <<Optional value not found in discovery>>
		 protocol = "6"
		 source = "**10.10.102.0/24**"
		 source_type = "CIDR_BLOCK"
		 stateless = "false"
		 #tcp_options = <<Optional value not found in discovery>>
		 #udp_options = <<Optional value not found in discovery>>
		 }
		 ingress_security_rules {
		 #description = <<Optional value not found in discovery>>
		 #icmp_options = <<Optional value not found in

	found in discovery>>
	    #udp_options = <<Optional value not found in discovery>>
	  }
	  ingress_security_rules {
	    #description = <<Optional value not found in discovery>>
	    #icmp_options = <<Optional value not found in discovery>>
	    protocol = "6"
	    source = "10.0.103.0/24"
	    source_type = "CIDR_BLOCK"
	    stateless = "false"
	    tcp_options {
	      max = "22"
	      min = "22"
	      #source_port_range = <<Optional value not found in discovery>>
	    }
	    #udp_options = <<Optional value not found in discovery>>
	  }
	  ingress_security_rules {
	    #description = <<Optional value not found in discovery>>
	    #icmp_options = <<Optional value not found in discovery>>
	    protocol = "6"
	    source = "10.0.103.0/24"
	    source_type = "CIDR_BLOCK"
	    stateless = "false"
	    tcp_options {
	      max = "1530"
	      min = "1521"
	      #source_port_range = <<Optional value not found in discovery>>
	    }
	    #udp_options = <<Optional value not found in discovery>>
	  }
	  vcn_id = oci_core_vcn.export_iad-cloudmaa-vcn.id
	}
		discovery>>
		    protocol = "1"
		    source = "**10.10.102.0/24**"
		    source_type = "CIDR_BLOCK"
		    stateless = "false"
		    #tcp_options = <<Optional value not found in discovery>>
		    #udp_options = <<Optional value not found in discovery>>
		  }
		  ingress_security_rules {
		    #description = <<Optional value not found in discovery>>
		    #icmp_options = <<Optional value not found in discovery>>
		    protocol = "6"
		    source = "**10.10.103.0/24**"
		    source_type = "CIDR_BLOCK"
		    stateless = "false"
		    tcp_options {
		      max = "22"
		      min = "22"
		      #source_port_range = <<Optional value not found in discovery>>
		    }
		    #udp_options = <<Optional value not found in discovery>>
		  }
		  ingress_security_rules {
		    #description = <<Optional value not found in discovery>>
		    #icmp_options = <<Optional value not found in discovery>>
		    protocol = "6"
		    source = "**10.10.103.0/24**"
		    source_type = "CIDR_BLOCK"
		    stateless = "false"
		    tcp_options {
		      max = "1530"
		      min = "1521"
		      #source_port_range = <<Optional value not found in discovery>>
		    }
		    #udp_options = <<Optional value not found in discovery>>
		  }
		  vcn_id = oci_core_vcn.**phx-cloudmaa-vcn**.id
		}

Subnet	resource oci_core_subnet export_exadb-private-subnet-ad2 {
	  availability_domain = "LoSv:US-ASHBURN-AD-2"
	  cidr_block = "10.0.101.0/24"
	  compartment_id = var.compartment_ocid
	  defined_tags = {
	    "Oracle-Tags.CreatedBy" = "ocid1.saml2idp.oc1..aaaaaaaatilj7lqztsx6jehhm7k5374c5jxg6uuhzvdehgbiprb55gnyejba/<oci user name>"
	    "Oracle-Tags.CreatedOn" = "2020-03-13T18:50:55.371Z"
	  }
	  dhcp_options_id = oci_core_vcn.export_iad-cloudmaa-vcn.default_dhcp_options_id
	  display_name = "exadb-private-subnet-ad2"
		Modifications include removing “export_” where it appears, changing CIDR to a subnet within the VCN for the Phoenix region, changing the availability domain, changing the route table and VCN references.
		
		resource oci_core_subnet exadb-private-subnet-**ad1** {
		  availability_domain = "LoSv:**US-PHOENIX-AD-1**"
		  cidr_block = "**10.10.101.0/24**"
		  compartment_id = var.compartment_ocid
		  defined_tags = {
		    "Oracle-Tags.CreatedBy" = "ocid1.saml2idp.oc1..aaaaaaaatilj7lqztsx6jehhm7k5374c5jxg6uuhzvdehgbiprb55gnyejba/<oci user name>"
		    "Oracle-Tags.CreatedOn" = "2020-03-13T18:50:55.371Z"
		  }
		  dhcp_options_id = oci_core_vcn.phx-cloudmaa-

	dns_label = "exadbprivate"
	freeform_tags = {
	}
	#ipv6cidr_block = <<Optional value not found in discovery>>
	prohibit_internet_ingress = "true"
	prohibit_public_ip_on_vnic = "true"
	route_table_id = oci_core_route_table.export_iad-db-private-RT.id
	security_list_ids = [
	oci_core_security_list.export_siteguard-seclist.id,
	oci_core_security_list.export_bastion-private-seclist.id,
	oci_core_security_list.export_iad-db-private-seclist.id,
	]
	vcn_id = oci_core_vcn.export_iad-cloudmaa-vcn.id
	}
		vcn.default_dhcp_options_id
		display_name = "exadb-private-subnet-**ad1**"
		dns_label = "exadbprivate"
		freeform_tags = {
		}
		#ipv6cidr_block = <<Optional value not found in discovery>>
		prohibit_internet_ingress = "true"
		prohibit_public_ip_on_vnic = "true"
		route_table_id = oci_core_route_table.phx-db-private-RT.id
		security_list_ids = [
		oci_core_security_list.siteguard-seclist.id,
		oci_core_security_list.bastion-private-seclist.id,
		oci_core_security_list.phx-db-private-seclist.id,
		]
		vcn_id = oci_core_vcn.phx-cloudmaa-vcn.id
		}

Table 35: Terraform resource definitions and modifications

As there are patterns to the items that must be changed, using editing tools such as *sed* can aid in automating the necessary changes.

If you provisioned some components using Terraform and others using the OCI console or other means, then you must adjust the Terraform resource definitions you plan to use. For example, if you provisioned the VCN and a NAT gateway using the OCI console, then any resource that references the VCN and the NAT gateway within the .tf files will need the following change:

1. In the vars.tf file, add and set the value of the two variables vcn_ocid and `nat_gateway_ocid` with these patterns:

   ```
   variable vcn_ocid { default = "<OCID of VCN>" }
   variable nat_gateway_ocid { default = "<OCID of NAT gateway>" }
   ```

2. Search all .tf files that have resources with definitions that have references to the VCN and/or the NAT gateway. For example, search for the pattern “vcn_id” and “network_entity_id”.   For each occurrence, set the variable to the new value, as shown below:

   ```
   vcn_id = "${var.vcn_ocid}"
   network_entity_id = "${var.nat_gateway_ocid}"
   ```

3. Modify the availability_domain.tf file to include all availability domains in the target region. Find the list of availability domains in OCI by clicking on Compute, then on Instance. The availability domains will be shown on the left side of your screen.

