Oracle® Communications Session Border Controller and Session Router

VNF Essentials Guide
Release S-CZ7.2.9

November 2016

ORACLE

Notices

Contents

About this Guide

Oracle® Communications Session Border Controller (SBC) applies core session control to reduce the complexity and cost of delivering high-value, revenue generating SIP multimedia services. Oracle Communications Session Border Controller can be used to support a broad range of SIP services including residential or business voice, GSMA-defined Rich Communication Suite (RCS) services and fixed mobile convergence (FMC) for small subscriber populations or initial service rollouts.

Oracle Communications Session Router (SR) is a next generation session routing proxy that incorporates both IETF SIP proxy and 3GPP Breakout Gateway Control Functions (BGCF). It is specifically design to control the routing of large volumes of SIP session signaling messages. Session Router provides high-performance SIP routing with scalable routing policies that increase overall network capacity and reduce costs. It plays a central role in Oracle's open session routing (OSR) architecture and helps service providers build a scalable, next-generation signaling core for SIP-based services.

Both the Oracle Communications Session Border Controller (OC-SBC) and Oracle Session Router (OC-SR) are capable of being deployed as VNFs. Support for the OC-SBC and OC-SR as VNFs is the same. This document refers to the OC-SBC, but the reader can assume that all information herein applies equally to the OC-SR.

Release Version S-Cz7.2.9 is supplied as virtual machine software or as a software-only delivery suitable for operation on server hardware. Refer to sales documentation updates for information further specifying hardware support.

Related Documentation

The following table lists the members that comprise the S-Cz7.1.2 documentation set for this release:

Document Name	Document Description
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Oracle Communications Session Border Controller.
ACLI Reference Guide	Contains explanations of how to use the ACLI, with alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Acme Packet’s enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about accounting support, including details about RADIUS and Rf accounting.

Document Name	Document Description
HDR Resource Guide	Contains information about the Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about support for the Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Session Border Controller family of products.

Revision History

Date	Description
April 2015	• Initial Release
June 2015	• Adds known issue about overlapping subnets • Updates Application Features section in Feature Support appendix
October 2015	• Updates Media Features - Unsupported list with SCTP in Feature Support appendix
October 2015	• Adds OC-SR support • Adds Interface architecture support • Adds VMware support • Updates KVM/Oracle Linux version support
November 2015	• Adds deploy on VMware configuration procedure • Adds tested CPU to resources information.
February 2016	• Clarifies VMware support as based on ESXI versions • States co-product support with AO
March 2016	• Removes outdated datapath licensing information • Adds AO to Co-Product Support section
November 2016	• Adds model type to KVM interface configuration • Adds kernel information to clarify Oracle Linux support requirements

1

The Oracle Communications SBC and SR as VNFs

The Oracle Communications Session Border Controller (OC-SBC) and Oracle Session Router (OC-SR) are capable of being deployed as VNFs. Support for the OC-SBC and OC-SR as VNFs is the same. This document refers to the OC-SBC, but the reader can assume that all information herein applies equally to the OC-SR.

VNF deployment types include:

- A single instance Oracle Communications Session Border Controller operating on a virtual machine,
- Machine(s) deployed within an Orchestrated Network Functions Virtualization (NFV) environment.

Single instance Oracle Communications Session Border Controller VNF deployments are always supported. Support within an orchestrated environment is dependent on version release.

Supported Platforms

For its applicable VNFs, Oracle supports a limited number of virtual machine architectures. These architectures translate into platforms, and consist of the supported operating environments.

The Oracle Communications Session Border Controller supports the following hypervisors:

- Oracle Virtual Machine (OVM): Using version 3.3.3
- KVM: Using version embedded in Oracle Linux 6.7 and 7.1 with UEK
- VMware: Using ESXI versions 5.5 and 6

Interface Architecture Support for Netra/Oracle Servers by Hypervisor

Interface architecture support for Netra servers varies by hypervisor. Applicable servers include the Netra Server X3-2, the Netra Server X5-2, and the Oracle Server X5-2.

The Oracle Communications Session Border Controller supports the following interfaces and interface architectures per platform.

Netra Server X3-2

Supports the X540 in the onboard slot, and/or the 82599 in the expansion slot.

Interface architecture support for KVM, using Oracle Linux 6.7, includes:

- Paravirtualized

- PCI Passthrough
- Single Root I/O Virtualization

Interface architecture support for VMware 5.5 includes:

- Paravirtualized
- PCI Passthrough
- Single Root I/O Virtualization

Interface architecture support for OVM 3.3.3 includes:

- PCI Passthrough
- Single Root I/O Virtualization

Netra Server X5-2

Supports the X540 and/or the 82599 in the expansion slot. The X710/XL710 in the onboard slot is not supported.

Interface architecture support for KVM, using Oracle Linux 7.1, includes:

- Paravirtualized
- PCI Passthrough

Interface architecture support for VMware 6 includes:

- Paravirtualized
- PCI Passthrough

Oracle Server X5-2

Supports the X540 in the onboard slot, and the X540 and/or the 82599 in the expansion slot.

Interface architecture support for KVM, using Oracle Linux 6.7, includes:

- Paravirtualized
- PCI Passthrough
- Single Root I/O Virtualization

Interface architecture support for VMware 5.5 includes:

- Paravirtualized
- PCI Passthrough
- Single Root I/O Virtualization

Note: Single Root I/O Virtualization does not support VLANs.

Virtual Machine Resources

An Oracle Communications Session Border Controller virtual machine requires CPU core, memory and disk size specified for operation. The system uses the DPDK kit for datapath design, which imposes VNF-specific resource requirements for CPU cores. Deployment details, such as the use of DoS, specify resource utilization beyond minimums. The user configures CPU core utilization from the ACLI based on their deployment. The user can also define memory and hard disk utilization based on deployment, but must configure the hypervisor with these settings prior to startup if they need them to be different then the machine defaults specified in the OVF.

Minimum hardware resources, with the exception of CPU cores, include:

- 4 GB RAM
- 40 GB hard disk (pre-formatted)
- 2 interfaces as follows:

  - 1 for management (wancom0)
  - 1 for signaling and media

Deployments commonly use more than 2 interfaces. Minimum recommended hardware resources for these deployments, which raises the required memory to 8 GB, include:

- 8 GB RAM
- 40 GB hard disk (pre-formatted)
- 8 interfaces as follows:
  - 1 for management (wancom0)
  - 2 for HA (wancom1 and 2)
  - 1 spare
  - 4 for signaling and media

The Oracle Communications Session Border Controller SCZ7.2.9 VNF is tested and confirmed as operational on the Intel Xeon "Ivy Bridge" CPU.

The Oracle Communications Session Border Controller VNF requires at least 3 CPU cores, and allows a maximum of 8. The system checks CPU core resources before every boot, as configuration can affect resource requirements. Examples of such resource requirement variations include:

- There is at least 1 CPU assigned to signaling.
- If DOS is required, then there are at least 2 CPUs assigned to the datapath.
- If DOS is not required, then there is at least 1 CPU assigned to the datapath.

The system performs resource utilization checks every time it boots for CPU, memory and hard-disk to avoid configuration/resource conflicts.

For HA systems, resource utilization on the backup must be the same as the primary.

Note: The hypervisor always reports the datapath CPU usage fully utilized. This isolates a physical CPU to this work load. However, this causes the hypervisor to generate a persistent alarm indicating that the VM is using an excessive amount of CPU, possibly triggering throttling.

Distribution Components

The Oracle Communications Session Border Controller VNF version SCz7.2.9 is a release suited for deployment over your virtual machine manager. Oracle provides software distributions specific to the virtual machine environment in which the product is deployed.

Oracle provides the version SCz7.2.9 Oracle Communications Session Border Controller VNF in one of two distributions:

- `nnSCZ729.64-img-vm_ovm.ova`—Open Virtualization Archive (.ova) distribution of the Oracle Communications Session Border Controller VNF for Oracle virtual machines.
- `nnSCZ729.64-img-vm_kvm.tar`—Compressed image file including Oracle Communications Session Border Controller VNF for KVM virtual machines.

Each package includes:

- Product software—Bootable image of the product allowing startup and operation as a virtual machine. This disk image is either in the vmdk or qcow2 format.
- `usbc.ovf`—XML descriptor information containing metadata for the overall package, including identification, hardware requirements, references to other files and so forth. The .ovf file format is specific to the supported hypervisor.
- `legal.txt`—Licensing information.

Configuration Overview

Oracle Communications Session Border Controller VNF deployments require configuration of the virtual machine environment and of the SBC itself. The user can consider VM environment and SBC configuration as separate from VNF configuration. VNF-specific configuration allows for resource tuning, such as CPU core allocation, based on the deployment's performance and capacity requirements.

When the user deploys the SBC VNF, they must provide operational information within the system's boot parameters, including:

- IP address
- Host name

The user can configure this information manually. Alternatively, your virtual machine or orchestration tools may provide a means of configuring this information.

Performance and capacity tuning configuration is not required to get the SBC VNF to function. Instead, this configuration improves the VNF's operation. Applicable configuration includes:

- Media manager traffic/bandwidth utilization tuning
- Datapath-related CPU core allocation

See the section on Configuring Your Device for NFV for VNF tuning configuration. Refer to your VM documentation for VM environment configuration. Refer to the documentation listed in the About This Guide section of this document for SBC-related configuration.

Co-Product Support

The products/features listed in this section run in concert with the Oracle Communications Session Border Controller for their respective solutions.

Oracle Communications Session Element Manager

Oracle Communications Session Element Manager versions 7.4M1 and later support this GA release of the Oracle Communications Session Border Controller

Oracle Communications Application Orchestrator

Oracle Communications Application Orchestrator versions 1.1 and later support this GA release of the Oracle Communications Session Border Controller

Provisioning Entitlements

VNF products licensing follows the standard C-series self-entitlements licensing model. Refer to the *ACLI Configuration Guide* for instructions on setting entitlements.

