



JunosV App Engine

Administration Guide

Release
13.3



Published: 2014-03-31

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2014, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

JunosV App Engine Administration Guide
13.3

Copyright © 2014, Juniper Networks, Inc.
All rights reserved.

Revision History
April 2014—JunosV App Engine Administration Guide 13.3R2

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	ix
	Documentation and Release Notes	ix
	Documentation Conventions	ix
	Documentation Feedback	xi
	Requesting Technical Support	xi
	Self-Help Online Tools and Resources	xii
	Opening a Case with JTAC	xii
Chapter 1	Configuration Topics	13
	Introduction to JunosV App Engine	13
	Deploying an Application	14
	Configuring the DHCP Server and TFTP Server for Compute Nodes	23
	SNMP and the JunosV App Engine MIB	30
	Configuring SNMP Traps for JunosV App Engine	31
	Configuring the Syslog Server for Compute Nodes	31
	VT-d Mode for Remote Applications	32
Chapter 2	JunosV App Engine Configuration Statements	35
	app-engine	36
	bridge	39
	compute-cluster (App Engine)	41
	compute-cluster (VM instance)	43
	compute-node	44
	ethernet	46
	family (Compute Cluster)	48
	family (Virtual Machines)	49
	instance	50
	interface	52
	interfaces	53
	local-management	54
	secondary-disk	55
	subagent	56
	syslog	57
	virtual-machines	59
Chapter 3	JunosV App Engine Operational Commands	61
	request app-engine offline compute-cluster compute-node	63
	request app-engine reboot compute-cluster compute-node	65
	request app-engine reboot compute-cluster compute-node virtual-machine-instance	67
	request app-engine service (start stop restart) compute-cluster compute-node	69

	show app-engine info	71
	show app-engine processes compute-cluster compute-node	74
	show app-engine resource-usage	78
	show app-engine status	85
	show app-engine virtual-machine package	88
	show app-engine virtual-machine instance	90
Chapter 4	Indexes	93
	Index	95
	Index of Statements and Commands	97

List of Figures

Chapter 1	Configuration Topics	13
	Figure 1: Interface Mapping to Ports on a VSE 1100 Device	16
Chapter 2	JunosV App Engine Configuration Statements	35
	Figure 2: Interface Mapping to Ports on a VSE 1100 Device	46

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	x
	Table 2: Text and Syntax Conventions	x
Chapter 1	Configuration Topics	13
	Table 3: DHCP Licenses for Juniper Routers	24
Chapter 3	JunosV App Engine Operational Commands	61
	Table 4: app-engine Operational Mode Commands	61
	Table 5: request app-engine-offline compute-cluster compute-node Output Fields	63
	Table 6: request app-engine-reboot compute-cluster compute-node Output Fields	65
	Table 7: request app-engine reboot compute-cluster compute-node virtual-machine-instance	67
	Table 8: request app-engine-(start stop restart) compute-cluster compute-node Output Fields	70
	Table 9: show app-engine info Output Fields	72
	Table 10: show app-engine processes compute-cluster compute-node Output Fields	74
	Table 11: show app-engine resource-usage Output Fields	79
	Table 12: show app-engine status Output Fields	86
	Table 13: show app-engine virtual-machine package Output Fields	88
	Table 14: show app-engine virtual-machine instance Output Fields	90

About the Documentation

- Documentation and Release Notes on page ix
- Documentation Conventions on page ix
- Documentation Feedback on page xi
- Requesting Technical Support on page xi

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Documentation Conventions

Table 1 on page x defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page x defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: <code>user@host> configure</code>
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> <i>RFC 1997, BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: <code>[edit]</code> <code>root@# set system domain-name <i>domain-name</i></code>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	<code>stub <default-metric <i>metric</i>>;</code>

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast <i>(string1 string2 string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract,

or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

CHAPTER 1

Configuration Topics

The following topics provide overview, conceptual, or detailed procedures related to the JunosV App Engine configuration statements:

- [Introduction to JunosV App Engine on page 13](#)
- [Deploying an Application on page 14](#)
- [Configuring the DHCP Server and TFTP Server for Compute Nodes on page 23](#)
- [SNMP and the JunosV App Engine MIB on page 30](#)
- [Configuring SNMP Traps for JunosV App Engine on page 31](#)
- [Configuring the Syslog Server for Compute Nodes on page 31](#)
- [VT-d Mode for Remote Applications on page 32](#)

Introduction to JunosV App Engine

JunosV App Engine enables third-party applications—applications written to run in Linux—to run on a remote virtualized system called a *guest OS*. These applications, which can be control-plane, management-plane, or data-plane applications, are referred to as *remote applications*. JunosV App Engine enables remote applications to run in their native environment without requiring porting to Junos OS.

JunosV App Engine provides a virtualized environment with a Kernel-based Virtual Machine (KVM) hypervisor, which runs on the host OS. The host OS controls the creation of virtual machines (VMs) on top of the hypervisor. The hypervisor and host OS run within a VSE device called a *compute node*. A JunosV App Engine compute node is in the form of an external device, provided by Juniper Networks, which runs CentOS, KVM, and Qemu. Compute nodes are used to spawn virtual machines (VMs), each of which runs either a Juniper Networks or third-party application within a guest OS. The external device is connected to a router using a Layer 2 or Layer 3 network. The compute node is connected to a device running Junos OS. This device can be a router, switch, or gateway.

Multiple remote applications can communicate with Junos OS. Remote applications run in a VM on the external system and communicate through a Service Broker that runs on the Junos device. The Service Broker on the Junos device functions as a front end for services provided by Junos daemons such as *rpd*, *mgd*, and so forth, and relays information between the Junos daemons and the remote applications on the external system. The remote applications on the external system can access the Junos daemons on the Junos device.

The *JunosV App Engine Administration Guide* provides CLI configuration information for configuring the compute node and provisioning the VMs. It also provides the operational commands that are available to monitor the compute node and VM settings. In the CLI configuration mode, there are two hierarchy levels to configure: the physical connection between the device running Junos OS and the hardware appliance (which is under the **[edit services app-engine compute-cluster]** hierarchy level) and the virtual machines (VMs), where the applications run, each on its own guest OS (in the **[edit services app-engine virtual-machines]** hierarchy level).

Currently, Linux and Ubuntu are the only supported guest OSes. The only device the compute node can connect to is a Juniper Networks M Series, MX Series, or T Series router. The type of connection is limited to a Layer 2 connection. Changes to the VM part of the CLI configuration are destructive and will cause the VM to reboot. In the compute cluster part of the CLI configuration, changes to management interfaces are destructive and require a reboot of the compute node.

- Related Documentation**
- [app-engine on page 36](#)
 - [compute-node on page 44](#)

Deploying an Application

This topic takes you through booting up the JunosV App Engine and deploying an application on the JunosV App Engine.

Required Hardware and Software In order to install and use the JunosV App Engine platform, please make sure you have the following items:

1. **Hardware and Software Platform Components:** To deploy a remote application—an application written using the JunosV App Engine development environment—you must first set up the JunosV App Engine platform. You will need the following hardware and software platform components.
 - Device running Junos: For the 13.2R1 release this is an M Series, an MX Series, or a T Series router. You need to make sure you have enough space to store the downloads, approximately 1.9 GB.
 - VSE device: This could be either a VSE1100 or VSE2100 device. For VSE device chassis specifications, see [Device Features and Specifications](#) in the *Virtual Services Engine Hardware Installation Guide*.
 - DHCP server: You need a DHCP server for the PXE boot-up of the VSE device. You can use a dedicated DHCP server or the Juniper Networks router your VSE device is tethered to as the DHCP server. You will need to purchase a DHCP license in the following circumstances:
 - If you are using the DHCP server functionality on a Juniper Networks M Series or MX Series router
 - If you have your own DHCP server and are relaying DHCP messages through a Juniper Networks M Series or MX Series router. If, however, the DHCP messages go directly to the VSE device, you do not need any additional licenses.

For more details on what licenses to buy, see the configuration topic “[Configuring the DHCP Server and TFTP Server for Compute Nodes](#)” on page 23.

- Console server: This device is optional and used mainly for debugging purposes.

2. **Software Packages:** You also need the following software:

- Junos OS package, Release 12.3R1 or later



NOTE: As of Junos OS Release 12.3, JunosV App Engine deployed applications will install only if the Junos OS release matches the release of the backing sandbox used to package the application. For example, an application packaged with Release 12.3R2 will only install on Junos OS Release 12.3R2 and will not install on Junos OS Release 12.3R1 or Junos OS Release 12.3R3 or Junos OS Release 13.1R1.

- JunosV App Engine Host Base OS: `jvae-release-version.tgz`

This package, also referred to as the JunosV App Engine Software, contains the JunosV App Engine Host Base operating system (OS). There is only one version of the JunosV App Engine Host Base OS that can be installed at one time. Make sure you are running the same release of Junos OS (`jinstall-release-version.tgz`) as the JunosV App Engine Host Base OS package you are going to install.



NOTE: To download this software package, see the [JunosV App Engine Download Site](#).

- A remote application package—optional if you just want to boot the VSE device

A remote application package is not supplied at the download site. You get this package from the provider of the application.

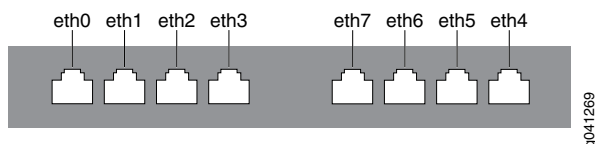
When you download the packages, verify that the MD5 and SHA1 checksums of the downloaded files match those of the corresponding files on the download website.

3. **Development Certificate:** If you have not already received a development certificate, please contact JunosV App Engine technical marketing team at sdk-cert@juniper.net. You will need the certificate when you begin packaging your custom application.

Boot the JunosV App Engine

On the VSE device, there are two sets of four ports for the VSE1100 device and one set of four ports and one set of two ports for the VSE2100 device. The set of four ports on the left side when you are facing the VSE device are *onboard ports*. The other ports are *IOC ports*. The onboard ports are interfaces named eth0 to eth3 going from leftmost to right as you face the device. The IOC ports are numbered eth7 down to eth4 going from leftmost to right. See [Figure 1 on page 16](#).

Figure 1: Interface Mapping to Ports on a VSE 1100 Device



To deploy an application you need boot the VSE device first and then install the application package.

To boot the JunosV App Engine:

1. Connect the router running Junos OS and the VSE device:
 - a. Choose any one of the VSE device interfaces as the management interface. Make sure there is a physical or Layer 2 connection from this interface to the router.
 - b. (Optional, but required for remote power cycling.) Connect the console port on the left side of the VSE device to a console server.
 - c. Write down the MAC address of the interface you have connected the router to.

The MAC address of the management interface is required for completing the installation (Step 2). There are labels on the back of the VSE chassis with the MAC addresses of the interfaces.

When the VSE device powers on and attempts to PXE boot, the MAC addresses of onboard interfaces are displayed. You can also monitor the console log for this information.

2. Install the software packages on the router and VSE device.
 - a. Verify you have the correct release of the Junos OS installed on the router (12.3R1 or later and a release matching rhw release of the backing sandbox used to package the application).
 - b. Install the JunosV App Engine Host Base OS package on the router. Go to the operational mode of the CLI to issue the following command.

```
user@router> request system software add jvae-release-version.tgz
```

This package contains the JunosV App Engine Host Base OS. The JunosV App Engine Host Base OS is installed on the VSE device based on the platform configuration in Step 3. Currently, only one version of the JunosV App Engine Host Base OS can be installed at one time.

3. Configure the platform layer.

Configure the physical connection between the router running Junos OS and the VSE device. You do this at the **[edit services app-engine compute-cluster]** hierarchy level. For details on options, see the [compute-cluster \(App Engine\)](#) configuration statement section. Links to child configuration statements are provided in the Syntax section of the configuration statement sections.

Here is an example of the configuration you must set. Use the configuration mode of the CLI to configure the settings. A minimum of one interface is required to boot the VSE device.

```
[edit services app-engine]
user@router# show
compute-cluster cc1 {
  local-management {
    family inet {
      address 192.168.1.20; # The IP address of the interface on the router
                          # tethered to the VSE device
    }
  }
  compute-node cn1 {
    mac-address 00:e0:81:ca:5a:76; # The MAC address of the interface on the VSE
    # device that is used to tether it to the router
    interfaces {
      bridge br0 {
        management;
        family inet {
          address 192.168.1.11/24; # IP address belongs to the bridge.
          # You can ssh to this address or use the console to monitor the VSE device
        }
        interface eth0; # Physical interface on the VSE device tethered to the router.
      }
    }
  }
}
```

The IP address under the **[edit services app-engine compute-cluster ccl local-management family inet address]** hierarchy level is the interface that is used to manage the entire compute cluster.

You can use the FreeBSD shell to check the VSE connection at the interface address 192.168.1.11/24. Type **ssh root@192.168.1.11** and enter the password when prompted. You can also use the console to monitor the VSE device.

