dd

Web and Application Tier on Oracle Private Cloud Appliance / Private Cloud at Customer

Oracle Private Cloud Appliance / Private Cloud at Customer Provides the Ideal Platform for Hosting Web and Application Tier

ENGINEERED SOLUTIONS WHITE PAPER / MARCH 22, 2020

PURPOSE STATEMENT

This document describes features and advantages for running web tier and application tier software on the Oracle Private Cloud Appliance (PCA). It focuses on methodology and best practices for migrating to PCA applications deployed on Oracle Exalogic.

Contents

INTRODUCTION

The Oracle Private Cloud Appliance (PCA) has emerged as the premier platform for WebLogic, Fusion Middleware and other application tier software, often in conjunction with Oracle Exadata - the premier database platform. Many of those applications have been deployed on Oracle Exalogic or commodity x86 servers, and are now being deployed on PCA for increased performance, scale, and manageability.

This white paper discusses advantages of moving those applications to PCA, describes how to migrate them, with emphasis on Exalogic workloads, and describes deployment methods and best practices. This paper emphasizes platform and performance characteristics of the PCA X8 over previous PCA versions. The methodology and architectural principles apply to all versions.

ADVANTAGES OF THE PRIVATE CLOUD APPLIANCE

The Oracle Private Cloud Appliance (PCA) is an Oracle Engineered System designed for application tier. PCA is an integrated hardware and software system that reduces infrastructure complexity and deployment time for virtualized workloads in private clouds. It is a complete platform for a wide range of application types and workloads, with built-in management, compute, storage and networking resources. PCA provides excellent performance and other system properties for a broad range of applications.

The Private Cloud Appliance is also available as the Private Cloud at Customer (PCC), a solution for on-premises private cloud that includes the PCA and Oracle services. Customers acquire PCC on a subscription basis, with Oracle operating the infrastructure so the customer can focus on applications. Except where noted, ‘PCA’ will be used in this document to describe either form of the product.

The PCA platform is ideal for both Oracle and third-party applications, with benefits especially suitable for WebLogic Fusion Middleware and similar application tier products. There are several reasons why this is so effective:

- Private Cloud Appliance provides ‘quick time to value’ for a robust virtualization platform, going from first power-up to starting VMs in a matter of hours. PCA automatically discovers hardware components and configures them to work with one another, reducing design and administrative effort, eliminating potential errors, and speeding time to application deployment. PCA’s automated configuration implements Oracle best practices for optimal performance and availability.

- Private Cloud Appliance provides high performance high speed 100Gb Ethernet, Oracle ZFS Storage Appliance ZS7-2, and Oracle X8-2 compute nodes, providing performance and scale improvements over previous product generations. See sections below for further description of the physical infrastructure.

- Private Cloud Appliance design eliminates single points of failure on management, network, storage, and compute resource, and permits 'zero-downtime' rolling upgrades to system infrastructure.
- Pre-built Oracle VM virtual appliances and templates quickly enable application instances. This is complementary to Private Cloud Appliance quickly provisioning physical infrastructure. You can see a list of pre-built virtual appliances at https://www.oracle.com/virtualization/technologies/virtual-appliances.html
- High performance inter-VM networking using the Private Cloud Appliance internal networks permits low-latency, high bandwidth, private communication between VMs in a clustered application. This is especially useful for clustered applications like WebLogic and Coherence, and frameworks like Kubernetes. Multiple private networks based on VLANs or custom PCA networks can provide independent isolated networks, and are ideal for hosting multiple application clusters on the same PCA. Each network carries traffic private to each cluster, without need to prevent IP address collision or data leakage between applications.
- Private Cloud Appliance provides load balancing (Dynamic Resource Scheduling - DRS) and High Availability (HA) features that provide higher performance and automates recovery from outages.
- Private Cloud Appliance can provide Infrastructure as a Service (IaaS) cloud function via Oracle Enterprise Manager 13c. Oracle Enterprise Manager is highly recommended for cloud function, administrative control and reporting.
- Application orchestration and automated workload deployment can be performed with Ansible or the Oracle VM API and scriptable command line interface.
- Customers can use Trusted Partitioning or Hard Partitioning to manage software license costs.
- PCA supports heterogeneous computing on Oracle Linux, Oracle Solaris, other Linux distributions, and Windows. This increases operational efficiency and ROI by letting the same system platform be used for multiple workloads, instead of requiring separate ones.

Oracle Private Cloud Appliance and Oracle Private Cloud at Customer fully support Oracle Linux Cloud Native Environment, including Oracle Container Runtime for Docker and Oracle Container Services for Use with Kubernetes. They provide an ideal runtime for Oracle WebLogic Server applications to run in Docker and Kubernetes with full, integrated system support from Oracle. We recommend that customers running Oracle WebLogic Server applications on Oracle Exalogic Elastic Cloud systems, and wishing to adopt cloud native infrastructure and DevOps practices, migrate to Oracle Private Cloud Appliance and Oracle Private Cloud at Customer.

The following sections discuss these items in further detail, and describe how to migrate applications from Exalogic to PCA.

Virtualization and cloud platform

The Private Cloud Appliance provides virtualization life cycle management using Oracle VM. Oracle VM consists of two parts: Oracle VM Manager and Oracle VM Server. These are included and pre-installed on PCA without additional license fees, representing a substantial cost savings over other virtualization platforms. Oracle Enterprise Manager is optional but highly recommended, and is available without additional cost for providing cloud capabilities.

Oracle VM Manager is an advanced and widely used virtualization management product for controlling multiple servers, VMs, networks and storage resources under a graphical browser user interface. Oracle VM Manager also provides programmable REST APIs and command line interfaces to permit automation. Each PCA has one active instance of Oracle VM Manager used as a central control plane to administer the entire PCA.

Oracle VM Server is a high performance hypervisor that runs virtual machines, based on commands sent from Oracle VM Manager. Oracle VM Server is automatically installed and configured on every compute node, both when a PCA rack is installed, and when additional compute nodes are added. Oracle VM efficiently runs virtual machines, which may run Oracle Linux, Oracle Solaris, other Linux versions, and Microsoft Windows, and the applications these operating systems support.

Oracle VM provides advanced functions that benefit applications.

- Administrators can set anti-affinity rules to ensure that virtual machines comprising a clustered application such as WebLogic Server or Coherence do not run on the same physical server, insulating them from single points of failure.
- High Availability (HA) settings can automatically reboot VMs if they or the server they are running on crash. For example, HA can bring up a WebLogic administrative instance or Kubernetes master after an outage.
- Live migration allows you to move running VMs from one server to another without stopping the VM and interrupting its applications. This is typically used for load balancing and to perform compute node maintenance without virtual machine outage,
- Dynamic Resource Scheduling (DRS) policies can provide load balancing among server based on several parameters, including CPU and network utilization according to policy settings. DRS automatically live migrates VMs from heavily loaded servers to less loaded ones to permit optimal performance.
- Resource management policies control CPU allocation for differential service between applications sharing the same servers, including CPU caps, and share-based priorities. This permits higher degrees of CPU oversubscription without affecting the service level objectives for production applications running alongside less-critical applications.
- Administrators can define tenant groups to dedicate PCA compute nodes to different clients. This ensures dedicated resources for applications or departments. PCA tenant groups leverage standard Oracle VM server pools, with additional automation and default configuration.

PCA customers can optionally use Oracle Enterprise Manager 13c to provide Infrastructure as a Service (IaaS) cloud capabilities, Oracle Enterprise Manager provides comprehensive management capabilities, including role-based access control, monitoring and chargeback. PCA can also leverage Ansible to provide “infrastructure as code”.

The Private Cloud Appliance includes the following components.

Management nodes

PCA uses two latest generation Oracle servers X8-2 function as the management nodes for Oracle Private Cloud Appliance X8. They serve as an active-passive cluster for management operations, providing resiliency in case of planned outage or server failure. Oracle VM Manager and other management functions run on the active management node. When a management node assumes the active role it takes over a virtual IP address (VIP) address, so clients of the management interface do not need to know which management node is currently active.

Integrated ZS7-2 Storage

Oracle Private Cloud Appliance features a fully integrated, enterprise-grade Oracle ZFS Storage Appliance ZS7-2 for centrally storing the management environment, virtual machine images and application data, providing superior performance and efficiency.

With dual ZS7-2 controllers in a high-availability (HA) configuration and one high-capacity DE-24C storage shelf, Oracle Private Cloud Appliance now includes 100 TB of customer usable storage in the rack. This storage can scale to 4.4 PB (petabytes) raw capacity (2.2PB mirrored), using a combination of DE3-24C or all-flash DE3-24P expansion storage trays depending on capacity and performance requirements. Further details on the Oracle ZFS Storage Appliance are at https://www.oracle.com/storage/nas/zs7-2/

Compute Nodes

Oracle Server X8-2 compute nodes in the Private Cloud Appliance provide the virtualization platform. Compute nodes run Oracle VM Server and provide processing power and memory capacity for virtual machines under Oracle VM Manager's control.

Each X8-2 compute node server has two 24-core Intel Xeon 8260 processors, and can be ordered in three different memory configurations - 384 GB, 768 GB, and 1.5 TB. With a 45% performance improvement over the previous compute node generation, Oracle Server X8-2 provides the optimal balance of CPU cores, memory, and I/O throughput for mission-critical enterprise applications. Customers can scale from 2 to 25 compute nodes in the same rack.

