



## **Cisco IOS XRv Router Installation and Configuration Guide, Release 4.3.2**

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-30492-01





## CONTENTS

---

### Preface

#### Preface v

Changes to This Document v

Obtaining Documentation and Submitting a Service Request v

---

### CHAPTER 1

#### Cisco IOS XRv Router Overview 1

Cisco IOS XRv Router 1

Features and Usages 2

Benefits of Virtualization Using Cisco IOS XRv Router 2

Software Configuration and Management 3

Cisco IOS XRv Router System Architecture 3

Cisco IOS XRv Router Components 4

Virtual Machine 4

Hypervisor 5

Router Interfaces 5

Server Requirements 6

Cisco IOS XRv Router Licensing 6

Supported Cisco IOS XRv Router Features 7

---

### CHAPTER 2

#### Hypervisor Requirements 9

General Virtual Machine Requirements 9

VMware ESXi Support Information 10

VMware vSphere 10

Supported VMware Features and Operations 10

KVM/QEMU Support Information 10

Supported KVM/QEMU Features and Operations 11

Hypervisor Limitations 11

---

**CHAPTER 3****Preparing for Installation 13**

- Obtaining Cisco IOS XRv Router Software 13
- Cisco IOS XRv Router Installation Package 14
- ROMMON and the Cisco IOS XRv Router 14
- CVAC - Bootstrap Configuration Support 14
  - Building the Bootstrap Configuration File 14

---

**CHAPTER 4****Deploying the Cisco IOS XRv Router 17**

- Deployment Types 17
- Hardcoded Username for Demo-Locked Image 17
- Cisco IOS XRv Router Deployment on VMware ESXi 18
  - Supported Parameters 18
  - Creating the Cisco IOS XRv Router Virtual Machine Using the vSphere GUI 19
- Cisco IOS XRv Router Deployment on KVM/QEMU 21
  - Supported Parameters 21
  - Creating and Modifying Disk Images 22
  - Creating Cisco IOS XRv Router Virtual Machine Using QEMU 23
  - Examples: Cisco IOS XRv Router Deployment on KVM/QEMU 24

---

**CHAPTER 5****Mapping the Cisco IOS XRv Router Network Interfaces to VM Network Interfaces 25**

- Mapping the Cisco IOS XRv Router Network Interfaces to Virtual Network Interface Cards (vNICs) 25
  - Adding and Deleting Network Interfaces on the Cisco IOS XRv Router 26
- Mapping Cisco IOS XRv Router Network Interfaces with vSwitch Interfaces 27

---

**CHAPTER 6****Troubleshooting the Cisco IOS XRv Router 31**

- Verifying the Cisco IOS XRv Router Hardware and VM Requirements 31
- Troubleshooting Network Connectivity Issues 32
- Troubleshooting VM Performance Issues 32
- Troubleshooting Crashes and Stack Traces 33



## Preface

---

This Preface contains these sections:

- [Changes to This Document](#), page v
- [Obtaining Documentation and Submitting a Service Request](#), page v

## Changes to This Document

This table lists the technical changes made to this document since it was first released.

Revision	Date	Summary
OL-30492-01	September 2013	Initial release of this document.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*, at: <http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.





## CHAPTER

# 1

## Cisco IOS XRv Router Overview

---

This chapter covers these topics:

- [Cisco IOS XRv Router](#) , page 1
- [Cisco IOS XRv Router System Architecture](#), page 3
- [Cisco IOS XRv Router Components](#), page 4
- [Cisco IOS XRv Router Licensing](#), page 6
- [Supported Cisco IOS XRv Router Features](#), page 7