Licensing detail with which the user should be aware includes:

- Peering vs Access—Only Peering licenses are available.
- Media Sessions—Media sessions are licensed as an entitlement.

SPL and Baseline Information

SPL Version Support

Current SPL Engine version supported:

- C2.0.0
- C2.0.1
- C2.0.2
- C2.0.9
- C2.1.0
- C2.1.1
- C2.2.0
- C2.2.1
- C3.0.0
- C3.0.1
- C3.0.2
- C3.0.3
- C3.0.4
- C3.0.6
- C3.1.0
- C3.1.1
- C3.1.2

Release and Patch Baseline

Current Patch Baseline: S-CZ7.1.2M3p3

2

Configuring Your Device for NFV

Operating the Oracle Communications Session Border Controller as a VNF introduces configuration requirements that define resource utilization by the virtual machine. The applicable configuration elements allow the user to optimize resource utilization based on the application need and shared VM resource operation.

Oracle Communications Session Border Controller configuration for VNF includes settings to:

media-manager—Media manager configuration elements, intended specifically to constrain bandwidth utilization based on traffic type.

datapath-config—Root level configuration element for specifying CPU and memory available to DoS, media, signaling and transcoding processes.

CPU core isolation—Identify appropriate cores for isolation, preventing the VM infrastructure from reserving core resources that must be exclusively reserved for DPDK support of forwarding, DoS or transcoding functions.

Media Manager Configuration

The Oracle Communications Session Border Controller provides the user with a means of tuning the media manager.

As the Oracle Communications Session Border Controller can classify traffic for use in DoS policing, bandwidth may be reserved for certain traffic types. Reserved bandwith is expressed as a percentage of maximum available system bandwidth. The system's maximum bandwidth is determined by the hardware configuration and the number of available signaling cores. The maximum system bandwidth is defined as the speed of ingress traffic sent to the host, measured in packets per second (PPS). It is reported in the **show platform limits** command, referring to the "Maximum Signaling rate:".

The following configuration options are available in the **media-manager-config**. these options are used to configure reserved bandwidth for application signaling, ARP, and untrusted traffic.

- **max-signaling-rate**—The percentage of maximum system bandwidth available for signaling application packets. Valid values are 0-100.
- **max-arp-rate** —The percentage of maximum system bandwidth available for ARP traffic . Valid values are 0-100.
- **min-untrusted-signaling**—The minimum percentage of maximum system bandwidth available for untrusted traffic.
- **max-untrusted-signaling**—The maximum percentage of maximum system bandwidth available for untrusted traffic. This is a floating highwater mark and is only available when not in use by trusted sources.

Datapath Configuration

The Oracle Communications Session Border Controller provides the user with a means configuring virtual resource allocations. These settings supercede those that may have been received by the orchestrator. The user makes these settings using the **datapath-config** command.

The user executes the **datapath-config** command from the enable prompt, followed by a parameter and setting. The Oracle Communications Session Border Controller provides a response showing the current core assignment and/or an error if there is an issue. The Oracle Communications Session Border Controller uses these values to update the system. A reboot is required for this setting to take effect.

Note: Configurations regarding core allocations are 0-based. Core-0 must always be set as a signaling core.

The **datapath-config** command supports the following arguments:

- **set-dos-cores**— Sets which core is assigned to the DOS functionality. A maximum of one core is allowed.
- **set-forwarding-cores**—Sets which cores are assigned to the forwarding engine. Valid values are from 1 to the number of cores currently active on the VM.
- **set-memory-size**—Specifies the amount of memory to request at boot for the application. When not set, the system allocates this size itself.
- **set-num-of-pages**— Sets the number of pages used by the application.
- **set-signaling-cores**— Sets which cores are assigned to signaling processing. These cores are also used for general system/Linux functionality. Valid values are from 0 to the number of cores currently active on the VM.
- **set-transcoding-cores**—Sets which cores are assigned to transcoding. Valid values are from 1 to the number of cores currently active on the VM.

To change core assignments as well as memory and page size settings, run the command with the newly desired setting. The Oracle Communications Session Border Controller overwrites existing settings, such as core assignment, based on the new command string.

For example, to reserve core 2 for DoS processing:

```
ORACLE# datapath-config set-dos-cores 1
New core assignment: S-D-F-F
```

CPU Isolation

The Oracle Communications Session Border Controller VNF requires that CPUs assigned to forwarding, DoS and transcoding need the user to reserve them. This prevents the VM infrastructure from using them, thereby isolating them for use by the Oracle Communications Session Border Controller. The user does this via configuration of the bootparameter named "**other**".

As a component of Oracle Communications Session Border Controller VNF installation and configuration, the user must identify CPU core usage within the context of datapath configuration. In addition to core usage specification, the user isolates the appropriate cores using the following bootparameter setting:

isolcpus=[comma separated list of cores to isolate for DPDK usage}

For example, the following configuration isolates CPU cores 1,2 and 3.

```
other:            isolcpus=1,2,3
```

Complementary configuration could include:

```
ORACLE# datapath-config set-dos-cores 1
New core assignment: S-D-?-?
ORACLE# datapath-config set-forwarding-cores 2
New core assignment: S-D-F-?
ORACLE# datapath-config set-transcoding-cores 3
New core assignment: S-D-F-X
```

These configurations isolate and specify core 1 for DoS, core 2 for forwarding, and core 3 for transcoding. Oracle recommends that the user does not isolate CPU core zero (0); a signaling core does not need to be isolated.

Perform this configuration prior to operation. Refer to the Boot Management chapter in this document for instructions on how to set a boot parameter. Reboot the system for your configuration to take effect.

Note: The boot parameter configuration does not provide any syntax or other error checking.

3

Introduction to Platform Preparation & Software Deployment

This section introduces the deployment of Oracle Communications software onto the platforms supported by this version of the Oracle Communications Session Border Controller. Refer to Hardware Installation documentation for rack and stack procedures. This documentation explains platform preparation, power-on and software deployment to the point where the user is ready to perform service configuration.

Platform support by Oracle Communications session delivery software extends to generic platforms. As such, preparation of these platforms requires that the user perform tasks independent of their session delivery product. Although the user needs to use platform documentation for this platform-specific information, Oracle distributes this documentation to provide the user with:

- Settings required by the session delivery software.
- Guidance to procedures that apply to session delivery software.

Virtual Machines

Virtual Machines (VMs) supported by Oracle Communications Session Delivery software varies across software version. Find specific version support within the context of your version's documentation.

Operation over VMs is roughly equivalent to deployment over COTS/Server hardware. Platform preparation, however, differs greatly. In addition, platform preparation differs greatly between VM platforms.

Preparation procedures that apply to all VM platforms include the following steps:

1. Make the VM template available to the VM manager.
2. Configure the VM manager to apply the template correctly for Oracle Communications Session Delivery software.
3. Power-on the VM. The system uses the VM template to automatically install onto the virtual drive, after which the server reboots.

VM deployment requires extensive knowledge about the specific platform that is not documented herein. The intent of this documentation is to provide information that helps the user navigate the deployment and perform tasks that are specifically related to Oracle Communications Session Deliver software.

4

Virtual Machine Platforms

Oracle distributes virtual machine templates, each containing a virtual disk image and sample configuration for the minimum supported profile of each VM platform. VM platform support is dependent on your Oracle product version.

This section addresses requirements associated with running applicable software as virtual machines. It also provides basic instructions on loading and starting machine templates.

VM distributors maintain extensive documentation sites. You must use those vendors' documentation for full explanations and instructions on VM deployment and operation.

Create and Deploy on Oracle VM Manager

This section provides detail on deploying Oracle Communications Session Delivery products in an Oracle Virtual Machine environment and assumes Oracle VM Manager 3.3. The section also assumes the user understands deployment in these environments and that the majority of deployment tasks, from hardware installation and startup to VM resource and management setup, is complete.

For information on Oracle OVM, including server and manager documentation, refer to the following links. The bottom link opens the page specifically to Oracle OVM version 3.3.

http://www.oracle.com/technetwork/documentation/vm-096300.html

http://docs.oracle.com/cd/E50245_01/index.html

Once hardware, VM resources and VM management are in place, the user prepares the VM management environment. High level steps include:

- Discover Oracle VM Servers
- Discover Storage (File Server or SAN)
- Create a Virtual Machine Network
- Create a Server Pool
- Create a Storage Repository

Note: The following procedure describes a typical deployment. The system may display different screens, depending on the deployment.

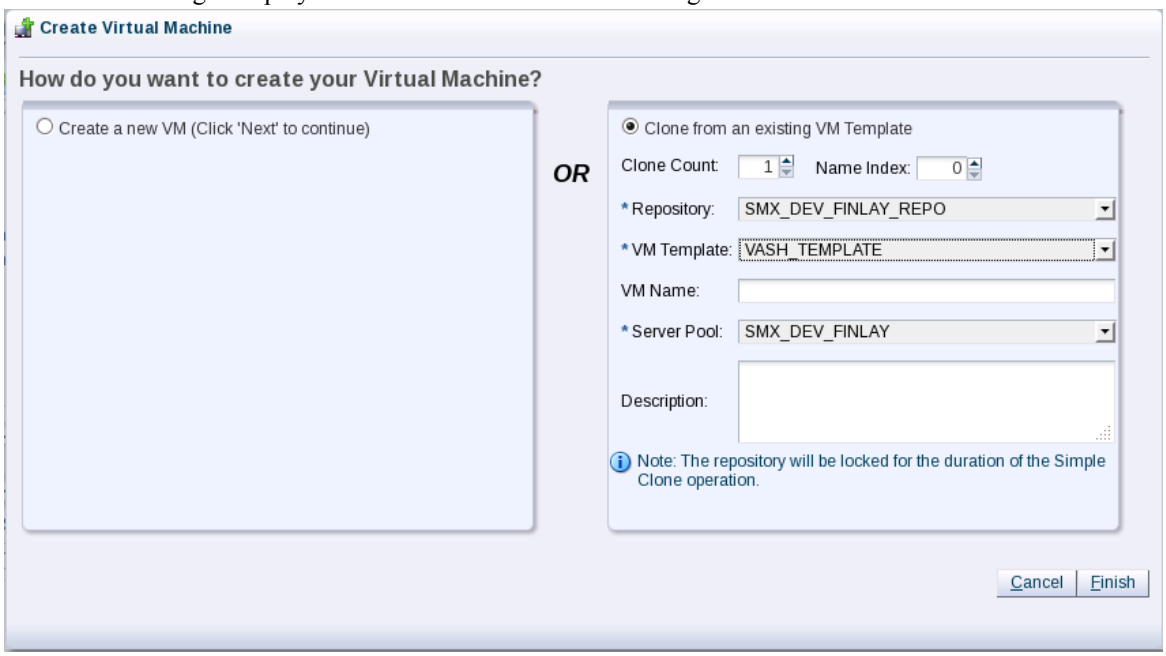
Oracle Communications Session Delivery product-specific setup steps include the following.