An automated provisioning process orchestrated by the active management node configures compute nodes into the Oracle VM environment. Private Cloud Appliance software installs Oracle VM Server software on each compute node, defines their network configurations, and places all compute nodes into an Oracle VM server pool.

Best practice: Oracle VM only uses a small percentage of the memory and CPU on each compute node, leaving the remainder for virtual machines. For capacity planning purposes, consider each X8-2 compute node as providing 48 cores for virtual machines and applications. CPU cores can be over-subscribed, with more virtual CPUs than physical CPUs on the server, which permits higher VM density for cloud operations, especially if VMs have different idle and peak periods. If CPUs are oversubscribed, administrators should use resource controls to ensure that important VMs get preferential service during periods of high CPU consumption.

Best practice: Spare CPU capacity should be provisioned for peak loads and to provide “N+1” redundancy to provide sufficient capacity when a server is down for maintenance or other reasons.

PCA administrators can optionally define “tenant groups”, which isolate compute and storage resources in separate Oracle VM server pools that can be assigned to different customers, to provide dedicated resources.

Network Infrastructure

The PCA network is an important differentiator from previous systems. The Private Cloud Appliance relies on “wire once” Software Defined Networking (SDN) that permits multiple isolated virtual networks to be created on the same physical network hardware components.

PCA physical networking

PCA networking uses 100Gb Ethernet leaf and spine switches. These offer 100Gb Ethernet connectivity for communication between internal PCA components and allow flexible 10/25/40/100GbE connectivity to data center networks. This provides a performance improvement over the 40Gb

InfiniBand internal networks and 10GbE external connectivity on earlier Oracle engineered systems, including earlier PCA generations, and faster networking to external hosts. PCA networks exceed the performance of the 1Gb or 10Gb Ethernet networks typically seen on commodity servers.

PCA uses redundant physical network hardware components, pre-cabled at the factory, to help ensure continuity of service during maintenance or in case of a failure.

Network connectivity

The Private Cloud Appliance provides external network access for connectivity to a datacenter's networks. The PCA connects to the datacenter network via a pair of next-level switches, also referred to TOR (top of rack) switches. This provides resiliency against a single point of failure. Software Defined Networks (SDN) based on the physical network devices connect virtual machines and bare metal servers to networks, storage and other virtual machines, maintaining the traffic separation traditionally provided by hard-wired connections. Optional custom external networks further isolate traffic and maximize bandwidth.

The PCA uses private, "internal" networks that are not exposed to the customer's datacenter network. This provides isolation, security, and the ability to use pre-defined IP address ranges for each networked component without conflict with existing datacenter network addresses. PCA uses internal networks for appliance management, storage access, and inter-VM communication. Every PCA rack component has a predefined IP address. Oracle storage, management and compute nodes have a second IP address for Oracle Integrated Lights Out Manager (ILOM) connectivity.

Compute nodes connect to the internal networks and to the customer datacenter networks. Oracle VM Server on each compute node communicates over Private Cloud Appliance internal networks for management, storage, heartbeat and live migration. By default, compute nodes do not have IP addresses on the customer datacenter network, which increases their isolation and reduces attack surface. Optional custom networks can give compute nodes IP addresses on the customer network, provide additional bandwidth and traffic separation, and be used to present Ethernet-based storage to each compute node.

Virtual Machine Networking

Guest virtual machines access the customer datacenter network using an Oracle VM network named "`default_external`", An additional virtual machine network named "`default_internal`" is internal to the Private Cloud Appliance and used for private network traffic between virtual machines. In previous PCA versions, the corresponding networks were named `vm_public_vlan` and `vm_private`. These networks are pre-defined in Oracle VM Manager with the "Virtual Machine" function (also called a "channel") - indicating they are used for guest VM TCP/IP traffic and not cluster management, storage, or live migration. This ensures that guest VMs do not see infrastructure network traffic.

PCA administrators can define VLANs based on the interfaces used for these networks, to comply with a datacenter's network standards and to permit traffic isolation. For example, a datacenter standard might require VM traffic be on VLANs 100 to 150, and an administrator would define separate networks with those VLAN tags. Private VLANs can also be built to isolate traffic between different virtual machines. Virtual machines do not need configuration to use VLANs – the use of VLANs is transparent to the virtual machine. This makes it easy to clone VMs or move them to different VLAN networks without having to change the virtual machines' internal VLAN configuration.

Best Practice: Application clusters such as WebLogic, Coherence, Kubernetes, and Oracle Real Application Clusters (RAC) benefit from PCA's internal VM networks, which provide high-performance, low-latency networking. Perform network traffic between members of an application cluster over a virtual network device based on `default_internal`. That is faster than sending inter-VM traffic to datacenter switches and provides better performance and isolation. This is ideal for Coherence and Java application servers like WebLogic, which have high rates of message traffic and depend on fast message delivery. This offers an improvement over the 40Gb networks available on Exalogic

Best Practice: If there are several application clusters, define VLAN-based internal virtual machine networks for each application based on the bond used for `default_internal`. That provides fast, low latency communication between members of an application cluster and insulates them from IP address collision or data leakage. You can have multiple independent application clusters without having to coordinate their IP address assignments or host names.

The result is that the PCA infrastructure stack replaces the Exalogic infrastructure stack.

MIGRATING WEBLOGIC APPLICATIONS FROM EXALOGIC TO PCA

There are two approaches to migrating applications from Exalogic to PCA: fresh OS install and "lift and shift", described later in this document. The result is that the Exalogic infrastructure stack is replaced by the PCA infrastructure stack

Figure 1. Exalogic stack with Exalogic Control and InfiniBand optimizations

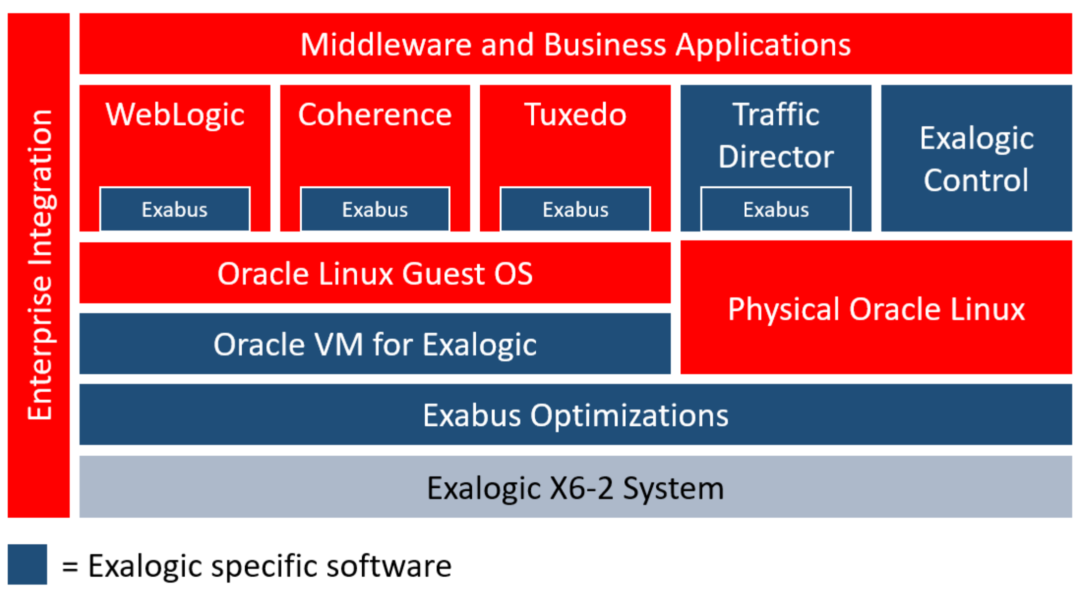
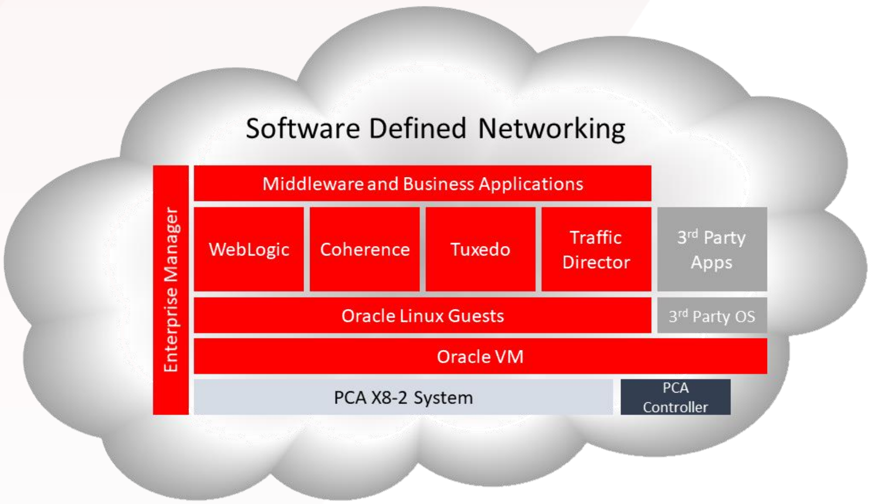


Figure 2. PCA infrastructure stack with Oracle VM and Enterprise Manager



In general, applications can be migrated without changes. Customers should consider the following general comparisons between the environments when planning migrations and evaluate whether they want to create a new environment that matches the old one as much as possible, or use the migration as an opportunity for an application and OS technical refresh as well as a hardware refresh. These considerations apply:

- Oracle WebLogic Server 10.3.6 and 12.1.3 versions are nearing end of life. New features such as updated REST support, JSON processing, auto-scaling and REST management in Oracle WebLogic Server 12.2.1.X enable better integration with cloud systems. Customers using prior versions should plan to migrate to 12.2.1.3 or later as part of the migration process.
- The migration will change the underlying compute infrastructure used by applications
- Although Oracle Traffic Director (OTD) is supported for migration to Oracle Private Cloud Appliance, native Kubernetes load balancers such as Traefik and Voyager are more appropriate for Kubernetes, and are recommended as replacements for Oracle Traffic Director. Simple load balancers can be implemented by using `haproxy`, which is included with Oracle Linux.
- Access to external systems via HTTP and T3 protocol is supported, including access to databases, and Oracle RAC clusters running in Oracle Exadata systems. SDP protocols are not supported on Oracle Private Cloud Appliance, so any existing usage of SDP within domains running on Oracle Exalogic Cloud systems must be removed. That is consistent with current recommendations for applications running on Exalogic.
- Oracle WebLogic Server, Oracle Coherence, and Oracle Application Development Framework are supported for use in Kubernetes with the WebLogic Kubernetes tools.

