Installing the Cisco CSR 1000v in KVM Environments

Kernel Virtual Machine Support Information

The CSR1000v supports the following Linux/KVM environments:

- Red Hat Enterprise Linux (RHEL)
- Red Hat Enterprise Virtualization (RHEV)
- Ubuntu (beginning with Cisco IOS XE Release 3.11S)

Red Hat Enterprise Linux (RHEL), an enterprise virtualization product produced by Red Hat, based on the Kernel-based Virtual Machine (KVM), is an open source, full virtualization solution for Linux on x86 hardware, containing virtualization extensions.

The Red Hat Enterprise Virtualization (RHEV) platform is a commercially packaged virtualization platform from Red Hat.

For more information on the KVM products and versions supported, see the "Installation Requirements for KVM" sections in the following few pages.

The Cisco CSR 1000v installation on KVM requires the manual creation of a VM and installation using the .iso file or the qcow2 file. Deploying the OVA template into a KVM environment is not supported.

The Cisco CSR 1000v supports the Virtio vNIC type on the KVM implementation. KVM supports a maximum of 26 vNICs.

KVM Support on OpenStack

(Cisco IOS XE 3.12S or later) The Cisco CSR 1000v supports the OpenStack environment. OpenStack support requires the .qcow2 installation file available on the Cisco.com download page. For more information, see Creating the Instance Using the OpenStack Command Line Tool , on page 10 and Creating the Instance Using the OpenStack Dashboard, on page 12.

Installation Requirements for KVM

The KVM requirements for Cisco CSR 1000V using Cisco IOS XE16.12 through 17.3 releases are as follows:

Note The KVM requirements for older versions of Cisco IOS XE (before IOS XE Denali 16.3) are shown in Installation Requirements for KVM—Cisco IOS XE 3.x, on page 21.

KVM Versions

Cisco IOS XE Release	KVM Version
Cisco IOS XE 17.3.x release	Linux KVM based on Red Hat Enterprise Linux 7.5 and 7.7 are recommended for release 17.3.1 - tested and meets performance benchmarks.
Cisco IOS XE 17.1.x and 17.2.x releases	Linux KVM based on Red Hat Enterprise Linux 7.4 is recommended for releases 17.1 and 17.2. These versions are tested and meets performance benchmarks.
Cisco IOS XE 16.12 release	Linux KVM based on Red Hat Enterprise Linux 7.4 is recommended - tested and meets performance benchmarks.

- vCPUs. The following vCPU configurations are supported:
  - 1 vCPU: requires minimum 4 GB RAM allocation
  - 2 vCPUs: requires minimum 4 GB RAM allocation
  - 4 vCPUs: requires minimum 4 GB RAM allocation
- Virtual CPU cores—1 vCPU is required
- Virtual hard disk size—8 GB minimum
- Supported vNICs—Virtio, ixgbe, ixgbevf or i40evf

Note If a vNIC with an i40evf driver is used, the maximum number of physical VLANs is limited to 512, shared across all VFs, and the number of VLANs for a VF can be further limited by the host (PF) driver for untrusted VFs. The latest Intel i40e PF driver limits untrusted VFs to a maxium of 8 VLANs/sub-interfaces.

- Maximum number of vNICs supported per VM instance—26
- Virtual CD/DVD drive installed (applicable only when installing using an .iso file)—required

Creating a Cisco CSR 1000v KVM Instance

Creating the Cisco CSR 1000v VM Using the Self-installing .Run Package

The Cisco CSR 1000v KVM Installer package (with **.run** extension) is a self-installing CSR package for KVM.

Default or Interactive

Installing a CSR instance using the **.run** package provides two options:

- Default mode

  Installation uses the bundled CSR image file and one of the default VM configuration options (small, medium, large, xlarge) described in the procedure below.

- Interactive mode

  Allows customization of VM configuration and option to install the bundled CSR image file or a separate .qcow2 image.

Installation Procedure

The following steps are performed on the KVM server.

Prerequisites

Download the .run executable from the Cisco CSR 1000v software installation image package and copy it onto a local device.

SUMMARY STEPS

1. Run the .run executable to launch the CSR VM.

DETAILED STEPS

Run the .run executable to launch the CSR VM.

Example:

```
csr1000v-universalk9.03.16.01a.S.155-3.S1a-ext.run <option>
```

Values for <option>:

- **interactive**—Interactive mode
- **small**—Deploy CSR with: 1 vCPU, 4 GB RAM, 3 vNICs
- **medium**—Deploy CSR with: 2 vCPU, 4 GB RAM, 3 vNICs
- **large**—Deploy CSR with: 4 vCPU, 4 GB RAM, 3 vNICs
- **xlarge**—Deploy CSR with: 4 vCPU, 8 GB RAM, 3 vNICs