- Add Resources to Storage Repository
  - Import an assembly

  - Import a virtual machine template (from the assembly)
- Create a virtual machine from a template
- Assign Networks
- Specify Boot Order
- Start your virtual machine
- Connect to the console

Detail on these steps is shown below. Having created your first machine, you can convert a virtual machine to a template, clone a virtual machine to a template, or clone another template.

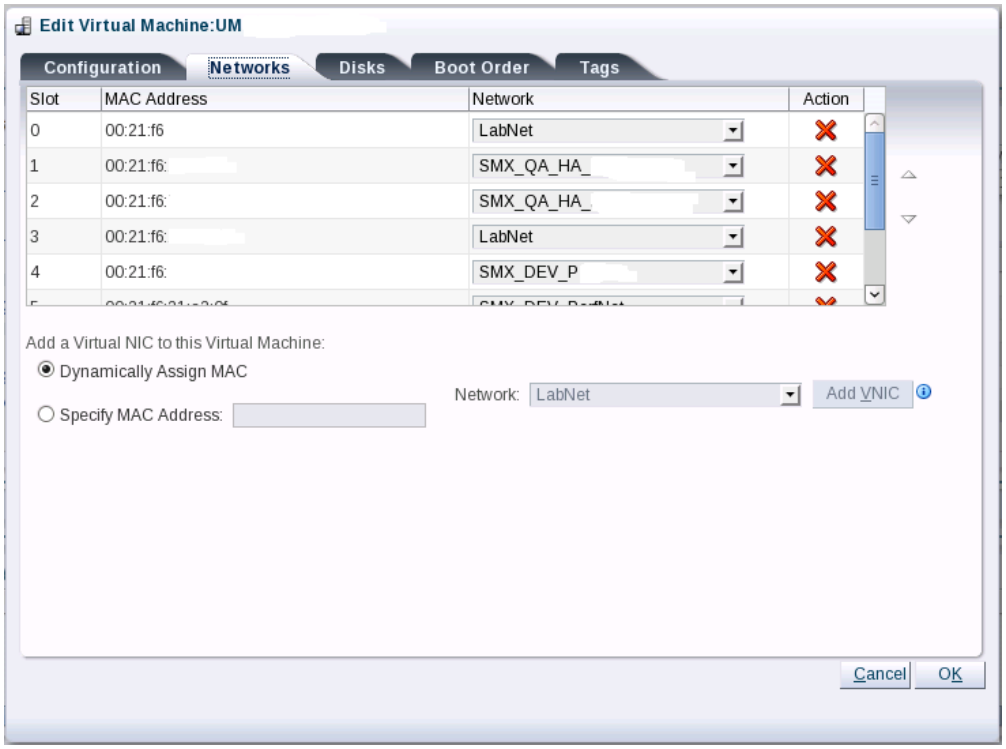
1. From the Oracle VM Manager application Home page, navigate to the **Repositories** tab.
2. Expand your target repository and highlight the Assemblies folder.
3. Click the **Import VM Assembly** button.
   Oracle VM Manager displays the Import VM Assembly dialog.
4. Type the URL, either http or ftp server, of your .ova assembly and click **OK**.
   Oracle VM Manager Assembly imports your assembly.
5. Click **Create VM Template**.
   Oracle VM Manager displays the Create VM Template dialog.
6. Select your Assembly from the Assembly Virtual Machines drop-down. Add a Name and Description, if desired, and click **OK**.
   Oracle VM Manager creates your template.
7. Navigate to the Servers and VMs tab, expand the Server Pools list and select your target server pool.
8. Click **Create Virtual Machines**.
   Oracle VM Manager displays the Create Virtual Machines dialog.



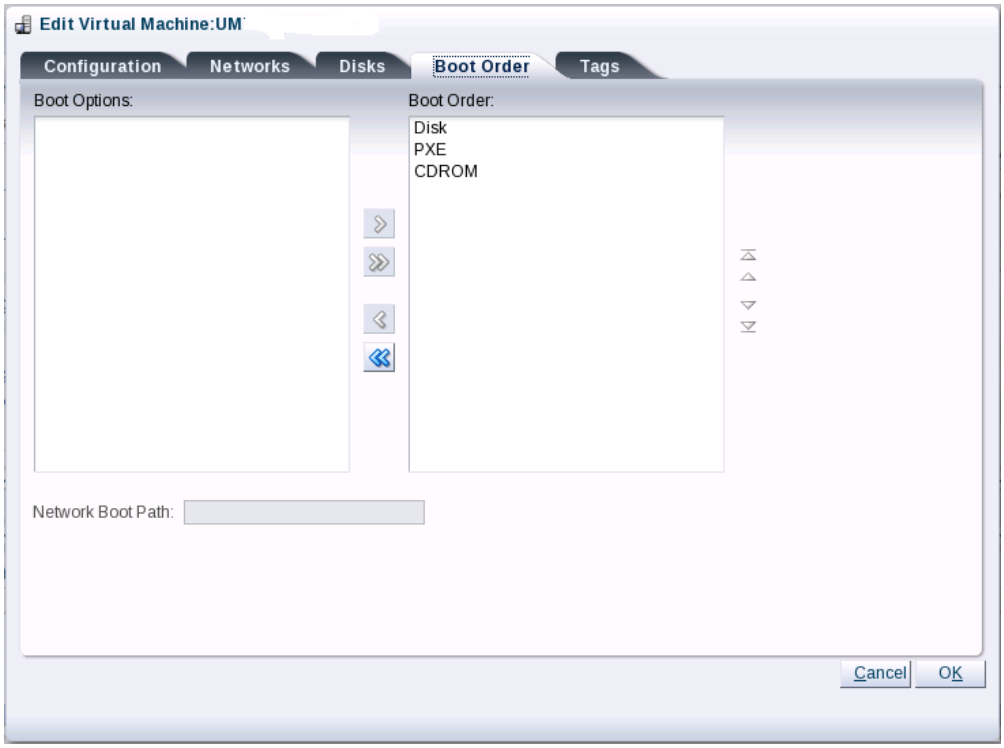
9. Select and configure the following, then click **Finish**.
   - Clone from an existing VM Template
   - Target Repository
   - VM Template
   - Target Server Pool
   - Name and Description (if desired)

   Oracle VM Manager creates and deploys your VM.

10. Navigate to the Servers and VMs tab, then expand the Server Pools folder, your target Server pool and select your target server.
    Oracle VM Manager displays the VMs available from this server.
11. Before starting your VM, highlight it and click the Edit icon, which appears as a pencil on the top of the VM list.
    Oracle VM Manager displays the Edit Virtual Machine dialog.
12. Click the **Networks** tab and assign ethernet port slots to your networks using the **Network** drop-down selection box.



13. Click the **Boot Order** tab and define you boot order to use Disk as its first option.



Oracle VM Manager knows to use the template for installation upon first startup. This setting ensures that subsequent startups use Disk as the first option.

14. Highlight the target VM, click **Start**.
    Oracle VM Manager starts your VM.
15. Click the **Console** button.
    Oracle VM Manager displays a terminal screen with the serial CLI operational.

Create and Deploy on KVM

This section provides detail on deploying Oracle Communications Session Delivery products in a KVM environment and shows Oracle version 7. The section assumes the user understands deployment in these environments and that the majority of deployment tasks, including hardware installation and startup, is complete.

For information on KVM, refer to the following link.*http://www.linux-kvm.org/page/Documents*

Any install procedure, using SR-IOV, includes the following initial steps to prepare the system for supporting virtual machines:

1. Verify BIOS settings:
   - VT-d enabled
   - Intel virtualization enabled
   - DVD first in the boot sequence (Linux installation from DVD)
2. Perform linux installation.
3. Include support of virtualization by the kernel, for example by applying the **`intel_iommu=on`** and **`iommu=force`** parameters to the kernel's configuration.
4. Configure a host management interface by setting a bridge that has network connectivity.
5. Install extra linux packages:
   - qemu
   - libvirt

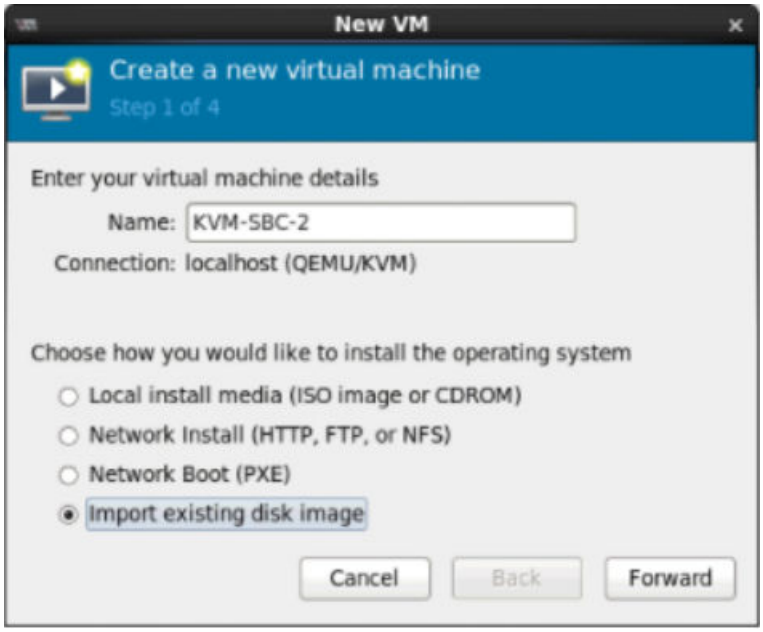
- virt-manager

6. Setup Single Root I/O Virtualization (SR-IOV) according to your deployment's interface requirements.
7. Assign dedicated MAC addresses to each of the virtual interfaces you plan to use for media/signaling interfaces.