- See the Licensing Information User Manual at https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.3/fmwlc/exalogic-elastic-cloud-software.html#GUID-FC2CC864-79A9-455A-8ADD-DEE80C05D72E for migration of Oracle Exalogic Elastic Cloud Software licenses.

If deploying to Kubernetes, see the white paper "Oracle WebLogic Server on Oracle Private Cloud Appliance and Kubernetes" at https://www.oracle.com/a/ocom/docs/engineered-systems/oracle-webLogic-server-on-pca.pdf

Fresh OS install method

This method deploys new virtual machines containing WebLogic Server or other application software, and then migrate application contents (binaries, scripts, tools) from current instances. This also applies to applications currently running on commodity servers.

The advantage of this approach is that it is a software technology refresh to complement the hardware technology refresh. This permits a more significant transformation and modernization of the application environment, including the opportunity to modernize applications, run them inside Docker containers and use Kubernetes. PCA fully supports container-based application delivery using Oracle Cloud Native Environment, representing the modern trend for delivering application systems.

The administrator would download the latest Oracle Linux OS and application versions from Oracle Software Delivery Network at https://edelivery.oracle.com to perform a fresh installation of the OS and application, and copy application data onto it.

This method brings the software stack fully up to date; however, this may require additional analysis of the current environment to capture the current system's contents.

Customers often prefer to move application environments to new hardware with as few changes as possible, as described in the next section.

Migrating to PCA with 'lift and shift' methodology

This paper focuses on a method often described as "lift and shift", which moves existing environments to PCA with as few changes as possible. This section describes how to download existing physical or virtual machines on Exalogic, then upload and transform them to operate on PCA.

STEP 1: Collect configuration from running system

First, collect userids and passwords to permit logging into the current OS instance. While logged in, collect network and disk mount details by issuing commands "`ifconfig -a`" and "`df -h`". Note details of any NFS shares or naming service (NIS, LDAP) in use, since they will be needed after the image is implemented on PCA. If those services are private to the Exalogic, they would be replaced by services available to the PCA, potentially in virtual machines.

STEP 2: Obtain OS Image

The next step is to copy the OS images and descriptive information from the current Exalogic system.

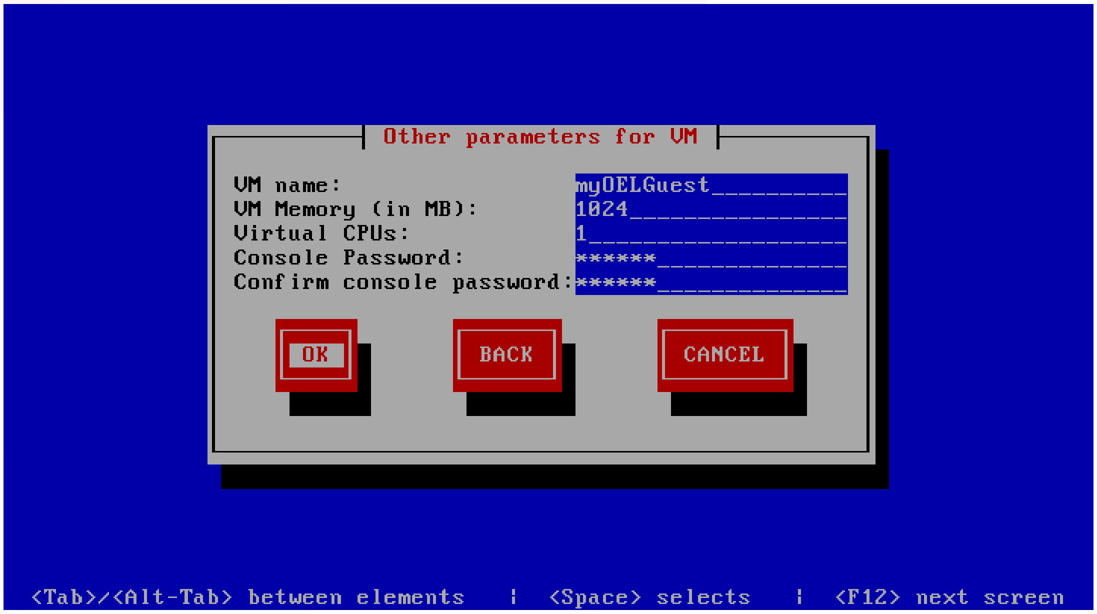
Physical OS images

Use the Oracle VM Physical to Virtual (p2v) utility to create an uploadable VM image from a physical OS instance. The utility is described in the Oracle VM Administrator's Guide for Release 3.4 at https://docs.oracle.com/cd/E64076_01/E64083/html/vmadm-ptov-intro.html

Cleanly shut down the OS instance. Then boot the Oracle VM install CD or ISO image on the physical server and enters ‘p2v’ at the first boot prompt. The utility will start up and present a panel asking the administrator which disks to include in the VM image. Select all of them unless you know there are unused disks in the physical machine.

The P2V utility will prompt for network information, and will present a screen requesting details about the new VM, as shown in Figure 3.

Figure 3: Virtual Machine Parameters



P2V will then display the network address of a web server it creates. Point a web browser to the IP address to download the `vm.cfg` file and disk image “`.img`” files that will comprise the VM. When that is complete, continue with Step 2 below, to upload the VM image into the Private Cloud Appliance.

Virtual OS images

When obtaining a virtual Exalogic image, start by logging into Oracle VM Manager on the Exalogc and getting the virtual machine details from the VM’s `vm.cfg` file, and locate the VM’s disk images.

Next, from the Oracle VM Manager user interface collect the VM descriptive information, as shown in Figure 4 and the text in Example 1 below. This includes the VM name, shape (number of virtual CPUs, memory size), disk and network devices.

Figure 4. Virtual machine details on Exalogic

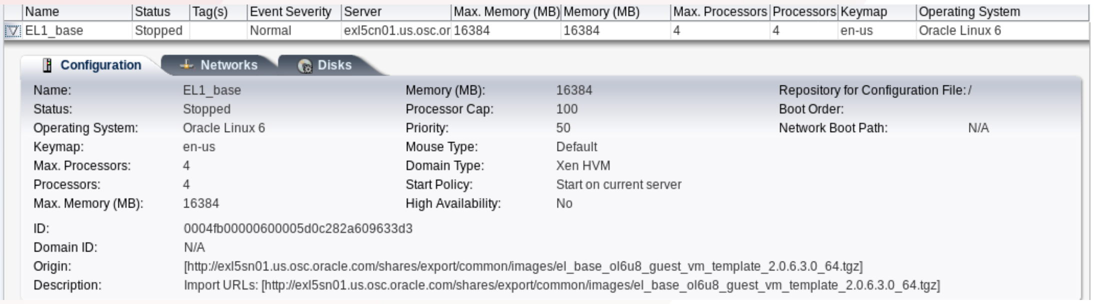


Figure 4 shows the `vm.cfg` location. Log into one of the Exalogic compute nodes and display its contents as in the Example 1. Note the highlighted lines, which describe the virtual disk and network configurations and Exalogic-specific details. The example shows the same VM displayed in Figure 3, with the name “EL1_base”, a single Ethernet virtual NIC, 16GB of RAM, 4 virtual CPUs, and a single virtual disk.

