



Deploying the Cisco IOS XRv Router

This chapter covers information about deploying the Cisco IOS XRv Router using VMware ESXi and KVM-QEMU hypervisors, and booting the Cisco IOS XRv Router as the VM.

- [Hard-coded Username for Demo-Locked Image, page 1](#)
- [Cisco IOS XRv Router Deployment on VMware ESXi, page 2](#)
- [Cisco IOS XRv Router Deployment on KVM-QEMU, page 3](#)

Hard-coded Username for Demo-Locked Image

As an anti-security feature for a demo-locked image, the Cisco IOS XRv Router provides hard-coded username-password combinations (see table below) that can be used to log in to the VM and administer the VM.

Table 1: Hard-coded username and password

Username	Password
lab	lab
cisco	cisco
root	root
admin	admin



Note

These hard-coded username-password combinations are available only in the demo-locked image. You cannot configure these user names (even at the initial Admin Setup Dialog). Attempting to do so will be rejected with an error message. If such scenerio occurs, the workaround is to pick a different username (other than those listed above) to configure.

Cisco IOS XRv Router Deployment on VMware ESXi

VMware ESXi is an enterprise-level computer virtualization product offered by VMware. It is VMware's enterprise software hypervisor for guest virtual servers that run directly on the host server hardware without requiring an additional underlying operating system. For more information, refer to [VMware documentation](#).

Prerequisites

- Before creating a new VM, upload the Cisco IOS XRv Router VMDK to your ESXi server.
- The Cisco IOS XRv Router is compatible with ESXi server version 5.0 or later.
- The ESXi server supports only E1000 network driver and does not support virtio. Hence, make sure you define all NICs as E1000 before running on ESXi.
- The Firewall options on the host server must be enabled to allow the VM serial port to be connected over the network. To enable Firewall option, select Configuration->Security Profile->Firewall Properties->'VM serial port connected over network' on the host server.

Creating the Cisco IOS XRv Router Virtual Machine Using the vSphere GUI

To create the Cisco IOS XRv Router Virtual Machine using the VMware vSphere, perform these tasks:



Note

The following procedure provides a general guideline for how to deploy the Cisco IOS XRv Router. However, the exact steps that you need to perform may vary depending on the characteristics of your VMware environment and setup.

Before You Begin

Make sure that:

- The vSphere Client is installed on your machine.
- You have set the correct Firewall Options to allow VM Serial port to be connect over network.

-
- Step 1** Download the **iosxrv-5.1.1.tar**, or **iosxrv-k9-5.1.1.tar** software installation image package and extract **iosxrv-5.1.1.ova** or **iosxrv-k9-5.1.1.ova** files. Copy OVF file to the VM Datastore.
- Step 2** In the vSphere client, select **File > Deploy OVF Template**.
- Step 3** Select the Datastore where the Cisco IOS XRv Router .ova file is stored and click **Next**.
- Step 4** Verify OVF template details, and click **Next**.
- Step 5** Specify the name of the VM, select Inventory Location, and click **Next**.
- Step 6** Select a hardware deployment configuration from the drop-down, and click **Next**.
- Note**
- Do not select configuration marked **non-ESXi**.
 - Do not select configurations that provide more than one vCPU in the production environment.

- Step 7** Select the host or cluster on which you want to create the Cisco IOS XRv Router VM. Click **Next**.
- Step 8** Select the datastore for the VM files, and click **Next**.
- Step 9** Select the format in which virtual disks are stored, and click **Next**.
- Step 10** Select the network mapping for each network adapter in the selected configuration, and click **Next**.
Note The Network Mapping window lists network mappings for all possible network adapters. Ignore the adapters that are not applicable for the selected configuration.
- Step 11** Review the VM settings, and click **Finish**.
- Step 12** Select the newly created VM and click **Edit Virtual Machine Settings**.
- Step 13** The **Virtual Machine Properties** window appears. You can add other hardware to the VM.
- Step 14** To add a serial port (Console Port), click **Add** under the Hardware tab.
- Step 15** Select **Serial Port** and click **Next**.
- Step 16** Select **Connect via Network** and click **Next**.
- Step 17** Select **Server** and add a telnet address (of the host) and a port higher than 1024. Click **Next**.
- Step 18** On Ready to Complete screen, click **Finish**.
- Step 19** (Optional) Repeat Step 14 to Step 18 to add another serial port (Auxiliary Port).
- Step 20** Both the serial ports are now added to the VM.
-

What to Do Next

Start the Cisco IOS XRv Router VM and telnet to the serial ports in the terminal. When LR-PLANE-READY DECLARATION is displayed, perform the following tasks:

- On the demo locked and production capable image, you will be prompted to configure a root-system username and password, which can be subsequently used to log in. Once configured, you may log in using hard-coded username and password combinations as well.

Interfaces are up and you can ping the default gateway. Now you are ready to configure the Cisco IOS XRv Router.

Cisco IOS XRv Router Deployment on KVM-QEMU

KVM is a virtualization infrastructure for the Linux kernel. QEMU is a free and open-source software product that performs hardware virtualization. You can run QEMU on a Cisco UCS server with KVM installed. The recommended version of QEMU for the Cisco IOS XRv Router reference platform is version 1.0.

Supported Parameters

The following table lists the parameters that are supported for the Cisco IOS XRv Router Deployment on KVM-QEMU.