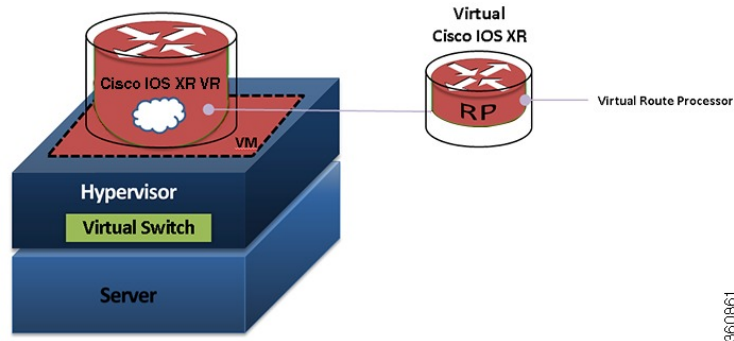
### Cisco IOS XRv Router

The Cisco IOS XRv Router runs the Cisco IOS XR Software and operating system on generic virtualized hardware based on classic (32-bit QNX) Cisco IOS XR Software. It is provided as a virtual machine (VM) supporting full virtualization, and can be deployed on any x86 hardware (server or laptop) running standard hypervisors. It provides the functionality of a route processor and a line card in a combined route processor line card (RPLC) with both the RP control plane functionality, and the network interfaces and associated LC functionality running on the same virtual card.

The Cisco IOS XRv Router is a representation of the Cisco IOS XR Software and operating system, and does not provide a virtual representation of any physical router. As a result, some physical system components,

such as line cards, fabric cards, and multichassis, that are not appropriate for the VM setup, are not present in the Cisco IOS XRv Router system.

**Figure 1: Cisco IOS XRv Router Virtual Form Factor**



## Features and Usages

The Cisco IOS XRv Router provides the features and usages described in this table.

Features	Description
SMUs and PIEs	Provides full support for SMUs and PIEs.
Cisco IOS XR Software Feature Set	Provides support for Cisco IOS XR Software feature set, including the manageability, control plane, routing, and forwarding features.
Multiple CPUs	Supports up to eight CPUs on a single VM. The number of CPUs are configured in the hypervisor and automatically detected by the Cisco IOS XRv Router.
Network Drivers	<p>Supports E1000 and VirtIO drivers to pass the traffic to support a wide array of hypervisors.</p> <p><b>Note</b> The Intel E1000 Driver is a suite of Linux kernel drivers for all Intel Ethernet adapters. Intel E1000 Ethernet chips are provided in most modern hypervisors.</p> <p>VirtIO is an abstraction layer over devices in a paravirtualized hypervisor. It provides an efficient abstraction for hypervisors and a common set of I/O virtualization drivers.</p>

## Benefits of Virtualization Using Cisco IOS XRv Router

The Cisco IOS XRv Router provides these benefits of virtualization in the cloud environment.



Benefits	Description
<b>Hardware independence</b>	The Cisco IOS XRv Router runs on a virtual machine, therefore, can be supported on any x86 hardware supported by the virtualization platform.
<b>Sharing of resources</b>	The resources used by the Cisco IOS XRv Router are managed by the hypervisor, and can be shared among VMs. The amount of hardware resources that the VM server allocates to a specific VM, can be reallocated to another VM on the server.
<b>Flexibility in deployment</b>	You can easily move a VM from one server to another. Thus, you can move the Cisco IOS XRv Router from a server in one physical location to a server in another physical location without moving any hardware resources.

## Software Configuration and Management

You can perform software configuration and management of the Cisco IOS XRv Router using these methods:

- Provision a serial port in the VM and connect to access the Cisco IOS XRv Router CLI commands.
- Use remote SSH/Telnet to connect to the management Ethernet interface to access the Cisco IOS XRv Router CLI commands.

## Cisco IOS XRv Router System Architecture

The Cisco IOS XRv Router is a single VM router that contains the combined functionality of a RPLC card with both the RP control plane functionality, and the network interfaces and associated LC functionality running on the same virtual card.