4. Configure the TFTP server and the DHCP server.

BIOS is enabled for PXE boot at the time of manufacture for all VSE device interfaces. This step shows how to enable PXE boot of the VSE device by configuring the Junos router as a DHCP server. For more examples and information about DHCP configuration, see the configuration topic [“Configuring the DHCP Server and TFTP Server for Compute Nodes”](#) on page 23.

a. To enable the TFTP server on Junos:

- Use the following **set** command:

```
user@router# set system services tftp-server
```

b. To configure the Junos router as a DHCP server:

- i. Configure the **dhcp-local-server group** statements at the **[edit system services]** hierarchy level.

```
[edit]
user@router# set system services dhcp-local-server group group-name interface
interface-name
```

For example:

```
user@router# set system services dhcp-local-server group dhcp-group1 interface
ge-4/2/0.0
```

- ii. Configure the DHCP attributes. For example, use these **set** commands:

```
# set access address-assignment pool dhcp-pool family inet network
192.168.1.0/24
# set access address-assignment pool dhcp-pool family inet dhcp-attributes
router 192.168.1.20
# set access address-assignment pool dhcp-pool family inet dhcp-attributes
boot-file pxelinux.0
# set access address-assignment pool dhcp-pool family inet dhcp-attributes
boot-server 192.168.1.20
# set access address-assignment pool dhcp-pool family inet dhcp-attributes
tftp-server 192.168.1.20
```

- iii. Configure the host interface MAC address and IP address. For example:

```
# set access address-assignment pool dhcp-pool family inet host dhcp-node
hardware-address 00:e0:81:ca:5a:76
# set access address-assignment pool dhcp-pool family inet host dhcp-node
ip-address 192.168.1.251
```

- iv. Commit your configuration.

```
[edit]
user@router# commit
commit complete
```

5. Boot the VSE device.

If you have not already done so, power on the VSE device. Depending on the interface you chose to connect to your router, you may have to wait for the VSE device to find the management connection and boot.

6. Verify your configuration.

- a. Verify the TFTP and DHCP server configuration with the show command.

```
[edit]
user@router# show interfaces
ge-4/2/0 {# interface on router to which VSE device is tethered
  unit 0 {
    family inet {
      address 192.168.1.20/24;
    }
  }
}
```

```

[edit]
user@router# show system services
tftp-server;
dhcp-local-server
    group dhcp-group1 {
        interface ge-4/2/0.0;
    }
}
[edit]
user@router# show access
address-assignment {
    pool dhcp-pool {
        family inet {
            network 192.168.1.0/24;
            dhcp-attributes {
                router {
                    192.168.1.20;
                }
                boot-file pxelinux.0;
                boot-server 192.168.1.20;
                tftp-server 192.168.1.20;
            }
            host dhcp-node {
                hardware-address 00:e0:81:ca:5a:76;
                ip-address 192.168.1.251;
            }
        }
    }
}

```

- b. Verify the status of your connection using the **show app-engine status** command. This command is an operational command. Exit from configuration mode in the CLI.

```

user@router> show app-engine status
Compute cluster: cc1
  Compute node   Status
  cn0            Online

```

This is the status of the App Engine. **Online** indicates that the node is booted and ready to accept virtual machines.

Deploy an Application The rest of this section gives step-by-step instructions on deploying an application on the JunosV App Engine.

To deploy an application on a running JunosV App Engine:

1. Authorize the application on the router and VSE device.

As the operator, you will need to authorize installation of the application package on your router and VSE device.

- a. Go to the configuration mode of Junos CLI and then go to the **[edit system extensions]** hierarchy level.

- b. Configure and commit the provider name, the license type, and the deployment scope associated with the application package. This information is supplied by the application package provider.

```
[edit system extensions]
user@router# set providers name license-type license deployment-scope
deployment
```

For example, if you were to install an application provided by Juniper Networks, you might configure and commit the following authorization settings:

```
[edit system extensions]
user@router# set providers juniper license-type juniper deployment-scope
commercial
```

If you do not authorize the application in this way, you will get this error message:

```
ERROR: validate-config: Packages from juniper are not allowed WARNING:
Current configuration not compatible with /var/tmp/package-name
```

2. Install the application package on the router.

This software is not provided on JunosV App Engine download site; it comes from the provider of the application.

```
user@router> request system software add app-package-name-release-version.tgz
The software is deployed on top of JunosV App Engine platform as a guest VM.
```

3. Configure and provision the virtual machines (VMs).

The VMs are where the applications run. Each application runs on its own guest OS. Configure and provision the VMs at the `[edit services app-engine virtual-machines]` hierarchy level. For details on options, see the [instance](#) configuration statement section. Links to child configuration statements are provided in the Syntax section of the configuration statement sections.

Here is an example of the configuration you must set:

```
[edit services app-engine]
user@router# show
virtual-machines {
  instance helloworld-vm {
    package sample-jvae-app-ve;
    compute-cluster cc1 {
      compute-node cn1;
    }
    interface eth0 {
      bridge br0;
      family inet {
        address 192.168.1.12/24;
      }
    }
  }
}
```



TIP: If the control traffic to and from the VMs and router is not too much, you should make the VM IP address be part of the same subnet as the

compute node management IP address. This strategy uses fewer physical ports than is required to form a separate VM network.

4. Commit the configuration and exit. When you commit your configuration, the system transfers the disk image of the VM, which is the part of the package installed on the Routing Engine, to the VSE device, and uses this disk image to boot up.

- a. Commit the changes to the configuration.

```
[edit]
user@router# commit
commit complete
```

- b. Exit the configuration mode.

```
[edit]
user@router# exit
Exiting configuration mode
```

5. Verify the configuration with show commands.

- a. Use the **show app-engine virtual-machine package** command to show the status of the download of the package.

```
user@router> show app-engine virtual-machine package
VM package: sample
VM disk image: sample.img.gz
Compute cluster      Package download status
cc1                  DOWNLOADED
```

- b. Use the **show app-engine virtual-machine instance** command to show the status of the VM. If the host manager is able to bring up the VM, the status is **Active**.

```
user@router> show app-engine virtual-machine instance
VM name      Compute cluster  VM status
sample      cc1              Active
```

The detail version of this command gives the following additional information:

```
user@router> show app-engine virtual-machine instance detail
VM name: sample
VM status: ACTIVE
Compute cluster name: cc1
Compute node name: cn1
VM UUID: cfe28222-f142-11e1-ba37-00e081cbbc57
VM disk image: sample.img
Number of CPUs: 1
Memory (GB): 1
Console: 192.168.1.11 15001
VM PID: 6824
eth0: 192.168.1.12
hda: 8 GB
```

The VM has a virtual console with the connectivity information as displayed (see the **Console** field in the output from the detail command above). You can log on to the VM console from the Juniper Networks device's FreeBSD shell. You can drop to the FreeBSD shell by typing **start shell**. From there you can log on to the VM console by typing **telnet 192.168.1.11 15001**, for example.

You can also ssh to the IP address of the management interface of the VM (see the **eth0** field in the output from the detail command above) if ssh has been enabled inside the VM.

You are now ready to run the application.

**Related
Documentation**

- [compute-cluster \(App Engine\) on page 41](#)
- [ethernet on page 46](#)
- [Configuring the DHCP Server and TFTP Server for Compute Nodes on page 23](#)
- [instance on page 50](#)

Configuring the DHCP Server and TFTP Server for Compute Nodes



NOTE: As of Junos OS Release 12.3, JunosV App Engine deployed applications will install only if the Junos OS release and the release of the backing sandbox used to package the application match. For example, an application built with Release 12.3R2 will only install on Junos OS Release 12.3R2 and will not install on Junos OS Release 12.3R1 or Junos OS Release 12.3R3 or Junos OS Release 13.1R1.

A compute node is the VSE device that runs the host operating system (OS) in the virtual plane—it connects the router or other device running Junos to the hardware device.

The VSE device does not have its own operating system. The only way to boot the compute node is through the network interface. The network interface, configured as a boot device, uses a protocol called Preboot eXecution Environment (PXE) to boot the compute node.

To boot the VSE device, you need to do the following:

- Install the JunosV App Engine Host Base OS package (jvae package).
- Configure the compute-cluster statement.
- Configure the TFTP server.
- Configure the DHCP server.

The Host Base OS package has the boot file and boot server as well as the entire Linux CentOS system. When the Host Base OS package is installed and the DHCP server is configured, the DHCP server sends the IP address of the interface, the boot filename, and the boot server to the VSE device. The VSE device can then boot and download the host operating system (CentOS) and the compute cluster configuration.

This topic details how to configure the TFTP server and the DHCP server. Installing the Host Base OS package and configuring the **compute-cluster** statement are covered in other steps in the [“Deploying an Application” on page 14](#) tutorial.

TFTP Server Configuration

The TFTP server is run on the Junos router, where the boot images are stored.

To enable the TFTP server on Junos:

- Use the following **set** command:

```
user@router# set system services tftp-server
```

DHCP License

Depending on the router, the VSE is connected to, here are the additional SKUs you will need to buy. Once you purchase the SKU you will need to load the license on Juniper router.

For Juniper Networks M Series and MX Series routers a license is required for DHCP. See [Table 3 on page 24](#).

Table 3: DHCP Licenses for Juniper Routers

Router Series	Platforms	DHCP SKU Needed *
MX Series	MX-80 MX-5, MX-10, MX-40	S-MX80-SA-FP S-MX80-SSM-FP
	MX-240, 480, 960, ...	S-SA-FP
M-Series	M7i, M10i, M320, M120	S-SA-FP
T-Series	T640, T320	No License required.
* Also would need a scale license (if number of clients > 1000)		

To add the license key:

1. Type the following CLI operational command at the prompt:

```
user@host> request system license add terminal key
```
2. Paste the license key for *key*. Then press Ctrl+d.

General DHCP Configuration

You can configure and use any DHCP server in the network. But you must configure the DHCP server as a PXE server to respond to the PXE clients.



NOTE: If the compute node receives multiple DHCP responses (which is possible if there are multiple DHCP servers configured in the network), it is up to the compute node to choose any DHCP response. In the case that the compute node chooses a DHCP response that is not the one the administrator has configured for PXE boot to work, the compute node will not boot with the host OS.

Because any router can be used as the DHCP server, the procedure to configure the DHCP server to send the appropriate DHCP reply to the network interface is specific to each deployment.

Whichever device you use as the DHCP server, you need to do the following:

1. Add the following two fields in the DHCP response:
 - tftp boot server—This is the tftp boot server field. Set it to the IP address of the network interface connected to the network boot interface of the compute node.
 - In the case of a Linux DHCP server, this field's name is **next-server** in **/etc/dhcpd.conf**.
 - In the case of a Junos DHCP server, this field's name is **tftp-server** at the **[access address-assignment pool *poolname* family inet dhcp-attributes]** hierarchy level.
 - boot filename—This is the filename of the boot file on the tftp boot server. Set boot filename to **pxelinux.0**.
 - In the case of a Linux DHCP server, this field's name is **filename** in **/etc/dhcpd.conf**.
 - In the case of a Junos DHCP server, this field's name is **boot-file** at the **[access address-assignment pool *poolname* family inet dhcp-attributes]** hierarchy level.
2. Configure the **forwarding-options** configuration statement on the router to which DHCP server is connected.



NOTE: The **forwarding-options** configuration in this step is required if and only if the compute node management network and DHCP network (assuming that DHCP is not running on Junos OS) are different. It is only in this case that the router needs to forward DHCP traffic (which are broadcast in nature) to another network where the DHCP server is running. For example, your VSE device may be in a different network than your DHCP server. If the DHCP server is in the same network as the compute node, then this configuration is not required.

```
[edit]
user@host# show interfaces
ge-4/2/0 {
    unit 0 {
        family inet {
            address 192.168.1.3/24;
        }
    }
}

[edit]
user@host# show forwarding-options
dhcp-relay {
    server-group {
        bng-vee;
    }
    active-server-group bng-vee;
    group all {
        interface ge-4/2/0.0;
    }
}
```

Following are a few example configurations.

Configuring DHCP on Junos

To configure a Junos router as a DHCP server:

1. Configure the **dhcp-local-server group** statements at the **[edit system services]** hierarchy level.

```
[edit]
user@router# set system services dhcp-local-server group group-name interface
interface-name
```

For example:

```
user@router# set system services dhcp-local-server group dhcp-group1 interface
ge-4/2/0.0
```

2. Configure the DHCP attributes. For example, use these set commands:

```
set access address-assignment pool dhcp-pool family inet network 192.168.1.0/24
set access address-assignment pool dhcp-pool family inet dhcp-attributes
boot-file pxelinux.0
set access address-assignment pool dhcp-pool family inet dhcp-attributes
boot-server 192.168.1.20
set access address-assignment pool dhcp-pool family inet dhcp-attributes
tftp-server 192.168.1.20
```

3. Configure the host interface MAC address and IP address. For example:

```
set access address-assignment pool dhcp-pool family inet host sdk-proto-bm2
hardware-address 00:e0:81:ca:5a:76
# Mac address of interface on VSE tethered to router.
set access address-assignment pool dhcp-pool family inet host sdk-proto-bm2
ip-address 192.168.1.251
```

4. Commit, and verify your configuration with the show command.

```
[edit]
user@router# show interfaces
ge-4/2/0 {
  unit 0 {
    family inet {
      address 192.168.1.20/24;
    }
  }
}
```

```
[edit]
user@router# show system services
tftp-server;
dhcp-local-server
  group dhcp-group1 {
    interface ge-4/2/0.0;
  }
}
```

```
[edit]
user@router# show access
address-assignment {
  pool dhcp-pool {
    family inet {
      network 192.168.1.0/24;
      dhcp-attributes {
        router {
```

```

        192.168.1.20;
    }
    boot-file pxelinux.0;
    boot-server 192.168.1.20;
    tftp-server 192.168.1.20;
}
host sdk-proto-bm2 {
    hardware-address 00:e0:81:ca:5a:76;
    ip-address 192.168.1.251;
}
}
}
}
}

```

Configuring forwarding-options for DHCP:

This example uses an external Linux machine as the DHCP server.