Example of Interactive Mode Installation

Example:

```
./csr1000v-universalk9.03.16.01a.S.155-3.S1a-ext.run interactive
Verifying archive integrity...  100%   All good.
Uncompressing KVM Installer Package  100%
Setting up installation of CSR on KVM
Tuning Daemon is already running
A tuning daemon process supports the automation of CSR VM vCPU and vhost vCPU affinity setting on
Ubuntu host OS and RHEL.
Please enter the appropriate values for the options below:
Enter the VM name [csr_29405]: my_csr_vm
Enter the image location
[/var/lib/libvirt/images/csr1000v-universalk9.03.16.01a.S.155-3.S1a-ext.qcow2_csr_29405]:
Enter the number of vCPUs [1]: 4
Enter the value for RAM [4096]: 8192
Enter number of vnics [3]: 2
Options for virtual bridge:
virbr0
vnet1
Enter bridge for vnic 1 [virbr0]:
Options for virtual bridge:
virbr0
vnet1
Enter bridge for vnic 2 [virbr0]:
Options for virtual bridge:
virbr0
vnet1
Creating VM my_csr_vm
Do you want to start the CSR?[y]
Starting CSR my_csr_vm...
VM my_csr_vm started
```

Creating the Cisco CSR 1000v VM Using the virt-manager GUI Tool

Creating the Cisco CSR 1000v VM Using virt-manager with qcow2 or ISO Image

Before you begin

Download and install the virt-manager RPM package on the KVM server.

Download the .qcow2 or .iso image from the Cisco CSR 1000v software installation image package and copy it onto a local or network device.

Step 1 Launch the virt-manager GUI. Click **Create a new virtual machine**.

Step 2 Do one of the following: (For **.qcow2**) Select **Import existing disk image**. (For **.iso**) Select **Local install media (ISO image or CDROM)**.

Step 3 Select the CSR qcow2 or iso file location.

Step 4 Configure the memory and CPU parameters.

Step 5 Configure virtual machine storage.

Step 6 Click **Finish**.

Note To add additional hardware before creating the VM, select **Customize configuration before install** before clicking **Finish**. If this option is selected, then the next screen displays an **Add Hardware** button that can be used one or more times to add various hardware options, such as additional disks or a serial port interface (see the following sections).

Step 7 Access the Cisco CSR 1000v console by using one of the following:

a) (If using the virtual console) Double-click the VM instance to access the VM console

b) (If using the serial console) See Booting the Cisco CSR 1000v and Accessing the Console

What to do next

Creating the Cisco CSR 1000v VM Using virt-manager—Add Serial Port

Enables access to the CSR by adding a serial console. See Booting the Cisco CSR 1000v as the VM.

SUMMARY STEPS

1. Click **Add Hardware**.
2. Select the **Serial** option from the menu.
3. From the **Device type** drop-down menu, select **TCP net console (tcp)**.
4. Specify the port number, and select the **Use Telnet** checkbox.
5. Click **Finish**.
6. After adding all necessary hardware, click **Begin Installation**.

DETAILED STEPS

Step 1 Click **Add Hardware**.

Step 2 Select the **Serial** option from the menu.

Step 3 From the **Device type** drop-down menu, select **TCP net console (tcp)**.

Step 4 Specify the port number, and select the **Use Telnet** checkbox.

Step 5 Click **Finish**.

Step 6 After adding all necessary hardware, click **Begin Installation**.

Creating the Cisco CSR 1000v VM Using virt-manager--Add Hard Disk

Describes the optional steps after selecting **Customize configuration before install** .

Before you begin

- Perform the task Creating a Bootstrap Day0 Configuration for virt-manager, on page 6 —Using .qcow2 or .iso Image, but during the task steps before clicking "Finish", select the **Customize configuration before install** option . A screen with "Add Hardware" button appears.

SUMMARY STEPS

1. Click **Add Hardware**.
2. Select the **Storage** option from the menu.
3. Select **Select managed or other existing storage** checkbox.
4. (Applicable only when adding a Bootstrap Day0 configuration) Click the **Browse** button and navigate to the **csr_config.iso** location. From the **Device type** drop-down menu, select the **IDE CDROM** option.
5. Click **Finish**.
6. After adding all necessary hardware, click **Begin Installation**.

DETAILED STEPS

Step 1 Click **Add Hardware**.

Step 2 Select the **Storage** option from the menu.

Step 3 Select **Select managed or other existing storage** checkbox.

Step 4 (Applicable only when adding a Bootstrap Day0 configuration) Click the **Browse** button and navigate to the **csr_config.iso** location. From the **Device type** drop-down menu, select the **IDE CDROM** option.

Step 5 Click **Finish**.

Step 6 After adding all necessary hardware, click **Begin Installation**.

Creating a Bootstrap Day0 Configuration for virt-manager

This procedure provides additional steps to be executed within Creating the Cisco CSR 1000v VM Using virt-manager with qcow2 or ISO Image, on page 4.

The following steps are performed on the KVM server.

SUMMARY STEPS

1. Create **iosxe_config.txt** or **ovf-env.xml** file. (For details, see Bootstrap Properties, on page 13.)
2. Create a disk image from this file using below command:
3. (This step must be performed within Creating the Cisco CSR 1000v VM Using virt-manager with qcow2 or ISO Image, on page 4 .)