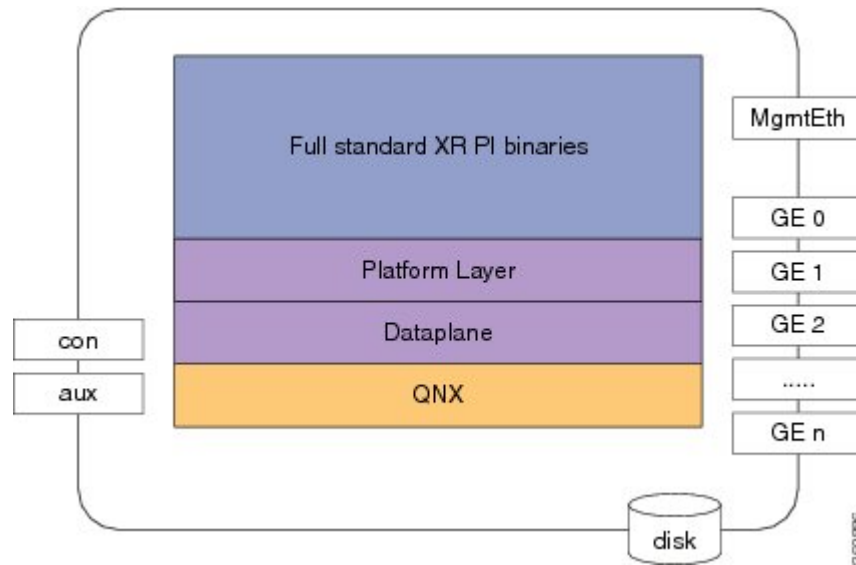
The Cisco IOS XRv Router physical resources are provided to the VM by the hypervisor, such that, from the Cisco IOS XRv Router perspective, it appears to the router as if it is running on a bare metal x86 based machine. These resources are:

- CPU and memory
- Standard PC hardware such as clock, IRQ controller, and PCI bus
- Serial console and auxiliary ports
- Management Ethernet interface
- Network interfaces
- An optional CD-ROM drive which can be mounted by the hypervisor and used to apply bootstrap configuration.

**Note**

The Cisco IOS XRv Router supports disk0 (default) and disk1 (optional additional disk). No support for additional hard disks.

**Figure 2: Cisco IOS XRv Router System Architecture**



The Cisco IOS XRv Router platform image (see Figure) is made up of the these major components:

- XR PI Binaries: The standard platform independent XR packages, built in the same manner as for other XR platforms.
- Platform layer: A virtual platform layer providing the minimal functionality for the PI code to run, including platform services such as node id and chassis management, and various capability and utility libraries.
- Data plane: The software data plane providing a software packet path for XR features, enabling forwarding and a host-stack to the router.
- QNX: The standard XR QNX kernel.

## Cisco IOS XRv Router Components

This section describes the components of the Cisco IOS XRv Router.

### Virtual Machine

A virtual machine (VM) is a software implementation of a computing environment in which an operating system or program can be installed and run. The VM typically emulates a physical computing environment,

but requests for CPU, memory, hard disk, network and other hardware resources are managed by a virtualization layer which translates these requests to the underlying physical hardware.

## Hypervisor

A hypervisor, also called a virtual machine manager (VMM), is a piece of computer software, firmware, or hardware that creates and runs virtual machines.

A hypervisor enables multiple operating systems to share a single hardware host machine. While each operating system appears to have the dedicated use of the host's processor, memory, and other resources; the hypervisor controls and allocates only needed resources to each operating system and ensures that the operating systems (VMs) do not disrupt each other. A computer on which a hypervisor is running one or more virtual machines is defined as a host machine. Each virtual machine is called a guest machine. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems.

The Cisco IOS XRv Router supports these hypervisors:

- **VMware ESXi 5.0 and higher**—VMware ESX and VMware ESXi are both bare-metal embedded hypervisors from VMware's enterprise software for guest virtual servers that run directly on host server hardware without requiring an additional underlying operating system.
- **QEMU 1.0**—Quick EMUlator (QEMU) is a free and open-source software product that performs hardware virtualization. QEMU is a hosted virtual machine monitor. It emulates central processing units through dynamic binary translation and provides a set of device models, enabling it to run a variety of unmodified guest operating systems. It also provides an accelerated mode for supporting a mixture of binary translation (for kernel code) and native execution (for user code), in the same fashion VMware Workstation and VirtualBox do. QEMU can also be used mainly for CPU emulation for user-level processes, allowing applications compiled for one architecture to be run on another.

Kernel-based Virtual Machine (KVM) is virtualization infrastructure for the Linux kernel that QEMU can use to improve performance. KVM requires a processor with hardware virtualization extension.