   Using Phoenix as an example:

   ```
   ## This configuration was generated by terraform-provider-oci
   ## then modified to include all ADs at the target

   data oci_identity_availability_domain LoSv-US-PHOENIX-AD-1 {
     compartment_id = var.compartment_ocid
   ```

```
          ad_number      = "1"
        }
        data oci_identity_availability_domain LoSv-US-PHOENIX-AD-2 {
          compartment_id = var.compartment_ocid
          ad_number      = "2"
        }
        data oci_identity_availability_domain LoSv-US-PHOENIX-AD-3 {
          compartment_id = var.compartment_ocid
          ad_number      = "3"
        }
```

Note: The OCID is obtained from the OCI console by clicking on the “Show” or “Copy” link of the OCID for the desired resource.

Here is an example of changes required to the core.tf file containing the definition of the route table resource that uses the variables defined above.

```
resource oci_core_route_table phx-db-private-RT {
  compartment_id = var.compartment_ocid
  defined_tags = {
  }
  display_name = "phx-db-private-RT"
  freeform_tags = {
  }
  route_rules {
    #description = <<Optional value not found in discovery>>
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    #network_entity_id = oci_core_nat_gateway.maa-phx-ngw.id
    network_entity_id = "${var.nat_gateway_ocid}"
  }
  #vcn_id = oci_core_vcn.ebs-maacloud2-vcn.id
  vcn_id = "${var.vcn_ocid}"
}
```

13.1.2 Deploying Resources with Terraform

Once all the resources that will be deployed with Terraform at the secondary region have been edited, collect the .tf files containing these resources. You must have the following files:

- vars.tf – This file contains all Terraform variables required to execute Terraform.
- availability_domain.tf – This file contains the definitions of all availability domains for the secondary region.
- One or more .tf files that contain the resource definitions for deploying the chosen resources.

It is not required to include all the .tf files that were generated by the Terraform discovery process at the primary site. Only the files mentioned above are required.

Follow these steps to use the OCI console to deploy the resources:

1. Zip up the required .tf files into a single ZIP file. This will be used to create your Terraform stack.
2. Log in to the OCI console and navigate to Development Services, then Stacks under Resource Manager.
3. Use the compartment combo-dropdown to specify the compartment you want the stack ZIP file to be placed.
4. Click the Create Stack button.
5. Select “My Configuration” option.
6. Under Terraform Source, choose Zip file then browse and select the ZIP file you created in step 1 above.

7. Optional: Provide a name for your stack.
8. Optional: Provide a description of your stack.
9. Select the compartment in which the stack is to be created.
10. Best practice: Select the latest version of Terraform.
11. Optional: Add any tags.
12. Click Next.
13. Verify that the variables listed on this page have the correct values. They can be changed in this screen. These variables were read from the vars.tf file.
14. Click Next.
15. The Review page is shown. As you are only creating a Terraform stack – which a definition of all resources to be deployed – do NOT check the checkbox for "Run Apply".
16. Click Create.

Once the Terraform stack is created, the Stack Details page is shown with several action buttons, one of which is Plan. Click the Plan button to create the plan.

Terraform will validate the stack while it is creating the plan. If creation of the plan fails, the OCI console will indicate that the job failed and will display the log, which will show which .tf files and which resource definitions had an error. You will need to change the .tf files to correct the errors, recreate the Terraform stack, and try to create the Plan again.

Once all errors have been resolved and the plan job runs successfully, click on Apply to start a job that will create all the resources defined in the Terraform stack. The amount of time the job will run depends on the type and number of resources being deployed. For example, creating compute instances or a database service (VM DB or ExaDB-D) will take a measurable amount of time.

14 Appendix B: Using PeopleSoft Cloud Manager to Provision Middle Tiers

This paper describes a manual method for moving the PeopleSoft application and middle tier software and configuration from the on-premises installation to the new OCI implementation. It is also possible to use the PeopleSoft Cloud Manager to provision the OCI middle tiers.

To use the PeopleSoft Cloud Manager to provision the OCI middle tiers, you will need:

- An OCI compute instance where the PeopleSoft Cloud Manager is installed and configured
- A separate Windows VM for running Change Assistant and PeopleSoft client tools
- The PeopleSoft database definition imported into the Cloud Manager.

With these options, you will use the PeopleSoft Cloud Manager to provision compute instances (VMs), which can each host one or more PeopleSoft domains. Each domain will host specific services. The application servers, process schedulers, and web server can be deployed together or separately. We recommend the web servers be deployed separately.

You can either do a fresh install of the PeopleSoft software or you can copy your software from your current on-premises installation to your new OCI environment, called “lift and shift”. Both the fresh install and the lift and shift follow the same pattern. With the fresh install, you specify the version of the new middle tiers, while with the lift and shift you will bring the existing application version over to the new environment.

14.1 Freshly Install PeopleSoft Application and Middle Tier Software Using the PeopleSoft Cloud Manager

On this path, you would do a fresh install of the application and web middle tier software then configure it to access the PeopleSoft database. The Cloud Manager must be subscribed to the correct PeopleSoft channels, specifically PeopleTools and all PeopleSoft applications in use. The PeopleTools version must be the same as your on-premises deployment.

This option does not reference the on-premises deployment. In the Cloud Manager, using “Manage Node”, you will create the compute instances one at a time, selecting:

- The version and shape of the new middle tier
- The compartment where the new middle tiers will be placed
- The VCN, availability domain, and subnet on which the new middle tier is to be deployed
- Which tier is to be configured: application server, process scheduler, or web server. Note the application servers and process scheduler can be deployed on shared servers
    - Settings specific to the tier selected (application server, process scheduler, or web server domain), and the number of PeopleSoft domains for each
    - Required credentials such as Access ID, Connect ID, WebLogic password, database passwords, etc.
    - The file system used for the shared PS_HOME, PS_APP_HOME, and PS_CUSTOM_HOME
    - The number of processes for each server type
- Other attributes that can be set if desired

Once all information has been provided, clicking on Submit will create a job that starts the provisioning process. If there are no failures, there will be a new middle tier running the services that were configured as described above. The new middle tier will show up in the OCI console.

14.2 Lift and Shift Application and Web Tiers Using the PeopleSoft Cloud Manager

With this choice, you will pull the application and middle tier software from the source system for installation on the new environment. You will first mine the existing environment – the "lift" portion – then use that data to build the OCI setup ("shift"). The application lift process will create a DPK (Deployment Puppet Kit) containing the contents of the PS_HOME, PS_APP_HOME, and PS_CUSTOM_HOME. Once the DPK is created, it is uploaded to an object storage bucket where the PeopleSoft Cloud Manager can access it for deploying new middle tiers in OCI.