Example 1. VM description file for VM on Exalogic

cat \
/OVS/Repositories/0004fb0000030000fb8bfb7348173fea/VirtualMachines/0004fb00000600005d0c282a609633d3/vm.cfg
kernel = '/usr/lib/xen/boot/hvmloader'
vif = ['mac=00:21:f6:7e:30:85,bridge=xenbr0']
OVM_simple_name = 'EL1_base'
vnclisten = '127.0.0.1'
vncconsole = 1
serial = 'pty'
disk = ['file:/OVS/Repositories/0004fb0000030000fb8bfb7348173fea/VirtualDisks/0004fb0000120000b912f85b31cb3130.img,hda,w']
vncunused = 1
uuid = '0004fb00-0006-0000-5d0c-282a609633d3'
on_reboot = 'restart'
EL_Template_Version = 1
cpu_weight = 27500
pae = 1
memory = 16384
cpu_cap = 0
maxvcpus = 4
OVM_high_availability = False
acpi = 1
timer_mode = 2
on_poweroff = 'destroy'
vnc = 1
OVM_os_type = 'Oracle Linux 6'
expose_host_uuid = 1
on_crash = 'restart'
exalogic_ipoib = [{'pkey': ['0x8004'], 'port': '1'}, {'pkey': ['0x8004'], 'port': '2'}]
apic = 1
name = '0004fb00000600005d0c282a609633d3'
guest_os_type = 'linux'

```
device_model = '/usr/lib/xen/bin/qemu-dm'
builder = 'hvm'
vcpus = 4
keymap = 'en-us'
exalogic_vnic = [{'pkey': ['0xffff'], 'guid': '0xbbb34d724b05dd73', 'port': '1'}, {'pkey': ['0xffff'], 'guid': '0xbbb34d724b05dd74', 'port': '2'}]
OVM_cpu_compat_group = ''
OVM_domain_type = 'xen_hvm'
```

Note the lines that describe the VM's virtual disks, network, CPU configuration, domain type, and lines for Exalogic features "`exalogic_ipoib`" and "`exalogic_vnic`".

Stop the VM to ensure stable disk contents, then use `scp` or `rsync` to copy disk image files listed in the "`disk =`" line to a host that has a web server and is on a network reachable from PCA. If you don't have a web server, select a system that hosts Python, open a terminal window, '`cd`' to the directory used for the copied VM disk ".img" files and issue the command "`python SimpleHttpServer 8000`" as shown in Example 2 below. That creates a web server you can use in the import step. Port 8000 is used in this example since it is unlikely to be in use. If that port number is in use you will see an error message "Address already in use". Just pick another port number and adjust the import URL accordingly.

Example 2. Using Python for an impromptu web server

```
$ ls vm.cfg *img
0004fb0000120000b912f85b31cb3130.img  vm.cfg
$ python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
10.80.108.245 - - [26/Dec/2019 12:39:57] "GET /0004fb0000120000b912f85b31cb3130.img HTTP/1.0" 200
10.80.108.245 - - [26/Dec/2019 12:51:38] "GET /vm.cfg HTTP/1.0" 200 -
```

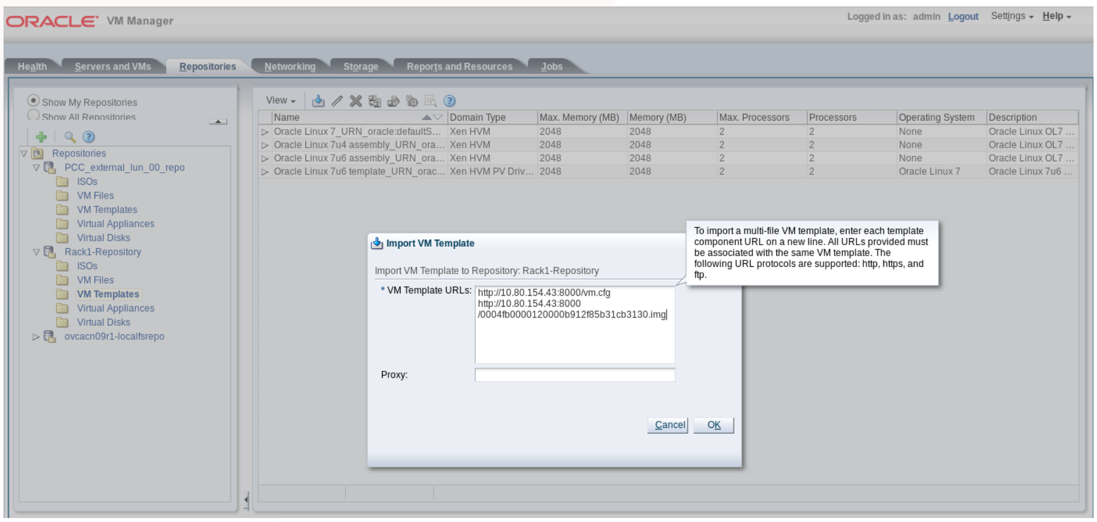
STEP 3: Upload Exalogic Virtual Machine into PCA

There are several ways to upload the Exalogic VM image into PCA. One way is to upload it as a template, which can be cloned to make multiple VMs.

First, remove Exalogic-specific details from the `vm.cfg` file that describes the virtual machine. Copy `vm.cfg` to the same directory as the VM disk image files. Then use an ASCII text editor to remove the "`exalogic_ipoib`" and "`exalogic_vnic`" lines highlighted above. This is a good time to remove the "`vif`" line in `vm.cfg` because it describes a network on Exalogic, not on PCA. If the "`vif`" line is not removed, there will be an "undefined" network in Oracle VM Manager after the import, which can be removed at that time. The files can optionally be combined into a single tar file using a command like "`tar cf MyTemplate.tgz vm.cfg *.img`"

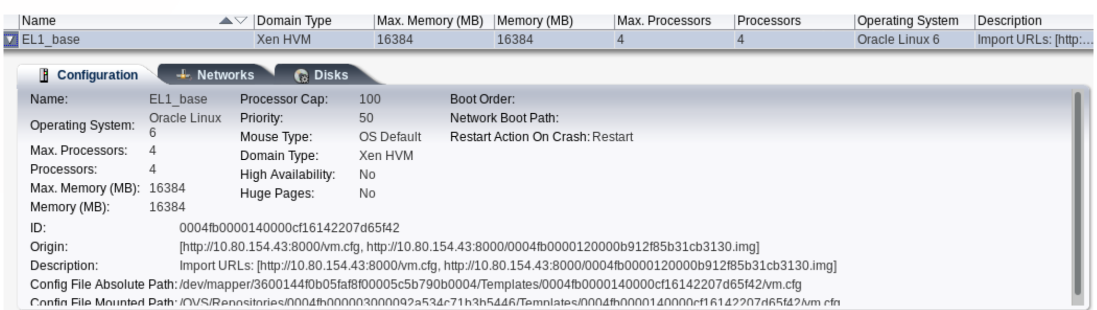
After the files are under a web server, log into Oracle VM Manager on the PCA, click on the Repositories tab, expand the desired target repository (typically Rack1-Repository) and click on "VM Templates". Then click on the "Import" icon and fill in details for the configuration and images files in the pop-up box, as shown in Figure 5. If a tar file was created, upload that file instead.

Figure 5. Import VM image as a Template



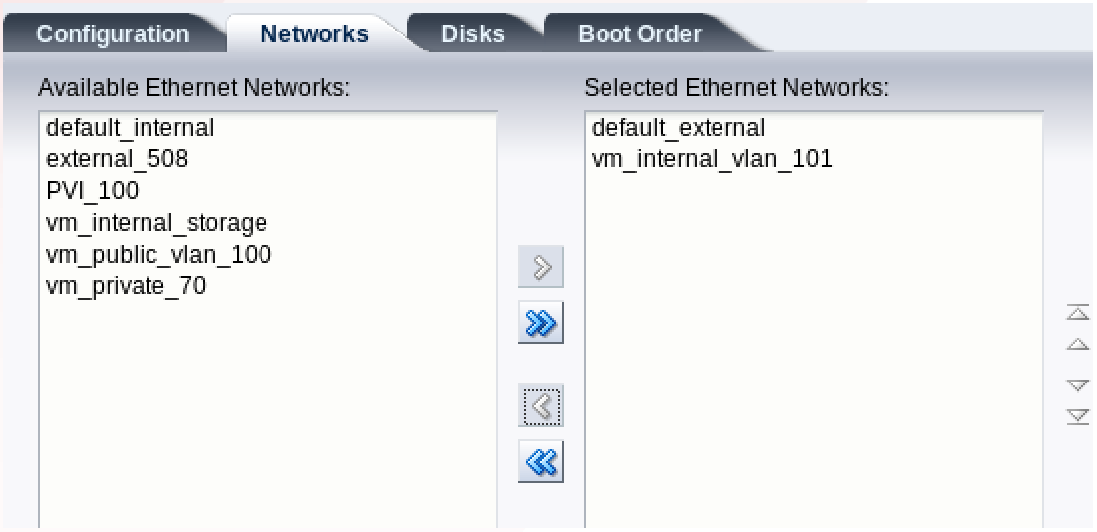
This uploads the files named in the pop-up box. When this job completes, you will have a VM template you can use as the basis for creating VMs, shown in Figure 6.

Figure 6. Imported Template



Edit the template using the Oracle VM user interface to fit site, in particular the network names. The convention used in this paper is that the first virtual network interface is used for external networks, and the second virtual network will be used for private, inter-VM communication. Figure 7 shows the “edit template” popup dialogue from Oracle VM Manager

Figure 7. Pop-up dialogue to select networks for template



The resulting template can now be cloned to create new virtual machines. See the Oracle VM Administrator's Guide at https://docs.oracle.com/cd/E64076_01/E64083/html/index.html for information on cloning VMs.