---

**Note** Ubuntu is the recommended Linux distribution to support KVM/QEMU hypervisors.

---

## Router Interfaces

The Cisco IOS XRv Router interfaces behave in a similar fashion as those on hardware-based Cisco routers. These router interfaces function as follows:

- The supported interfaces are Management Ethernet and Gigabit Ethernet (GE) interfaces.
- Interface port numbering from 0 and up to a maximum of 128 interfaces (including Management Ethernet interface) are supported. The maximum interfaces count depends on the hypervisor used.
- The first interface 0 is reserved for the Management Ethernet interface and subsequent interfaces become the Cisco IOS XRv Router data interfaces.
- The Cisco IOS XRv Router interfaces map to vNIC interfaces on the VM.

For more information, see the [Mapping the Cisco IOS XRv Router Network Interfaces to Virtual Network Interface Cards \(vNICs\)](#), on page 25

## Server Requirements

The Cisco IOS XRv Router can run on Cisco Unified Computing System (UCS) server or servers from leading vendors that support VMWare ESXi 5.0 or the combination of Ubuntu Linux 12.04LTS and QEMU/KVM 1.0. The server must support at least the following:

- Intel Nehalem CPU with clock frequency 2.0 GHz.
- Gigabit Ethernet interfaces.

## Cisco IOS XRv Router Licensing

The license model for Cisco IOS XRv Router includes these images:

- Demo Image (xrvr-full-demo.vmdk)
- Production Image (xrvr-full-prod.vmdk)

License Model	Description
Demo Image	<p><b>Demo Locked</b>—A portable, downloadable virtual machine that is hamstrung to limit its usefulness, but enables a number of internal and external use cases including IOS XR training and familiarization, demonstrations, sales tool, and early field trial (EFT) for control plane features.</p> <ul style="list-style-type: none"> <li>• Available free for users</li> <li>• AAA hardcoded users</li> <li>• Rate Limit of 2 Mbps</li> </ul>

License Model	Description
Production Image	<p><b>Production</b>—Provides a platform for IOS XR based virtual appliances, such as a virtualized route reflector (vRR), or as a network positioning system virtual appliance.</p> <ul style="list-style-type: none"> <li>• No hardcoded AAA users</li> <li>• No Rate Limit</li> </ul>
	<p><b>Simulation</b>—Provides large-scale, high-fidelity control-plane network simulations.</p> <ul style="list-style-type: none"> <li>• No hardcoded AAA users</li> <li>• Rate Limit of 50 Mbps</li> </ul>
	<p><b>Demo Unlocked</b>—A portable, downloadable virtual machine enables a number of internal and external use cases including IOS XR training and familiarization, demonstrations, sales tool, and early field trial (EFT) for control plane features.</p> <ul style="list-style-type: none"> <li>• No hardcoded AAA users</li> <li>• Rate Limit of 2 Mbps</li> <li>• Typically not used, kept as a placeholder when Smart licensing is used</li> </ul>

**Note**

- To move from demo unlocked image to simulation or productions image, configure this command with the xrvr-full-prod.vmdk image:

```
platform mode {simulation | production} accept-eula
```

- To return to demo unlocked image, configure this command:

```
no platform mode
```

## Supported Cisco IOS XRv Router Features

The Cisco IOS XRv Router supports the general IOS XR features as described at a high level in the table below. This list is not intended to be a fully complete list of IOS XR features, but rather a representative presentation of the types of features that are supported in Cisco IOS XRv Router.

**Table 1: Supported Cisco IOS XRv Router Features**

Features	Supported from release
BGP	4.3.2

<b>Features</b>	<b>Supported from release</b>
OSPF	4.3.2
IS-IS	4.3.2
BVI	4.3.2
Syslog	4.3.2



## CHAPTER 2

# Hypervisor Requirements

The Cisco IOS XRv Router supports selected hypervisors for installation. This chapter covers information about virtual machine requirements for hypervisors, VMware ESXi and KVM/QEMU hypervisor's support, and limitations.