DETAILED STEPS

Step 1 Create **iosxe_config.txt** or **ovf-env.xml** file. (For details, see Bootstrap Properties, on page 13.)

Step 2 Create a disk image from this file using below command:

Example:

```
mkisofs -l -o /my/path/csr_config.iso <configuration_filename>
```

Step 3 (This step must be performed within Creating the Cisco CSR 1000v VM Using virt-manager with qcow2 or ISO Image, on page 4 .)

Mount the **csr_config.iso** as an additional disk during creation of the CSR virtual machine.

Creating the Cisco CSR 1000v VM Using virt-install with qcow2 Image

- Download and install the virt-install RPM package on the KVM server.
- Download the **.qcow2** image from the Cisco CSR 1000v software installation image package and copy it onto a local or network device.

Before you begin

Using the virt-install command, create the instance and boot, using the following syntax:

Example:

```
virt-install                        \
     --connect=qemu:///system       \
     --name=my_csr_vm               \
     --os-type=linux                \
     --os-variant=rhel4             \
     --arch=x86_64                  \
     --cpu host                     \
     --vcpus=1,sockets=1,cores=1,threads=1    \
     --hvm                          \
     --ram=4096                     \
     --import                       \
     --disk path=<path_to_csr1000v_qcow2>,bus=ide,format=qcow2    \
     --network bridge=virbr0,model=virtio                         \
     --noreboot
```

(Optional) To configure a Bootstrap Day0 configuration, perform the steps described in Creating a Bootstrap Day0 Configuration for virt-install, on page 9.

After the installation is complete, the CSR VM will be shutdown. You can start the CSR VM using the **virsh start** command.

What to do next

Red Hat Enterprise Linux—Setting Host Mode

Due to an issue specific to Red Hat Enterprise Linux, when launching the Cisco CSR1000v in a Red Hat Enterprise Linux environment using **virt-install**, set the host mode as follows:

- In Red Hat Enterprise Linux 6, use:

  ```
  --cpu host
  ```

- In Red Hat Enterprise Linux 7, use:

  ```
  --cpu host-model
  ```

Creating the Cisco CSR 1000v VM Using virt-install with ISO Image

Before you begin

- Download and install the virt-install RPM package on the KVM server.
- Download the **.iso** image from the Cisco CSR 1000v software installation image package and copy it onto a local or network device.

Step 1 Create a 8G disk image in **.qcow2** format using the **qemu-img** command.

Example:

```
qemu-img create -f qcow2 csr_disk.qcow2 8G
```

Step 2 Use the **virt-install** command to install the CSR. This requires the correct permissions to create a new VM. The following example creates a 1 vCPU CSR with 4G of RAM, one network interface, and one serial port.

Example:

```
   virt-install               \
    --connect=qemu:///system  \
    --name=my_csr_vm          \
    --description "Test VM"   \
    --os-type=linux           \
    --os-variant=rhel4        \
    --arch=x86_64             \
    --cpu host                \
    --vcpus=1,sockets=1,cores=1,threads=1     \
    --hvm                     \
    --ram=4096                \
    --cdrom=<path_to_csr1000v_iso>            \
    --disk path=csr_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2   \
    --network bridge=virbr0,model=virtio      \
    --noreboot
```

(Optional) To configure a Bootstrap Day0 configuration, perform the steps described in Creating a Bootstrap Day0 Configuration for virt-install, on page 9.

The **virt-install** command creates a new VM instance and the CSR installs the image onto the specified disk file. After the installation is complete, the CSR VM will be shutdown. You can start the CSR VM using the **virsh start** command.

Step 3

What to do next

Red Hat Enterprise Linux—Setting Host Mode

Due to an issue specific to Red Hat Enterprise Linux, when launching the Cisco CSR1000v in a Red Hat Enterprise Linux environment using **virt-install**, set the host mode as follows:

- In Red Hat Enterprise Linux 6, use:

```
--cpu host
```

- In Red Hat Enterprise Linux 7, use:

```
--cpu host-model
```

Creating a Bootstrap Day0 Configuration for virt-install

This procedure provides additional steps to execute within one of the following procedures, as noted within the procedures:

- Creating the Cisco CSR 1000v VM Using virt-install with qcow2 Image, on page 7
- Creating the Cisco CSR 1000v VM Using virt-install with ISO Image, on page 8

Note The CSR1000v will read the day 0 configuration during initial bootup and will save the configuration after bootup.

In order to bootstrap, perform the following steps on the KVM server.

SUMMARY STEPS

1. Create an **iosxe_config.txt** or **ovf-env.xml** file. (For details, see Bootstrap Properties, on page 13.)
2. Create a disk image from the file using following command:
3. (This step must be performed within the VM creation procedure (see the options indicated above).

DETAILED STEPS

Step 1 Create an **iosxe_config.txt** or **ovf-env.xml** file. (For details, see Bootstrap Properties, on page 13.)

Step 2 Create a disk image from the file using following command:

Example:

```
mkisofs -l -o /my/path/csr_config.iso <configuration_filename>
```

Step 3 (This step must be performed within the VM creation procedure (see the options indicated above).