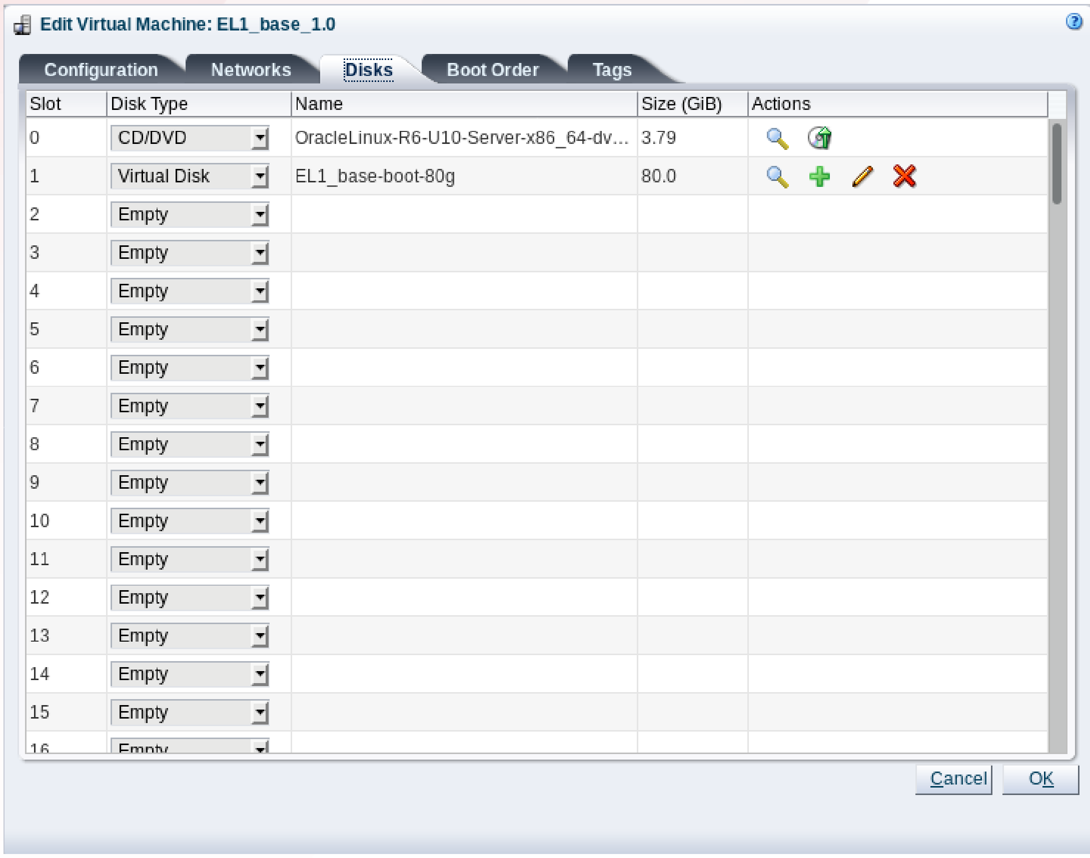
An alternative is to import the virtual disk images and then create new virtual machines that use them. In that case, use a similar import step to upload the disk images, then go to the "Servers and VMs" tab, select the desired server pool (typically Rack1_ServerPool) and click on the "Create VM" icon. In the dialogue there, use the same VM shape (number of CPUs, memory size) and networks as in the original VM, and select the just-uploaded virtual disk images.

A useful method is to add an Oracle Linux boot CD ROM ISO image to the virtual machine, and boot into it to perform the transformation step below. Edit the VM, move to the Disks tab and add a CDROM ISO image as shown in Figure 8. Use the Boot Order tab to make CD/DVD the first boot device.

It can be even more convenient to create a utility VM running Oracle Linux, and temporarily add the imported Exalogic VM's disks to it: Boot up the utility VM and then mount the imported VM's disks. For example, assuming the imported boot disk is the utility VM's second disk:: `mkdir /mnt/sysimage ; mount /dev/xvdb1/ /mnt/sysimage`). A pre-defined utility VM based on Oracle Linux would help automate transformation: it would have an IP address defined so you could login and not rely on the VM console, and you could create scripts in advance for the transformation steps below.

Or, simply boot up the VM using the imported disk, and work from the VM console.

Figure 8. Disk tab showing CD-ROM image

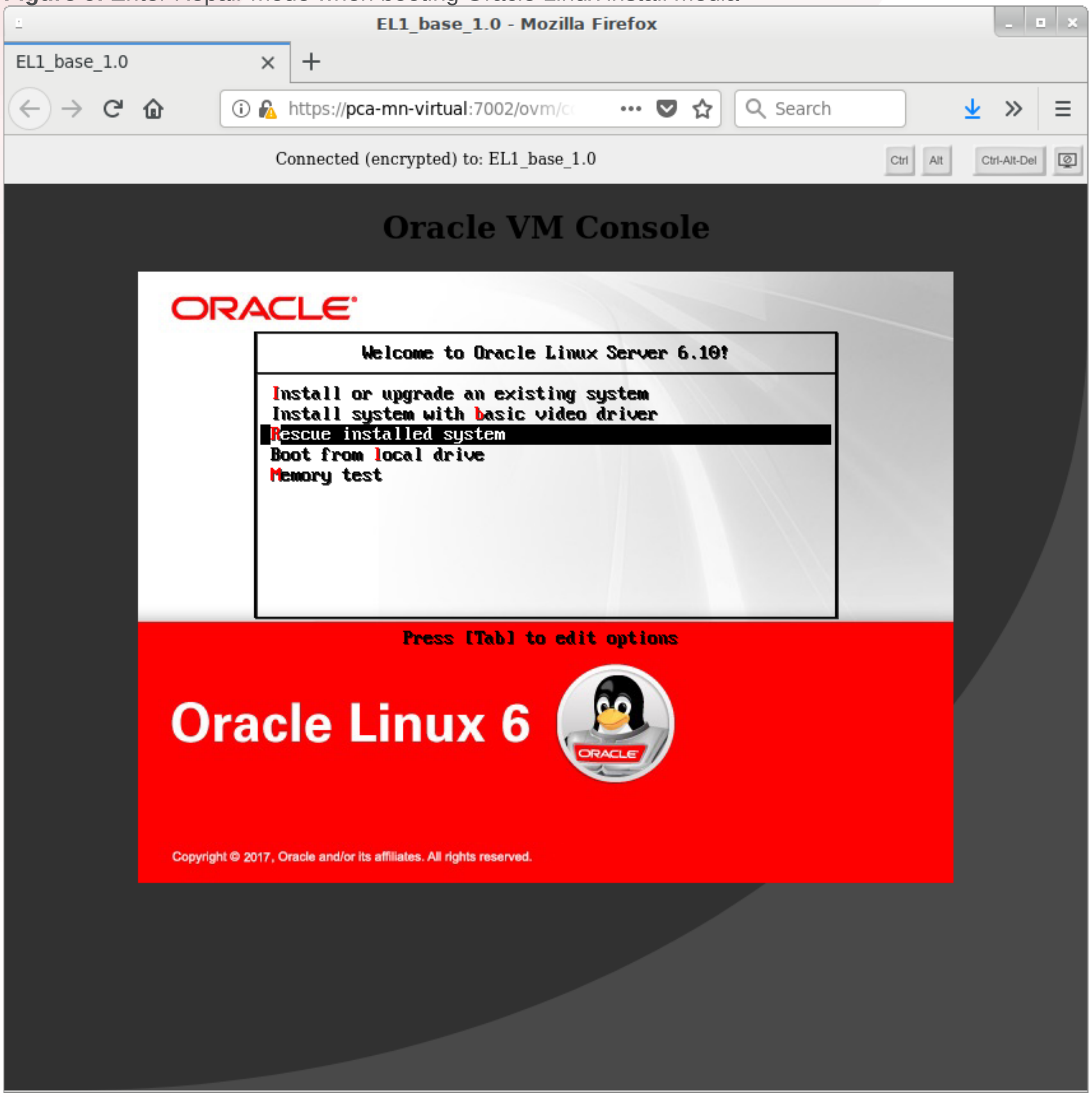


At this point, you have a VM that corresponds to the VM on Exalogic. The following steps will remove Exalogic-specific customization so it can work on PCA.

STEP 4: Transform Software and Configuration inside the VM

Boot up the VM, either from the Oracle Linux install CD image or from the VM's boot disk, and launch the virtual machine console from the Oracle VM user interface. If booting the install CD image, select the "Rescue Installed System" option and proceed through the prompts to have the imported VM's boot disk mounted at `/mnt/sysimage`. You can `chroot` to that directory to simplify typing. If booting from the imported disk images, expect network failures for the InfiniBand devices that are not present, and for the different network definition.

Figure 9. Enter Repair mode when booting Oracle Linux install media



At this point, remove the Exalogic OS components. Assuming that we've booted from the install media or utility VM boot disk, and that the imported OS image is at `/mnt/sysimage`, perform the following steps. If simply booting the imported VM boot disk, leave out the `chroot` commands.

1. Remove the Infiniband RPM used with Exalogic. First, display the RPMs as shown here

```
# chroot /mnt/sysimage /bin/bash -c "rpm -qa \"infinibus*\""
infinibus-1.1-226.x86_64
```

   then remove the rpm that was found in the above line

```
# chroot /mnt/sysimage /bin/bash -c "rpm -ev infinibus-1.1-226.x86_64”
```

2. Remove or move away the network definition files for the Infiniband devices:

```
# cd /mnt/sysimage/etc/sysconfig/network-scripts
# rm ifcfg-ib0.0004; rm ifcfg-ib1.0004
```

3. Remove the OFED component:

```
# chroot /mnt/sysimage /bin/bash -c "/usr/sbin/ofed_uninstall.sh --force"
```

Now exit from the repair mode: unmount `/mnt/sysimage` and exit from the repair shell if it was used, then shutdown.