To configure **forwarding-options** on a Junos router for DHCP:

1. Configure the following settings:

- Configure the router's interfaces:

```

[edit]
set interfaces ge-0/1/3 unit 0 family inet address 10.5.5.254/24;
set interfaces ge-0/1/6 unit 0 family inet address 10.1.1.254/24;

```

- Configure the **forwarding-options dhcp-relay** statements:

```

[edit]
set forwarding-options dhcp-relay server-group vee 10.5.5.10
set forwarding-options dhcp-relay active-server-group vee
set forwarding-options dhcp-relay group all interface ge-0/1/6.0

```

- Configure the compute cluster and compute node.

```

[edit]
set services app-engine compute-cluster dhcp-cluster local-management family
  inet address 10.1.1.254
set services app-engine compute-cluster dhcp-cluster compute-node dhcp-node
  mac-address 00:e0:81:ca:5a:74
set services app-engine compute-cluster dhcp-cluster compute-node dhcp-node
  interfaces bridge mgmt management
set services app-engine compute-cluster dhcp-cluster compute-node dhcp-node
  interfaces bridge mgmt family inet address 10.1.1.50/24
set services app-engine compute-cluster dhcp-cluster compute-node dhcp-node
  interfaces bridge mgmt interface eth1

```

- Configure TFTP server.

```

[edit]
set system services tftp-server

```

2. Commit, and verify the settings.

```

user@router# show interfaces
ge-0/1/3 {
  unit 0 {
    family inet {

```

```
        address 10.5.5.254/24;
    }
}
ge-0/1/6 {
    unit 0 {
        family inet {
            address 10.1.1.254/24;
        }
    }
}

[edit]
user@router# show forwarding-options dhcp-relay
server-group {
    vee {
        10.5.5.10;
    }
}
active-server-group vee;
group all {
    interface ge-0/1/6.0;
}

[edit]
user@router# show services
app-engine {
    compute-cluster dhcp-cluster {
        local-management {
            family inet {
                address 10.1.1.254;
            }
        }
        compute-node dhcp-node {
            mac-address 00:e0:81:ca:5a:74;
            interfaces {
                bridge mgmt {
                    management;
                    family inet {
                        address 10.1.1.50/24;
                    }
                }
                interface eth1;
            }
        }
    }
}

[edit]
user@router# show system
services {
    tftp-server;
}
```

Configuring DHCP on Linux

To install and configure DHCP on a Linux machine:

Make sure yum updatedd is running on the Linux machine.

1. Run the **yum install dhcpd** command on the Linux machine. The dhcp package is installed.
2. Update the **dhcp.conf** file. See the following sample file:

```
[root@host ~]# cat /etc/dhcpd.conf
#
# Sample DHCP Server Configuration file.
#
ddns-update-style ad-hoc;
subnet 10.1.1.0 netmask 255.255.255.0 {
    group {
        option subnet-mask 255.255.255.0;
        host node {
            hardware ethernet 00:e0:81:ca:5a:74;
            fixed-address 10.1.1.50;
            next-server 10.1.1.254;
            filename "pxelinux.0";
        }
    }
}

subnet 10.5.5.0 netmask 255.255.255.0
```

3. Issue the **service dhcpd restart** command so that the machine restarts with the new configuration.

The following configuration is a different example DHCP server configuration in Linux:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    group {
        option subnet-mask 255.255.255.0;
        host cn2 {
            hardware ethernet 00:E0:81:CA:5A:77;
            fixed-address 192.168.1.20;
            next-server 192.168.1.3;
            filename "pxelinux.0";
        }
    }
}
```

This list gives a description of the configured parameters:

subnet 192.168.1.0 netmask 255.255.255.0—This is the subnet to which the compute node's network boot interface is connected.

hardware ethernet 00:E0:81:CA:5A:77—This is the MAC address of the network boot interface. This should be same as the **mac-address** option we configure for compute node on the router.

fixed-address 192.168.1.20—This is the IP address to be given in the DHCP response.

next-server 192.168.1.3—This is the boot server IP address. Should be set to the IP address or the router connected to the network boot interface of compute node.

filename "pxelinux.0"—This is the boot filename.

- Related Documentation
- [compute-node on page 44](#)
 - *Using Junos OS Defaults Groups*

SNMP and the JunosV App Engine MIB

As of Junos OS Release 13.2R2, SNMP-based management and monitoring of the JunosV App Engine infrastructure is introduced using a new Juniper Networks proprietary MIB. This MIB is a collection of tables that gives users access to information such as an inventory of compute nodes and application instances, resource usage, and hardware.

System and network administrators are the intended users of this feature.

The new MIB is split into two parts: **jnx-jvae-infra-mib** and **jnx-jvae-node-mib**. **jnx-jvae-infra-mib** is implemented on the router and contains information about various nodes and virtual machines (VMs). **jnx-jvae-node-mib** is implemented on the compute nodes as a subagent of the Junos SNMP daemon (snmpd) on the router.

The subagent is added to vhostd on the compute node and registers with snmpd on the router to service SNMP queries for the MIB and generate traps for monitored events. The router responds to the queries with information on compute nodes and VMs that are passed on to the configured trap receiver. To receive and act on this information, at least one SNMP trap receiver must be configured.

CLI configuration statements are provided to specify thresholds for low storage (disk) space, high processing load, and low free memory conditions on the compute node. SNMP traps are raised when the values exceed or fall below these thresholds. These threshold statements are at the following CLI hierarchy levels:

- **[edit services app-engine]**
- **[edit services app-engine compute-cluster *compute-cluster-name*]**
- **[edit services app-engine compute-cluster *compute-cluster-name* compute-node *compute-node-name*]**

Threshold configuration statements at the **[edit services app-engine compute-cluster *compute-cluster-name* compute-node *compute-node-name*]** hierarchy level override those at the **[edit services app-engine compute-cluster *compute-cluster-name*]** and **[edit services app-engine]** hierarchy levels, and statements at the **[edit services app-engine compute-cluster *compute-cluster-name*]** hierarchy level override those at the **[edit services app-engine]** hierarchy level.

- Related Documentation
- [Configuring SNMP Traps for JunosV App Engine on page 31](#)
 - [subagent on page 56](#)
 - [app-engine on page 36](#)
 - [compute-cluster \(App Engine\) on page 41](#)
 - [compute-node on page 44](#)

Configuring SNMP Traps for JunosV App Engine

SNMP-based management and monitoring of the JunosV App Engine infrastructure is implemented using the Juniper Networks proprietary MIB `jnx-jvae-infra-mib`. CLI configuration statements are provided to specify thresholds for low storage (disk) space, high processing load, and low free memory conditions on the compute node. SNMP traps are raised when the values exceed or fall below these thresholds.

To enable this SNMP-based monitoring, two CLI configurations are required:

- You must define at least one system as the recipient of the SNMP traps (that is, configure the **trap-group** statement).
- You must enable the subagent to connect to router.

To create an SNMP trap group:

1. Create and name a group of SNMP traps.

```
set snmp trap-group group-name
```
2. For each trap group that you define, you must include the **targets** statement to define at least one system as the recipient of the SNMP traps in the trap group. Specify the IPv4 or IPv6 address of each recipient, not its hostname.

```
set snmp trap-group group-name targets address
```
3. Configure other optional statements as required to define the type of traps in the trap group and the behavior of the traps.

You also must configure the Agent-X TCP port so the subagent can connect to the router.

To enable the subagent to connect to router:

- Configure the **subagent** statement at the **[edit snmp subagent]** hierarchy level.

```
set snmp subagent tcp routing-instance default
```

This command allows connections over the default routing instance.

Related Documentation

- [SNMP and the JunosV App Engine MIB on page 30](#)
- [subagent on page 56](#)
- *trap-group*

Configuring the Syslog Server for Compute Nodes

The syslog server configured on the router running Junos OS is used to receive the log messages generated in compute nodes and virtual machines. Therefore, all the messages are logged at one place.

In a router running Junos OS, to enable syslog server in a default routing instance:

- Use the CLI set command:

```
[edit system]
user@host# set syslog server
```

To send the log messages to the router from the compute node or virtual machine (VM):

- In the `/etc/syslog.conf` file of the compute node or VM, manually add the following:

facility.priority@ip-address

where

- ***facility*** is a variable that can be one of several options listed in the man page of `syslog.conf`.
- ***priority*** is a variable that can be one of several options listed in the man page of `syslog.conf`.
- ***ip-address*** is the IP address of the management interface of the router.

Using this configuration, you can control the level of syslog messages visible in the router.

Here are some examples:

example 1:

To send critical kernel log messages to a router running Junos OS (with management interface 1.1.1.2), add the following entry to `syslog.conf`:

kern.crit @1.1.1.2

example 2:

To send all log messages (any facility and any priority) to a router running Junos OS, add the following entry to `syslog.conf`:

.* @1.1.1.2

VT-d Mode for Remote Applications

Virtualization solutions allow multiple operating systems and applications to run in independent partitions all on a single computer. Intel Virtualization Technology for Directed I/O (VT-d) extends Intel's Virtualization Technology (VT) by providing hardware support for isolating and restricting device accesses to the owner of the partition managing the device. With VT-d, virtual machines (VMs) access host resources directly through physical interfaces.

VT-d provides VM management software with the following capabilities:

- Device isolation using hardware-assisted remapping to improve reliability and security
- Direct assignment of devices to improve I/O performance and availability

Using the CLI, you can configure interfaces for VT-d mode. You need to configure both the platform and the VM for passthrough. See the **ethernet** and **interface** configuration statements for the syntax and configuration options.

Advantages of VT-d VT-d gives much better performance of the network bandwidth than simulating interfaces in a VM and attaching them through a bridge. VT-d can also improve security and reliability of the systems. The incoming and outgoing network packets land directly at the VM and completely avoid the host network stack.

Disadvantages of VT-d Using VT-d for an interface means that the network interface is not sharable. Therefore, the number of network interfaces is limited to the number of physical interfaces in the VSE device.

Limitations of VT-d VT-d can only be supported on a real physical interface and not on any other software interfaces, like bridges or aggregates. Additionally, once the host interface is reserved for VT-d mode of operation, it cannot be used for another purpose; for example, it cannot be used to attach to a bridge interface or an aggregated interface. Once the interface mode is reserved in VT-d, the interface is removed from the host system for any use until the VT-d mode is disabled.

Related Documentation

- [ethernet on page 46](#)
- [interface on page 52](#)

CHAPTER 2

JunosV App Engine Configuration Statements

This section is a reference for each of the configuration statements you use to configure the JunosV App Engine. The configuration statements are organized alphabetically. For each configuration statement, there are options listed or links to child statements from the syntax section of the configuration statement.

- [app-engine on page 36](#)
- [bridge on page 39](#)
- [compute-cluster \(App Engine\) on page 41](#)
- [compute-cluster \(VM instance\) on page 43](#)
- [compute-node on page 44](#)
- [ethernet on page 46](#)
- [family \(Compute Cluster\) on page 48](#)
- [family \(Virtual Machines\) on page 49](#)
- [instance on page 50](#)
- [interface on page 52](#)
- [interfaces on page 53](#)
- [local-management on page 54](#)
- [secondary-disk on page 55](#)
- [subagent on page 56](#)
- [syslog on page 57](#)
- [virtual-machines on page 59](#)

app-engine

```
Syntax  app-engine {
    compute-cluster compute-cluster-name {
        compute-node compute-node-name {
            interfaces {
                bridge bridge-name {
                    family inet {
                        (address ip-address | dhcp);
                    }
                    interface bridge-interface-list;
                    management;
                }
                ethernet ethernet-interface-name {
                    enable-passthrough;
                    family inet {
                        (address ip-address | dhcp);
                    }
                    management;
                }
            }
            mac-address mac-address;
            monitor-cpu normal threshold;
            monitor-memory normal threshold;
            monitor-storage normal threshold;
            syslog {
                facility level;
            }
        }
        local-management {
            family inet {
                address ip-address;
            }
        }
        monitor-cpu normal threshold;
        monitor-memory normal threshold;
        monitor-storage normal threshold;
    }
    monitor-cpu normal threshold;
    monitor-memory normal threshold;
    monitor-storage normal threshold;
    virtual-machines {
        instance instance-name {
            compute-cluster compute-cluster-name {
                compute-node compute-node-name;
            }
            cpu cpu-number;
            interface interface-name {
                bridge bridge-name;
                family inet {
                    [ address ip-address ];
                }
            }
            host-interface interface-name {
                enable-passthrough;
            }
        }
    }
}
```