Add an additional disk parameter to the **virt-install** command to include the csr_config.iso disk image, as follows:

Example:

```
--disk path=/my/path/csr_config.iso,device=cdrom,bus=ide
```

Creating the Cisco CSR 1000v on OpenStack

Selecting a Cisco CSR 1000v Installation Image

There are two different installation image packages available for downloading. Each package contains a different qcow2 file.

- `csr1000v-universalk9.16.03.01a.qcow2`

  —choose an image such as this (for Cisco IOS XE Denali 16.3.1) to use a virtual console. This is recommended for later use with the OpenStack dashboard.

- `csr1000v-universalk9.16.03.01a-serial.qcow2`

  —choose an image such as this (for Cisco IOS XE Denali 16.3.1) to use a serial console to access the VM. This is useful in the lab or if you are using the Cisco Modeling Tool.

Creating the Instance Using the OpenStack Command Line Tool

Although the following procedure provides a general guideline for how to create the Cisco CSR 1000v tenant instance, the exact steps that you need to perform may vary depending on the characteristics of your KVM environment and setup. For more information, see the OpenStack documentation. See "Installation Requirements" sections.

The following steps are performed using the Nova (OpenStack Compute) console on your server.

Step 1 Download the .qcow2 file from the Cisco CSR 1000v software installation image package and copy it onto a local or network device

Step 2 Create the Nova flavor using the **nova flavor-create**command

nova flavor-create *<flavor_name> <flavor_id> <ram size MB> <disk size GB> <num_ vCPUs>*

The disk size should be set to 0 for the Cisco CSR 1000V to boot. The following command example creates a KVM instance with 4096 MB RAM, a disk size of 0 and 2 vCPUs configured:

nova flavor-create csr_flavor 6 4096 0 2

Step 3 Enter the **nova flavor-list** command to verify that the nova flavor created the previous step is available

Step 4 Use the **glance** command, to create the OpenStack image

glance image-create --name <*image_name*> **--disk-format qcow2 --container-format bare --file** <*Location-of-img-file*>

The following example creates an OpenStack image using the Cisco CSR 1000v installation file:

Example:

```
glance image-create --namecsr_image --disk-format qcow2 --container-format bare
  --file /opt/stack/csr/files/images/csr1000v-universalk9.03.12.00.S.154-2.S-std.qcow2
```

Step 5 Use the **nova boot** command, to create the instance and boot

nova boot <*instance_name*> **--image** <*image_id*> **--flavor** <*flavor_id*> **--nic net-id=**<*uuid*> **--config-drive=**<**true**/**false**> **--file**<*configuration_file_name*>

The **--config-drive** option can be used to specify that the configuration is loaded on the Cisco CSR 1000v when it comes up. Set the **--config-drive** option to “true” and specify the name of the configuration file in which you enter the router configuration to be booted. There are two possible formats for the configuration file:

- “ovf-env.xml” (OVF format)
- “iosxe_config.txt”

Note For details , see Bootstrap Properties, on page 13 and subsequent sections.

Note These file names are hard-coded and required for the config-drive settings to boot.

Prior to Cisco IOS XE 3.16S, you could specify only one of the two configuration files in the **nova boot** command. Beginning with Cisco IOS XE 3.16S, and including Cisco IOS XE Denali 16.3.1 and later, you can specify both configuration files in the **nova boot** command line—for example:

Example:

```
nova boot csr-vm-316 --image csr-316 --flavor csr.2vcpu.4gb
  --nic port-id=6773be11-7b95-48cd-b372-fb8a3cae2b50 --config-drive=true
  --file ovf-env.xml=/home/stack/conf_files/ut/ovf-env.xml
   --file iosxe_config.txt=/home/stack/conf_files/ut/iosxe_config.txt
```

Example:

This example shows the booting of the Cisco CSR 1000v image on OpenStack with the “ovf-env.xml” file containing the router configuration:

```
nova boot csr_instance --image csr_image --flavor 6 --nic net-id=546af738-bc0f-43cf-89f2-1e2c747d1764

 --config-drive=true --file ovf-env.xml=/opt/stack/csr/files/ovf-env.xml
```

Example:

The following example boots the Cisco CSR 1000v image on OpenStack with the “iosxe_config.txt” file containing the router configuration:

```
nova boot csr_instance --image csr_image --flavor 6 --nic net-id=546af738-bc0f-43cf-89f2-1e2c747d1764

 --config-drive=true --file iosxe_config.txt=/opt/stack/iosxe_config.txt
```

The Cisco CSR 1000v begins the boot process. See Booting the Cisco CSR 1000v as the VM.

After the OpenStack image is created, you can access the instance on your OpenStack dashboard.

Creating the Instance Using the OpenStack Dashboard

Perform the following steps to create the instance using the OpenStack dashboard.

Note To configure the KVM to run with config-drive, you must select the serial image and use the procedure described in the Creating the Instance Using the OpenStack Command Line Tool , on page 10.

Step 1 Download the .qcow2 file from the Cisco CSR 1000v software installation image package and copy it onto a local or network device.

Step 2 From the OpenStack dashboard, access the OpenStack console.