- [General Virtual Machine Requirements, page 9](#)
- [VMware ESXi Support Information, page 10](#)
- [KVM/QEMU Support Information, page 10](#)
- [Supported KVM/QEMU Features and Operations, page 11](#)
- [Hypervisor Limitations, page 11](#)

## General Virtual Machine Requirements

Regardless of hypervisor, all Cisco IOS XRv Router virtual machines must meet the following requirements:

Parameter	Minimum	Maximum
Memory (RAM)	3 GB	8 GB
Hard Disk	1 disk, 2 GB	Primary disk must be 2 GB, secondary disk of arbitrary size can be added
CPUs	1 CPU	8 CPUs
Serial Ports	1 serial port (IOS XR console)	4 serial ports (XR console, XR aux port, 2 debugging ports)
NICs	1 NIC	128 (depending on hypervisor's capabilities)

## VMware ESXi Support Information

The Cisco IOS XRv Router runs on the VMware ESXi hypervisor. You can use the same VMware ESXi hypervisor to run several VMs. Use the VMware vSphere Client GUI to create and manage VMs.

**Note**

---

The Cisco IOS XRv Router is compatible with VMware ESXi Server version 5.0 and later.

---

## VMware vSphere

VMs run on the VMware vSphere Hypervisor. You can use the same VMware vSphere hypervisor to run several VMs. Use the VMware vSphere Client GUI to create and manage VMs.

The VMware vSphere Client is an application for creating, configuring and managing VMs on the VMware vCenter Server. The Cisco IOS XRv Router can boot from a virtual disk located on the data store. You can perform basic administration tasks such as starting and stopping the Cisco IOS XRv Router, using the VMware vSphere Client.

VMware vCenter Server manages the vSphere environment and provides unified management of all the hosts and VMs in the data center from a single console.

For more information about how Cisco and VMware work together, see <http://www.vmware.com/cisco>.

## Supported VMware Features and Operations

VMware supports various features and operations that allow you to manage your virtual applications and perform operations.

Below is the list of VMware features and operations that are supported on the Cisco IOS XRv Router:

- Template
- Power On
- Power Off
- vSwitch

For more information about VMware features and operations, see the [VMware Documentation](#).

## KVM/QEMU Support Information

Kernel-based Virtual Machine (KVM) is an open source, full virtualization solution for Linux on x86 hardware containing virtualization extensions. It consists of a loadable kernel module, `kvm.ko` that provides the core virtualization infrastructure and a processor specific module, `kvm-intel.ko` or `kvm-amd.ko`.

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



## Supported KVM/QEMU Features and Operations

Below table lists KVM/QEMU features and operations that are supported/not-supported on the Cisco IOS XRv Router.

Supported	Not-supported
Power on/off	Suspend/Resume
E1000	Snapshots
VirtIO NICs	Cloning

## Hypervisor Limitations

The following are hypervisor limitations for Cisco IOS XRv Router, Release 4.3.2:

- Cisco IOS XRv Router supports an maximum transmission unit (MTU) range of 1500-9216 bytes. However, the maximum MTU supported on your hypervisor version may be lower. The MTU value configured on any interface on Cisco IOS XRv Router should not exceed the maximum MTU value supported on the hypervisor.
- Cisco IOS XRv Router is aware of the emulated virtual NICs (VirtIO/E1000) provided to it by the hypervisor, but is unaware of the underlying physical NICs on the host. This has various implications, including:
  - Cisco IOS XRv Router interfaces default to 1 GB bandwidth, irrespective of the hypervisor's physical NIC(s) bandwidth. The routing protocols (OSPF, EIGRP) use the Cisco IOS XRv Router interface bandwidth values for calculating the costs, not the physical NIC(s) bandwidth.
  - If the physical NIC(s) on the host goes down, this state change may not necessarily propagate to the virtual NICs in Cisco IOS XRv Router, which may remain in an up/up state. This behavior is expected.





## Preparing for Installation

---

This chapter covers information about the prerequisites before installing the Cisco IOS XRv Router.