```

    }
    hw-model model-name;
  }
  local-management {
    family (Compute Cluster) inet {
      address ip-address;
    }
  }
  memory gigabytes;
  package package-name;
  secondary-disk (hdb | hdc | hdd) disk-size gigabytes;
}
}
}

```

Hierarchy Level [edit services]

Release Information Statement introduced in Junos OS Release 12.3.
 Option **monitor-cpu normal threshold** introduced in Junos OS Release 13.2R2.
 Option **monitor-memory normal threshold** introduced in Junos OS Release 13.2R2.
 Option **monitor-storage normal threshold** introduced in Junos OS Release 13.2R2.

Description Configure the App Engine. The [edit services app-engine] hierarchy level has two components to configure: the physical connection between the device running Junos and the hardware appliance (which is under the [edit services app-engine compute-cluster] hierarchy level) and the virtual machines, where the applications run, each on its own guest OS (the [edit services app-engine virtual-machines] hierarchy level).

SNMP traps can be set at this hierarchy level or at the [edit services app-engine compute-cluster *compute-cluster-name* compute-node *compute-node-name*] or [edit services app-engine compute-cluster *compute-cluster-name*] hierarchy levels. Settings at the [edit services app-engine compute-cluster *compute-cluster-name*] hierarchy level override those at this level and settings at the [edit services app-engine compute-cluster *compute-cluster-name* compute-node *compute-node-name*] hierarchy level override those at any other level.

Options **monitor-cpu *normal threshold***—Normal and maximum processing loads as percentages of the CPU. If the overall system CPU usage or per-CPU usage exceeds the *threshold* percentage, the system raises the **jnxVECNProcessingLoadHigh** or **jnxVECNProcessorLoadHigh** trap. When the system detects a **jnxVECNProcessingLoadHigh** or **jnxVECNProcessorLoadHigh** condition and then drops to or below the *normal* percentage, the **jnxVECNProcessingLoadOk** or **jnxVECNProcessorLoadOk** alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-memory *normal threshold*—Normal and maximum RAM usages as percentages. If the RAM usage rises above the *threshold* percentage, the system raises the **jnxVECNMemoryLow** trap. When the RAM usage rises above the maximum and then drops below the *normal* percentage, the **jnxVECNMemoryOk** alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-storage *normal threshold*—Normal and maximum storage usages as percentages. If the space usage under volume LFS (/volume/lfs) or volume SFS (/volume/sfs) rises above the *threshold* percentage, the system raises the **jnxVECNStorageLow** trap. When the storage usage recovers and drops below the *normal* percentage, the **jnxVECNStorageOk** alarm is triggered.

Default: normal default is 80; threshold default is 90

Required Privilege Level admin—To view this statement in the configuration.
 admin-control—To add this statement to the configuration.

Related Documentation • *extensions*
 • [virtual-machines on page 59](#)

bridge

Syntax	<pre>bridge <i>bridge-name</i> { family inet { (address <i>ip-address</i> dhcp); } interface <i>bridge-interface-list</i>; management; }</pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i> interfaces]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	<p>Create a bridge interface on a compute node. A bridge interface is used to connect virtual machine interfaces to a network. If the bridge-interface-list option is specified, then the virtual machine interface attached to this bridge is connected to the physical network in the bridge-interface list.</p> <p>Both bridge and Ethernet interfaces are referred to as <i>network interfaces</i>. If only one network interface is configured, it is the management network interface by default. If more than one network interfaces are configured, you must specify one, and only one, as the management network interface.</p> <p>Configuration changes to network interfaces are nondestructive, that is, such changes do not require a reboot of the compute node. The exception to this is if it is a management interface. Changes to management interfaces are destructive and require a reboot of the compute node.</p>
Options	<p>bridge <i>bridge-name</i>—Name of the bridge interface.</p> <p>interface <i>bridge-interface-list</i>—A list of interfaces on the host OS. All the interfaces in the bridge-interface list are in the same network. There can be a maximum of four interfaces in a bridge-interface list.</p> <p>For management interfaces, changing this option is destructive.</p> <p>management—(Optional) Configures the specified interface as the management interface of the compute node. There can be only one management interface in a compute node. If there is only one network interface, it is by default the management interface. Configuration changes made to a management interface require a reboot of the compute node.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • interface on page 52 • ethernet on page 46

- [interfaces on page 53](#)

compute-cluster (App Engine)

```
Syntax  compute-cluster compute-cluster-name {
        compute-node compute-node-name {
            interfaces {
                bridge bridge-name {
                    family inet {
                        (address ip-address | dhcp);
                    }
                    interface bridge-interface-list;
                    management;
                }
                ethernet ethernet-interface-name {
                    enable-passthrough;
                    family inet {
                        (address ip-address | dhcp);
                    }
                    management;
                }
            }
            mac-address mac-address;
            monitor-cpu normal threshold;
            monitor-memory normal threshold;
            monitor-storage normal threshold;
            syslog {
                facility level;
            }
        }
        local-management {
            family inet {
                address ip-address;
            }
        }
        monitor-cpu normal threshold;
        monitor-memory normal threshold;
        monitor-storage normal threshold;
    }
```

Hierarchy Level [edit services app-engine]

Release Information Statement introduced in Junos OS Release 12.3.
 Option **monitor-cpu *normal threshold*** introduced in Junos OS Release 13.2R2.
 Option **monitor-memory *normal threshold*** introduced in Junos OS Release 13.2R2.
 Option **monitor-storage *normal threshold*** introduced in Junos OS Release 13.2R2.

Description Configure a compute cluster.

The values for **monitor-cpu**, **monitor-memory**, and **monitor-storage** can be set at any of these hierarchy levels:

- [edit services app-engine]
- [edit services app-engine compute-cluster *compute-cluster-name*]

- `[edit services app-engine compute-cluster compute-cluster-name compute-node compute-node-name]`

Settings at the `[edit services app-engine compute-cluster compute-cluster-name]` hierarchy level override those at the `[edit services app-engine]` hierarchy level and settings at the `[edit services app-engine compute-cluster compute-cluster-name compute-node compute-node-name]` hierarchy level override those at any other level.

Options *compute-cluster-name*—Name of the compute cluster. You can configure multiple compute clusters.

monitor-cpu normal threshold—Normal and maximum processing loads as percentages of the CPU. If the overall system CPU usage or per-CPU usage exceeds the *threshold* percentage, the system raises the `jnxVECNProcessingLoadHigh` or `jnxVECNProcessorLoadHigh` trap. When the system detects a `jnxVECNProcessingLoadHigh` or `jnxVECNProcessorLoadHigh` condition and then drops to or below the *normal* percentage, the `jnxVECNProcessingLoadOk` or `jnxVECNProcessorLoadOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-memory normal threshold—Normal and maximum RAM usages as percentages. If the RAM usage rises above the *threshold* percentage, the system raises the `jnxVECNMemoryLow` trap. When the RAM usage rises above the maximum and then drops below the *normal* percentage, the `jnxVECNMemoryOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-storage normal threshold—Normal and maximum storage usages as percentages. If the space usage under volume LFS (`/volume/lfs`) or volume SFS (`/volume/sfs`) rises above the *threshold* percentage, the system raises the `jnxVECNStorageLow` trap. When the storage usage recovers and drops below the *normal* percentage, the `jnxVECNStorageOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

The remaining statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
 admin-control—To add this statement to the configuration.

Related Documentation • [app-engine on page 36](#)

compute-cluster (VM instance)

Syntax	<code>compute-cluster <i>compute-cluster-name</i> { compute-node <i>compute-node-name</i>; }</code>
Hierarchy Level	[edit services app-engine virtual-machines instance <i>instance-name</i>]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	Specify the compute cluster and compute node on which the virtual machine runs.
Options	<p><i>compute-cluster-name</i>—Name of the compute cluster that contains the compute node on which the VM is deployed.</p> <p><i>compute-node-name</i>—Name of the compute node on which the VM is deployed.</p>
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• instance on page 50

compute-node