At this point, bring up the virtual machines and configure virtual networks to your datacenter standards, creating `/etc/sysconfig/network-scripts/ifcfg-eth`* files to bring up the VM's public and private networks. They should be different from the Exalogic VMs to permit parallel testing.

If Weblogic is in use, connect to the WLS console, and turn off the Exalogic enhancements by reversing the steps used to turn them on for Exalogic, in particular disabling Socket Direct Protocol (SDP). This is a current best practice for Exalogic, and is not supported for PCA. See the Oracle Fusion Middleware Exalogic Enterprise Deployment Guide at https://docs.oracle.com/cd/E18476_01/doc.220/e18479/optimization.htm#ELEDG779

When Exalogic VMs are identical other than hostname and IP address, you can stop the VM and clone it for additional members of the application cluster. Boot each VM up in turn to change its hostname and address to make them unique.

MIGRATING WITH ORACLE ENTERPRISE MANAGER

The preceding examples showed how to import and transform an Exalogic virtual machine by using Oracle VM Manager. This section will demonstrate using Oracle Enterprise Manager to set up a VM for an Infrastructure as a Service (IaaS) end user.

Oracle Enterprise Manager (OEM for short) works with PCA and Oracle VM to provide cloud services, and enhances management, compared to Oracle VM Manager, by providing Role Based Access Control, in which different user roles have different privileges. This provides security, auditability, and insulation from error. Users with administrative privileges can administer the system, while end users have simpler screens that display only the items they own and control.

In this example, an administrator will import a template based on an Exalogic virtual machine into a software library, and an end user (an "IaaS user") will clone the template to make a VM, and then launch it. In the interest of brevity, some of the screens used for this process are omitted.

Import template into a software library

Oracle Enterprise Manager uses software libraries that are associated with different roles. Figure 10 shows the screen an administrator would have when about to add to the software library.

Figure 10. Create Virtualization Entity in Software Library

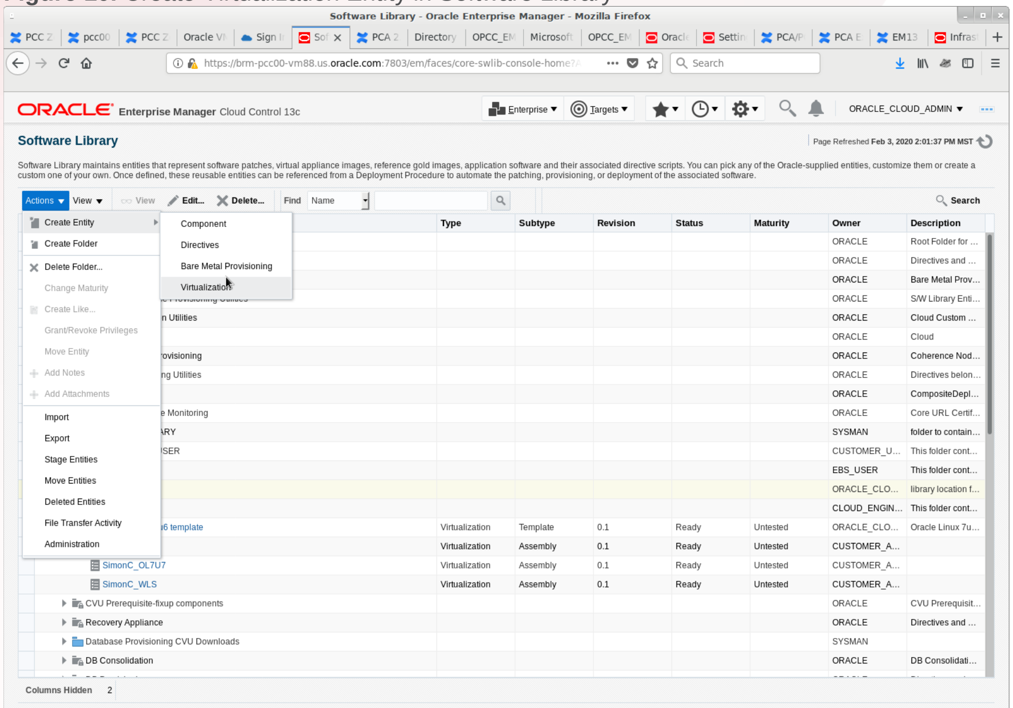
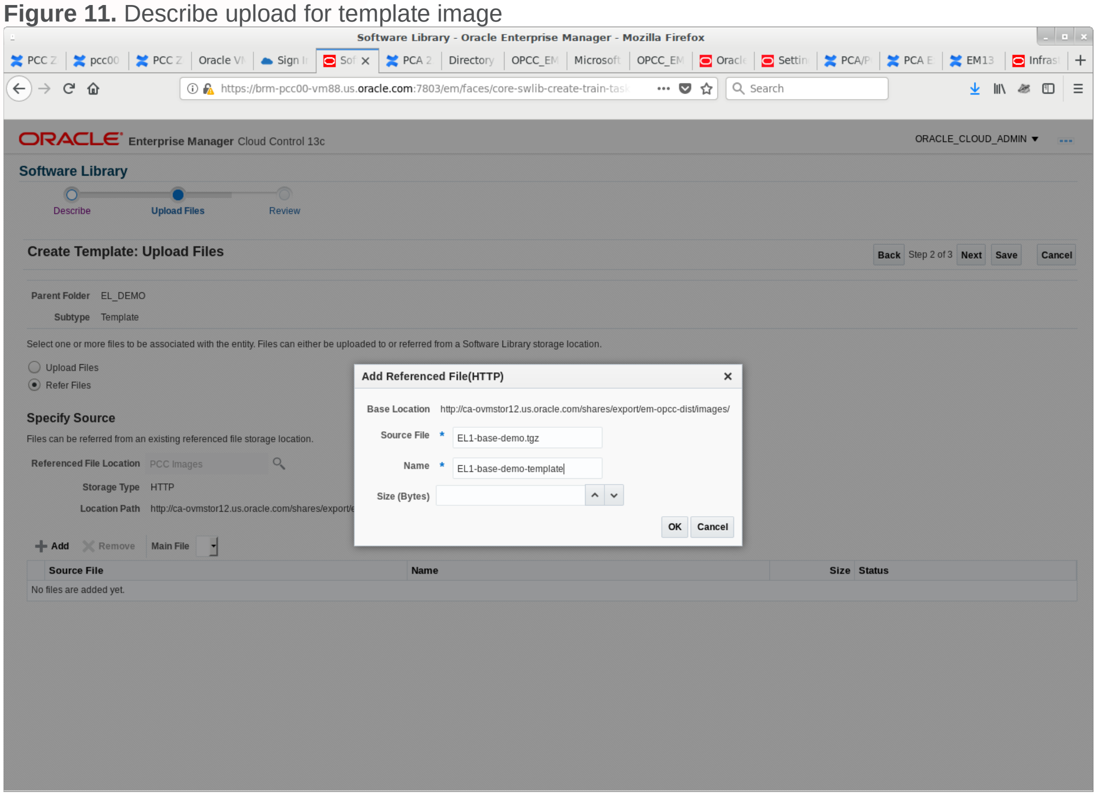


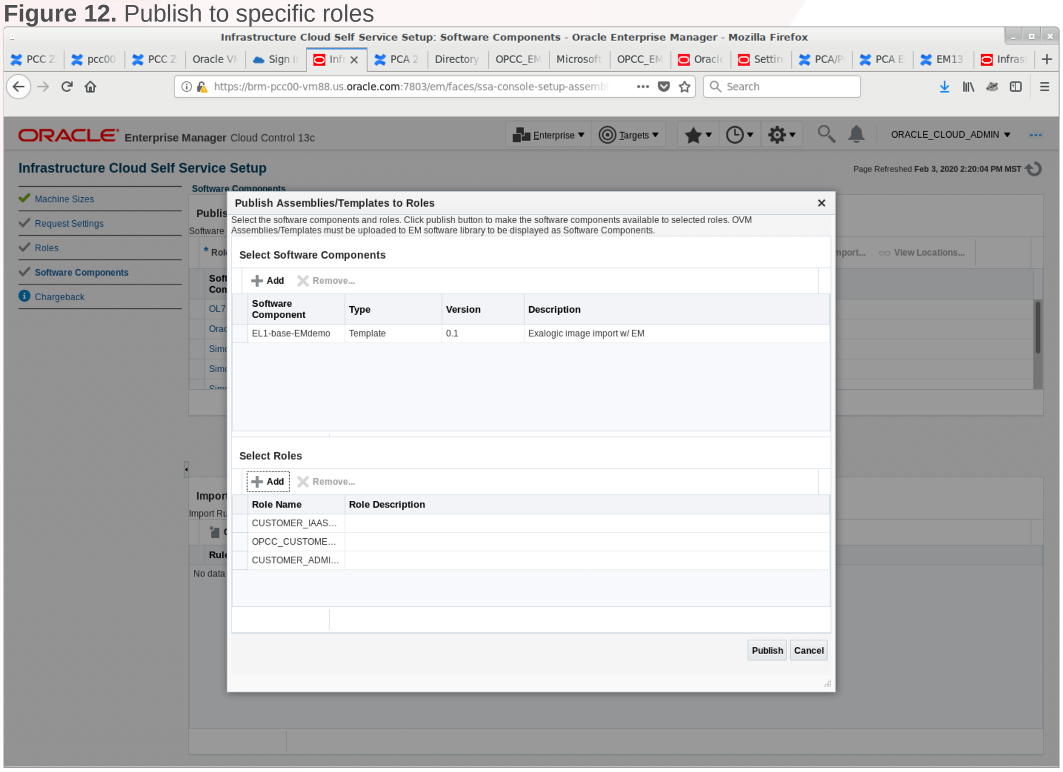
Figure 11 shows the part of the dialog, used to specify upload details.