- [Obtaining Cisco IOS XRv Router Software](#), page 13
- [Cisco IOS XRv Router Installation Package](#), page 14
- [ROMMON and the Cisco IOS XRv Router](#), page 14
- [CVAC - Bootstrap Configuration Support](#), page 14

## Obtaining Cisco IOS XRv Router Software

To obtain the Cisco IOS XRv Router software:

### Before You Begin

Before starting your installation of the Cisco IOS XRv Router, you must first set up your virtual environment, including the necessary host and client software. For example, if you are installing the Cisco IOS XRv Router in a VMware ESXi environment, you must first install the vSphere Client.

For more information, see the vendor documentation for the supported hypervisors.

- 
- Step 1** Go to the product page for Cisco Routers at:  
<http://www.cisco.com/en/US/products/hw/routers/index.html>
- Step 2** Navigate to the Cisco IOS XRv Router product page.
- Step 3** Click the **Download Software** link.
- Step 4** Select the Cisco IOS XR release package and click **Download Now** or **Add to Cart**.  
Follow the instructions for downloading the software.
-

# Cisco IOS XRv Router Installation Package

The following file type is supported in the Cisco IOS XRv Router software image package.

- `.vmdk`—Virtual machine disk image (vmdk) containing Cisco IOS XRv Router kernel and grand unified bootloader (GRUB) configured for turboboot, and package installation.

## ROMMON and the Cisco IOS XRv Router

The Cisco IOS XRv Router does NOT include a ROMMON image similar to the one included in many Cisco hardware-based routers.

**Note**

---

Although the Cisco IOS XRv Router does not include ROMMON, the platform uses a GRUB-based bootloader. However, some Cisco IOS XRv Router commands, such as **show version** may indicate the presence of ROMMON.

---

## CVAC - Bootstrap Configuration Support

Cisco Virtual Appliance Configuration (CVAC) is an out-of-band configuration mechanism supported by Cisco IOS XRv Router. CVAC receives configuration injected into the Cisco IOS XRv Router environment on a CD-ROM or other disk image provided by the hypervisor. The configuration is detected and applied at startup time.

**Note**

---

Real-time configuration is not supported by CVAC.

---

## Building the Bootstrap Configuration File

The Cisco IOS XRv Router supports these plain-text configuration file(s) on a single CD-ROM drive:

- `iosxr_config.txt`—provides standard configuration
- `iosxr_config_admin.txt`—provides admin configuration

These text files provide a simple list of configuration CLIs for CVAC to apply automatically. This operation is functionally equivalent to manually issuing a **copy iosxr\_config.txt running-config** command.

Given a configuration file(s), you can create an ISO image suitable for insertion into Cisco IOS XRv Router with any of the following commands:

**Note**

The mkisofs commands should be executed from a Linux shell.

```
mkisofs -o xrconfig.iso -l --iso-level 2 iosxr_config.txt
mkisofs -o xrconfig.iso -l --iso-level 2 iosxr_config_admin.txt
mkisofs -o xrconfig.iso -l --iso-level 2 iosxr_config.txt iosxr_config_admin.txt
```

If the correct configuration files are provided and CVAC runs successfully, these syslog messages are displayed:

```
RP/0/0/CPU0:Dec 14 09:10:14.560 : config[65699]: %MGBL-CONFIG-6-DB_COMMIT_ADMIN :
Configuration committed by user 'CVAC'. Use 'show configuration commit changes 2000000001'
to view the changes.
RP/0/0/CPU0:Dec 14 09:10:15.330 : cvac[163]: %MGBL-CVAC-5-CONFIG_DONE : Configuration was
applied from file /cd0/iosxr_config_admin.txt.
RP/0/0/CPU0:Dec 14 09:10:22.719 : config[65699]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'CVAC'. Use 'show configuration commit changes 1000000001' to view the
changes.
RP/0/0/CPU0:Dec 14 09:10:23.619 : cvac[163]: %MGBL-CVAC-5-CONFIG_DONE : Configuration was
applied from file /cd0/iosxr_config.txt.
```