Step 3 Login as the admin onto the OpenStack console.

Step 4 Create a new flavor using the **Flavor Create** tab on the screen, and specify the *<flavor_name> <flavor_id> <ram size MB> <disk size GB> <num_ vCPUs>*.

See the "Installation Requirements" section required for your version of IOS XE Installing the Cisco CSR 1000v in KVM Environments, on page 1. The disk size should be set to 0 for the Cisco CSR 1000v to boot. as in the tables 6-1 and 6-2.

Select the required flavor from the **System Panel** > **Flavors** tab.

Step 5 Create a new image using the **Image Create** tab on the screen.

Specify the location of the image, the disk format (qcow2) and container-format (raw).

Select the **System Panel** > **Images** tab. The image should show up on the list of images shown on the screen.

Step 6 Create a new instance using the **Instance Create** tab on the screen.

Specify the image, the flavor, and the appropriate network interfaces to be attached to the instance.

Select the **System Panel** > **Instances** tab. The instance should show up on the list of instances shown on the screen, and you should be able to access the console by clicking on the instance name.

Step 7 To launch the instance, select the instance and select **Launch Instance**.

Click the **Details** tab. Review the instance information to ensure it is correct. When you ready to launch the instance, click **Launch**.

The instance is launched and the Cisco CSR 1000v begins the boot process. See Booting the Cisco CSR 1000v as the VM.

Troubleshooting for Creating the Instance using OpenStack

The following issues may occur when creating the instance using the OpenStack Dashboard (see Creating the Instance Using the OpenStack Dashboard, on page 12).

- If you do not see any output on the OpenStack dashboard’s console it may be due to you having previously selected the incorrect type of image. Select the virtual qcow2 image (not the serial qcow2 image) before following the steps in Creating the Instance Using the OpenStack Dashboard, on page 12.

Bootstrapping the CSR Configuration

Bootstrap Properties

The Cisco CSR 1000v bootstrap properties are specified in the **ovf-env.xml** file. For an example **ovf-env.xml** file, see the *Example ovf-env.xml File* section.

Table 1: Bootstrap Properties

Property	Description
console	(Cisco IOS XE 3.17S and later) Configures the console mode. Possible values: auto, virtual, serial
domain-name	Domain name of the router.
enable-scp-server	Enables the IOS SCP feature.
enable-ssh-server	Enables remote login using SSH and disables remote login via Telnet. Requires that the login username and password are set.
hostname	Hostname of the router.

ios-config	Enables execution of a Cisco IOS command.
	To execute multiple commands, use multiple instances of ios-config, with a number appended to each instance—for example, ios-config-1, ios-config-2. The commands are executed in numerical order according to the appended number.
	Example
	`ios-config-1="username cisco priv 15 pass ciscoxyz"`
	`ios-config-2="ip scp server enable"`
	`ios-config-3="ip domain lookup"`
	`ios-config-4="ip domain name cisco.com”`

license	(Cisco IOS XE 3.13S and later) Configures the license technology level that is available when the Cisco CSR 1000v boots.
login-password	Login password for the router.
login-username	Login username for the router.
mgmt-interface	Designates the management interface for the Cisco CSR 1000v. The format must be GigabitEthernetx or GigabitEthernetx.xxx. **Note** Beginning with Cisco IOS XE 3.11S, the GigabitEthernet0 interface is no longer supported.

Property	Description
mgmt-ipv4-addr	Management gateway address/mask in IPv4 format for the GigabitEthernet0 management interface.
mgmt-ipv4-gateway	IPv4 management default gateway address. If using DHCP, enter “dhcp” in the field.
mgmt-ipv4-network	Configures the IPv4 Network (such as “192.168.2.0/24” or “192.168.2.0 255.255.255.0”) that the management gateway should route to. If not specified, the default route (0.0.0.0/0) is used.
mgmt-vlan	Configures the dot1Q VLAN interface. Requires the management interface to be configured using the GigabitEthernetx.xxx format.
pnsc-agent-local-port	(Optional) Configures the Cisco Prime Network Services Controller service agent SSL port on the local Cisco CSR 1000v to receive policies from the service manager. This setting is used if you plan to remotely manage the Cisco CSR 1000v using the Cisco Prime Network Services Controller.
pnsc-ipv4-addr	Configures the IP address of the Cisco Prime Network Services Controller. This setting is used if you plan to remotely manage the Cisco CSR 1000v using the Cisco Prime Network Services Controller.
pnsc-shared-secret-key	Configures the Cisco Prime Network Services Controller shared secret key for the Cisco Prime Network Services Controller agent to set the SSL certificate from the controller. This setting is used if you plan to remotely manage the Cisco CSR 1000v using the Cisco Prime Network Services Controller.
privilege-password	Configures the password for privileged (enable) access.
remote-mgmt-ipv4-addr	(Optional) Configures the IP address used for remote management of the Cisco CSR 1000v by the REST API or by the Cisco Prime Network Services Controller. The address must be in the same subnet as the management interface address. **Note** Beginning with Cisco IOS XE 3.13S, this option is not used if configuring the shared management interface to support REST API.
resource-template	(Cisco 3.16S2 and later) Configures the Resource Template. Possible values: default, service_plane_medium, service_plane_heavy