```
Syntax  compute-node compute-node-name {
        interfaces {
            bridge bridge-name {
                family inet {
                    (address ip-address | dhcp);
                }
                interface bridge-interface-list;
                management;
            }
            ethernet ethernet-interface-name {
                enable-passthrough;
                family inet {
                    (address ip-address | dhcp);
                }
                management;
            }
        }
        mac-address mac-address;
        monitor-cpu normal threshold;
        monitor-memory normal threshold;
        monitor-storage normal threshold;
        syslog {
            facility level;
        }
    }
```

Hierarchy Level [edit services app-engine compute-cluster *compute-cluster-name*]

Release Information Statement introduced in Junos OS Release 12.3.
 Option **monitor-cpu *normal threshold*** introduced in Junos OS Release 13.2R2.
 Option **monitor-memory *normal threshold*** introduced in Junos OS Release 13.2R2.
 Option **monitor-storage *normal threshold*** introduced in Junos OS Release 13.2R2.

Description Create a compute node. A compute node is the interface with the VSE device that runs the host operating system (host OS). A compute node is a member of a compute cluster. Each compute node requires that you configure network interfaces and the media access control (MAC) address of the compute node that you want to create. Configuring the **syslog** statement is optional.

When you commit the compute node configuration, the compute node is booted. Compute nodes bootstrap using network boot, or PXE. For instructions on how to enable PXE and complete setting up the compute node for applications, see [“Configuring the DHCP Server and TFTP Server for Compute Nodes” on page 23](#).

The values for **monitor-cpu**, **monitor-memory**, and **monitor-storage** can be set at any of these hierarchy levels:

- [edit services app-engine]
- [edit services app-engine compute-cluster *compute-cluster-name*]

- `[edit services app-engine compute-cluster compute-cluster-name compute-node compute-node-name]`

Settings at the `[edit services app-engine compute-cluster compute-cluster-name]` hierarchy level override those at the `[edit services app-engine]` hierarchy level and settings at the `[edit services app-engine compute-cluster compute-cluster-name compute-node compute-node-name]` hierarchy level override those at any other level.

Options *compute-node-name*—Name of the compute node. When you commit the compute-node name, the compute node is booted with that name as the operational node name. Any time you change the compute-node name in the CLI, the compute node needs to be rebooted.

mac-address—MAC address of the network boot interface of the compute node. The MAC address is essential so that the compute node can PXE-boot, or boot through the network interface, rather than from the local hard disk. The compute node identifies itself to the router with the network boot interface mac-address.

monitor-cpu normal threshold—Normal and maximum processing loads as percentages of the CPU. If the overall system CPU usage or per-CPU usage exceeds the *threshold* percentage, the system raises the `jnxVECNProcessingLoadHigh` or `jnxVECNProcessorLoadHigh` trap. When the system detects a `jnxVECNProcessingLoadHigh` or `jnxVECNProcessorLoadHigh` condition and then drops to or below the *normal* percentage, the `jnxVECNProcessingLoadOk` or `jnxVECNProcessorLoadOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-memory normal threshold—Normal and maximum RAM usages as percentages. If the RAM usage rises above the *threshold* percentage, the system raises the `jnxVECNMemoryLow` trap. When the RAM usage rises above the maximum and then drops below the *normal* percentage, the `jnxVECNMemoryOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

monitor-storage normal threshold—Normal and maximum storage usages as percentages. If the space usage under volume LFS (/volume/lfs) or volume SFS (/volume/sfs) rises above the *threshold* percentage, the system raises the `jnxVECNStorageLow` trap. When the storage usage recovers and drops below the *normal* percentage, the `jnxVECNStorageOk` alarm is triggered.

Default: normal default is 80; threshold default is 90

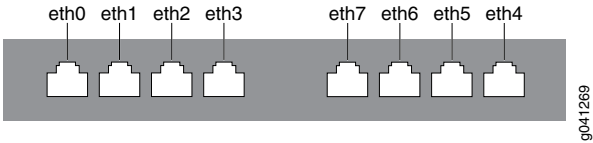
The remaining statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- [compute-cluster \(App Engine\) on page 41](#)
- [Configuring the DHCP Server and TFTP Server for Compute Nodes on page 23](#)

ethernet

Syntax	<pre>ethernet <i>ethernet-interface-name</i> { enable-passthrough; family inet { (address <i>ip-address</i> dhcp); } management; }</pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i> interfaces]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	<p>Create an Ethernet interface on a compute node. An Ethernet interface connects one interface on the VSE device to the device running Junos OS.</p> <p>On the VSE device, there are two sets of four ports for the VSE 1100 device and one set of four ports and one set of two ports for the VSE 2100 device. The set of four ports on the left side when you are facing the VSE device are <i>onboard ports</i> and the ports on the left are data or <i>IOC ports</i>. The onboard ports are interfaces named eth0 to eth3 going from leftmost to right as you face the device. The IOC ports are interfaces named eth4 to eth7 going from the rightmost to the leftmost. See Figure 2 on page 46, which shows that interfaces eth3 and eth7 are next to each other in the middle of the device face (on a VSE 1100 device).</p>
<p>Figure 2: Interface Mapping to Ports on a VSE 1100 Device</p> 	
<p>Configuration changes to network interfaces are nondestructive, that is, such changes do not require a reboot of the compute node. The exception to this is if it is a management interface. Changes to management interfaces are destructive and require a reboot of the compute node.</p>	
Options	<p>enable-passthrough—Required for VT-d mode. Configure the Ethernet interface in VT-d mode.</p> <p>If the Ethernet interface is configured for VT-d mode, the following constraints for the ethernet statement apply:</p> <ul style="list-style-type: none"> • The management option must not be configured. • The interface cannot be part of a bridge interface. • The family and address options must not be configured. <p>Default: Not configured</p>

ethernet *ethernet-interface-name*—Name of the Ethernet interface. The Ethernet interface name must be of the form eth#, where # can be any valid network interface number (for example, from eth0 to eth7 for a VSE 1100 device).

management—(Optional) Specifies the interface as the management interface of the compute node. There can be only one management interface per compute node. If there are more than one network interfaces, specify **management** for one of the network interfaces. Configuration changes to a management interface require a reboot of the compute node.

The remaining statements are explained separately.

Required Privilege	admin—To view this statement in the configuration.
Level	admin-control—To add this statement to the configuration.

Related Documentation	<ul style="list-style-type: none">• bridge on page 39• interfaces on page 53• VT-d Mode for Remote Applications on page 32
------------------------------	--

family (Compute Cluster)

Syntax	<pre>family inet { (address <i>ip-address</i> dhcp); }</pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i> interfaces bridge <i>bridge-name</i>], [edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i> interfaces ethernet <i>ethernet-interface-name</i>],
Release Information	Statement introduced in Junos OS Release 12.3. dhcp option introduced in Junos OS Release 13.2R1.
Description	<p>Configure a static IP address or enable DHCP to get the IP addresses for management or data ports.</p> <p>For a management interface, changing the family options in the platform configuration is destructive and requires a reboot of the compute node.</p>
Options	<p>address <i>ip-address</i>—Static IPv4 address (with prefix) of the network interface being configured. This option and the dhcp option are mutually exclusive.</p> <p>dhcp—Enable the VSE devices's management port and data ports IPv4 addresses to be assigned by DHCP. You can do one of the following:</p> <ol style="list-style-type: none">1. Enable DHCP for data ports.2. Enable DHCP for the management port.3. Enable DHCP for both the data and management ports. <p>This option and the address option are mutually exclusive.</p> <p>family inet—Only the IPv4 protocol is supported.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• bridge on page 39• ethernet on page 46

family (Virtual Machines)

Syntax	<pre>family inet { [address <i>ip-address</i>]; }</pre>
Hierarchy Level	[edit services app-engine virtual-machines instance <i>instance-name</i> interface <i>interface-name</i>]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	Configure a static IP address for the interface.
Options	<p>address <i>ip-address</i>—Static IPv4 address (with prefix) of the network interface being configured. There can be multiple IP addresses set.</p> <p>family inet—Only the IPv4 protocol is supported.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• interface on page 52

instance

Syntax `instance instance-name {
 compute-cluster compute-cluster-name {
 compute-node compute-node-name;
 }
 cpu cpu-number;
 interface interface-name {
 bridge bridge-name;
 family inet {
 [address ip-address];
 }
 host-interface interface-name {
 enable-passthrough;
 }
 hw-model model-name;
 }
 local-management {
 family inet {
 address ip-address;
 }
 }
 memory gigabytes;
 package package-name;
 secondary-disk (hda | hdb | hdc | hdd) size gigabytes;
 }`

Hierarchy Level [edit services app-engine virtual-machines]

Release Information Statement introduced in Junos OS Release 12.3.

Description Describe the basic attributes a virtual machine (VM) needs as a VM instance. These attributes are called *virtual machine descriptors*. An instance description is used to deploy and instantiate a virtual machine. The description is configured using the Junos CLI. The user specifies on which compute node a VM will run and what resources are allocated to it.

The virtual machine management (VMM) infrastructure allows for pre-provisioning. You can configure a VM instance without the compute nodes being physically present in the network. When the nodes are available, the virtual machines are instantiated based on policy and configuration. Any change in VM configuration will result in restart of the VM before the changed configuration takes effect.

Options `instance instance-name`—Name of the VM instance.

`cpu cpu-number`—(Optional) Number of CPUs assigned to the VM.

Default: 1

Range: 1 to 16; do not configure more than 8 CPUs for a VSE 1100 tethered device

`memory gigabytes`—(Optional) Memory for the VM, in gigabytes.

Default: 1

Range: 1 to 32; do not configure more than 16 GB for a VSE 1100 tethered device

package *package-name*—Name of the VM package.

The remaining statements are explained separately.

Required Privilege	admin—To view this statement in the configuration.
Level	admin-control—To add this statement to the configuration.

Related Documentation	<ul style="list-style-type: none">• virtual-machines on page 59
------------------------------	---

interface

Syntax	<pre>interface <i>interface-name</i> { bridge <i>bridge-name</i>; family inet { [address <i>ip-address</i>]; } host-interface <i>interface-name</i> { enable-passthrough; } hw-model <i>model-name</i>; }</pre>
Hierarchy Level	[edit services app-engine virtual-machines instance <i>instance-name</i>]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	Configure a virtual machine (VM) interface. Every interface in a VM is associated with either a bridge interface or a host interface. The host-interface configuration statement is configured only in the case that VT-d mode is enabled.
Options	<p>bridge <i>bridge-name</i>—Required unless VT-d is enabled on the VM. Assign the VM interface to a bridge using the name of one of the bridge interfaces created on the compute node. The bridge interface is used to connect the VM interface with the physical interface on the compute node. If the bridge-interface list on the bridge interface has more than one physical interface listed, the VM interface is linked to that network of interfaces.</p> <p>enable-passthrough—Required for VT-d mode. Enable VT-d mode. Default: Not configured</p> <p>host-interface <i>interface-name</i>—Required for VT-d mode. Name of the host interface that has VT-d mode enabled.</p> <p>hw-model <i>model-name</i>—(Optional) Interface hardware model.</p> <p>interface <i>interface-name</i>—Name of the VM interface. VM interfaces are named eth# up to a maximum of eight interfaces (as in eth0, eth1, and so on up to eth7) per VM instance. These interfaces must be configured sequentially, that is, eth0 must be configured before eth1 and so on.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• instance on page 50

interfaces

Syntax	<pre> interfaces { bridge <i>bridge-name</i> { family inet { (address <i>ip-address</i> dhcp); } interface <i>bridge-interface-list</i>; management; } ethernet <i>ethernet-interface-name</i> { enable-passthrough; family inet { (address <i>ip-address</i> dhcp); } management; } } </pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	<p>Create a network interface for a compute node. Both bridge and Ethernet interfaces are referred to as <i>network interfaces</i>. Configuration changes to network interfaces are nondestructive, that is, such changes do not require a reboot of the compute node. The exception to this is if it is a management interface. Changes to management interfaces are destructive and require a reboot of the compute node.</p> <p>The statements are explained separately.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • compute-node on page 44


local-management

Syntax	<pre>local-management { family inet { address <i>ip-address</i>; } }</pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i>], [edit services app-engine virtual-machines instance <i>instance-name</i>]
Release Information	Statement introduced in Junos OS Release 12.3.
Description	<p>Specify the management address of the device running Junos to which the compute cluster is connected. This is the Routing Engine counterpart to the management interface address on the compute node.</p> <p>To configure local-management, configure the IPv4 address (without a prefix) of the interface on the Junos device that is connected to the management interface of all the compute nodes in the compute cluster. <i>Management interface</i> means the network interface (either bridge or ethernet) on the compute node that has the management option configured.</p> <p>Configuration changes to the local-management option are nondestructive, that is, such changes do not require a reboot of the compute node.</p> <p>If local-management is not configured under [edit services app-engine virtual-machines instance <i>instance-name</i>] hierarchy level, the local management interface defaults to the compute cluster's local management interface under which that VM is provisioned.</p>
Options	<p>family inet—Only the IPv4 protocol is supported.</p> <p>address <i>ip-address</i>—The IPv4 address without prefix of the network interface being configured.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	

secondary-disk

Syntax	<code>secondary-disk (hdb hdc hdd) size <i>gigabytes</i>;</code>
Hierarchy Level	<code>[edit services app-engine virtual-machines instance <i>instance-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 12.3.
Description	<p>Create storage pools. You can configure multiple secondary disks to the virtual machine (VM). Any data written to these disks are persistent until the VM destroyed (when the VM is deleted or the VM's storage disk deconfigured) at which time the storage space allocated to the VM is automatically deallocated.</p> <p>There are two storage pools that get configured on the system: the external pool and the internal pool. The hardware device is shipped with four external slots (no external disks) and two internal disks. The external pool configuration is not visible until you add a pair of mirrored disks. At least two disks (in slots 0 and 1) are needed to initialize the external pool. The external pool can consist of up to two pairs of mirrored disks. Each mirrored pair is called a <i>volume</i>. Slots 0 and 1 are paired to form one volume, and slots 2 and 3 are grouped formed another second volume. The plug order of the disks is 0, 1, 2, 3. At least two disks (in slots 0 and 1) are needed to initialize the external pool. A mirrored volume gets configured only when both the drives are added.</p> <p>The internal pool volume group has one pair of disks in RAID-1 configuration. These disks are shipped with the system, so the internal pool should always be present in the system. The system first tries to allocate a raw disk from the external pool if it can be allocated there. If the external pool is not present or does not have enough space for the raw disk, the system tries to allocate from the internal pool. If it is not possible to allocate the disk from either pool, the VM is left in the inactive state.</p> <p>You can see information about the secondary storage by issuing the show app-engine resource-usage operational command.</p>
Options	<p>secondary-disk (hdb hdc hdd)—Name of optional secondary disk. You must specify size in order to configure a secondary disk.</p> <p>Values: hdb, hdc, or hdd</p> <p>size <i>gigabytes</i>—Size of disk in gigabytes. This option is required to configure a secondary disk.</p> <p>Range: 1 to 160</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • instance on page 50 • show app-engine resource-usage on page 78

subagent

Syntax	<pre>subagent { tcp { routing-instance { default; } } }</pre>
Hierarchy Level	[edit snmp]
Release Information	Statement introduced in Junos OS Release 13.2R2.
Description	Configure an SNMP subagent. The JunosV App Engine MIB is implemented in the compute node as a subagent of snmpd on the router. To enable the subagent to connect to the router, you must configure the subagent statement at the [edit snmp subagent] hierarchy level.
<div> NOTE: To enable the SNMP-based monitoring, you must define at least one system as the recipient of the SNMP traps. You must also configure the trap-group statement at the [edit snmp] hierarchy level.</div>	
Options	<p>tcp—Enable SNMP subagent TCP connection.</p> <p>routing-instance—Specify routing-instance name for TCP connection</p> <p>default—Allow connections over default routing instance.</p>
Required Privilege Level	<p>snmp—To view this statement in the configuration.</p> <p>snmp-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring SNMP Traps for JunosV App Engine on page 31• <i>trap-group</i>• SNMP and the JunosV App Engine MIB on page 30

syslog

Syntax	<pre>syslog { facility level; }</pre>
Hierarchy Level	[edit services app-engine compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>]
Description	<p>Configure system logging facility.</p> <p>Configuration changes to syslog are nondestructive, that is, such changes do not require a reboot of the compute node. The exception to this is if it is a management interface.</p>
Options	<p>facility—Group of messages that are either generated by the same software process or concern a similar condition or activity. The following are possible values for facility:</p> <ul style="list-style-type: none"> any—All facilities. authorization—Authorization system. cron—cron daemon. daemon—Various system processes. kernel—Kernel. privileged—Privileged authorization events. syslog—Syslog messages. user—User processes. uucp—UUCP system. local0—Local 0 messages. local1—Local 1 messages. local2—Local 2 messages. local3—Local 3 messages. local4—Local 4 messages. local5—Local 5 messages. local6—Local 6 messages. local7—Local 7 messages. <p>severity—Classification of effect on functioning. The following are possible values for severity:</p> <ul style="list-style-type: none"> • any—Include all severity levels.

- **emergency**—System panic or other condition that causes the routing platform to stop functioning.
- **alert**—Conditions that require immediate correction, such as a corrupted system database.
- **critical**—Critical conditions, such as hard errors. Hard errors are permanent, unrecoverable errors, such as a disk read error
- **error**—Error conditions that generally have less serious consequences than errors in the **emergency**, **alert**, and **critical** levels.
- **warning**—Conditions that warrant monitoring.
- **notice**—Conditions that are not errors but might warrant special handling.
- **info**—Events or nonerror conditions of interest.
- **debug**—Debug messages.

Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
---------------------------------	---

Related Documentation	• compute-node on page 44
------------------------------	---

virtual-machines

```
Syntax  virtual-machines {
        instance instance-name {
            compute-cluster compute-cluster-name {
                compute-node compute-node-name;
            }
            cpu cpu-number;
            interface interface-name {
                bridge bridge-name;
                family inet {
                    [ address ip-address ];
                }
                host-interface interface-name {
                    enable-passthrough;
                }
                hw-model model-name;
            }
            local-management {
                family inet {
                    address ip-address;
                }
            }
            memory gigabytes;
            package package-name;
            secondary-disk (hdb | hdc | hdd) size gigabytes;
        }
    }
```

Hierarchy Level [edit services app-engine]

Release Information Statement introduced in Junos OS Release 12.3.

Description Manage the lifecycle of the virtual machines and the resource allocation on compute nodes. Configuration changes to **[edit services app-engine virtual-machines]** hierarchy level are destructive, that is, CLI changes at this hierarchy level. will cause the VM to reboot.

The statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- [app-engine on page 36](#)

CHAPTER 3

JunosV App Engine Operational Commands

Table 4 on page 61 summarizes the command-line interface (CLI) operational commands for JunosV App Engine. The commands are listed in alphabetical order.

Table 4: app-engine Operational Mode Commands

Task	Command
Take a compute node offline.	<code>request app-engine offline compute-cluster compute-node</code>
Restart compute node.	<code>request app-engine reboot compute-cluster compute-node</code>
Restart a virtual machine.	<code>request app-engine reboot compute-cluster compute-node virtual-machine-instance</code>
Request a start, stop, or restart of a system service.	<code>request app-engine service (start stop restart) compute-cluster compute-node</code>
Show the basic information of a compute node.	<code>show app-engine info</code>
Show virtual engine appliances processes.	<code>show app-engine processes compute-cluster compute-node</code>
Show the resource usage information of the compute node.	<code>show app-engine resource-usage</code>
Show the status of a compute node.	<code>show app-engine status</code>
Show the download status of a VM package.	<code>show app-engine virtual-machine package</code>
Show the status of a virtual machine (VM) instance.	<code>show app-engine virtual-machine instance</code>

- `request app-engine offline compute-cluster compute-node`
- `request app-engine reboot compute-cluster compute-node`

- `request app-engine reboot compute-cluster compute-node virtual-machine-instance`
- `request app-engine service (start | stop | restart) compute-cluster compute-node`
- `show app-engine info`
- `show app-engine processes compute-cluster compute-node`
- `show app-engine resource-usage`
- `show app-engine status`
- `show app-engine virtual-machine package`
- `show app-engine virtual-machine instance`

request app-engine offline compute-cluster compute-node

Syntax	<code>request app-engine offline compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i></code>
Release Information	Command introduced in Junos OS Release 12.3.
Description	Take a compute node offline.
Options	<p><i>compute-cluster-name</i>—Name of compute cluster.</p> <p><i>compute-node-name</i>—Name of compute node.</p>
Additional Information	In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, , if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine processes compute-cluster compute-node on page 74 • request app-engine service (start stop restart) compute-cluster compute-node on page 69 • request app-engine reboot compute-cluster compute-node on page 65
List of Sample Output	request app-engine offline (request successful) on page 64 request app-engine offline (compute node offline) on page 64 request app-engine offline (request failed) on page 64
Output Fields	For a description of the output fields, see Table 5 on page 63 . Output fields are listed in the approximate order in which they appear.

Table 5: request app-engine-offline compute-cluster compute-node Output Fields

Field Name	Field Description
Compute cluster	Name of the compute cluster.
Compute node	Name of the compute node.
Status	Status of the request. See the sample output for the possible values of this field.

Sample Output

request app-engine offline (request successful)

```
user@host> request app-engine offline compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               Successful
```

request app-engine offline (compute node offline)

```
user@host> request app-engine offline compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Offline ---
```

request app-engine offline (request failed)