To download and install the lift toolkit then perform an application lift, see the PeopleSoft Cloud Manager documentation, starting with the section "Download the Lift Utility". Make sure you review "Installing Lift Prerequisites" and "Performing Application Lift". You would be following these detailed steps to perform an application-only lift since the database was migrated using ZDM.

Use the "Manage Node" action to add nodes to this environment, as described in the previous section.

This option will allow you to select the DPK that was uploaded and the shape of the new middle tier node. The cloud manager will present settings discovered from the source environment for your review and adjustment. The list of settings is the same as described in the previous section.

Once all information has been provided, clicking on Submit will create a job that starts the provisioning process. If there are no failures, there will be a new middle tier running the services that were configured as described above. This new middle tier will also show up in the OCI console.

Please see the PeopleSoft Cloud Manager for further details of these options.

15 Appendix C: Basic Tasks and Scripts

15.1 Accessing OCI Cloud Resources with SSH

This section briefly describes how to access resources that are provisioned within OCI, such as ExaDB-D domUs and the compute instances hosting the middle and application tier services.

15.1.1 Get the ExaDB-D IP addresses

To connect to the ExaDB-D compute domUs requires having their IP addresses. To obtain their IP addresses:

1. Log in to the OCI console
2. From the menu, select Oracle Databases → Exadata at Oracle Cloud
3. Select Exadata VM Cluster
4. Make sure the correct compartment is selected, in our case psft_exa_compartment
5. Click on the Exadata VM cluster listed in the table, ours is called IAD-Exa-VMCluster-1
6. Click on Virtual Machines
7. A table is displayed with the virtual machines (domUs). For this project, private subnets were used so the IP addresses under the column “Private IP Addresses & DNS name” are the IP addresses we need for accessing the domUs.

15.1.2 Get the Compute Instances IP addresses

You can obtain the IP addresses of your middle tier compute instances as follows:

1. Log in to the OCI console
2. From the main menu select Compute, then Instances
3. Change to the compartment where the compute instances reside e.g., psft-app-compartment
4. A table is displayed that includes the private IP addresses.

15.1.3 Connect Using FastConnect or IPSec

We expect most customers to use FastConnect or IPSec. For these connection services, follow these steps:

1. On the client system, load the private key to access the ExaDB-D domUs

```
$ eval `ssh-agent`
$ ssh-add <path-to-private-key-file/private-key-file>
```

2. Now connect to either of the domUs:

```
$ ssh opc@<domu IP address>
```

If you do not use the ssh-agent, then you can issue the following:

```
$ ssh -I <path-to-private-key-file/private-key-file> opc@10.0.101.2
```

3. Once connected, you can change to the oracle user:

```
$ sudo su – oracle
```

15.2 PeopleSoft Startup / Shutdown Scripts

IMPORTANT: The scripts provided below serve as examples. They are not part of any Oracle product nor are they under any warranty. They are intended for customers to use and modify for their deployment.

ALWAYS test these scripts on a test environment before promoting them to a production environment.

The scripts for this project fall into one of the following categories:

1. Stand-alone PeopleSoft startup and shutdown scripts
2. Rsync scripts that replicate middle tier file system contents from one site to another
3. Wrapper scripts that call the stand-alone scripts plus enable or disable rsync replication, depending on which wrapper script is run

These scripts can be used by OCI Full Stack DR Cloud Service (FSDR) to automate switchover and failover. The rsync scripts described below will handle file system role transition for the application and web tiers.

The scripts should be placed in a common location at each site that all application and web tier compute instances can access. In this project, we created the following directory location on each site's shared storage, and labeled it $SCRIPT_DIR in our .env file:

/u01/app/psft/pt/custom_admin_scripts

As these scripts are used by administrators, it is advisable to add the scripts directory to the administrator account's PATH.

15.2.1 Application Domain Server

This section holds simple scripts to start and stop the application and the process scheduler domains. These are stand-alone scripts that are run on each PeopleSoft middle tier node. You can specify the domain name as a parameter to these scripts.

15.2.1.1 startAPP.sh:

```
############################################################################
#!/bin/sh
# File name:   startAPP.sh
#
# Description: Start PeopleSoft application server on one node
#
# Usage: startAPP.sh <app server domain>
#        If no parameter, look in $PS_CFG_HOME/appserv for the domain
#
# Errors: Domain not set.  Could not determine domain.
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley  Created
############################################################################

source ~/psft.env
```

```
DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n != 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single App Server domain."
  DOMAIN=`ls -l $PS_CFG_HOME/appserv | grep ^d | grep -v prcs | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the domain set?
if { $DOMAIN = "" ]; then
   echo "Domain not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

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

15.2.1.2stopAPP.sh:

```
############################################################################
#!/bin/sh
# File name: stopAPP.sh
#
# Description: Stop the PSFT application servers
#
# Usage: stopAPP.sh <app server domain>
#        If no parameter, look in $PS_CFG_HOME/appserv for the domain
#
# Errors: Domain not set
#         More than one domain found.
#
# Revisions:
```

```
# Date       Who         What
# 7/1/2023   DPresley    Created
############################################################################

source ~/psft.env

DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n -!= 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single App Server domain."
  DOMAIN=`ls -l $PS_CFG_HOME/appserv | grep ^d | grep -v prcs | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the DOMAIN set?
if { $DOMAIN" = "" ]; then
   echo $DOMAIN not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

EGREP_STRING="rmiregistry"
PROCESS_COUNT=0
PID_LIST=""

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

# Explicitly stopping rmiregistry due to a bug in PeopleTools 8.57.
# This is not needed for later versions of PeopleTools.
# Note that there can be more than one rmiregistery process running.  All must
# be terminated when Tuxedo is shut down.

echo ""
echo "Stopping rmiregistry processes..."
PROCESS_COUNT=`ps -elf | grep psadm2 | egrep "${EGREP_STRING}" | grep -v grep | wc -l`
echo "Number of remaining process : ${PROCESS_COUNT}"

i=1
while [ ${PROCESS_COUNT} -ne 0 ];
do
   PROCESS_COUNT=`ps -elf | grep psadm2 | egrep  "${EGREP_STRING}" | grep -v grep | wc -l `
   echo "Number of remaining process : ${PROCESS_COUNT}"
   if [[ $i -gt 0 && ${PROCESS_COUNT} -ne 0 ]]; then        -- WHAT IS THE PURPOSE OF i? WANT TO
ECHO IT?
        PID_LIST=`ps -elf | grep psadm2 | egrep  "${EGREP_STRING}" | grep -v grep | awk '{print
$ }'`
        echo "Killing processes:"
        echo "${PID_LIST} "
        kill -9 ${PID_LIST}
   fi
   i=`expr $i + 1`

done
```

15.2.1.3startPS.sh:

```
############################################################################
#!/bin/sh
# File name: startPS.sh
#
# Description: Start the PSFT process scheduler
#
# Usage: startPS.sh <process scheduler domain>
#        If no parameter, look in $PS_CFG_HOME/appserv/prcs for the domain
#
# Errors: Domain not set
#         More than one domain found.
#
# Revisions:
# Date       Who
# 7/1/2023   DPresley
############################################################################

source ~/psft.env

DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n -!= 0 ]; then
```

```
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single Process Scheduler domain."
  DOMAIN=`ls -l $PS_CFG_HOME/appserv/prcs | grep ^d | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the DOMAIN set?
if { $DOMAIN" = "" ]; then
   echo $DOMAIN not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

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

15.2.1.4 stopPS.sh:

```
############################################################################
#!/bin/sh
# File name:  stopPS.sh
#
# Description: Stop the PSFT process scheduler
#
# Usage: stopPS.sh <process scheduler domain>
#        If no parameter, look in $PS_CFG_HOME/appserv/prcs for the domain
#
# Errors: Domain not set
#         More than one domain found.
#
# Revisions:
# Date       Who
# 7/1/2023   DPresley
############################################################################

source ~/psft.env

DOMAIN=$1
```

```
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n -!= 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single Process Scheduler domain."
  DOMAIN=`ls -l $PS_CFG_HOME/appserv/prcs | grep ^d | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the DOMAIN set?
if { $DOMAIN" = "" ]; then
   echo $DOMAIN not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

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

15.2.2 PIA Web Server Domain

This section holds scripts that start and stop the PeopleSoft WLS server and the Coherence*Web cache servers. Similar to the application server scripts, you can pass a domain name to these scripts. Note that the domain name is required for the startCacheServer.sh and stopCacheServer.sh scripts.

Note: If Coherence*Web has been configured with the PeopleSoft PIA, we must start the Coherence*Web cache server FIRST on ALL compute nodes that host Coherence*Web before starting the PeopleSoft Weblogic server (PIA).

In the Java command that starts the cache server, note the Java heap initial and max size of 2 Gigabytes: -Xms2g -Xmx2g. This should be a good starting point. Refer to the Coherence*Web documentation.

15.2.2.1startCacheServer.sh:

```
############################################################################

#!/bin/sh

# File name: startCacheServer.sh
```

```
#
# Description: Start Coherence*Web cache server
#
# Usage: startCacheServer.sh <PeopleSoft coherence domain>
#        We don't have a good way to derive the domain name/build a
#        directory location for Coherence*Web log files, so it is
#        required.
#
# Errors: Could not determine Coherence domain
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley  Created
############################################################################

source ~/psft.env

# export DOMAIN=HR92U033
DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n != 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
   echo "No domain passed in. Domain required."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`
export COHERENCE_HOME=$BASE_DIR/pt/bea/coherence
export COHERENCE_CONFIG=$PS_CFG_HOME/coherence/config
export COHERENCE_LOG=$PS_CFG_HOME/coherence/log
export CWEB_LOG_NAME=pia_${DOMAIN}_${HOSTNAME}
export CWEB_LOG_LEVEL=9


date
echo "------ Starting Coherence*Web Cache Server for domain: $DOMAIN on host: $HOSTNAME ----"


echo ""
echo "tangosol.coherence.override=${COHERENCE_CONFIG}/tangosol-coherenceoverride.xml"
echo "Log file can be found at: ${COHERENCE_LOG}/cweb_coherence_server_${CWEB_LOG_NAME}.log "


java -Xms2g -Xmx2g -Dtangosol.coherence.distributed.localstorage=true -
Dtangosol.coherence.session.localstorage=true -
Dtangosol.coherence.override=${COHERENCE_CONFIG}/tangosol-coherence-override.xml -
Dtangosol.coherence.cacheconfig=default-session-cache-config.xml -
Dtangosol.coherence.log=${COHERENCE_LOG}/cweb_coherence_server_${CWEB_LOG_NAME}.log -
Dtangosol.coherence.log.level=9 -classpath
${COHERENCE_CONFIG}:${COHERENCE_HOME}/lib/coherence.jar:${COHERENCE_HOME}/lib/coherence-web.jar
com.tangosol.net.DefaultCacheServer -Djava.net.preferIPv6Addresses=false -
Djava.net.preferIPv4Stack=true -Dcoherence.log.level=${CWEB_LOG_LEVEL}  &


# Sleep for 30 seconds to allow for cache server to fully start.
sleep 30
```

15.2.2.2 stopCacheServer.sh:

Note: If Coherence*Web has been configured with the PeopleSoft PIA, you need to shut the PIA down on ALL compute instances hosting the PIA before stopping the Coherence*Web cache server.

```
############################################################################
#!/bin/sh
# File name: stopCacheServer.sh
#
# Description: Stop Coherence*Web cache server
#
# Usage: stopCacheServer.sh
#        NOTE: THIS STOPS ALL COHERENCE PROCESSES RUNNING ON THIS SERVER
#
# Errors:
```

```
#
# Revisions:
# Date       Who
# 7/1/2023   DPresley
############################################################################

EGREP_STRING="coherence"
PROCESS_COUNT=0
PID_LIST=""

echo "Stopping Coherence*Web Cache Server..."
PROCESS_COUNT=`ps -elf | grep psadm2 | egrep  "${EGREP_STRING}" | grep -v grep | wc -l `
echo "Number of remaining process : ${PROCESS_COUNT}"


while [ ${PROCESS_COUNT} -ne 0 ];
do
     PROCESS_COUNT=`ps -elf | grep psadm2 | egrep  "${EGREP_STRING}" | grep -v grep | wc -l `
     echo "Number of remaining process : ${PROCESS_COUNT}"
     if [ ${PROCESS_COUNT} -ne 0 ]; then
          PID_LIST=`ps -elf | grep psadm2 | egrep  "${EGREP_STRING}" | grep -v grep | awk '{ print $4 }' `
          echo "Killing processes:"
          echo "${PID_LIST} "
          kill -9 ${PID_LIST}
     fi
     i=`expr $i + 1`
done
```

15.2.2.3 startWS.sh:

```
############################################################################
#!/bin/sh
# File name:   startWS.sh
#
# Description: Start the PeopleSoft PIA / WLS server
```

```
#
# Usage:       startWS.sh <domain>
#
# Errors:
#
# Revisions:
# Date       Who
# 7/1/2023   DPresley
############################################################################

# Start the PeopleSoft PIA.

source ~/psft.env

DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n != 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single WLS Server domain."
  DOMAIN=`ls -l $PS_CFG_HOME/webserv | grep ^d | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the domain set?
if { $DOMAIN = "" ]; then
   echo "Domain not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

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

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

15.2.2.4 stopWS.sh:

```
############################################################################
#!/bin/sh
# File name:  stopWS.sh
#
# Description: Stop the PeopleSoft PIA / WLS server
#
# Usage:     stopWS.sh <domain>
#
# Errors:    No domain passed in, more than one found
#
# Revisions:
#
# Date       Who        What
# 7/1/2023   DPresley   Created
############################################################################

source ~/psft.env

DOMAIN=$1
# get the length of the parameter
n=${#DOMAIN}

# did they pass in a parameter?  it is the domain
if [ $n != 0 ]; then
   echo "Domain passed in as parameter: $DOMAIN"
else
  echo "No domain passed in. Look for single WLS Server domain."
  DOMAIN=`ls -l $PS_CFG_HOME/webserv | grep ^d | awk '{print $9}'`
  n=`echo $DOMAIN | wc -w`
  if [ $n != 1 ]; then
     echo "More than one domain directory found: $DOMAIN . Stopping run."
     echo "Count: $n"
     exit 1
  fi
fi

# is the domain set?
if { $DOMAIN = "" }; then
   echo "Domain not set. Stopping run."
   exit 1
fi

export $DOMAIN
export HOSTNAME=`hostname`

date
```

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

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

15.3 rsync Log and Report Files

Keep the standby report and job log file system in sync with the primary.