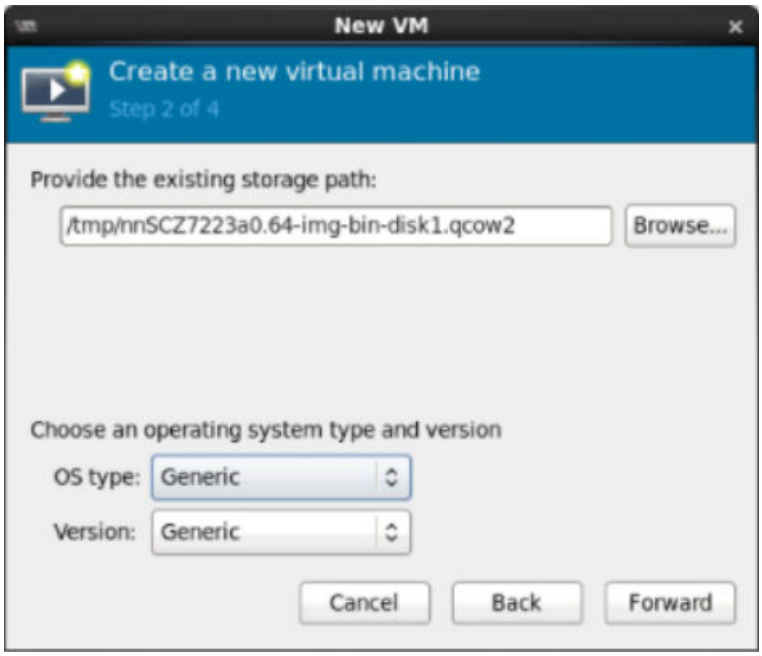
You can now create virtual machines for your systems. The steps below show the process using virt-manager.

1. Start KVM and initiate the New VM configuration routine.

   The system starts the New VM steps at the **Create a new virtual machine** dialog.
2. Name your machine, click the **Import existing disk image** radio button, and click **Forward.**

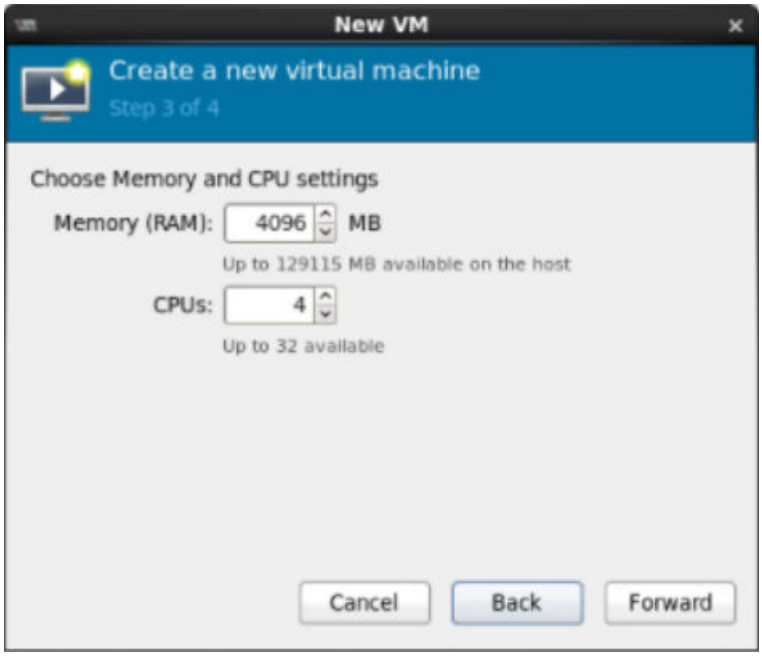


   The system displays the next **Create a new virtual machine** dialog.
3. On the next **Create a new virtual machine** dialog, browse to select the target file. After selecting the filename, click **Forward**.



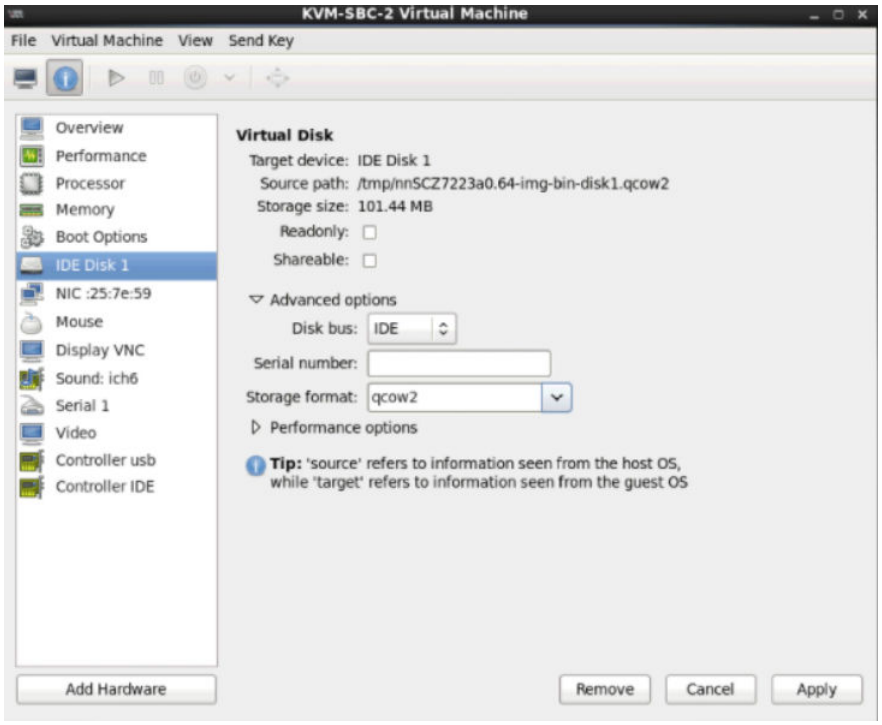
The system displays the next **Create a new virtual machine** dialog.

4. Set your machine to use 4MB or memory and 4 CPUs, and click **Forward.**

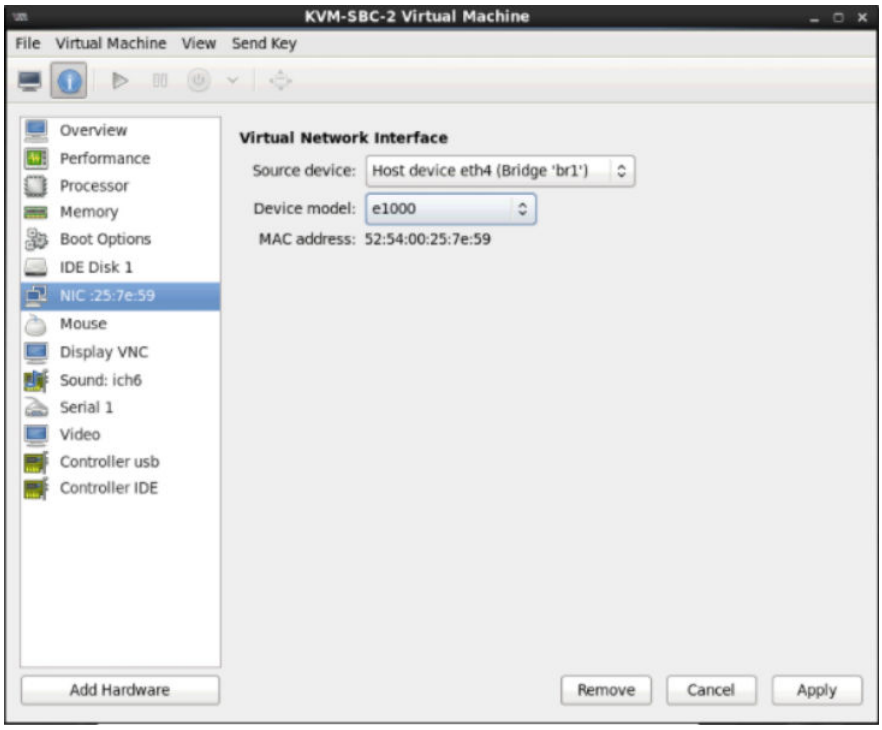


The system displays the a virtual machine configuration review dialog, allowing you to verify your settings and complete the VM creation process.

5. Finish adding your machine, then select it from KVM's VM list.
   Perform the next steps on the VM itself.
6. Access machine details by double clicking it on the VM list and setting the View to Details.
   Virtual Machine manager displays machine setting and controls dialogs.
7. From your VM's **Disk 1** dialog, select **qcow2** from the **Storage format** dropdown under **Advanced options**, and click **Apply**.