```
user@host> request app-engine offline compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Error executing command ---
```


request app-engine reboot compute-cluster compute-node

Syntax	<code>request app-engine reboot compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i></code>
Release Information	Command introduced in Junos OS Release 12.3.
Description	Restart the specified compute node. When the compute node is updated, the compute node needs to be restarted for the configuration to take effect.
Options	<p><code>compute-cluster <i>compute-cluster-name</i></code>—Request reboot of the specified compute cluster.</p> <p><code>compute-node <i>compute-node-name</i></code>—Request reboot of the specified compute node in the specified compute cluster.</p>
Additional Information	In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, , if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine status on page 85
List of Sample Output	<p>request app-engine reboot compute-cluster compute-node (request successful) on page 66</p> <p>request app-engine reboot compute-cluster compute-node (compute node offline) on page 66</p> <p>request app-engine reboot compute-cluster compute-node (request failed) on page 66</p>
Output Fields	For a description of the output fields, see Table 6 on page 65 . Output fields are listed in the approximate order in which they appear.

Table 6: request app-engine-reboot compute-cluster compute-node Output Fields

Field Name	Field Description
Compute cluster	Name of the compute cluster.
Compute node	Name of the compute node.
Status	Status of the request. See the sample output for the possible values of this field.

Sample Output

request app-engine reboot compute-cluster compute-node (request successful)

```
user@host> request app-engine reboot compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               Successful
```

request app-engine reboot compute-cluster compute-node (compute node offline)

```
user@host> request app-engine reboot compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Offline ---
```

request app-engine reboot compute-cluster compute-node (request failed)

```
user@host> request app-engine reboot compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Error executing command ---
```

request app-engine reboot compute-cluster compute-node virtual-machine-instance

Syntax	<code>request app-engine reboot compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i> virtual-machine-instance <i>instance-name</i></code>
Release Information	Command introduced in Junos OS Release 12.3.
Description	Restart a virtual machine.
Options	<p>compute-cluster <i>compute-cluster-name</i>—Request reboot of the specified compute cluster.</p> <p>compute-node <i>compute-node-name</i>—Request reboot of the specified compute node in the specified compute cluster.</p> <p>virtual-machine-instance <i>instance-name</i>—Request reboot of the specified VM instance in the specified compute node.</p>
Additional Information	In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine virtual-machine instance on page 90
List of Sample Output	request app-engine reboot compute-cluster compute-node virtual-machine-instance on page 68
Output Fields	For a description of the output fields, see Table 7 on page 67 . Output fields are listed in the approximate order in which they appear.

Table 7: request app-engine reboot compute-cluster compute-node virtual-machine-instance

Field Name	Field Description
Virtual machine	Name of the virtual machine.
Status	Status of the request. See the sample output for the possible values of this field.

Sample Output

`request app-engine reboot compute-cluster compute-node virtual-machine-instance`

```
user@host> request app-engine reboot compute-cluster cluster1 compute-node cn2
virtual-machine-instance vm1
  Virtual machine      Status
  vm1                  Successful
```

request app-engine service (start | stop | restart) compute-cluster compute-node

Syntax	<code>request app-engine service (start stop restart) <i>service-name</i> compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i></code>
Release Information	Command introduced in Junos OS Release 12.3.
Description	Start, stop, or restart a system service.
Options	<p><i>service-name</i>—Name of the service to be acted upon.</p> <p>The possible services are:</p> <ul style="list-style-type: none"> • <code>vm-manager</code>—VM manager service • <code>syslog</code>—Syslog service • <code>sshd</code>—SSH service <p><i>compute-cluster-name</i> —Name of the compute cluster.</p> <p><i>compute-node-name</i>—Name of the compute node.</p>
Additional Information	In the operational mode of the CLI when you type <code>?</code> for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine status on page 85 • show app-engine info on page 71
List of Sample Output	request app-engine service restart compute-cluster compute-node (request successful) on page 70 request app-engine service restart compute-cluster compute-node (compute node offline) on page 70 request app-engine service restart compute-cluster compute-node (request failed) on page 70
Output Fields	For a description of the output fields, see Table 8 on page 70 . Output fields are listed in the approximate order in which they appear.

Table 8: request app-engine-(start | stop | restart) compute-cluster compute-node Output Fields

Field Name	Field Description
Compute cluster	Name of the compute cluster.
Compute node	Name of the compute node.
Status	Status of the request. See the sample output or the possible values of this field.

Sample Output

request app-engine service restart compute-cluster compute-node (request successful)

```

user@host> request app-engine service restart syslog compute-cluster compute-cluster-name
compute-node cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               Successful

```

request app-engine service restart compute-cluster compute-node (compute node offline)

```

user@host> request app-engine service restart syslog compute-cluster cluster1 compute-node
cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Offline ---

```

request app-engine service restart compute-cluster compute-node (request failed)

```

user@host> request app-engine service restart syslog compute-cluster cluster1 compute-node
cn2
Compute cluster: cluster1
  Compute node      Status
  cn2               --- Error executing command ---

```

show app-engine info

Syntax	<pre>show app-engine info <compute-cluster <i>compute-cluster-name</i>> <compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>></pre>
Release Information	<p>Command introduced Junos OS Release 12.3.</p> <p>Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.</p>
Description	Show the basic information of a compute node.
Options	<p><i>compute-cluster-name</i>—(Optional) Name of the compute cluster.</p> <p><i>compute-node-name</i>—(Optional) Name of the compute node. Specifying a compute-node alone will not return any result. Always specify the compute-cluster with a compute node.</p>
Additional Information	<p>In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, , if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.</p> <p>For commands with an optional compute-cluster <i>compute-cluster-name</i> option, if that option is omitted, the command will be executed on all compute nodes of all compute clusters. For commands with an optional compute-node <i>compute-node-name</i> option, if that option is omitted, the command will be executed on all compute nodes of the specified compute cluster.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine status on page 85 • show app-engine resource-usage on page 78 • show app-engine processes compute-cluster compute-node on page 74
List of Sample Output	<p>show app-engine info on page 72</p> <p>show app-engine info compute-cluster compute-node on page 72</p> <p>show app-engine info (Command failed) on page 72</p> <p>show app-engine info (QFX5100 Switch) on page 73</p>
Output Fields	For a description of the output fields, see Table 9 on page 72 . Output fields are listed in the approximate order in which they appear.

Table 9: show app-engine info Output Fields

Field Name	Field Description
Compute cluster	Name of the compute cluster.
Compute node	Name of the compute node.
Model	Model name of the compute node.
Kernel release	Kernel release of the Linux which the compute node is running.
Machine	The machine architecture of the compute node.
Management IP	The management IP address of the compute node.

Sample Output

show app-engine info

In the following example, notice there are multiple compute clusters and multiple compute nodes shown because there was no compute cluster or compute node specified in the command.

```
user@host> show app-engine info
Compute cluster: cluster1
  Compute node  Model      Kernel release  Machine  Management IP
  cn2           VXE1001        2.6.18-238.el5  x86_64   192.168.1.29/24

Compute cluster: new-cluster
  Compute node  Model      Kernel release  Machine  Management IP
  new-node      --- Offline ---                10.1.1.1/24
```

show app-engine info compute-cluster compute-node

The output when the optional **compute-cluster** and **compute-node** options are used is the same (both online and offline) as for the **show app-engine info** command except that basic information is displayed for only the compute node specified in the command.

```
user@host> show app-engine info compute-cluster cluster1 compute-node cn2
Compute cluster: cluster1
  Compute node  Model      Kernel release  Machine  Management IP
  cn2           VXE1001        2.6.18-238.el5  x86_64   192.168.1.29/24

user@host> show app-engine info compute-cluster new-cluster compute-node new-node
Compute cluster: new-cluster
  Compute node  Model      Kernel release  Machine  Management IP
  new-node      --- Offline ---                10.1.1.1/24
```

show app-engine info (Command failed)

```
user@host> show app-engine info compute-cluster new-cluster compute-node new-node
```



```
Compute cluster: new-cluster
Compute node      Model      Kernel release    Machine    Management IP
new-node          --- Error getting information --- 10.1.1.1/24
```

show app-engine info (QFX5100 Switch)

```
user@switch> show app-engine info
Compute cluster: default-cluster
Compute node      Model      Kernel release    Machine    Management IP
default-node      QFX        2.6.32-279.22.1.el6.x86_64x86_64 192.168.1.1/24
```

show app-engine processes compute-cluster compute-node

Syntax	show app-engine processes compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>
Release Information	introduced in Junos OS Release 12.3. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
Description	Display information about the processes currently running on the compute node, including the environment information.
Options	<i>compute-cluster-name</i> —Name of compute cluster. <i>compute-node-name</i> —Name of compute node.
Additional Information	In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, , if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show app-engine info on page 71 • show app-engine status on page 85
List of Sample Output	show app-engine processes compute-cluster compute-node on page 75 show app-engine processes compute-cluster compute-node (QFX5100 switches) on page 75
Output Fields	For a description of the output fields, see Table 10 on page 74 . Output fields are listed in the approximate order in which they appear.

Table 10: show app-engine processes compute-cluster compute-node Output Fields

Field Name	Field Description
PID	Process ID number.
TT	Terminal Type.
STAT	
TIME	Time the process started.
COMMAND	Process.

Sample Output

show app-engine processes compute-cluster compute-node

```

user@host> show app-engine processes compute-cluster cc1 compute-node cn0
user@host%ps
  PID  TT  STAT      TIME COMMAND
 37375  qp  I+    0:00.06 /bin/csh
 58992  qp  Is    0:00.04 /bin/csh
 87272  qt  Ss    0:00.03 -csh (csh)
 87308  qt  R+    0:00.00 ps
user@host%ps -e
  PID  TT  STAT      TIME COMMAND
 37375  qp  I+    0:00.06 SB_PATH=/volume/frame/latest/bin:/volume/frame/latest
 58992  qp  Is    0:00.04 STY=58989.ttyqo.svl-junos-pool57 TERM=screen TERMCAP=

```

Sample Output

show app-engine processes compute-cluster compute-node (QFX5100 switches)

```

root> show app-engine processes compute-cluster compute-node
Compute cluster: default-cluster

```

```

Compute node: default-node
  PID  TTY      TIME CMD
   1 ?      00:00:01 init
   2 ?      00:00:00 kthreadd
   3 ?      00:00:00 migration/0
   4 ?      00:00:00 ksoftirqd/0
   5 ?      00:00:00 migration/0
   6 ?      00:00:00 watchdog/0
   7 ?      00:00:00 migration/1
   8 ?      00:00:00 migration/1
   9 ?      00:00:00 ksoftirqd/1
  10 ?      00:00:00 watchdog/1
  11 ?      00:00:00 migration/2
  12 ?      00:00:00 migration/2
  13 ?      00:00:00 ksoftirqd/2
  14 ?      00:00:00 watchdog/2
  15 ?      00:00:00 migration/3
  16 ?      00:00:00 migration/3
  17 ?      00:00:00 ksoftirqd/3
  18 ?      00:00:00 watchdog/3
  19 ?      00:00:01 events/0
  20 ?      00:00:00 events/1
  21 ?      00:00:00 events/2
  22 ?      00:00:01 events/3
  23 ?      00:00:00 cgroup
  24 ?      00:00:00 khelper
  25 ?      00:00:00 netns
  26 ?      00:00:00 async/mgr
  27 ?      00:00:00 pm
  28 ?      00:00:00 sync_supers
  29 ?      0
0:00:00 bdi-default
  30 ?      00:00:00 kintegrityd/0
  31 ?      00:00:00 kintegrityd/1
  32 ?      00:00:00 kintegrityd/2
  33 ?      00:00:00 kintegrityd/3
  34 ?      00:00:00 kblockd/0
  35 ?      00:00:00 kblockd/1

```

```

36 ?      00:00:00 kblockd/2
37 ?      00:00:00 kblockd/3
38 ?      00:00:00 ata/0
39 ?      00:00:00 ata/1
40 ?      00:00:00 ata/2
41 ?      00:00:00 ata/3
42 ?      00:00:00 ata_aux
43 ?      00:00:00 ksuspend_usbd
44 ?      00:00:00 khubd
45 ?      00:00:00 kseriod
46 ?      00:00:00 md/0
47 ?      00:00:00 md/1
48 ?      00:00:00 md/2
49 ?      00:00:00 md/3
50 ?      00:00:00 md_misc/0
54 ?      00:00:00 khungtaskd
55 ?      00:00:00 kswapd0
56 ?      00:00:00 ksmd
57 ?      00:00:00 khugepaged
58 ?      00:00:00 aio/0
59 ?      00:00:00 aio
/1
60 ?      00:00:00 aio/2
61 ?      00:00:00 aio/3
62 ?      00:00:00 crypto/0
63 ?      00:00:00 crypto/1
64 ?      00:00:00 crypto/2
65 ?      00:00:00 crypto/3
70 ?      00:00:00 kthrotld/0
71 ?      00:00:00 kthrotld/1
72 ?      00:00:00 kthrotld/2
73 ?      00:00:00 kthrotld/3
74 ?      00:00:00 pciehp
76 ?      00:00:00 kpsmoused
273 ?     00:00:00 scsi_eh_0
274 ?     00:00:00 scsi_eh_1
275 ?     00:00:00 scsi_eh_2
276 ?     00:00:00 scsi_eh_3
277 ?     00:00:00 scsi_eh_4
278 ?     00:00:00 scsi_eh_5
846 ?     00:00:00 kstriped
887 ?     00:00:00 kdmflush
890 ?     00:00:00 kdmflush
892 ?     00:00:00 kdmflush
1053 ?    00:00:00 udevd
1360 ?    00:00:00 kvm-irqfd-clean
1426 ?    00:00:00 jbd2/dm-2-8
1427 ?    00:00:00 ext4-dio-unwrit
1428 ?    00:00:00 ext4-dio-unwrit
1429 ?    00:00:00 ext4-dio-unwrit
1430 ?    00:00:00 ext4-dio-unwr
it
1431 ?    00:00:00 jbd2/dm-0-8
1432 ?    00:00:00 ext4-dio-unwrit
1433 ?    00:00:00 ext4-dio-unwrit
1434 ?    00:00:00 ext4-dio-unwrit
1435 ?    00:00:00 ext4-dio-unwrit
1436 ?    00:00:00 jbd2/dm-1-8
1437 ?    00:00:00 ext4-dio-unwrit
1438 ?    00:00:00 ext4-dio-unwrit
1439 ?    00:00:00 ext4-dio-unwrit

```