Data Guard manages replication of data stored in the database from the primary site to the standby. Certain file systems also need to be replicated – the application and middle tier software, and the file-based output generated by the application programs. The files holding output from application software need to be as close to in sync with the data inside the database as possible, so that if a site failover is required it is as easy as possible to resolve any differences between the state of data in the database and the reports and interfaces to other systems outside the database.

These rsync scripts replicate specific file systems from the primary to the standby, and manage the rsync direction through role transitions. We honored these basic requirements in the design:

- Keep the file systems holding the relatively unchanging application and middle tier software current at the standby
- Keep the data for file-based output as up to date as we can at the standby
- Allow file system replication to continue while using the standby for snapshot standby tests

To accomplish this, we need:

- A pair of OCI compute instances. This pair of compute instances must be able to access and mount the file systems being replicated. In addition, this pair should not host or run any PeopleSoft components, only Oracle client software, including SQL*Plus. To simplify your install, use the same Oracle client software that is used by PeopleSoft.
- This pair of OCI compute instances must have the psadm2 user created with the exact same uid and gid as the PeopleSoft middle tiers. Otherwise, permission issues will be encountered.
- OCI command-line interface (CLI) must be installed and configured on this pair of compute instances and on all compute instances hosting the PeopleSoft application server and process scheduler. This was done in Install OCI CLI, earlier in this document.

The rsync scripts will access the PeopleSoft database at the CDB level and will require the SYS password. The OCI region-local vault service must be provisioned at each site / region to securely store these database credentials (secrets).

IMPORTANT:

These scripts are built with the assumption that the PeopleSoft application and web tiers are deployed on shared file systems in OCI, e.g., on File System Service (FSS).

The scripts in PeopleSoft Startup / Shutdown Scripts run on the PeopleSoft application and web tiers. They do not run on the pair of compute instances that run only the rsync replication scripts.

The rsync scripts can run on the PeopleSoft application or web tiers, but doing so as normal practice prevents you from doing snapshot DR site testing, since the target "production" file systems would need to be unmounted.

The startPSFTAPP.sh and stopPSFTAPP.sh wrapper scripts in Wrapper Scripts, which run on the application tier compute instances, need to have access to the rsync scripts described here when performing a full stack switchover.

Do not replicate anything under $PS_CFG_HOME, the COBOL run-time environment, or the COBOL license manager and its database

Script design:

The core replication of a given file system is done by a simple script that replicates the contents of a directory to a mirrored location at the standby site.

In our implementation, two replication scripts are deployed in order to manage the push of file system changes at a rate appropriate to the rate of change at the source - the report logging and output being kept as current as possible via frequent execution, and the relatively unchanging software image less frequently. These scripts are executed as cron jobs.

The scripts are deployed at each region/site, and are designed to be running at both sites. The script's behavior depends on whether the rsync process is enabled or disabled, and whether the site is in the PRIMARY or STANDBY role. These scripts respond to role transitions and automatically change the direction of the replication.

The main replication scripts are:

- enable_psft_rsync.sh
- disable_psft_rsync.sh
- rsync_psft.sh

There is an environment file called psrsync.env at each site that the rsync scripts use, to limit the need for hard coding. It contains the following:

```
SCRIPT_DIR=<path to directory containing the replication scripts>

LOG_DIR=${SCRIPT_DIR}/log

USER=<file system owner - typically psadm2>

TNS_CONNECT_STRING=<TNS connect string alias to the PeopleSoft database at the local site>

TARGET_HOST=<IP address of remote rsync host>

COMPARTMENT_OCID=<OCID of compartment containing the region-local vault>

SECRET_NAME="<name of secret within region-local vault>"
```

There is a second set of environment files used to specify the directory structure being replicated and the name of the file where execution information will be logged. The name of this log file will be passed in as a parameter to the enable / disable scripts as well as to the rsync_psft.sh script. This file contains:

```
FS_ALIAS=<short name / alias for this file system for example, fs1>

LOG_FILE_NAME=<file to write rsync execution log info to, in $LOG_DIR>

SOURCE_RSYNC_DIR=<name of directory to replicate>

TARGET_RSYNC_DIR=<name of target directory to write to>
```

On our system, we have two file systems we are replicating, thus two of these second environment files. Our file fs1 holds:

```
FS_ALIAS=fs1

LOG_FILE_NAME=fs1_psft_rsync.log
```

SOURCE_RSYNC_DIR=/u02/app/psft/ps

TARGET_RSYNC_DIR=/u02/app/psft/ps

Our file fs2 holds:

FS_ALIAS=fs2

LOG_FILE_NAME=fs2_psft_rsync.log

SOURCE_RSYNC_DIR=/u01/app/psft/pt

TARGET_RSYNC_DIR=/u01/app/psft/pt

The replication scripts require the name of this run-specific environment file to be passed in on the command line - e.g.:

$ ./enable_psft_rsync.sh /<path>/fs1

Note the use of the ./ since we do not have the current script directory in the PATH.

The replication scripts call these additional scripts:

- get_site_role.sh
- get_db_session_count.sh

The get_site_role.sh script queries the database to determine the site role, which is used for determining replication direction. It uses the OCI CLI and SQL*Plus from the Oracle client software to access the PeopleSoft database at the local site. The get_db_session_count.sh script is used to determine when all PeopleSoft application and process scheduler sessions have shut down , also using OCI cli. This script is called by the stopPSFTAPP.sh script.