If any configuration from the config file(s) is rejected, this additional syslog message is displayed:

```
RP/0/0/CPU0:Dec 14 09:10:23.619 : cvac[163]: %MGBL-CVAC-3-CONFIG_ERROR : Errors were
encountered while applying configs from file /cd0/iosxr_config.txt. Please inspect 'show
configuration failed' for details
```





# 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.

- [Deployment Types, page 17](#)
- [Hardcoded Username for Demo-Locked Image, page 17](#)
- [Cisco IOS XRv Router Deployment on VMware ESXi, page 18](#)
- [Cisco IOS XRv Router Deployment on KVM/QEMU, page 21](#)

## Deployment Types

Like other IOS XR platforms, Cisco IOS XRv Router supports **turboboot** pre-installed hard disks (VMDK).

**Turboboot**—Cisco IOS XRv Router can turboboot from a .vm or .vmdk image. When booting Cisco IOS XRv Router for the first time from a hard disk image, IOS XR undergoes a normal turboboot process that involves expanding the IOS XR software on disk, followed by an automatic reload. After this reboot, IOS XR is fully installed and boots normally from this disk.

## Hardcoded Username for Demo-Locked Image

As an anti-security feature for demo-locked image, Cisco IOS XRv Router provides hard-coded username-password combinations that can be used to login at and administer the VM:

Username	Password
lab	lab
cisco	cisco
root	root
admin	admin

**Note**

These hardcoded 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 appropriate error message. The workaround is simply to pick a different username (other than above listed) to configure instead.

## 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.

## Supported Parameters

While creating a new virtual machine using vSphere, specify these parameters with the recommended settings:

Parameters	Recommendation
Configuration	<b>Custom</b>
Name and Location	as with any other VM
Storage	as with any other VM
Virtual Machine Version	<b>Virtual Machine Version: 8</b>
Guest Operating System	<b>Other</b> , Version: <b>Other (32-bit)</b>
CPUs	1 virtual socket, 1 core per virtual socket
Memory	minimum 4 GB, maximum 8 GB



Parameters	Recommendation
Network	1-4 NICs, each NIC must use "E1000" adapter type. First NIC will be MgmtEthernet0/0/CPU0/0 while subsequent NICs will be GigabitEthernet  <b>Note</b> VMware ESXi only allows up to 4 NICs when initially creating the VM, but you can later add additional NICs as needed.
SCSI Controller	<b>LSI Logic Parallel</b> (default)
Select a Disk	<b>Use an existing virtual disk</b>
Select Existing Disk	select Cisco IOS XRv Router VMDK image
Advanced Options	(default)
Ready to Complete	select <b>Edit the virtual machine settings before completion</b>

## 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

Before you go ahead, 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 **xrvr-full-demo.vmdk** or **xrvr-full-prod.vmdk** file from the Cisco IOS XRv Router software installation image package and copy it to the VM Datastore.
- Step 2** In the vSphere client, first select the host on which you want to create the Cisco IOS XRv Router VM, and then select **Create a New Virtual Machine**.
- Step 3** Select the **Custom** configuration for the VM, and click **Next**.
- Step 4** Specify the name of the VM, select Inventory Location, and click **Next**.
- Step 5** Select the Datastore where the Cisco IOS XRv Router .vmdk file is stored and click **Next**.
- Step 6** Select the VMware version as **Virtual Machine Version: 8**. Click **Next**.

**Note** The Cisco IOS XRv Router is compatible with ESXi Server versions 5.0 and later.

**Step 7** Select the guest operating system to use with this VM as **Other** and version as **Other (32-bit)** from the drop-down menu. Click **Next**.

**Step 8** Select the number of virtual CPUs for the VM.

- Number of virtual sockets (virtual CPUs)
- Number of cores per virtual socket (Currently only 1 core per CPU is supported)

Click **Next**.

**Step 9** Configure the VM's memory size. Click **Next**.

**Note** Supported memory size range is from 2 GB to 8 GB.

**Step 10** Select the number of NICS required and the Network that each interface needs to be connected to. Select the **E1000** adapter. Click **Next**.