Example ovf-env.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
   xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
    <PropertySection>
        <Property oe:key="com.cisco.csr1000v.license.1" oe:value="security"/>
        <Property oe:key="com.cisco.csr1000v.console.1" oe:value="serial"/>

<Property oe:key="com.cisco.csr1000v.config-version.1" oe:value="1.0"/>
        <Property oe:key="com.cisco.csr1000v.domain-name.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.enable-scp-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.csr1000v.enable-ssh-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.csr1000v.hostname.1" oe:value="lab"/>
        <Property oe:key="com.cisco.csr1000v.license.1" oe:value="ax"/>
        <Property oe:key="com.cisco.csr1000v.login-password.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.login-username.1" oe:value="lab"/>
       <Property oe:key="com.cisco.csr1000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>

       <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-addr.1" oe:value="172.25.223.251/25"/>

       <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-gateway.1" oe:value="172.25.223.129"/>

        <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-network.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.mgmt-vlan.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-agent-local-port.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-shared-secret-key.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.privilege-password.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.resource-template.1"
oe:value="service_plane_medium"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0001" oe:value="logging buffered
10000"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0002" oe:value="hostname uut-ovf"/>

        <Property oe:key="com.cisco.csr1000v.ios-config-0003" oe:value="ip domain-name
cisco.com"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0004" oe:value="crypto key generate
 rsa modulus 1024"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0005" oe:value="interface
GigabitEthernet2"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0006" oe:value="ip address 10.0.0.5
 255.255.255.0"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0007" oe:value="no shut"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0008" oe:value="exit"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0009" oe:value="ip route 0.0.0.0
0.0.0.0 10.0.0.1"/>
    </PropertySection>
</Environment>
```

Example iosxe_config.txt File

```
hostname ultra-ios_cfg
license smart enable
username lab privilege 15 password lab
ip domain-name cisco.com
crypto key generate rsa modulus 1024
interface GigabitEthernet1
ip address 10.0.0.5 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
 login local
exit
```

Increasing the KVM Configuration Performance

You can increase the performance for a Cisco CSR 1000v running in a KVM environment by changing some settings on the KVM host. These settings are independent of the IOS XE configuration settings on the CSR 1000v instance.

Note In Cisco IOS XE Release 3.13S and earlier, the CSR 1000v instance does not support jumbo packets larger than 1518 bytes for KVM on a Virtio interface. The packets larger than 1518 bytes are dropped.

To improve the KVM configuration performance, Cisco recommends that you:

- Enable vCPU pinning
- Enable emulator pinning
- Enable numa tuning. Ensure that all the vCPUs are pinned to the physical cores on the same socket.
- Set hugepage memory backing
- Use virtio instead of IDE
- Use graphics VNC instead of SPICE
- Remove unused devices USB, tablet etc.
- Disable memballoon

Note These settings might impact the number of VMs that you can be instantiate on a server.

Tuning steps are most impactful for a small number of VMs that you instantiate on a host.

In addition to the above mentioned, do the following:

Enable CPU Pinning

Increase the performance for the KVM environments by using the KVM CPU Affinity option to assign a virtual machine to a specific processor. To use this option, configure CPU pinning on the KVM host.

In the KVM host environment, use the following commands:

- **virsh nodeinfo**: To verify the host topology to find out how many vCPUs are available for pinning by using the following command.
- **virsh capabilities**: To verify the available vCPU numbers.
- **virsh vcpupin** *<vmname> <vcpu#> <host core#>*: To pin the virtual CPUs to sets of processor cores.

  This KVM command must be executed for each vCPU on your Cisco CSR 1000v. The following example pins virtual CPU 1 to host core 3:

  virsh vcpupin csr1000v 1 3

The following example shows the KVM commands needed if you have a Cisco CSR 1000v configuration with four vCPUs and the host has eight cores:

virsh vcpupin csr1000v 0 2

virsh vcpupin csr1000v 1 3

virsh vcpupin csr1000v 2 4

virsh vcpupin csr1000v 3 5

The host core number can be any number from 0 to 7. For more information, see the KVM documentation.

Note When you configure CPU pinning, consider the CPU topology of the host server. If you are using a CSR 1000v instance with multiple cores, do not configure CPU pinning across multiple sockets.

BIOS Settings

Optimize the performance of the KVM configuraiton by applying the recommended BIOS settings as mentioned in the following table:

Configuration	Recommended Setting
Intel Hyper-Threading Technology	Disabled
Number of Enable Cores	ALL
Execute Disable	Enabled
Intel VT	Enabled
Intel VT-D	Enabled
Intel VT-D coherency support	Enabled
Intel VT-D ATS support	Enabled
CPU Performance	High throughput
Hardware Perfetcher	Disabled
Adjacent Cache Line Prefetcher	Disabled
DCU Streamer Prefetch	Disable
Power Technology	Custom
Enhanced Intel Speedstep Technology	Disabled
Intel Turbo Boost Technology	Enabled
Processor Power State C6	Disabled
Processor Power State C1 Enhanced	Disabled

Configuration	Recommended Setting
Frequency Poor Override	Enabled
P-State Coordination	HW_ALL
Energy Performance	Performance

For information about Red Hat Enterprise Linux requirements, see Bootstrap Properties, on page 13 and the subsequent sections.

Host OS Settings

In the host side, Cisco recommends that you use hugepages and enable emulator pinning. The following are some of the recommended settings in the host side:

- Enable IOMMU=pt
- Enable intel_iommu=on
- Enable hugepages
- Use SR-IOV if your system supports it for higher networking performance. Please check SR-IOV limitations your system might have.

In addition to enabling hugepages and emulator pinning, the following settings are also recommended: nmi_watchdog=0 elevator=cfq transparent_hugepage=never

Note If you use Virtio VHOST USER with VPP or OVS-DPDK, you can increase the buffer size to 1024 (rx_queue_size='1024') provided the version of your QEMU supports it.

IO Settings

You can use SR-IOV for better performance. However, note that this might bring in some limitations such as number of virtual functions (VF), OpenStack limitations for SR-IOV like QoS support, live migration and security group support.

If you use a modern vSwitch like fd.io VPP or OVS-DPDK, reserve at least 2 cores for the VPP worker threads or the OVS-DPDK PMD threads.

Configure the following parameters to run the VPP through command line:

- -cpu host: This parameter causes the VM to inherit the host OS flags. You require libvirt 0.9.11 or greater for this to be included in the xml configuration.
- -m 8192: You require 8GB RAM for optimal zero packet drop rates.
- rombar=0: To disable PXE boot delays, set rombar=0 to the end of each device option list or add "<rom bar=off />" to the device xml configuration.

Sample XMLs for KVM Performance Improvement

Sample XML for numa tuning