Table 2: Supported Parameters for Cisco IOS XRv Router Deployment on KVM-QEMU

Parameter	Description
-nographic	Recommended, as Cisco IOS XRv Router does not support VGA.
-m memory	-m 4096 - Minimum memory supported is 3072 (3 GB), maximum is 8192 (8 GB)
-hda disk-image	Required. Minimum and preferred size is 3 GB. Note Do not start multiple VM instances using the same hard disk image file.
-hdb disk-image-2	This is optional for disk1.
-serial ...	Requires at least 1; can use up to 4 (IOS console + 3 ksh consoles)
-cdrom ...	Supported for configuration from CVAC. Since the CD-ROM is read-only, it is safe for multiple VM instances to share a CD-ROM if desired.
-net ... -net or -netdev ... -device	The -netdev ... -device... syntax is preferred. Note Prior to QEMU 1.4, -netdev socket may fail with the following error: <i>Property 'e1000.netdev' can't find value 'mgmt'</i> . If you are using sockets you must use the -net parameter instead. For other networking modes (e.g., tap) you may use -netdev without issue.

Creating and Modifying a Disk Image

The Cisco IOS XRv Router supports a secondary disk as a means to provide files beyond the base operating system.

To create and modify a disk image perform these tasks:

SUMMARY STEPS

1. Format the disk in MSDOS (FAT32) format.
2. Open the disk image in guestfish.
3. List disk partitions.
4. Mount the desired partition as root filesystem.
5. Add the desired file to the disk.
6. (Optional) Verify the file is present.

DETAILED STEPS

Step 1 Format the disk in MSDOS (FAT32) format.

Example:

```
mkfs.msdos filedisk.vmdk
```

Step 2 Open the disk image in guestfish.

Example:

```
guestfish -a filedisk.vmdk  
run
```

Step 3 List disk partitions.

Example:

```
list-filesystems /dev/vda: vfat
```

Step 4 Mount the desired partition as root filesystem.

Example:

```
mount /dev/vda/
```

Step 5 Add the desired file to the disk.

Example:

```
upload xrvr-4.3.2.08I.pie /xrvr.pie
```

Step 6 (Optional) Verify the file is present.

Example:

```
ls /xrvr.pie
```

Creating Cisco IOS XRv Router Virtual Machine Using QEMU

To create Cisco IOS XRv Router Virtual Machine using QEMU, perform these tasks:

Before You Begin

Make sure you have QEMU emulator version 1.0.

SUMMARY STEPS

1. Download the **iosxrv-5.1.1.tar**, or **iosxrv-k9-5.1.1.tar** software installation image package and extract **iosxrv-5.1.1.vmdk** or **iosxrv-k9-5.1.1.vmdk** files. Copy VMDK file to the VM Datastore.
2. (Optional) Create a disk1.
3. Start QEMU with IOS XRv image.
4. QEMU waits for first serial.
5. Start telnet sessions to the configured serial ports in separate windows.
6. The Cisco IOS XRv Router deployment proceeds.

DETAILED STEPS

Step 1 Download the **iosxrv-5.1.1.tar**, or **iosxrv-k9-5.1.1.tar** software installation image package and extract **iosxrv-5.1.1.vmdk** or **iosxrv-k9-5.1.1.vmdk** files. Copy VMDK file to the VM Datastore.

Step 2 (Optional) Create a disk1.

Example:

```
qemu-img create -f vmdk blank.vmdk 1G
```

Step 3 Start QEMU with IOS XRv image.

Example:

```
qemu-system-x86_64\  
-smp cores=1,sockets=8\  
-nographic\  
-m 8192\  
-hda iosxrv-k9-5.1.1.vmdk\  
-hdb blank.vmdk\  
-serial telnet::13101,server,wait\  
-serial telnet::13102,server,nowait\  
-serial telnet::13103,server,nowait\  
-net nic,model=e1000,vlan=1,macaddr=00:01:00:ff:00:0\
```

Step 4 QEMU waits for first serial.

Step 5 Start telnet sessions to the configured serial ports in separate windows.

Example:

```
telnet localhost 13101 (IOS XRv Console Port)  
telnet localhost 13102 (IOS XRv Auxiliary (AUX) Port)
```

Step 6 The Cisco IOS XRv Router deployment proceeds.

Examples: Cisco IOS XRv Router Deployment on KVM-QEMU

The following examples show the deployment of the Cisco IOS XRv Router on KVM-QEMU for two VMs, two NICs each, back-to-back, socket transport.

Table 3: Cisco IOS XRv Router deployment examples

VM 1	VM 2
<pre> qemu-system-x86_64 \ -nographic \ -m 4096 \ -hda iosxrv-k9-1.vmdk \ -serial telnet::9101,server,nowait \ -serial telnet::9102,server,nowait \ -net socket,listen=localhost:9001,vlan=1 \ -net nic,model=e1000,vlan=1,macaddr=00:01:00:ff:00:00 \ -net socket,listen=localhost:9002,vlan=2 \ -net nic,model=e1000,vlan=2,macaddr=00:01:00:ff:00:01 </pre>	<pre> qemu-system-x86_64 \ -nographic \ -m 4096 \ -hda iosxrv-k9-2.vmdk \ -serial telnet::9111,server,nowait \ -serial telnet::9112,server,nowait \ -net socket,connect=localhost:9001,vlan=1 \ -net nic,model=e1000,vlan=1,macaddr=00:01:00:ff:00:10 \ -net socket,connect=localhost:9002,vlan=2 \ -net nic,model=e1000,vlan=2,macaddr=00:01:00:ff:00:11 </pre>