Figure 11. Describe upload for template image



The next step, shown in Figure 12 publishes the new template to roles – in effect, to the specific users who can use it.

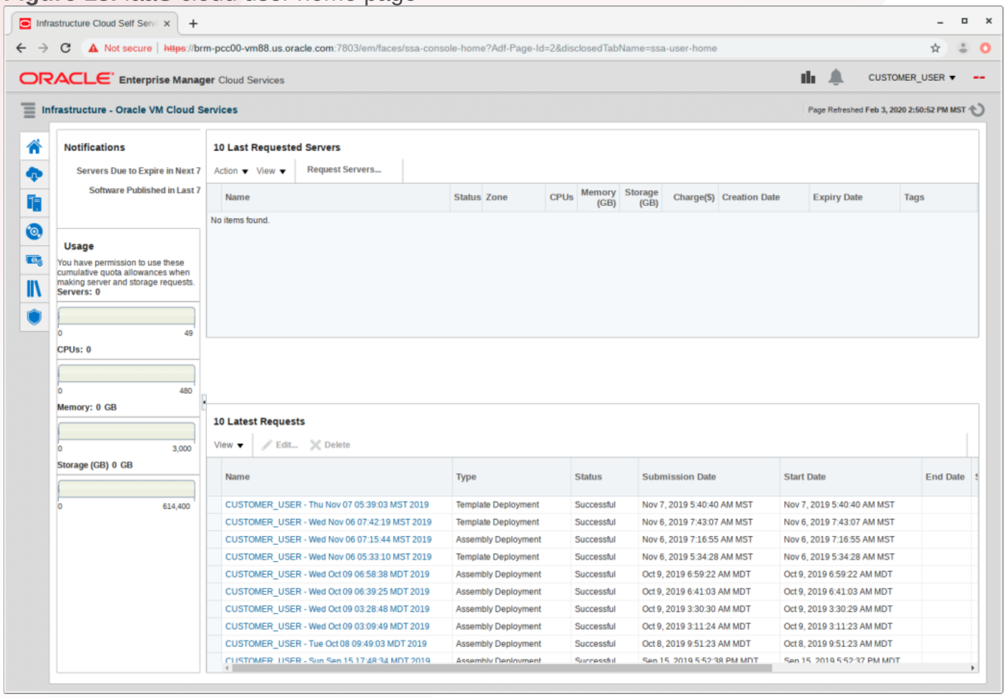
Figure 12. Publish to specific roles



One the template is published, the end user – referred to as an “IaaS cloud user” -- sees the template in their private software library, and can request a VM from it, subject to their quota.

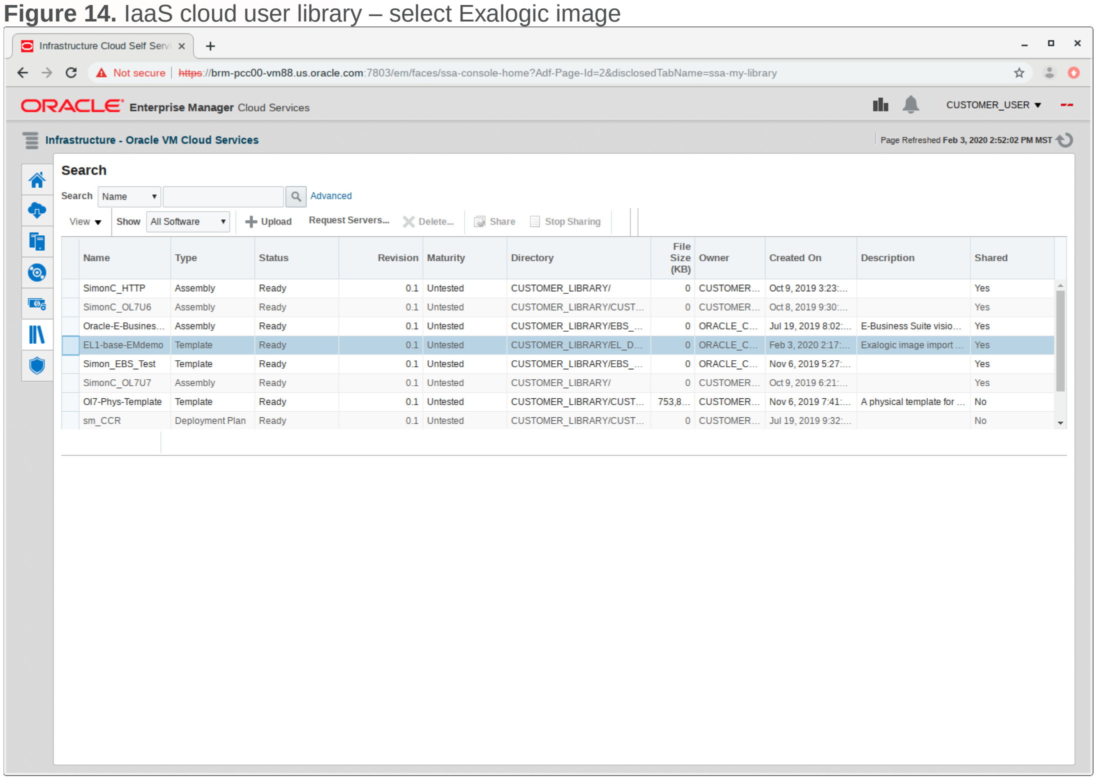
The cloud user’s home page shown in Figure 13 provides simple navigation to a home screen showing requested servers (Enterprise Manager’s word used for a virtual machine.

Figure 13. IaaS cloud user home page



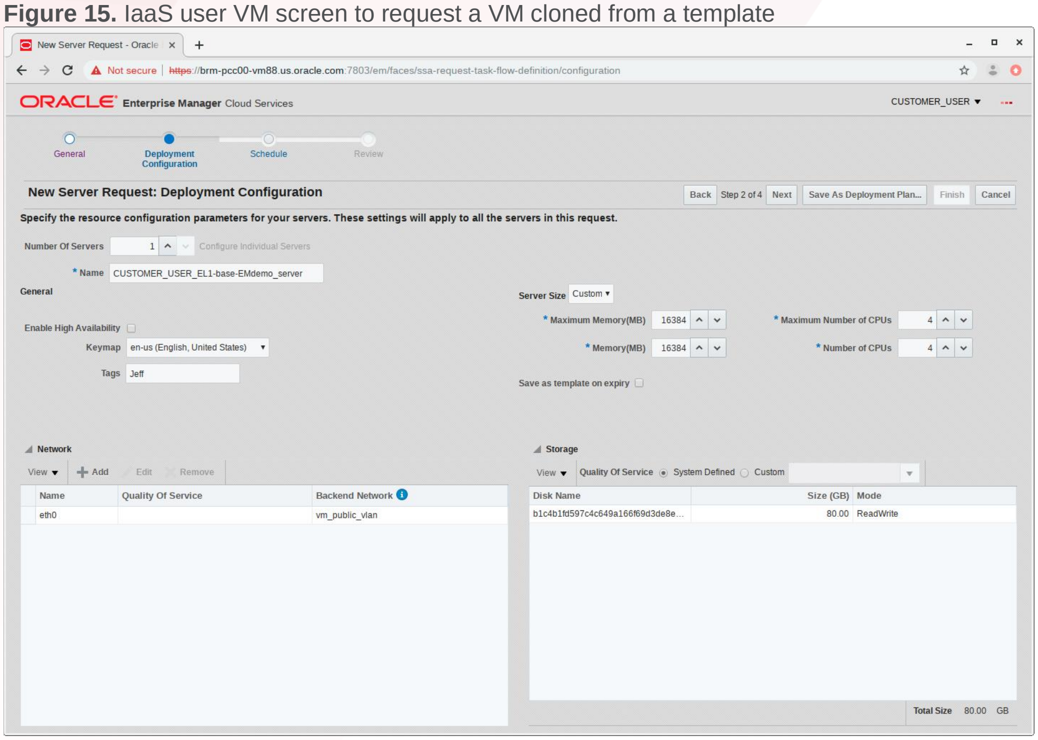
From this page the user navigates to the library (the icon looking like books on a bookshelf, with the rightmost book leaning over) and selects the desired images as shown in Figure 14, and clicks “Request Servers”