```

1440 ?      00:00:00 ext4-dio-unwrit
1441 ?      00:00:00 jbd2/sda1-8
1432 ?      00:00:00 ext4-dio-unwrit
1433 ?      00:00:00 ext4-dio-unwrit
1434 ?      00:00:00 ext4-dio-unwrit
1435 ?      00:00:00 ext4-dio-unwrit
1436 ?      00:00:00 jbd2/dm-1-8
1437 ?      00:00:00 ext4-dio-unwrit
1438 ?      00:00:00 ext4-dio-unwrit
1439 ?      00:00:00 ext4-dio-unwrit
1440 ?      00:00:00 ext4-dio-unwrit
1441 ?      00:00:00 jbd2/sda1-8
1442 ?      00:00:00 ext4-dio-unwrit
1443 ?      00:00:00 ext4-dio-unwrit
1444 ?      00:00:00 ext4-dio-unwrit
1445 ?      00:00:00 ext4-dio-unwrit
1514 ?      00:00:00 kauditd
1570 ?      00:00:00 flush-253:0
1572 ?      00:00:00 flush-253:2
2623 ?      00:00:00 sshd
2652 ?      00:00:00 xinetd
2671 ?      00:00:00 sntpc
2789 ?      00:00:00 dnsmasq
2793 ?      00:00:00 crond
2825 ?      00:00:00 watchdog
2869 tty1    00:00:00 mingetty
2874 tty2    00:00:00 mingetty
2879 tty3    00:00:00 mingetty
2881 tty4    00:00:
00 mingetty
2885 tty5    00:00:00 mingetty
2887 tty6    00:00:00 mingetty
4209 ?      00:00:00 udevd
5724 ?      00:00:00 vhostd
5738 ?      00:00:00 ntpd
6014 ?      00:00:00 udevd
6017 ?      00:02:41 qemu-kvm
6018 ?      00:00:00 kvm-pit-wq
6031 ?      00:00:00 socat
6178 ?      00:00:00 in.rlogind
6179 ?      00:00:00 login
6180 pts/1   00:00:00 sh
6423 ?      00:00:00 rsyslogd
7012 ?      00:00:00 libvirtd
7086 ?      00:00:21 qemu-kvm
7087 ?      00:00:00 kvm-pit-wq
7144 pts/1   00:00:00 virsh
7260 ?      00:00:00 ps

```

show app-engine resource-usage

Syntax	<code>show app-engine resource-usage</code> <code><compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>></code>
Release Information	Command introduced in Junos OS Release 12.3. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
Description	Show the resource usage information of the compute node. Display used and free space available and other information for storage pools, physical disks, and so on.
Options	<ul style="list-style-type: none"><code>compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i></code>—Name of the compute cluster and name of the compute node.
Additional Information	<p>In the operational mode of the CLI when you type <code>?</code> for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, , if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.</p> <p>For commands with an optional <code>compute-cluster <i>compute-cluster-name</i></code> option, if that option is omitted, the command will be executed on all compute nodes of all compute clusters. For commands with an optional <code>compute-node <i>compute-node-name</i></code> option, if that option is omitted, the command will be executed on all compute nodes of the specified compute cluster.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">show app-engine info on page 71show app-engine processes compute-cluster compute-node on page 74show app-engine status on page 85
List of Sample Output	show app-engine resource-usage on page 81 show app-engine resource-usage compute-cluster compute-node (compute node offline) on page 83 show app-engine resource-usage compute-cluster compute-node (failure to get information requested) on page 83 show app-engine resource-usage (QFX5100 Switches) on page 83
Output Fields	For a description of the output fields, see Table 11 on page 79 . Output fields are listed in the approximate order in which they appear.

Table 11: show app-engine resource-usage Output Fields

Field Name	Field Description
Compute Cluster	Name of compute cluster
Compute Node	Name of compute node
CPU Usage	<p>This section of the display shows the CPU utilization since system startup. Each row displays the CPU utilization of individual CPUs.</p> <ul style="list-style-type: none"> • First column—No heading. This column shows the time at which the CPU usage was taken. • CPU—Processor number. The keyword all indicates that statistics are calculated as averages among all processors. • %use—Percentage of CPU utilization that occurred while executing at the user level (application). • %nice—Percentage of CPU utilization that occurred while executing at the user level with nice priority. UNIX "niceness" value. A lower number indicates a higher priority. • %sys—Percentage of CPU utilization that occurred while executing at the system level (kernel). Note that this does not include time spent servicing interrupts or softirqs. A softirq (software interrupt) is one of up to 32 enumerated software interrupts which can run on multiple CPUs at one time. • %iowait—Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request. • %irq—Percentage of time spent by the CPU or CPUs to service interrupts. • %soft—Percentage of time spent by the CPU or CPUs to service softirqs. A softirq (software interrupt) is one of up to 32 enumerated software interrupts which can run on multiple CPUs at once. • %steal—Percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor. • %idle—Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request. • intr/s—Total number of interrupts received per second by the CPU or CPUs.
Memory Usage	<p>This section shows the current memory usage in megabytes. Below are the fields in each row.</p> <ul style="list-style-type: none"> • total—Total memory. • used—Used memory. • free—Free memory. • shared—Ignore this field. • buffers—Amount of physical RAM used for file buffers. • cached—Amount of physical RAM used as cache memory.
Mem	RAM usage.
Swap	Swap usage.

Table 11: show app-engine resource-usage Output Fields (*continued*)

Field Name	Field Description
Disk Usage	<p>This section displays the disk usage of the compute node in Gigabytes.</p> <ul style="list-style-type: none"> • Filesystem—Name of the filesystem. • Size—Total space. • Used—Used space. • Avail—Available space. • Use%—Used percentage. • Mounted on—Directory in which the filesystem is mounted.
Storage Information	<p>Volume information:</p> <ul style="list-style-type: none"> • Volume #—Number of the volume as in Volume0, Volume1, etc. • Volume OS name—Drive name that the compute node uses. The drive name consists of the path, drive name, and PCI part number. • Volume State—Whether the volume is enabled or disabled and if both disks are working properly. The different values for the Volume State are: <ul style="list-style-type: none"> • degraded, enabled • optimal, enabled • degraded, enabled, resync in progress Resync Progress: total blocks 1757917184, blocks remaining 1488220472, 84% • Volume Size—Size of the volume <p>Physical disk information:</p> <ul style="list-style-type: none"> • slot#—Disk slot number. • Bus—SAS bus number; information used for debugging. • Tgt—SAS target number; information used for debugging. • Size(GB)—Size of the disk. • Volume #—Number of the volume as in Volume0, Volume1, etc. • Role—Primary or secondary disk. • Type—Model of disk: SAS or SATA. • State—Online or offline. <p>Internal disk information:</p> <ul style="list-style-type: none"> • Slot#—Disk slot number. • Bus—SAS bus number; information used for debugging. • Tgt—SAS target number; information used for debugging. • Size(GB)—Size of the disk. • Type—Model of disk: ships with SATA. • Product—Product part number of the disk; information used for debugging. • Rev—Rev for the disk; information used for debugging. • SASAddress—SAS address allocated to the disk; information used for debugging. • PhyNum—Information used for debugging.

Table 11: show app-engine resource-usage Output Fields (*continued*)

Field Name	Field Description
Volume Groups	<p>All the volumes on the system go into a volume group:</p> <ul style="list-style-type: none"> • VG—Name of the volume group. This is vg01. • #PV—Number of physical volumes in use. • #LV—Number of logical volumes in use. (These are the volumes created with the secondary-disk configuration statement.) • #SN—Serial number. • Attr—Attributes. The vg_attr bits are: <ul style="list-style-type: none"> 1 Permissions: (w)riteable, (r)ead-only 2 Resi(z)eable 3 E(x)ported 4 (p)artial: one or more physical volumes belonging to the volume group are missing from the system 5 Allocation policy: (c)ontiguous, c(l)ing, (n)ormal, (a)nywhere, (i)nherited 6 (c)lustered • VSize—Size of the total volume. • VFree—Size of the available portion of the volume.

Sample Output

show app-engine resource-usage

```
user@host> show app-engine resource-usage
Compute cluster: cc3
```

```
Compute node: cn4
```

```
CPU Usage
```

```
=====
```

	CPU	%user	%nice	%sys	%iowait	%irq	%soft	%steal	%idle	intr/s
08:11:33 all		0.01	0.00	0.04	0.02	0.01	0.00	0.00	99.93	1326.41
08:11:33 0		0.10	0.00	0.43	0.16	0.12	0.01	0.00	99.19	1326.41
08:11:33 1		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 2		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 3		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 4		0.00	0.00	0.01	0.00	0.00	0.00	0.00	99.98	0.00
08:11:33 5		0.00	0.00	0.01	0.00	0.00	0.00	0.00	99.99	0.00
08:11:33 6		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 7		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 8		0.01	0.00	0.02	0.09	0.00	0.00	0.00	99.88	0.00
08:11:33 9		0.01	0.00	0.03	0.00	0.00	0.00	0.00	99.96	0.00
08:11:33 10		0.00	0.00	0.01	0.00	0.00	0.00	0.00	99.99	0.00
08:11:33 11		0.00	0.00	0.01	0.00	0.00	0.00	0.00	99.99	0.00
08:11:33 12		0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
08:11:33 13		0.02	0.00	0.05	0.00	0.00	0.00	0.00	99.93	0.00
08:11:33 14		0.01	0.00	0.02	0.00	0.00	0.00	0.00	99.97	0.00
08:11:33 15		0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.99	0.00

```
Memory Usage
```

```
=====
```

	total	used	free	shared	buffers	cached
Mem:	48354	442	47912	0	0	315
Swap:	66767	0	66767			

Disk Usage

=====

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	24G	0	24G	0%	/dev/shm
/dev/mapper/vg00-sfs	404G	11G	373G	3%	/volume/sfs
/dev/mapper/vg00-lfs	450G	234M	427G	1%	/volume-lfs

Storage Information

=====

1 volume is active, 1 physical disk is active

Volume 0 is Bus 0 Target 16, Type IM (Integrated Mirroring)

Volume #: Volume0

Volume OS name: /dev/sdd [8:1:16:0]

Volume State: degraded, enabled

Volume Size 476798 MB, 2 Members

Primary is Disk in slot 2 Secondary Disk in slot 3 *has failed or is missing*

=====

1 physical disk is active

slot#	Bus	Tgt	Size(GB)	Volume#	Role	Type	State
-------	-----	-----	----------	---------	------	------	-------

2	0	18	465.76	Volume0	Primary	SATA	online
---	---	----	--------	---------	---------	------	--------

=====

Internal disks

Slot#	Bus	Tgt	Size(GB)	Type	Product	Rev	SASAddress	PhyNum
-------	-----	-----	----------	------	---------	-----	------------	--------

1	0	19	838.36	SAS	ST9900805SS	0003	5000c50043e5f0b5	1
0	0	20	838.36	SAS	ST9900805SS	0003	5000c50043e5e701	0

=====

Unused disks: (Will be added to storage system when its pairing disk is installed)

Slot#	Bus	Tgt	Size(GB)	Type	Product	Rev	SASAddress	PhyNum
-------	-----	-----	----------	------	---------	-----	------------	--------

=====

Volume Groups

=====

VG	#PV	#LV	#SN	Attr	VSize	VFree
vg01	1	0	0	wz--n-	465.76G	465.76G

show app-engine resource-usage compute-cluster compute-node (compute node offline)

```
user@host> show app-engine resource-usage compute-cluster new-cluster compute-node
new-node
```

```
Compute cluster: new-cluster
```

```
Compute node: new-node
```

```
--- Offline ---
```

show app-engine resource-usage compute-cluster compute-node (failure to get information requested)

```
user@host> show app-engine resource-usage compute-cluster new-cluster compute-node
new-node
```

```
Compute cluster: new-cluster
```

```
Compute node: new-node
```

```
--- Error getting information ---
```

show app-engine resource-usage (QFX5100 Switches)

```
root> show app-engine resource-usage
```

```
Compute cluster: default-cluster
```

```
Compute node: default-node
```

CPU Usage

```
=====
```

10:18:26	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal
%guest %idle								
10:18:26	all	0.66	0.00	2.26	1.08	0.00	0.02	0.00
4.78 91.19								
10:18:26	0	0.58	0.00	0.79	3.02	0.00	0.01	0.00
1.25 94.36								
10:18:26	1	0.79	0.00	0.83	0.21	0.00	0.00	0.00
0.35 97.82								
10:18:26	2	0.00	0.00	0.01	0.00	0.00	0.00	0.00
0.00 99.99								
10:18:26	3	1.29	0.00	7.49	1.11	0.00	0.08	0.00
17.66 72.36								

Memory Usage

```
=====
```

	total	used	free	shared	buffers	cached
Mem:	15178	6875	8303	0	8	988
Swap:	0	0	0			

Disk Usage

```
=====
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	7.5G	4.0K	7.5G	1%	/dev/shm
/dev/mapper/vg0_vjunos-lv_var	11G	4.2G	5.7G	43%	/var
/dev/mapper/vg0_vjunos-lv_junos	12G	4.2G	7.1G	37%	/junos
/dev/mapper/vg0_vjunos-lv_junos_recovery	4.0G	960M	2.9G	26%	/recovery
/dev/sda1	962M	312M	602M	35%	/boot

Storage Information

=====

VG	#PV	#LV	#SN	Attr	VSize	VFree
vg0_vjunos	2	3	0	wz--n-	26.53g	0

show app-engine status

Syntax	<code>show app-engine status</code> <code><compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i>></code>
Release Information	Command introduced in Junos OS Release 12.3. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
Description	Show the status of a compute node.
Options	<code>compute-cluster <i>compute-cluster-name</i> compute-node <i>compute-node-name</i></code> —(Optional) Name of computer cluster and compute node.