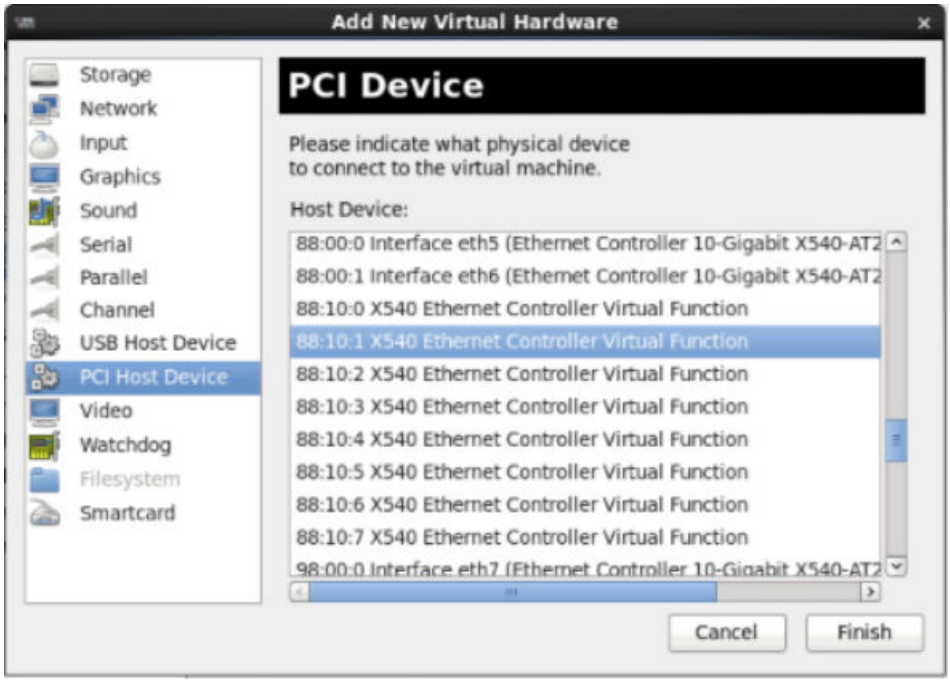
8. To assign an IP address to wancom0, change the existing NIC device model to e1000. This also allows the assigned MAC address to appear in the interface-mapping menu.



- Create a minimum of 2 management interfaces, one for wancom0 and one for wancom1. The system support a maximum of 4 management interfaces, including wancom2 and a spare. All management interfaces must use e1000 as the Device Model of the Virtual Network Interface.

From your VM's management port **NIC** dialog, select **e1000** from the **Device model** dropdown, and click **Apply**.

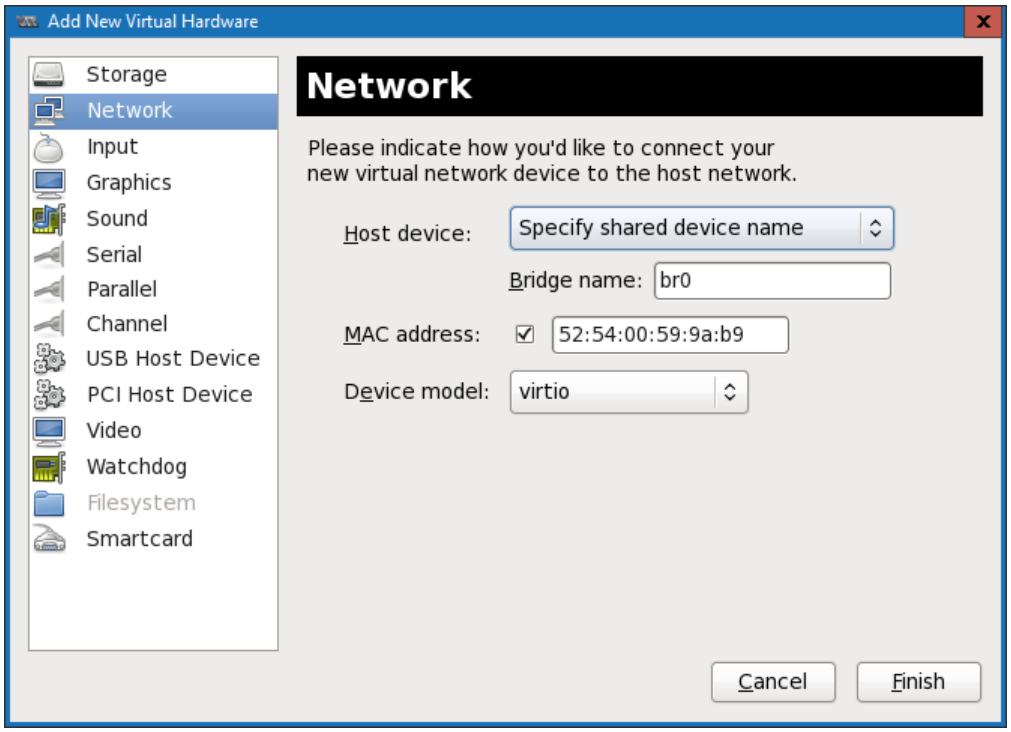
9. Click the **Add Hardware** button.
KVM displays the **Add New Virtual Hardware** dialog. From here, you create PCI-passthru or SR-IOV media interfaces. The minimum is 2 media interfaces; the maximum is 8.
10. From your VM's **PCI Host Device** dialog, select and add a device you plan to use as a media port. Click **Finish** after you have added your device.



Repeat this step for all media ports.

This completes the interface selection procedure for PCI-passthru or SR-IOV. PV media interfaces, however, require that you set their device model to **virtio**.

For PV media interfaces, access the **Network** dialog, select the Host device, and select **virtio** from the **Device model** dropdown.



Click **Finish** after you have configured each PV device.

11. Access your linux CLI.

12. Ensure that Generic Receive Offload (GRO) is not running on every interface that the device must use.

Perform this step using the ethtool command, **`ethtool -K <dev> gro off`**, where **`<dev>`** is the network-interface name.

```
ethtool -K eth0 gro off
ethtool -K eth1 gro off
```

13. Start your VM.
14. Double click your machine and set the **View** to **Console**.
    The Virtual Manager displays your VM's console, which operates similarly to a serial console.

Create and Deploy on VMware®

This section provides detail on deploying Oracle Communications Session Delivery products in an Oracle Virtual Machine environment and assumes VMware 5.5. The section also assumes the user understands deployment in these environments and that the majority of deployment tasks, from hardware installation and startup to VM resource and management setup, is complete.

For information on VMware 5.5, refer to the following link.

http://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html

For information on VMware 6, which is also supported, refer to the following link.

https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html

Before You Begin:

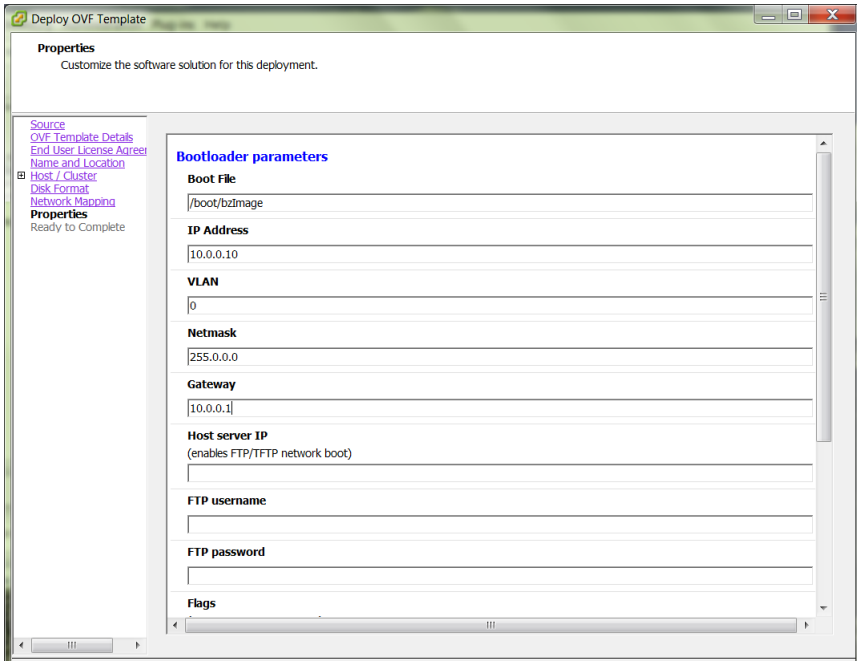
- Confirm that the VMware 5.5 Hypervisor is installed on an appropriate network server.
- Confirm that the server has 40GB of space for this installation.

Note: The following procedure describes a typical deployment. The system may display different screens, depending on the deployment.

Detail on Oracle Communications Session Delivery product-specific setup steps is shown below. Having created your first machine, you can convert a virtual machine to a template, clone a virtual machine to a template, or clone another template.

1. On the vSphere Client application Home page, go to File > Deploy OVF Template File.
2. On the Source screen, browse to the target .ova file, and click Next.
3. On the End User License Agreement screen, click Accept and click Next.
4. On the Name and Location screen, do the following and click Next.
   - Name. Enter a name for the template.
   - Inventory Location. Select the location where you want to deploy the template.
5. On the Host / Cluster screen, select the host or cluster where you want to run the deployed template, and click Next.
6. If applicable to your deployment, select the resource, from the Resource Pool screen, in which you want to deploy this template, and click Next.
7. On the Storage screen, select the destination storage for the virtual machine files, and click Next.
8. On the Disk Format screen, select Thick Provisioned Lazy Zeroed, and click Next.
9. On the Network Mapping screen, map the networks used in this OVM template to networks in your inventory, and click Next.

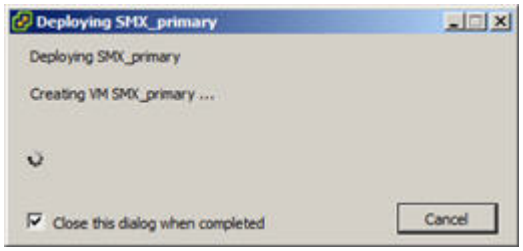
10. On the properties screen, enter the bootloader parameters, and click Next.



11. On the Ready to Complete screen, review the selections, and do one of the following:

- Edit a previous screen by way of the Back button.
- When the edits are completed, or if no edits are required, click Finish

The system displays the Deploying screen.



When the Deploying screen closes, VM creation and deployment is completed.