15.3.1 enable_psft_rsync.sh:

```
############################################################################

#!/bin/sh

# File name:    enable_psft_rsync.sh

#

# Description:  Set the environment up so that the file system will be

#               replicated

#

# Usage:        enable_psft_rsync.sh <fully qualified name of run env file>

#

# Errors:       No run environment file specified

#               Cannot find run environment file

#

# Revisions:

# Date       Who       What
# 7/1/2023   DPresley   Created
############################################################################

# Enable the PeopleSoft rsync job.

if [ $# -eq 0 ]
then
    echo "No run env file supplied.  Please provide a run env. file."
    echo "Usage:     enable_psft_rsync.sh <run env file>"
    echo "Example:   enable_psft_rsync.sh fs1"
    exit -1
fi

if  [ ! -f "$1" ]
then
     echo "File $1 does not exist."
     exit -1
fi

source $SCRIPT_DIR/psrsync.env
source $1

if [ -f "${SCRIPT_DIR}/.${FS_ALIAS}_rsync_disabled" ]
then
     rm -f ${SCRIPT_DIR}/.${FS_ALIAS}_rsync_disabled
     echo "PeopleSoft rsync job for ${FS_ALIAS} enabled."
else
     echo "PeopleSoft rsync job for ${FS_ALIAS} already enabled."
fi
```

15.3.2 disable_psft_rsync.sh

```
############################################################################
#!/bin/sh
# File name:     disable_psft_rsync.sh
```

```
#
# Description: Set the file system up to disable rsync.  Do this when
#              switching the roles of primary and standby sites
#
# Usage:        disable_psft_rsync.sh <fully qualified name of run env file>
#
# Errors:       No run environment file specified
#               Cannot find run environment file
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley  Created
############################################################################

if [ $# -eq 0 ]
then
    echo "No run env file supplied.  Please provide a run env. file."
    echo "Usage:     disable_psft_rsync.sh <run env file>"
    echo "Example:   disable_psft_rsync.sh fs1"
    exit -1
fi

if  [ ! -f "$1" ]
then
     echo "File $1 does not exist."
     exit -1
fi

source $SCRIPT_DIR/psrsync.env
source $1

if [ -f "${SCRIPT_DIR}/.${FS_ALIAS}_rsync_disabled" ]
then
     echo "PeopleSoft rsync job for ${FS_ALIAS} already disabled."
else
     touch ${SCRIPT_DIR}/.${FS_ALIAS}_rsync_disabled
     echo "PeopleSoft rsync job for ${FS_ALIAS} disabled."
fi
```

15.3.3 rsync_psft.sh

```
############################################################################

#!/bin/sh

# File name:    rsync_psft.sh

#

# Description:  rsync the file system / directory defined in the run env.
#               file.  If the -f is used, this will cause the script to
#               perform a FORCED rsync.  The -f option should only be used
#               in situations where a forced rsync is required such as when
#               a site failover is required but the file systems at both
#               sites are still intact.
#
# Usage:        rsync_psft.sh <fully qualified name of run env file> [ -f ]
#
# Errors:       No run environment file specified
#               Cannot find run environment file
#
# Revisions:
# Date      Who      What
# 7/1/2023   DPresley  Created
############################################################################

if [ $# -eq 0 ]
then
    echo "No run env. file supplied.  Please provide a run env. file."
    echo "Usage:     rsync_psft.sh <run env file> [ -F|f ]"
    echo "If a forced rsync is required, use -f."
    echo "Example:   rsync_psft.sh fs1"
    exit -1
fi

if  [ ! -f "$1" ]
then
     echo "File $1 does not exist."
     exit -1
fi

x=$2
FORCE_RSYNC=0
RSYNC_DISABLED=0
```

```
if [ ${#x} -ne 0 ]; then
     if [ $x = "-f" ] || [ $x = "-F" ]; then
          FORCE_RSYNC=1
     else
          echo "Invalid argument $x"
          exit 1
     fi
fi

source $SCRIPT_DIR/psrsync.env
source $1

pcount=0

# Check to see if rsync is disabled.

if [ -f "${SCRIPT_DIR}/.${FS_ALIAS}_rsync_disabled" ]
then
     RSYNC_DISABLED=1
fi

if [ ${RSYNC_DISABLED} = 1 ] && [ ${FORCE_RSYNC} = 1 ]; then

     proceed=""
     while [[ -z ${proceed} ]];
     do
          read -p "Rsync is disabled for ${FS_ALIAS} (${SOURCE_RSYNC_DIR}). OK to continue?
[Y|y|N|n] :" proceed
          [[ ${proceed} = 'Y' || ${proceed} = 'y' || ${proceed} = 'N' || ${proceed} = 'n' ]] &&
break
          proceed=""
     done
     if [[ ${proceed} = 'N' || ${proceed} = 'n' ]]; then
          echo "User response was N.  Exiting..."
          exit 0
     else
          echo "User response was Y.  Proceeding with FORCED rsync..."
     fi

fi

if [ ${RSYNC_DISABLED} = 1 ] && [ ${FORCE_RSYNC} = 0 ]; then
     date >> ${LOG_DIR}/${LOG_FILE_NAME}
     echo "PeopleSoft rsync is disabled for ${FS_ALIAS} (${SOURCE_RSYNC_DIR}). Re-enable with
enable_psft_rsync.sh." >> ${LOG_DIR}/${LOG_FILE_NAME}
     exit 0
fi

# If rsync is enabled and we are not forcing an rsync, check to see what role the site is in.
If not in the PRIMARY role, then exit.

if [ ${FORCE_RSYNC} = 0 ]; then
     SITE_ROLE=$( ${SCRIPT_DIR}/get_site_role.sh | egrep "PRIMARY|PHYSICAL STANDBY|SNAPSHOT
STANDBY" )
     if [ "${SITE_ROLE}" != "PRIMARY" ]; then
          date >> ${LOG_DIR}/${LOG_FILE_NAME}
          echo "This site is in the ${SITE_ROLE} role and not in the PRIMARY role.  Rsync will
not be performed."  >> ${LOG_DIR}/${LOG_FILE_NAME}
          exit 0
     fi
fi

# Run rsync on the file system/directory passed in to this script - unless one
# is already running for this file system.
# Exit if there is an rsync process already running for this file system.

pcount=$(ps -elf | grep "rsync -avzh --progress ${SOURCE_RSYNC_DIR}" | grep -v grep | wc -l)

if [[ ${pcount} -gt 0 ]]
```

```
then
    date >> ${LOG_DIR}/${LOG_FILE_NAME}
    echo "psft_rsync.sh is already running." >> ${LOG_DIR}/${LOG_FILE_NAME}
    exit 1
fi

date >> ${LOG_DIR}/${LOG_FILE_NAME}
[[ ${FORCE_RSYNC} = 0 ]] && echo "Site role is: ${SITE_ROLE}" >> ${LOG_DIR}/${LOG_FILE_NAME}
[[ ${FORCE_RSYNC} = 0 ]] && echo "Running rsync..."  >> ${LOG_DIR}/${LOG_FILE_NAME}
[[ ${FORCE_RSYNC} = 1 ]] && echo "Running rsync (FORCED)..."  >> ${LOG_DIR}/${LOG_FILE_NAME}
echo " FS Alias: ${FS_ALIAS} "  >> ${LOG_DIR}/${LOG_FILE_NAME}
echo " Source: ${SOURCE_RSYNC_DIR} "  >> ${LOG_DIR}/${LOG_FILE_NAME}
echo " Target: ${TARGET_RSYNC_DIR} " >> ${LOG_DIR}/${LOG_FILE_NAME}
echo "" >> ${LOG_DIR}/${LOG_FILE_NAME}

#time rsync -avzh --progress ${SOURCE_RSYNC_DIR}/ ${USER}@${TARGET_HOST}:${TARGET_RSYNC_DIR}/ |
tee -a ${LOG_DIR}/${LOG_FILE_NAME}
( time rsync -avzh --progress ${SOURCE_RSYNC_DIR}/ ${USER}@${TARGET_HOST}:${TARGET_RSYNC_DIR}/ )
>> ${LOG_DIR}/${LOG_FILE_NAME} 2>&1
echo "##################################" >> ${LOG_DIR}/${LOG_FILE_NAME}

exit 0
```