Additional Information In the operational mode of the CLI when you type ? for a name, for example a compute-node name, you would expect to get a list of available compute nodes plus the option to type in a name not listed. This is the auto-complete feature in the CLI. However, in JunosV App Engine, if you specify compute cluster and compute node in the operational command, the auto-complete works only if the compute cluster is put before the compute node.

For commands with an optional `compute-cluster compute-cluster-name` option, if that option is omitted, the command will be executed on all compute nodes of all compute clusters. For commands with an optional `compute-node compute-node-name` option, if that option is omitted, the command will be executed on all compute nodes of the specified compute cluster.

When the compute node configuration has been updated or the jvae package is upgraded, this command shows the following note that the compute node configuration or the jvae package has been updated and the compute node needs to be rebooted for the configuration to take effect.

NOTE: Configuration and/or jvae package updated. Please reboot to take effect.

The note also displays the details of what exactly has been updated on the Routing Engine. Once the compute node is rebooted, the note automatically goes off.

If the operational compute node name (hostname) is different than the administrative compute node name, then the following error message is displayed.

```
ERROR: The compute node has the following errors:
- Operational node name: cn
  Administrative node name: newname
  Either reboot the node OR change the
  administrative node name to 'cn' to
  continue VMM operations.
```

Where `cn` is the previous compute node name.

A mismatch between the operational node name and the administrative node name could happen if you rename the compute node name in the CLI. For example:

```
[edit]
user@router# rename services app-engine compute-cluster cc compute-node cn to
compute-node newname
```

```

[edit]
user@router# set services app-engine virtual-machines instance vmx1 compute-cluster cc
compute-node newname
[edit]
user@router# commit
commit complete

[edit]
user@router# run show app-engine status
Compute cluster: cc
Compute node      Status
newname           Online      ERROR: The compute node has the following errors:
                  - Operational node name: cn
                  Administrative node name: newname
                  Either reboot the node OR change the administrative
                  node name to 'cn' to continue VMM operations.

[edit]
user@router#

```

If the compute node has any error messages displayed, the VM operations will not work until the error is fixed.

```

user@router# run show app-engine virtual-machine instance
VM name      Compute cluster  VM status
vmx1         cc              --- VMM service is not available ---

```

To fix the error, either reboot the node or change the administrative node name (in both the physical and VM configurations) back to the operational (previous) compute node name.

Required Privilege Level

view

Related Documentation

- [show app-engine info on page 71](#)
- [show app-engine processes compute-cluster compute-node on page 74](#)
- [show app-engine resource-usage on page 78](#)

List of Sample Output

[show app-engine status on page 87](#)
[show app-engine status compute-cluster compute-node on page 87](#)

Output Fields

For a description of the output fields, see [Table 12 on page 86](#). Output fields are listed in the approximate order in which they appear.

Table 12: show app-engine status Output Fields

Field Name	Field Description
Compute cluster	Name of compute cluster.
Compute node	Name of compute node.

Table 12: show app-engine status Output Fields (*continued*)

Field Name	Field Description
Status	<p>Online or offline.</p> <p>If the status is Offline, a note is displayed. The XX:XX:XX:XX:XX:XX variable in the note will contain the mac-address attribute of the compute node.</p> <p>NOTE: If you see the node as Offline, please check if the compute node is booted and is properly connected to the router.</p>

Sample Output

show app-engine status

```

user@host> show app-engine status
Compute cluster: cc1
  Compute node   Status
  cn0            Online
Compute cluster: new-cluster
  Compute node   Status
  new-node       Offline

```

NOTE: Please check if the compute node with network boot interface MAC address XX:XX:XX:XX:XX:XX is UP and is properly connected to the router

show app-engine status compute-cluster compute-node

The output for the **show app-engine status compute-cluster compute-node** command is identical to that for the **show app-engine status** command except that usage information displayed is for only the compute node specified whereas the **show app-engine status** command displays all the compute nodes.

show app-engine virtual-machine package

Syntax	show app-engine virtual-machine package <vm-package-name>
Release Information	Command introduced in Junos OS Release 12.3. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
Description	Show the download status of a virtual-machine (VM) package
Options	<i>vm-package-name</i> —(Optional) Name of the VM package.
Additional Information	If the <i>vm-package-name</i> is not included in the command, the command is executed for all VM package names.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> compute-cluster (VM instance) on page 43 show app-engine virtual-machine instance on page 90
List of Sample Output	show app-engine virtual-machine package on page 88 show app-engine virtual-machine package (QFX5100 Switches) on page 89
Output Fields	For a description of the output fields, see Table 13 on page 88 . Output fields are listed in the approximate order in which they appear.

Table 13: show app-engine virtual-machine package Output Fields

Field Name	Field Description
VM package	Name of the VM package.
VM disk image	Name of the virtual machine disk image. Includes the package-name, version, and disk image name.
Compute cluster	Name of the compute cluster.
Package download status	Download status of VM package for the cluster.

Sample Output

show app-engine virtual-machine package

```

user@host> show app-engine virtual-machine package
VM package: centos_vm
VM disk image: centos_vm/12.3B2/centos.img.gz
Compute cluster          Package download status
cc1                      DOWNLOADED

```


Sample Output

show app-engine virtual-machine package (QFX5100 Switches)

```
root> show app-engine virtual-machine package
VM package: cust-vm-ve
VM disk image: cust-vm-ve/13.2/cust-vm.img.gz
  Compute cluster      Package download status
  default-cluster      DOWNLOADED
```

show app-engine virtual-machine instance

Syntax	show app-engine virtual-machine instance <i><instance-name></i> <brief detail>
Release Information	Command introduced in Junos OS Release 12.3. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
Description	Display the status of a virtual machine (VM) instance.
Options	<i>instance-name</i> —(Optional) Name of the VM instance. (brief detail)—(Optional) Level of description of display.
Additional Information	For commands with an optional <i>instance-name</i> , if the <i>instance-name</i> is omitted, the command will be executed on all VM instances. If neither brief nor detail option is specified, the default is brief .
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • compute-cluster (VM instance) on page 43 • show app-engine virtual-machine package on page 88
List of Sample Output	show app-engine virtual-machine instance on page 91 show app-engine virtual-machine instance for a Single VM Instance on page 91 show app-engine virtual-machine instance detail on page 91 show app-engine virtual-machine instance (QFX5100 Switches) on page 92
Output Fields	For a description of the output fields, see Table 14 on page 90 . Output fields are listed in the approximate order in which they appear.

Table 14: show app-engine virtual-machine instance Output Fields

Field Name	Field Description
VM name	Name of the VM instance.
Compute cluster	Name of the compute cluster.
VM status	<p>Status of VM instance: active (the VM is up) or inactive (the VM is down).</p> <p>When the status is listed as ACTIVE, it does not mean that the operating system on the VM has come up. You need to telnet to the VM console to verify if the OS has come up or not and if everything is running properly.</p>
Compute node name	Name of the compute node.

Table 14: show app-engine virtual-machine instance Output Fields (*continued*)

Field Name	Field Description
VM UUID	Unique identifier of the VM instance.
Number of CPUs	Number of CPU in the VM instance.
VM disk image	Name of VM disk image including package, version, and image (<i>package-name/version/image-name</i>).
Memory (GB)	Memory in gigabytes.
Console	Identifier of console.
VM PID	Process ID of VM instance.
Name of the VM interface	IP address of the VM.
Name of the secondary disk or disks	Size of secondary disk.

Sample Output

show app-engine virtual-machine instance

```

user@host> show app-engine virtual-machine instance
VM name      Compute cluster  VM status
attacker      cc1              ACTIVE
host          cc1              ACTIVE
snort         cc1              ACTIVE

```

show app-engine virtual-machine instance for a Single VM Instance

```

user@host> show app-engine virtual-machine instance attacker
VM name      Compute cluster  VM status
attacker      cc1              ACTIVE

```

show app-engine virtual-machine instance detail

```

user@host> show app-engine virtual-machine instance attacker detail
VM name: attacker
VM status: ACTIVE
Compute cluster name: cc1
Compute node name: cn1
VM UUID: 34938d5c-6c45-11e1-8b1f-00e081ca5959
VM disk image: centos_vm/12.3B2/centos.img
Number of CPUs: 1
Memory (GB): 1
Console: 10.5.5.9 15144
VM PID: 9792
eth0: 10.5.5.29
hda: 10 GB

```

Sample Output

show app-engine virtual-machine instance (QFX5100 Switches)

```
root> show app-engine virtual-machine instance
VM name          Compute cluster  VM status
vm                default-cluster  ACTIVE
```

CHAPTER 4

Indexes

- [Index on page 95](#)
- [Index of Statements and Commands on page 97](#)

Index

Symbols

#, comments in configuration statements.....	xi
(), in syntax descriptions.....	xi
< >, in syntax descriptions.....	x
[], in configuration statements.....	xi
{ }, in configuration statements.....	xi
(pipe), in syntax descriptions.....	xi

A

app-engine statement.....	36
---------------------------	----

B

braces, in configuration statements.....	xi
brackets	
angle, in syntax descriptions.....	x
square, in configuration statements.....	xi
bridge interface.....	39, 52
bridge statement.....	39

C

comments, in configuration statements.....	xi
compute node	
configuration of.....	44
defined.....	13
reboot not required.....	39, 46, 53, 54, 57
reboot required.....	39, 46, 48, 53, 57
compute-cluster statement	
app engine.....	41
vm instance.....	43
compute-node statement.....	44
conventions	
text and syntax.....	x
curly braces, in configuration statements.....	xi
customer support.....	xi
contacting JTAC.....	xi

D

dhcp option for configuring host OS interfaces.....	48
DHCP server	
and PXE boot.....	23
how to configure.....	23

documentation	
comments on.....	xi

E

Ethernet interface.....	39, 46
ethernet statement.....	46
usage.....	32

F

family statement.....	48, 49
font conventions.....	x
forwarding-options statement	
usage.....	23

G

guest OS	
defined.....	13

H

Host Base OS package	
contents of.....	23
host OS	
overview of.....	13

I

instance See virtual machine instance	
instance statement.....	50
interface statement.....	52
usage.....	32
interfaces statement.....	53

J

JunosV App Engine	
overview of.....	13
physical and virtual configuration of.....	36

L

Linux CentOS	
how to configure for DHCP.....	23
local management interface	
defined.....	54
local-management statement.....	54

M

MAC address	
configuration of.....	44
management interface.....	48
configuration of.....	39, 46, 48
defined.....	54

manuels		
comments on.....	xi	
MIB for the JunosV App Engine.....	30	
N		
network interface		
configuration of.....	46, 53	
defined.....	39	
P		
parentheses, in syntax descriptions.....	xi	
PXE boot		
booting the VSE device.....	23	
R		
remote applications		
and VT-d mode.....	32	
defined.....	13	
request app-engine offline compute-cluster		
compute-node command.....	63	
request app-engine reboot compute-cluster		
compute-node command.....	65	
request app-engine reboot compute-cluster		
compute-node virtual-machine-instance		
command.....	67	
S		
secondary-disk statement.....	55	
show app-engine info command.....	71	
show app-engine processes compute-cluster		
compute-node command.....	74	
show app-engine resource-usage command.....	78	
usage.....	55	
show app-engine status command.....	85	
show app-engine virtual-machine package		
command.....	88	
show app-engine virtual-machine-instance		
command.....	90	
SNMP		
monitoring based on.....	30, 31, 56	
traps.....	30, 31, 56	
storage		
secondary, creation of.....	55	
subagent.....	56	
support, technical See technical support		
syntax conventions.....	x	
syslog server, configuring.....	31	
syslog statement.....	57	
T		
technical support		
contacting JTAC.....	xi	
TFTP server		
and PXE boot.....	23	
how to configure.....	23	
V		
virtual machine		
using VT-d mode.....	32	
virtual machine instance		
configuration of.....	50	
virtual machine interface		
configuration of.....	52	
configuration of secondary storage.....	55	
connected to a network by a bridge		
interface.....	39	
virtual-machines statement.....	59	
VMM.....	50	
VSE device		
and compute node interface.....	44	
defined.....	23	
ports, order of.....	46	
VT-d mode		
configuration of.....	32, 46, 52	
overview of.....	32	

Index of Statements and Commands

A

app-engine statement.....36

B

bridge statement.....39

C

compute-cluster statement
 app engine.....41
 vm instance.....43
compute-node statement.....44

E

ethernet statement.....46

F

family statement.....48, 49

I

instance statement.....50
interface statement.....52
interfaces statement.....53

L

local-management statement.....54

R

request app-engine offline compute-cluster
 compute-node command.....63
request app-engine reboot compute-cluster
 compute-node command.....65
request app-engine reboot compute-cluster
 compute-node virtual-machine-instance
 command.....67

S

secondary-disk statement.....55

show app-engine info command.....71
show app-engine processes compute-cluster
 compute-node command.....74
show app-engine resource-usage command.....78
show app-engine status command.....85
show app-engine virtual-machine package
 command.....88
show app-engine virtual-machine-instance
 command.....90
SNMP
 monitoring based on.....30, 31, 56
 traps.....30, 31, 56
subagent.....56
syslog statement.....57

V

virtual-machines statement.....59