5

Boot Management

Boot Management includes the tasks needed to ensure the system is operating according to the users requirements as it starts up. Requirements met by properly managing system boot include defining management access IP, specifying the load to boot and specifying a system name. The user may set this information manually or configure the operational environment to provide it.

Boot management consists of tasks working with the following:

- Boot Loaders—The user needs to perform file management tasks to ensure that the software used to boot the system is compatible with the application system software itself. This typically includes verifying boot loader and application system software version for compatibility and placing the correct boot loader software in the correct location.
- Boot Parameters—The user sets boot parameters to specify their requirements for boot, including defining management access IP, specifying the load to boot and specifying a system name.
- Boot Flags—The user can, optionally, set special boot parameters called boot flags to further define how the system boots. The user may also set boot flags for diagnostic purposes under the guidance of Oracle support personnel.

Boot Parameters

Boot parameters specify the information that your device uses at boot time when it prepares to run applications.

This section explains how to view, edit, and implement device's boot parameters, and boot flags. Boot parameters:

- Allow you to set the IP address for the management interface (wancom0).
- Allow you to set a system prompt. The target name parameter also specifies the title name displayed in your web browser and SNMP device name parameters.
- Specify the software image to boot and from where the system boots that image.

Boot flags are arguments to a specific boot parameter, and allow functional settings, such as the use of DHCP for acquiring a management port address, as well as various diagnostic startup configurations.

Configuring boot parameters has repercussions on your system's physical and network interface configurations. When you configure these interfaces, you can set values that might override the boot parameters.

The bootparam configuration list is shown below.

```
[Acme Boot]: p
Boot File         : /boot/bzImage-bones64
IP Address        : 172.44.12.89
VLAN              :
```

```
Netmask          : 255.255.0.0
Gateway          : 172.44.0.1
IPv6 Address     :
IPv6 Gateway     :
Host IP          :
FTP username     :
FTP password     :
Flags            : 0x00000030
Target Name      : ACMEPACKET
Console Device   : COM1
Console Baudrate : 115200
Other            :


[Acme Boot]: ?
 ?                     - print this list
 @                     - boot (load and go)
 p                     - print boot params
 c                     - change boot params
 v                     - print boot logo with version
 r                     - reboot
 s                     - show license information
```

Boot Parameter Definitions

The system displays all boot parameters when the user configures them after a boot interrupt. The system hides some boot parameters from the ACLI because the user should not configure them. If changed improperly, these parameters can cause the system to fail.

The following table defines each of the parameters that are visible when the user configures after a boot interrupt

Boot Parameter	Description
Boot File	The name and path of the software image you are booting. Include the absolute path for a local boot from the local /boot volume and for a net boot when a path on the FTP server is needed.
IP Address	IP address of wancom0.
VLAN	VLAN of management network over which this address is accessed. Note: VLANs over management interfaces are not supported on the Acme Packet 6000
Netmask	Netmask portion of the wancom0 IP Address.
Gateway	Network gateway that this wancom0 interface uses.
IPv6 address	Version 6 IP address of wancom0.
IPv6 Gateway	Network gateway that this wancom0 interface uses.
Host IP	IP Address of FTP server from which to download and execute a software image.
FTP Username	FTP server username
FTP password	FTP server password
Flags	Codes that signal the system from where to boot. Also signals the system about which file to use in the booting process. This sequence always starts with 0x (these flags are hexadecimal).

Boot Parameter	Description
Target Name	Name of the Oracle Communications Session Border Controller as it appears in the system prompt. For example, ORACLE> or ORACLE#. You need to know the target name if you are setting up an HA node. This name is required to be unique among Oracle Communications Session Border Controllers in your network. This name can be 64 characters or less.
Console Device	Set this to COM1
Console Baud Rate	The speed in bits per second which the console port operates at. It operates at 115200 BPS, 8 data bits, no stop bit, parity NONE.
other (o)	For Oracle use only.

Changing Boot Parameters

You can access and edit boot parameters by using either the ACLI or by interrupting the system boot process.

Note: Changes to boot parameters do not go into effect until you reboot the system.

Oracle recommends that you use management port 0 (wancom0) as the boot interface, and that your management network is either:

- directly a part of your LAN for management port 0
- accessible through management port 0

Change Boot Parameters from the ACLI

To access and change boot parameters from the ACLI:

1. In Superuser mode, type configure terminal, and press Enter.

```
ORACLE# configure terminal
```

2. Type bootparam, and press Enter. The boot device parameters display.

```
ORACLE(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device          : eth0
```

To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one. You can clear the contents of a parameter by typing a period and then pressing Enter.

```
ORACLE(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device     : eth0
processor number    : 0
host name    : goose
file name    : /boot/nnPCz100.bz /boot/nnPCz200.bz
```

When you have scrolled through all of the boot parameters, the system prompt for the configure terminal branch displays.

```
ORACLE(configure)#
```

4. Exit the configure terminal branch.

5. Reboot the system for the changes to take effect.

   The ACLI **reboot** and **reboot force** commands initiate a reboot. With the **reboot** command, you must confirm that you want to reboot. With the **reboot force** command, you do not have make this confirmation.

   ```
   ORACLE# reboot force
   ```

   The system completes the full booting sequence. If necessary, you can stop the auto-boot at countdown to fix any boot parameters.

   If you configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

   Note: If you configured the boot parameters incorrectly, the system goes into a booting loop and displays an error message.

   ```
   Error loading file: errno = 0x226.
   Can't load boot file!!
   ```

   Press the space bar to stop the loop. Correct the error in the boot parameter, and reboot the system.

Change Boot Parameters by Interrupting a Boot in Progress

To access and change boot parameters by interrupting a boot in progress:

1. When the system is in the process of booting, you can press the space bar on your keyboard to interrupt when you see the following message appear:

   ```
   Press the space bar to stop auto-boot...
   ```

2. After you stop the booting process, you can enter the letter p to display the current parameters, the letter c to change the boot parameters or the @ (at-sign) to continue booting.

   ```
   [Acme Packet Boot]: c
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   boot device          : wancom0
   ```

   To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

   You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one.

   ```
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   boot device     : wancom0
   processor number     : 0
   host name     : goose
   file name     : /code/nnPCz100.gz /code/nnPCz200.gz
   ```

4. After you have scrolled through the complete list of boot parameters, you return to the boot prompt. To reboot with your changes taking effect, type @ (the at-sign), and press Enter.

   ```
   [Acme Packet Boot]: @
   ```

   The system completes the full booting sequence, unless there is an error in the boot parameters.

   If you have configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

   Note: If you have configured the boot parameters incorrectly, the system goes into a booting loop and displays an error message.

   ```
   Error loading file: errno = 0x226.
   Can't load boot file!!
   ```

   Press the space bar to stop the loop. Correct the error, and reboot your system.

6

Formatting Disks for VM Deployments

Default OVA distributions come with 40GB virtual hard disk. Oracle pre-formats the system disk area as follows:

- 2GB **/boot**
- 2GB **/code**
- 16GB **/opt**

User disk area, however, is not pre-formatted. If the user wishes to use this space they must run the **format data-disk** command to make the additional 20GB available for user data, such as CDRs.

If the user deploys a non-default configurations, such as one with a larger amount of memory, they need to reformat the disk using the **format system-disk** or **format hard-disk**command. During this reformat, the user must be sure they allocate enough space to support the system disk by reserving 2GB for /boot, 2GB for /code, 8GB for /opt and an additional 2xRAM.

After the drive(s) are formatted, the system mounts the newly created partitions.

Note: If you find that, after the first reboot, the system has not created new partitions, perform another reboot to resolve this issue, which is created by an incorrect dynamic partition table refresh.

Formatting Procedure for VM Deployments

The **format** command requires one of the following arguments:

- system-disk — formats and creates the 2 system partitions: `/opt` and `/opt/crash`
- data-disk — formats and creates 1 or more data partitions with the default (`/mnt/sys` and `/mnt/app`) or user-defined volumes
- hard-disk — formats and creates both the system partition and data partition

After the drive(s) are formatted, the system mounts the newly created partitions.

The following example shows the **format data-disk** command process. In this case, the user formats and mounts the **/sys** and **/app** partitions.

Note: The format command may only be executed if certain tasks like local CDR and HDR generation are not active. Remove any boot time configuration for these features and reboot the system before attempting to format the hard-disk. In addition, ensure that your device is not passing traffic while you format the any partition.

```
ORACLE# format data-disk
```

```
WARNING: Please ensure device is not currently in use by any applications
before proceeding

Continue [y/n]?: y

Use factory default data partitions [y/n]?: y

The following data partitions will now be created:
/sys    3894968320 bytes

/app    35054714880 bytes

Create the data partitions and filesystems as configured above [y/n]?: y

*******************************************************
WARNING: All non-system data on the disk will be
permanently erased and unrecoverable.

Are you sure [y/n]?: y

The format process will take a few minutes. Once
the format process begins, it cannot be stopped.
Please do not power down or reboot the system until
the format process is complete.

Continue [y/n]?: y

*** Beginning format process ***

*** Removing previous data partitions - please wait ***

*** Creating new data partitions - please wait ***

*** Formatting partition /sys. Please wait... ***

*** Formatting completed successfully ***

*** Formatting partition /app. Please wait... ***

*** Formatting completed successfully ***

*** Format finished successfully
New partitions have been created ***

*** Mounting partitions ***/sys mounted/app mounted
```

This section of the format hard-drive walk-through shows the data partition creation. The following system output shows that the user has chosen to define a custom data partition scheme by typing "n" at the following prompt.

```
Use factory default data partitions [y/n]?:n
```