15.3.4 get_site_role.sh

```
############################################################################
#!/bin/sh
# File name:    get_site_role.sh
#
# Description: Determine whether this is the primary or a standby site
#
# Notes:       This script requires oci cli be installed on the server it runs on.
#
# Usage:       get_site_role.sh
```

```
#
# Errors:      Can return a database error (ORA-XXXXX) if database is unavailable or access failure.
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley  Created
############################################################################

source ~/psft.env
source $SCRIPT_DIR/psrsync.env

date +"%d-%b-%Y %T"
echo "Running get_site_role.sh"
echo ""

export SECRET_OCID=$(oci vault secret list -c $COMPARTMENT_OCID --raw-output --query "data[?\"secret-name\" == '$SECRET_NAME'].id | [0]")
export PSFT_SECRET=$(oci secrets secret-bundle get --raw-output --secret-id $SECRET_OCID --query "data.\"secret-bundle-content\".content" | base64 -d )

sqlplus -s /nolog  <<EOF!
connect sys/${PSFT_SECRET}@${TNS_CONNECT_STRING} as sysdba
set heading off
set feedback off
select rtrim(database_role) from v\$database;
exit
EOF!
```

15.3.5 get_db_session_count.sh

```
############################################################################
#!/bin/sh
# File name:    get_db_session_count.sh
#
```

```
# Description:  Get the number of user connections from the application
#
# Parameters:   None
#
# Output:       Returns only the number of sessions.
#
# Errors:       Can return a database error (ORA-XXXXX) if database is unavailable or access
failure.
#
# Notes: This script requires oci cli be installed
#
# Revisions:
# Date       Who         What
# 7/1/2023   DPresley    Created
############################################################################


source ~/psft.env
source $SCRIPT_DIR/psrsync.env


export SECRET_OCID=$(oci vault secret list -c $COMPARTMENT_OCID --raw-output --query
"data[?\"secret-name\" == '$SECRET_NAME'].id | [0]")
export PSFT_SECRET=$(oci secrets secret-bundle get --raw-output --secret-id $SECRET_OCID --query
"data.\"secret-bundle-content\".content" | base64 -d )


sqlplus -s /nolog  <<EOF!
connect sys/${PSFT_SECRET}@${TNS_CONNECT_STRING} as sysdba
set heading off
set feedback off
select ltrim(count(*))
from gv\$instance a, gv\$session b
where a.inst_id = b.inst_id
and service_name in ('HR92U033_BATCH','HR92U033_ONLINE')
/
exit
EOF!
```

15.3.6 Example Cron Job Entries

In this project, we are replicating two FSS file systems:

- /u02/app/psft/ps, defined in the file fs1, that contains job logs and the report repository.
- /u01/app/psft/pt, defined in the file fs2, that contains the PeopleSoft PeopleTools software installation.

The file system for fs1 needs to match the contents of the database as closely as possible, thus needs to have a higher frequency of replication. In the below crontab entry, it is set to every 5 minutes (*/5) for the minutes column.

The file system for fs2 is more static and is set up to replicate once a day, at 2:00 AM (0 2).

```
*/5 * * * * psadm2 /u01/app/psft/pt/custom_admin_scripts/rsync_psft.sh
/u01/app/psft/pt/custom_admin_scripts/fs1

0 2 * * * psadm2 /u01/app/psft/pt/custom_admin_scripts/rsync_psft.sh
/u01/app/psft/pt/custom_admin_scripts/fs2
```

15.4 Wrapper Scripts

These wrapper scripts coordinate the execution of the basic scripts above. They are designed to be called from external components such as Full Stack DR cloud service but can be run manually if needed.

NOTE: If you need to stop or start PeopleSoft components without enabling or disabling the rsync replication process, then use the stand-alone scripts in Application Domain Server.

15.4.1 startPSFTAPP.sh

```
############################################################################
#!/bin/sh
# File name:    startPSFTAPP.sh
#
# Description:  Call the scripts to start the application server and process
#               scheduler on this server.
#               Start the rsync process for both types of file systems being
#               replicated.
#
# Usage:        startPSFTAPP.sh
#
# Errors:
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley  Created
############################################################################

source $SCRIPT_DIR/psrsync.env
```

```
PS_DOMAIN=HR92U033

# Set the process scheduler report distribution node before starting the
# app and process scheduler.
# DO NOT run set_ps_rpt_node.sh in the background.  The set_ps_rpt_node.sh
# script must complete before startPS.sh and startAPP.sh scripts are executed.

$SCRIPT_DIR/set_ps_rpt_node.sh
$SCRIPT_DIR/startPS.sh $PS_DOMAIN  &
$SCRIPT_DIR/startAPP.sh $PS_DOMAIN &

# Enable the rsync scripts.
$SCRIPT_DIR/enable_psft_rsync.sh $SCRIPT_DIR/fs1
$SCRIPT_DIR/enable_psft_rsync.sh $SCRIPT_DIR/fs2
```