```
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
```

Sample XML for vCPU and emulator pinning

```
 <cputune>
    <vcpupin vcpu='0' cpuset='3'/>
    <emulatorpin cpuset='3'/>
  </cputune>
```

Sample XML for hugepages

```
 <currentMemory unit='KiB'>4194304</currentMemory>
  <memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB' nodeset='0'/>
    </hugepages>
    <nosharepages/>
  </memoryBacking>
```

Sample XML for virtio instead of IDE

```
<devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/rhel7.0.qcow2'/>
      <backingStore/>
      <target dev='vda' bus='virtio'/>
      <boot order='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </disk>
```

Sample XML for VNC graphics

```
<graphics type='vnc' port='5900' autoport='yes' listen='127.0.0.1' keymap='en-us'>
      <listen type='address' address='127.0.0.1'/>
    </graphics>
```

XML for disabling memballon

```
<memballon model='none'>
```

Cloning the VM

In a KVM environment, cloning the Cisco CSR1000v virtual machine using the **virt-manager** virtual machine manager creates a Cisco CSR1000v virtual machine that may be un-bootable.

The issue is caused by an increase in the size of the cloned image size created by **virt-manager**, compared with the original Cisco CSR1000v VM image. The extra bytes (in the KB range) cause the boot failure.

Workarounds

There are three workarounds:

- Use the **virt-clone** command to clone the Cisco CSR 1000v VM image.
- For a cloned Cisco CSR 1000v VM image created by **virt-manager**, during the bootup, select the GOLDEN image to boot instead of packages.conf.
- In the “Create a new virtual machine” window, deselect “Allocate entire disk now” before the new Cisco CSR1000v VM is created. This ensures that the cloned Cisco CSR1000v VM image is able to boot up. However, this workaround does not support nested cloning. Use this method only on the first cloned Cisco CSR1000v VM image.

Configure the halt_poll_ns Parameter

On newer kernel releases (3.10.0-375.el7 and later), a KVM module parameter - halt_poll_ns has been introduced. You can use this parameter to alter the behaviour of how idle KVM guest virtual CPUs (vcpus) are handled.

When a virtual CPU in a KVM guest has no threads to run, the QEMU traditionally halts the idle CPU. This setting specifies a period of 400 nanoseconds by default, where a virtual CPU waits and polls before entering a CPU Idle state.

When new work arrives during the polling period before the vcpu is halted, the vcpu is immediately ready to execute the work. If the vcpu has been idle when new work arrives, the vcpu must be brought out of the idle state before the new work can be started. The time taken from idle to running state induces additional latency which negatively impacts latency sensitive workloads.

With the default kernel parameters, the guest Cloud Services Router (CSR1000v) CPU consumes 100% of the host CPU.

You can configure halt_poll_ns in two ways:

- Large halt_poll_ns: In this case, more CPU is spent busy-spinning for events that wake the virtual CPU, and less acpi deep sleeps occur. This means more power is consumed. However, there are less wakeups from deep states states, which depending on the state that’s configured, can cause issues like cache misses etc.
- Small halt_poll_ns: In this case, less CPU time is spent busy-spinning for events that wake the CPU, more acpi deep sleeps occur. Here, less power consumed, but more wakeups from deep sleep states are required. More wakeups can cause large amounts of deep sleep instances, which depending on the configuration, can cause large amounts of cache misses and long wakeup time.

Configuring the halt_poll_ns parameter

You can configure the halt_poll_ns parameter in the following ways:

1. At run time, run the following: `echo 0 > /sys/module/kvm/parameters/halt_poll_ns`.
2. When you load the module, perform the following configuration:

```
# rmmod kvm_intel
# rmmod kvm
# modprobe kvm halt_poll_ns=0
# mpdprobe kvm_intel
```

3. When you boot the device, add `kvm.halt_poll_ns=<specify value>` in the parameters section of grub2.

Installation Requirements for KVM—Cisco IOS XE 3.x

This section contains information about VMware requirements for older Cisco IOS XE releases (before IOS XE Denali 16.3.1).

The following table lists the installation requirements for KVM environments. For installation procedures, see Creating the Cisco CSR 1000v VM Using the Self-installing .Run Package, on page 3 and subsequent sections.

Table 2: Installation Requirements for KVM Environments (Cisco IOS XE versions 3.x)

KVM Requirements	Cisco IOS XE Release 3.10S	Cisco IOS XE Release 3.11S	Cisco IOS XE Release 3.12S and 3.13S	Cisco IOS XE Release 3.14S	Cisco IOS XE Release 3.15S, 3.16S, 3.17S
KVM versions supported	• Linux KVM based on Red Hat Enterprise Linux 6.3 (Requires Kernel version 2.6.32 and QEMU 0.12.1.2.) • Red Hat Enterprise Virtualization 3.1	• Linux KVM based on Red Hat Enterprise Linux 6.3 (Requires Kernel version 2.6.32 and QEMU 0.12.1.2.) • Red Hat Enterprise Virtualization 3.1 • Ubuntu 12.04.03 LTS Server 64 Bits(Requires QEMU-x86_64 version 1.0 (qemu-kvm-1.0))	• Linux KVM based on Red Hat Enterprise Linux 6.3 (Requires Kernel version 2.6.32 and QEMU 0.12.1.2.) • Ubuntu 12.04.04 LTS Server 64 Bits 2 (Requires QEMU-x86_64 version 1.0 (qemu-kvm-1.0))	• Linux KVM based on Red Hat Enterprise Linux 6.3 (Requires Kernel version 2.6.32 and QEMU 0.12.1.2.) • Ubuntu 14.04 LTS Server 64 Bits 2 (Requires QEMU-x86_64 version 1.0 (qemu-kvm-1.0))	• Linux KVM based on Red Hat Enterprise Linux 6.6 (Requires Kernel version 2.6.32-504 and QEMU 0.12.1.2.) (RHEL 7.1 (Requires Kernel version 3.10.0 and QEMU 1.5.3) supported in 3.17S) • Ubuntu 14.04 LTS Server 64 Bits 2 (Requires QEMU-x86_64 version 1.0 (qemu-kvm-1.0))

KVM Requirements	Cisco IOS XE Release 3.10S	Cisco IOS XE Release 3.11S	Cisco IOS XE Release 3.12S and 3.13S	Cisco IOS XE Release 3.14S	Cisco IOS XE Release 3.15S, 3.16S, 3.17S
Supported vCPU configurations(The required vCPU configuration depends on the throughput license and technology package installed. See the data sheet for your release for more information.)	4 vCPUs: requires 4 GB RAM minimum allocation	• 1 vCPU: requires minimum 2.5 GB RAM allocation • 2 vCPUs: requires minimum 2.5 GB RAM allocation • 4 vCPUs: requires minimum 4 GB RAM allocation	• 1 vCPU: requires minimum 2.5 GB RAM allocation • 2 vCPUs: requires minimum 2.5 GB RAM allocation • 4 vCPUs: requires minimum 4 GB RAM allocation	• 1 vCPU: requires minimum 4 GB RAM allocation • 2 vCPUs: requires minimum 4 GB RAM allocation • 4 vCPUs: requires minimum 4 GB RAM allocation	• 1 vCPU: requires minimum 4 GB RAM allocation • 2 vCPUs: requires minimum 4 GB RAM allocation • 4 vCPUs: requires minimum 4 GB RAM allocation
Virtual CPU cores required	1	1	1	1	1
Virtual hard disk size	8 GB minimum	8 GB minimum	8 GB minimum (Applies only to creating the VM using the .iso file. If using the .qcow2 file to install in an OpenStack environment, the hard disk size must be set to 0)	8 GB minimum (Applies only to creating the VM using the .iso file. If using the .qcow2 file to install in an OpenStack environment, the hard disk size must be set to 0)	8 GB minimum (Applies only to creating the VM using the .iso file. If using the .qcow2 file to install in an OpenStack environment, the hard disk size must be set to 0)
Supported vNICs	Virtio	Virtio	Virtio	Virtio	Virtio
Maximum number of vNICs supported per VM instance	26	26	26	26	26
Virtual CD/DVD drive installed (applicable only when installing using an .iso file)	Required	Required	Required	Required	Required