In this case, the user creates three partitions.

```
Suspending logging to RAM drive
Stopping tLogCleaner task
Relocating logging onto hard disk
Initializing /opt/ Cleaner
Starting tLogCleaner task
Disk space used by system:
        16226317824 bytes
Use factory default data partitions [y/n]?: n
Enter the number of data partitions to create: 3
Total unallocated space = 100 %
Enter the name of volume 1 (or 'q' to quit): VOLUME1
```

```
Enter the size of the volume (in %): 20
Total unallocated space = 80 %
Enter the name of volume 2 (or 'q' to quit): VOLUME2
Enter the size of the volume (in %): 40
Total unallocated space = 40 %
Enter the name of volume 3 (or 'q' to quit): VOLUME3
Enter the size of the volume (in %): 40
The following data partitions will now be created:
/VOLUME1  96776308838 bytes
/VOLUME2  193552617676 bytes
/VOLUME3  193552617676 bytes
Create the data partitions and filesystems as configured above [y/n]?: y
*******************************************************
WARNING: All non-system data on the disk will be
permanently erased and unrecoverable.
Are you sure [y/n]?: y
The format process will take a few minutes. Once
the format process begins, it cannot be stopped.
Please do not power down or reboot the system until
the format process is complete.
Continue [y/n]?: y
*** Beginning format process ***
*** Removing previous data partitions - please wait ***
*** Creating new data partitions - please wait ***
*** Formatting partition /VOLUME1. Please wait... ***
mke2fs 1.41.14 (22-Dec-2010)
[...]
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
*** Formatting completed successfully ***
*** Formatting partition /VOLUME2. Please wait... ***
mke2fs 1.41.14 (22-Dec-2010)
[...]
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
*** Formatting completed successfully ***
*** Formatting partition /VOLUME3. Please wait... ***
mke2fs 1.41.14 (22-Dec-2010)
[...]
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 31 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
*** Formatting completed successfully ***
*** Format finished successfully
New partitions have been created ***
*** Mounting partitions ***
e2fsck 1.41.14 (22-Dec-2010)
VOLUME1: clean, 11/5914624 files, 418265/23626953 blocks
/VOLUME1 mounted
e2fsck 1.41.14 (22-Dec-2010)
VOLUME2: clean, 11/11821056 files, 789884/47254150 blocks
/VOLUME2 mounted
e2fsck 1.41.14 (22-Dec-2010)
VOLUME3: clean, 11/11821056 files, 789884/47253628 blocks
/VOLUME3 mounted
```

Power cycle the system after format is complete. You can re-enable any tasks that may have conflicted with the format, including local CDR and HDR generation, after the power cycle is complete.

Note: If you find that, after the first reboot, the system has not created new partitions, perform another reboot to resolve this issue, which is created by an incorrect dynamic partition table refresh.

7

VM Interfaces

Virtual machines need to interface with their virtual machine management systems to reach and share physical interfaces with other virtual machines. The Oracle Communications Session Border Controller assigns a unique pseudo MAC address from a virtual machine management system and maps it to the Oracle Communications Session Border Controller configuration naming syntax to direct network traffic to physical interfaces based on the type of service they support.

The MACTAB File

Interface mapping on the Oracle Communications Session Border Controller is statically defined by a configuration file named MACTAB.

Sample default Oracle Communications Session Border Controller interface mapping is presented below:

```
Device2# show interface-mapping
Interface Mapping Info-------------------------------------------
Eth-IF  MAC-Addr                Label
wancom0 00:50:88:BC:11:12       #generic
wancom1 00:50:88:BC:61:6C       #generic
wancom2 00:50:88:BC:11:C7       #generic
spare   00:50:88:BC:61:12       #generic
s0p0    00:50:88:BC:71:79       #generic
s1p0    00:50:88:BC:21:FF       #generic
s0p1    00:50:88:BC:41:A2       #generic
s1p1    00:50:88:BC:31:AC       #generic
s0p2    FF:FF:FF:FF:FF:FF       #dummy
s1p2    FF:FF:FF:FF:FF:FF       #dummy
s0p3    FF:FF:FF:FF:FF:FF       #dummy
s1p3    FF:FF:FF:FF:FF:FF       #dummy
```

The user creates physical-interface configurations using the names shown under the Eth_IF column, within service configuration elements. There are two types of physical interfaces that apply to the Oracle Communications Session Border Controller, segregated by the naming conventions:

- Media interfaces, shown above as s0p0 and s1p0.
- Management interfaces, shown above as wancom0 and wancom1

It is recommended that the user configure physical interfaces using the same naming conventions used in Oracle Session Border Controller that operates on proprietary platforms. These conventions, which simply use 's' for slot and 'p' for port, are visible in the MACTAB file.

Working with the MACTAB File

Interface identification on the Oracle Communications Session Border Controller is based on a system-level file called MACTAB that maps interface MAC addresses to interface naming that can be applied within Oracle Communications Session Border Controller configuration. In most cases, users retain the default mapping. The **show interface-mapping** command provide access to commands that allow the user see, change and even locate each interface based on this mapping. The MACTAB file is stored in the `/boot` folder. The MACTAB file ensures that interface mapping is persistent, and therefore usable, by your configuration regardless of changes to the system.

The **show interface-mapping** command displays the current mapping. An example of a MACTAB file that a user has configured is provided below.

```
CMS1# show interface-mapping
Interface Mapping Info
============================================================
Eth-IF       MAC-Addr                Label
wancom0      00:16:3E:30:00:2A    # ctrl port, onboard MAC
wancom1      00:16:3E:30:00:2B    # 2nd ctrl port, onboard MAC
s0p0         00:16:3E:30:00:2C    # First media interface
s1p0         00:16:3E:30:00:2D    # Second media interrface
============================================================
```

Serial Interfaces

In lieu of IP management access, serial access provides the user with direct access to the Oracle Communications Session Border Controller ACLI. The user must identify how their system allows for serial access. The serial interface can be a critical component of VM and COTS physical interface configuration as the user can make MACTAB changes via the serial interface without interrupting their own access during that management procedure.

Access to the Oracle Communications Session Border Controller serial interface is dependent on platform. Familiarity with the platform is required to understand physical location and default serial configuration.

Virtual machine management software provides a simulated serial port service from which the user gets direct serial access to each system. See your virtual machine manager's documentation for instructions on accessing these serial ports. COTS systems provide serial interfaces, such as RS232, from which the user can get serial access to bare metal installations. See your COTS documentation for the location and requirements of these interfaces.

Serial port configuration, via boot parameters, is the same across all platforms.

8

References and Debugging

ACLI Configuration Elements

The following sections describe the Oracle Communications Session Border Controller's configuration elements that are unique to S-CZ7.2.9.

datapath-config

The **datapath-config** commands configure VM core and memory utilization on the Oracle Communications Session Border Controller.

datapath-config set-dos-cores

This command sets which core is assigned to the DoS functionality. Valid values from 1 to the number of cores currently active on the VM. DoS cores are optional. To remove a DOS core use one of the other commands to overwrite the specified core.

Syntax

```
datapath-config set-dos-cores <Core-ID","Core-ID >
```

Arguments

The argument(s) entered for this command comprise a comma-separated list of 1 or more core IDs. The number of cores available to the Oracle Communications Session Border Controller are shown in the **show datapath-config** command.

Core IDs are 0-based. DoS core notation is marked by an "D".

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

datapath-config set-forwarding-cores

This command sets which cores are assigned to the forwarding engine. Valid values are from 0 to the number of cores currently active on the VM. More than 1 core can be specified.

Syntax

```
datapath-config set-forwarding-cores <Core-ID","Core-ID >
```

Arguments

The argument(s) entered for this command comprise a comma-separated list of 1 or more core IDs. The number of cores available to the Oracle Communications Session Border Controller are shown in the **show datapath-config** command.

Core IDs are 0-based. Forwarding core notation is marked by an "F".

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

datapath-config set-memory-size

This command sets the amount of memory to request at boot for the datapath. In the absence of this setting the application will calculate a reasonable memory amount. This amount is displayed in the **show datapath-config** command. Note that this setting may not be honored depending on circumstances such as available memory.

Syntax

```
datapath-config set-memory-size <MB of memory>
```

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

datapath-config set-signaling-cores

This command sets which cores are assigned to signaling processing and general system functionality. Valid values are from 0 to the number of cores currently active on the VM. More than 1 core can be specified.

Syntax

```
datapath-config set-signaling-cores <Core-ID","Core-ID >
```

Arguments

The argument(s) entered for this command comprise a comma-separated list of 1 or more core IDs. The number of cores available to the Oracle Communications Session Border Controller are shown in the **show datapath-config** command.

Core IDs are 0-based. Signaling core notation is marked by an "S".

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

datapath-config set-transcoding-cores

This command sets which cores are assigned to transcoding. Valid values are from 1 to the number of cores currently active on the VM. More than 1 core can be specified.

Enter the syntax information of your reference here (optional).

Syntax

```
datapath-config set-transcoding-cores <Core-ID","Core-ID >
```

Arguments

The argument(s) entered for this command comprise a comma-separated list of 1 or more core IDs. The number of cores available to the Oracle Communications Session Border Controller are shown in the **show datapath-config** command.

Core IDs are 0-based. Transcoding core notation is marked by an "X".

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

datapath-config set-num-of-pages

This command sets the number of pages used by the application.

Enter the syntax information of your reference here (optional).

Syntax

```
datapath-config set-num-of-pages <num-of-page>
```

Arguments

The argument(s) entered for this command is a single integer, specifying the number of pages to request from the system.

Mode

Superuser

Platform Restrictions

This command is only used with VM-based products.

RTC Status

A reboot is required to enact this command.

SNMP MIBs and Traps

The following MIBs and traps are supported for the Oracle Communications Session Border Controller. Please consult the Oracle Communications Session Border Controller MIB Reference Guide for more SNMP information.

apUsbcSysDPDKObjects

This group of objects, found in the `ap-usbcsys.mib`, provide a listing of DPDK statistics.