15.4.2set_ps_rpt_node.sh

The set_ps_rpt_node.sh script will set the distribution node for the PeopleSoft report repository. This is a site specific setting as it is based on the name of the PIA web server node at each site. Below is the ps_rpt.env file that is used by the set_ps_rpt_node.sh script. Place both the script and the ps_rpt.env files at each sites. Edit the ps_rpt.env file according to each site.

```
ps_rpt.env

# The following environment variables are used by the set_ps_rpt_node.sh script.
#
# Modify the following environment variables accordingly.
# Set the RPT_URL_HOST to the distribution hostname.network-domain of one of a PIA web servers
e.g., myhost.mycompany.com
RPT_URL_HOST=phx-psft-hcm-web01.appprivad1.maacloud2vcn.oraclevcn.com
# Set RPT_URI_PORT to the http or https port of the PIA web server.
RPT_URI_PORT=8080
# SITE_NAME is the PIA web deployment site typically 'ps'.
SITE_NAME=ps
# PSFT_DOMAIN is set per the product.  For HCM, it is HRMS.
PSFT_DOMAIN=HRMS
# Set the PDB_NAMNE to the name of the Pluggable Database Name in which the PeopleSoft schema is
stored.
PDB_NAMNE=HR92U033
# Set SCHEMA_NAME to the database schema name within the pluggable database wherre the
PeopleSoft schema is stored.
SCHEMA_NAME=EMDBO

# Adjust the following two environment variables IF AND ONLY IF required.  Otherwise, leve them
as they are set.
# If SSL is enabled on the PIA web server, then you will need to change the protocol scheme to
https for both URL and RPT_URI.
```

```
# NOTE: if SSL termination is at the load balancer, then the protocol should be set to http.
URL="http://${RPT_URL_HOST}:${RPT_URI_PORT}/psreports/${SITE_NAME}"
RPT_URI="http://${RPT_URL_HOST}:${RPT_URI_PORT}/psc/${SITE_NAME}/EMPLOYEE/${PSFT_DOMAIN}/c/CDM_R
PT.CDM_RPT.GBL?Page=CDM_RPT_INDEX&Action=U&CDM_ID="


############################################################################
#!/bin/sh
# File name:    set_ps_rpt_node.sh
#
# Description: Sets the distribution node for the process scheduler.  Called by startPSFTAPP.sh.
#              This script is site-specific and should be modified according to each site.
#
# NOTE:        OCI CLI must be installed for this script to run.
#
# Usage:       set_ps_rpt_node.sh
#
# Errors:      Can return a database error (ORA-XXXXX) if database is
#              unavailable or there is access failure.
#
# Revisions:
# Date       Who        What
# 7/1/2023   DPresley   Created
############################################################################

source ~/psft.env
source $SCRIPT_DIR/psrsync.env
source $SCRIPT_DIR/ps_rpt.env

date +"%d-%b-%Y %T"
echo "Running set_ps_rpt_node.sh"
echo ""

export SECRET_OCID=$(oci vault secret list -c $COMPARTMENT_OCID --raw-output --query
"data[?\"secret-name\" == '$SECRET_NAME'].id | [0]")
export PSFT_SECRET=$(oci secrets secret-bundle get --raw-output --secret-id $SECRET_OCID --query
"data.\"secret-bundle-content\".content" | base64 -d )

echo "URL = ${URL}"
echo "RPT_URI =  ${RPT_URI}"

# Update the PS_CDM_DIST_NODE table to set the site specific report
# distribution node.  There should only be one row in this table.
sqlplus -s /nolog  <<EOF!
connect sys/${PSFT_SECRET}@${TNS_CONNECT_STRING} as sysdba
alter session set container = ${PDB_NAMNE};
set heading off
set define off
set feedback on
```

```
update ${SCHEMA_NAME}.PS_CDM_DIST_NODE
   set URL = '${URL}',
       URI_HOST = '${RPT_URL_HOST}',
       URI_PORT = ${RPT_URI_PORT},
       URI_RPT = '${RPT_URI}';

commit;

exit
EOF!
```

15.4.3stopPSFTAPP.sh

```
############################################################################
#!/bin/sh
# File name:   stopPSFTAPP.sh
#
# Description: This script shuts down PSFT app servers and process
#              schedulers.
#              It is also integrated with the rsync scripts.
#
#              Note: Because this deployment of PeopleSoft uses a shared file
#              system for interface files, job logs, and reports, only one
#              rsync process should be running, and only one final execution
#              after all app servers and process scheduler processes across
#              all nodes have completed their shutdown.
#
#              We use a simple lock file on the shared file system to
#              manage that process. The first script in creates the lock
#              file.  That session will also run the final rsync, then
#              will remove the lock file.
#
#              NOTE: If you do not want to run rsync but only shut down
#              either the app servers or the process scheduler, use the
#              individual scripts:
#              stopAPP.sh and stopPS.sh
#
# Errors:
```

```
#
# Revisions:
# Date       Who       What
# 7/1/2023   DPresley   Created
############################################################################

source $SCRIPT_DIR/psrsync.env
PS_DOMAIN= HR92U033

if [ -f "${SCRIPT_DIR}/psftrsync.lck" ]
then
     SKIP_RSYNC=1
else
     hostname > ${SCRIPT_DIR}/psftrsync.lck
     SKIP_RSYNC=0
fi

# Stop application server and process scheduler.
$SCRIPT_DIR/stopPS.sh  $PS_DOMAIN &
$SCRIPT_DIR/stopAPP.sh $PS_DOMAIN &

# If SKIP_RSYNC is 0, we must wait until all sessions have been shut down.
# We can then do one final rsync.

if [ ${SKIP_RSYNC} -eq 0 ]
then
  echo "Checking number of remaining sessions before performing rsync...."
  SESSION_COUNT=1
  while [ ${SESSION_COUNT} -ne 0  ];
  do
     SESSION_COUNT=$(${SCRIPT_DIR}/get_db_session_count.sh | sed 's/[^0-9]*//g')
     echo "Number of remaining sessions: " ${SESSION_COUNT}
     sleep 3
  done

# Do one final rsync then disable rsync. If there is an existing rsync
# process running, wait until the rsync process completes.
# We need to source the fs1 file to get the SOURCE_RSYNC_DIR env
# variable set.

   source $SCRIPT_DIR/fs1

   pcount=1
   while [ $pcount -gt 0 ];
   do
        pcount=$(ps -elf | grep "rsync -avzh --progress ${SOURCE_RSYNC_DIR}" | grep -v grep | wc
-l)
        sleep 3
   done
   ${SCRIPT_DIR}/rsync_psft.sh $SCRIPT_DIR/fs1
   ${SCRIPT_DIR}/disable_psft_rsync.sh $SCRIPT_DIR/fs1
   ${SCRIPT_DIR}/disable_psft_rsync.sh $SCRIPT_DIR/fs2
   rm -f ${SCRIPT_DIR}/psftrsync.lck
fi
```

15.4.4 startPSFTWEB.sh

```
############################################################################

#!/bin/sh

#

# File name:    startPSFTWEB.sh

#

# Description:  Start the Coherence*Web cache server then the PIA web server

#

# Usage:        startPSFTWEB.sh

#

# Errors:

#

# Revisions:

# Date       Who

# 7/1/2023   DPresley

############################################################################

PS_DOMAIN=HR92U033


# Start the Coherence*Web cache server first.

$SCRIPT_DIR/startCacheServer.sh $PS_DOMAIN


# Start the PIA web server.

$SCRIPT_DIR/startWS.sh $PS_DOMAIN


# Don't return control until Coherence and the web server have started

wait
```

15.4.5stopPSFTWEB.sh

```
############################################################################
```

```
#!/bin/sh
#
# File name:    stopPSFTWEB.sh
#
# Description: Stop the PIA web server then the Coherence*Web cache server
#
# Usage:       stopPSFTWEB.sh
#
# Errors:
#
# Revisions:
# Date       Who
# 7/1/2023   DPresley
############################################################################

PS_DOMAIN=HR92U033

# Stop the PIA web server
$SCRIPT_DIR/stopWS.sh $PS_DOMAIN

# Stop the Coherence*Web cache server
$SCRIPT_DIR/stopCacheServer.sh $PS_DOMAIN
```

Disclaimer

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

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

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