Figure 14. IaaS cloud user library – select Exalogic image



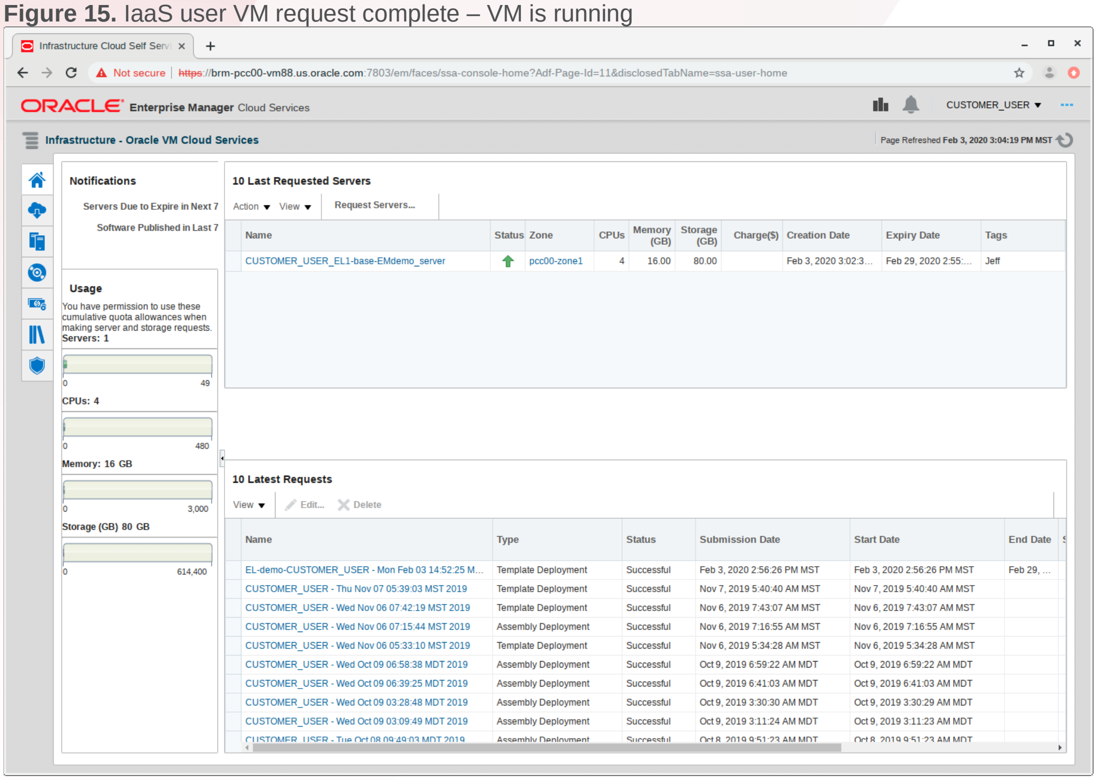
The “Request Servers” action starts a series of screens to describe the requested VMs: the number of VMs to create,, when to create them (now, or schedule in the future), and their networks, CPU and memory configuration. Part of that dialog is shown in Figure 15.

Figure 15. IaaS user VM screen to request a VM cloned from a template



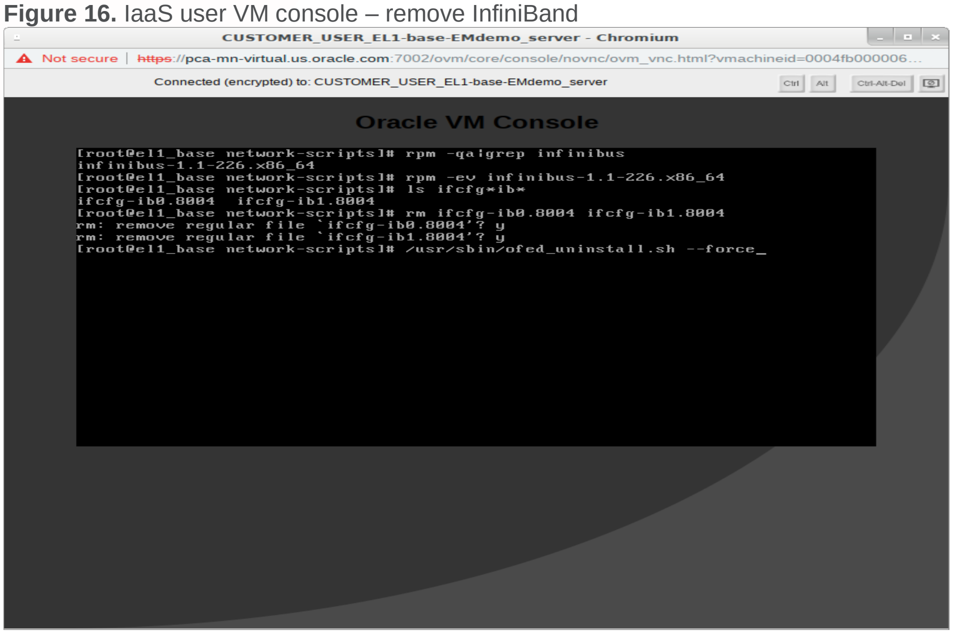
The last screen of the dialogue submits the request, which then executes by cloning the template to a VM (or more than one if requested) and launches it. Figure 16 shows the completed request.

Figure 15. IaaS user VM request complete – VM is running



The user can connect to the VM's console (select the VM with mouse, and click on Action tab), and then configure the VM as described earlier to remove Exalogic specific features like InfiniBand networks. In this example, we simply boot off the imported virtual machine's disk, shown in Figure 16.

Figure 16. IaaS user VM console – remove InfiniBand



Once the InfiniBand features are removed and the VM given an IP address, the user can start operating the VM.

BEST PRACTICES FOR APPLICATIONS MIGRATED FROM EXALOGIC TO PCA

This section describes best practices for applications migrated from Exalogic. These recommendations also apply to applications migrated from other platforms, and completely new applications:

- Keep software levels up to date with current versions, to reduce security exposures and incorporate bug fixes. This is a general Best Practice, but sometimes omitted when moving an application to a new platform since there is often a goal to “change as little as possible”. At the very least, current patch levels should be installed. For example, if Exalogic applications use Oracle Linux 5 virtual machines, update them to the most recent Oracle Linux 5 version if upgrading to a later release of Oracle Linux is not planned during migration.
- Use the Unbreakable Enterprise Kernel on Oracle Linux guests for best performance, and install the Oracle VM Guest Additions to improve operational flexibility. Guest Additions are described at https://docs.oracle.com/cd/E64076_01/E64083/html/vmadm-guestadd.html
- Use PCA internal networks for inter-VM communication. This provides better performance and isolation. Use different VLANs for independent applications so they can operate without interference or the need to coordinate network design.
- Set HA attributes on key virtual machines, such as WebLogic admin servers or Kubernetes masters. HA setting automatically brings the VMs up after an outage. Note that HA settings require that administrators stop virtual machines from Oracle VM Manager or Oracle Enterprise Manager rather than performing a VM shutdown from within the guest. That prevents a “false positive” in which Oracle VM considers a VM stop as an outage.
- Establish anti-affinity rules to ensure that virtual machines that comprise an application cluster are not on the same compute node. This prevents ‘Single Point of Failure” where a cluster member and its backup are on the same server and can be impacted by the same failure
- Consider oversubscribing virtual CPUs onto physical CPU resources if that is consistent with your service level requirements and the resource demands of your applications. This is very useful for increasing VM density on private clouds, but requires understanding application requirements and monitoring performance.
- If oversubscribing CPUs, use Oracle VM resource controls to ensure the most important virtual machines get sufficient CPU access.
- Provision sufficient physical capacity to support expected compute needs, and leave headroom to handle loss of capacity in case of outage or server maintenance.

SIZING CONSIDERATIONS

Sizing PCA systems to replace Exalogic, or any system, depends on measuring the current system to understand its resources, and adjusting for changes in platform and expected growth. Oracle has performed internal tests on WebLogic applications moved to PCA X8 and has seen performance improvements of up to 2.25X compared to Exalogic X6. That degree of improvement may not occur on all migrations, but indicates that PCA works well as a replacement for Exalogic that can enhance performance. Customers have moved from Exalogic and commodity servers with excellent results. We observed that:

- The 100Gb Ethernet networks on PCA provide substantially higher performance for applications with high network traffic between members of an application cluster. The performance improvement mentioned above was obtained by comparing performance between applications nodes on Exalogic and on PCA. This is important for applications like Coherence that use a high rate of network message traffic between cluster members.
- Applications with high message traffic to external hosts benefit from the faster external connectivity PCA has to datacenter networks. PCA network connections can be 100, 40, 25 or 10 Gb Ethernet, and can use LACP for load balancing. If throughput between Exalogic and other hosts was the gating factor for performance, PCA provides the opportunity for much higher throughput. This includes connectivity to database servers like Exadata.
- For CPU-bound components of workload, factor in the relative CPU speeds of the current platform and PCA X8. PCA X8 has more and faster cores than predecessor systems like Exalogic, and the CPU component of application performance – the application payload – should be proportionally faster. Calculate the number of cores on PCA from the number of cores on the current system adjusted by their performance ratio, and factor in peak CPU utilization if it is available.
- PCA compute nodes offer more memory – up to 1.5TB RAM per server. This provides opportunities for using more 'data in memory' buffering to reduce I/O waits.

Application performance depends on specific applications' performance characteristics and requirements, and is not accurately predicted by a micro-benchmark or synthetic test. For sizing assistance, customers should work with their Oracle sales consultant to characterize workloads requirements and establish the correct PCA configuration. The Oracle Solution Center can be leveraged to stress test and tune applications on Oracle facilities to get the most accurate sizing.

CONCLUSION

The Oracle Private Cloud Appliance is the ideal platform to host applications which formerly ran on Exalogic and commodity servers. This paper describes the advantages of PCA and techniques to migrate applications from earlier platforms to PCA.

ABOUT THE AUTHORS

This paper was produced by the Engineered Solutions team, part of Oracle's System Product Management organization.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

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

Integrated Cloud Applications & Platform Services

White Paper **Web and Application Tier on Oracle Private Cloud Appliance / Private Cloud at Customer**
March 2020