MIB Object	Object ID: 1.3.6.1.4.1.9148.3.17.1.1.13 +	Description
apUsbcSysDPDKFwdPurpose	.1	A bitset representing Forwarding cores. 1s represent forwarding cores, while 0s represent non-forwarding cores.
apUsbcSysDPDKDOSPurpose	.2	A bitset representing DoS cores. Bits set to 1 represent DoS cores, while 0s represent non-DoS cores.
apUsbcSysDPDKSigPurpose	.3	A bitset representing signaling cores. Bits set to 1 represent signaling cores, while 0s represent non-signaling cores.
apUsbcSysDPDKTransPurpose	.4	A bitset representing transcoding Cores. Bits set to 1 represent transcoding cores, while 0s represent non-transcoding cores.
apUsbcSysDPDKCmdLine	.5	System CmdLine string - as defined in /proc/cmdline. (including relevant bootparams.)
apUsbcSysDPDKFileMem	.6	Total DPDK File Memory.
apUsbcSysDPDKSysMem	.7	Total DPDK System Memory
apUsbcSysDPDKNum1G	.8	Number of 1GB Hugepages allocated.
apUsbcSysDPDKNum2MB	.9	Number of 2MB hugepages allocated.

MIB Object	Object ID: 1.3.6.1.4.1.9148.3.17.1.1.13 +	Description
apUsbcSysDPDKHypervisorType	.10	The description regarding the system type and what hypervisor the system is running on (OVM, KVM, VMWare,...).
apUsbcSysDPDKAddFwdCores	.11	Number of additional cores that may be used for forwarding.
apUsbcSysDPDKAddSigCores	.12	Number of additional cores that may be used for signaling.
apUsbcSysDPDKAddTransCores	.13	Number of additional cores that may be used for transcoding.

apUsbcSysScalingObjects

This group of objects, found in the `ap-usbcsys.mib`, provide a listing of objects relating to scaling VMs.

MIB Object	Object ID: 1.3.6.1.4.1.9148.3.17.1.1.12+	Description
apUsbcSysEstSessions	.1	Estimated number of unecrypted media sessions.
apUsbcSysEstG711G729Trans	.2	Estimated number of G711<->G729 transcoded media sessions.
apUsbcSysEstSigTPS	.3	Estimated number of signaling TPS.
apUsbcSysEstACLs	.4	Estimated number of ACLs.
apUsbcSysEstTCP	.5	Estimated number of TCP connections.
apUsbcSysEstTLS	.6	Estimated number of TLS connections.
apUsbcSysEstVLANs	.7	Estimated number of VLANs.

Show Commands

show datapath-config

This command displays the current VM operating parameters. The command has no arguments.

Syntax

```
show datapath-config
```

Output Description

Field	Description
Number of cores assigned to VM	The number of cores that this VM can use.

Field	Description
Current core assignment	The core assignment, based on function type: • S - system • D - DoS • F - Forwarding • X - Transcoding The position of the function indicates the core number.
Requested Page size	Configured page size.
Current Page size	Actual page size.
Number of 1 GB pages	Number of 1GB Hugepages allocated.
Number of 2 MB pages	Number of 1GB Hugepages allocated.
Total system memory	Total DPDK System Memory.
Memory reserved for datapath	Actual memory reserved for datapath.
Memory requested for datapath	Configured memory reserved for datapath.

show platform limits

This command displays the current limits for a variety of operating capacities. The output of **show platform limits** is based on the platform this command is executed from and the software version running. The command has no arguments.

Syntax

Sample output is displayed below.

```
ORACLE# show platform limits
Maximum number of sessions:3000
Maximum number of ACLS: 60000
Maximum number of common PAC buffers: 8000
Maximum number of kernel-rules: 216256
Maximum CPS rate: 300
Maximum number of TCP Connections: 60000
Maximum number of TLS Connections: 10
Maximum number of packet buffers: 30000
Maximum Signaling rate: 4000
Maximum number of session agents: 125
Maximum number of System ACLs: 256
Maximum number of VLANs: 4096
Maximum number of ARPs: 4104
Maximum number of INTFC Flows: 4096
Maximum number of Static Trusted Entries: 8192
Maximum number of Untrusted Entries: 4096
Maximum number of Media Entries: 6000
Maximum number of Deny Entries: 8192
Maximum number of Internal Flows: 32
Maximum number of Sip Rec Sessions: 512
Maximum number of RFC 2833 Flows: 6000
Maximum number of SRTP Sessions: 500
Maximum number of QoS Sessions: 3000
```

```
Maximum number of Xcoded Sessions: 100
Maximum number of HMU Flows: 6000
Maximum number of Transport Sessions: 0
Maximum number of MSRP Sessions: 0
Maximum number of SLB Tunnels: 0
Maximum number of SLB Endpoints: 0
Maximum number of IPSec SAs: 0
Maximum Licensed Capacity: 256000
```

Supporting Configuration

The following configuration elements which are not mentioned in this guide are required for the Oracle Communications Session Border Controller to function. Please refer to the Oracle Communications Session Border Controller ACLI Configuration Guide for details about configuring all supporting elements.

- network-interface
- physical-interface
- realm-config
- sip-config
- system-config

The following configuration elements are mentioned in this guide briefly and still require configuration:

- local-policy
- session-agent
- sip-interface

Log Files for the VNF

The following new log files have been added:

- log.usdp: DPDK processing log messages
- log.usdpClient: Log messages related to overall system

A

Known Issues and Caveats

Known Issues

This section lists the known issues that the customer needs to be aware of when deploying the Oracle Communications Session Border Controller version SCZ7.2.9.

- When the Oracle Communications Session Border Controller approaches or exceeds maximum concurrent sessions, call quality will degrade for both new and existing sessions.
- OVM-based deployments do not support para-virtualized (PV) drivers. As a workaround, OVM-based deployments can use SR-IOV drivers.
- When configured with more than 1 forwarding core, calls on the Oracle Communications Session Border Controller may not be balanced across all forwarding cores.
- Media and management (wancom) interfaces may not be configured with the same subnet, regardless of VLAN.

Caveats

This section lists caveats to be aware of when deploying the Oracle Communications Session Border Controller.

- The actual maximum signaling value for an S-Cz7.2.9 deployment, may diverge from the value reported in the **show platform limits** command.
- It may be necessary to disable hypervisor / vSwitch anti-spoofing security features on media interfaces in order for HA fail-overs to be successful.

B

SCz7.2.9 Feature Support

Release SCz7.2.9 feature support differs from all other Oracle Communications Session Border Controller releases based on the introduction of the product as a VNF.

Management Features

The Management features consist of all items related to the management interface including the HA interfaces.

This release supports the following:

- Alarm Manager
- SFTP Server
- Interface Configuration
- IPv4 for wancom
- IPv6 for wancom
- SNMP IPv4 monitoring
- SSH
- System ACLs - IPv4 / IPv6
- Telnet
- VLAN

This release does not support the following:

- FTP
- IPSec for wancom0
- SNMP IPv6 monitoring

Install/Boot Features

The Install/Boot/Upgrade features consist of items related to installation of the software on the target platform, booting that platform, and any issues with upgrading that platform.

This release supports the following:

- Support for OVM Hypervisor (3.3)
- Support of KVM Hypervisor

This release does not support the following:

- Interface Re-arrangement

- [Re-]Install from PXE server

Signaling Features

The Signaling Features consist of items related to and pre-processing of signaling packets on media interfaces as well as their delivery to the application stack. In addition it includes the ICMP protocol that use the media interfaces.

This release supports the following:

- High-capacity SCTP (ASCTP)
- ICMP - IPv4 / IPv6
- SIP-TLS
- Traceroute

This release does not support the following:

- Source based routing/OLIP

Datapath Features

The Datapath Features consist of items related to the media interface and datapath framework, which is responsible for packets accepted from physical interfaces through their processing, whether that is to the application (i.e. signaling) or fully processed (i.e. media).

This release supports the following:

- Configurable MTU
- DPWD
- GW Heartbeat
- High Availability
- Interface Configuration
- Jumbo packets
- Packet Trace - local
- VLAN support
- Trusted/Untrusted DDOS
- SA Minimum Reserved Bandwidth
- SLB Cluster Member

This release does not support the following:

- Lawful Interception (UDP/RTP) IPv4, inner, outer
- Lawful Interception (UDP/RTP) IPv6 inner, outer
- IP Routing to a 3rd-party on-board process
- IPSEC
- MAC Filtering
- Packet Matching - Dest Port - PQ
- Packet Trace - remote - IPv4
- Packet Trace - remote - IPv6
- PHY Link Redundancy

Media Features

The Media features consist of items related to the processing of media packets on media interfaces. This release supports the following:

- RFC 2833 Support - Detect / Dual / Generate
- Collapsed Flows
- Fragmentation and Reassembly

- Guard timers
- Latching - IPv4 / IPv6
- NAT - IPv4 / IPv6 / IWF
- Payload Type Mapping
- QoS per flow IPv4 / IPv6 statistics collection & reporting
- TOS - IPv4 / IPv6
- UDP checksum recalculation
- Hide media update

This release does not support the following:

- IP Loopback (Latch and late drop)
- Media Policing
- Non-terminated TCP media
- SIPREC
- SRTP
- Software Transcoding
- Hardware Transcoding
- Hardware based RFC 2833 In-band detection / generation
- Comfort Noise Generation
- Lawful Interception - X3 TCP - IPv4 (inner, outer)
- Lawful Interception - X3 TCP - IPv6 (inner)
- Lawful Interception - X3 TCP - IPv6 (outer)
- Legacy Call Recording (SRR)
- MSRP - TCP / TCP LI / TLS
- Ringback tone generation (low scale)
- RTCP generation on non-xcoded calls
- RTCP generation on non-xcoded calls (DSP)
- RTCP generation on Xcoded calls (DSP)
- Terminated TCP media
- TSCF - TLS / DTLS
- SCTP

Application Features

The following application features are supported for this release:

- Peering SBC
- SIP/UDP and SIP/TCP
- Pooled Transcoding

The following application features are NOT supported for this release:

- Access Session Border Controller
- Geo-redundancy

Coproduct Support

This release is compatible with:

- Oracle Communications Application Orchestrator
- Oracle Communications Session Delivery Manager

Please see those products' documentation for version compatibility.

This release is not compatible with:

- Oracle Communications Operations Monitor