**Note** The first NIC added to a Cisco IOS XRv Router VM is mapped to the Management Ethernet interface (MgmtEth0/0/CPU0/0). Additional NICs are mapped to Gigabit Ethernet interfaces such as GigabitEthernet0/0/0/0, GigabitEthernet0/0/0/1, GigabitEthernet0/0/0/2, etc.

**Step 11** Select **LSI Logic Parallel** for the SCSI Controller and click **Next**.

**Step 12** Select **Use an existing virtual disk** option and click **Next**.

**Step 13** Browse to the VMDK file copied to the Datastore as the virtual disk.

**Step 14** Make sure you have selected the correct VMDK image file and click **Next**.

**Step 15** Keep the Advanced Options as is and click **Next**.

**Step 16** On Ready to Complete screen, click the checkbox to **Edit the virtual machine settings before completion**. Click **Continue**.

**Step 17** The **Virtual Machine Properties** window gets opened. You can add other hardware to the VM.

**Step 18** To add a serial port (Console Port), click **Add** under the Hardware tab.

**Step 19** Select **Serial Port** and click **Next**.

**Step 20** Select **Connect via Network** and click **Next**.

**Step 21** Select **Server** and add a telnet address (of the host) and a port higher than 1000. Click **Next**.

**Step 22** On Ready to Complete screen, select **Finish**.

**Step 23** (Optional) Repeat Step 18 to Step 21 to add another serial port (Auxiliary Port).

**Step 24** Both the serial ports are now added to the VM.

---

### What to Do Next

Start Cisco IOS XRv Router VM and telnet to the serial ports in the terminal. Wait for the LR-PLANE-READY DECLARATION and then:

- on the demo locked image, login with hardcoded username/password combinations.
- on the production image, you will be prompted to configure a root-system username and password, which can subsequently use to log in.

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

Kernel-based Virtual Machine (KVM) is a virtualization infrastructure for the Linux kernel. QEMU (Quick EMUlator) 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 Cisco IOS XRv Router reference platform is QEMU 1.0.

### Supported Parameters

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

Parameters	Cisco IOS XRv Router Comments
-nographic	Recommended as Cisco IOS XRv Router does not support VGA.
-m memory	-m 4096 - minimum supported is 3072 (3 GB), maximum is 8192 (8 GB)
-hda disk-image	Required. Minimum/preferred size is 2 GB. <b>Note</b> Do not start multiple VM instances using the same hard disk image file.
-hdb disk-image-2	Optional. For disk1
-serial ...	Requires at least one, can use up to 4 (IOS console + 3 ksh consoles)
-cdrom ...	Supported for configuration from CVAC. As 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. <b>Note</b> Prior to QEMU 1.4, -netdev socket may fail with the following error: Property 'e1000.netdev' can't find value 'mgmt'. 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 Disk Images

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

To create and modify disk images perform these steps:

### 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

Before you go ahead, make sure you have QEMU emulator version 1.0.

### SUMMARY STEPS

1. Download **xrvr-full-demo.vmdk** or **xrvr-full-prod.vmdk** file from Cisco IOS XRv Router software installation image package and copy it 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. Cisco IOS XRv Router proceeds with TURBOBOOT.

### DETAILED STEPS

**Step 1** Download **xrvr-full-demo.vmdk** or **xrvr-full-prod.vmdk** file from Cisco IOS XRv Router software installation image package and copy it 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 xrvr-full-turboboot-prod.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** Cisco IOS XRv Router proceeds with TURBOBOOT.

## Examples: Cisco IOS XRv Router Deployment on KVM/QEMU

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

VM 1	VM 2
<pre>qemu-system-x86_64 \ -nographic \ -m 4096 \ -hda xrvr-full-turboboot-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 xrvr-full-turboboot-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>



## Mapping the Cisco IOS XRv Router Network Interfaces to VM Network Interfaces

---

This chapter covers information about mapping the Cisco IOS XRv Router network interfaces to vNICs, adding and deleting network interfaces, and mapping them with vSwitch interfaces.

- [Mapping the Cisco IOS XRv Router Network Interfaces to Virtual Network Interface Cards \(vNICs\) , page 25](#)
- [Mapping Cisco IOS XRv Router Network Interfaces with vSwitch Interfaces , page 27](#)

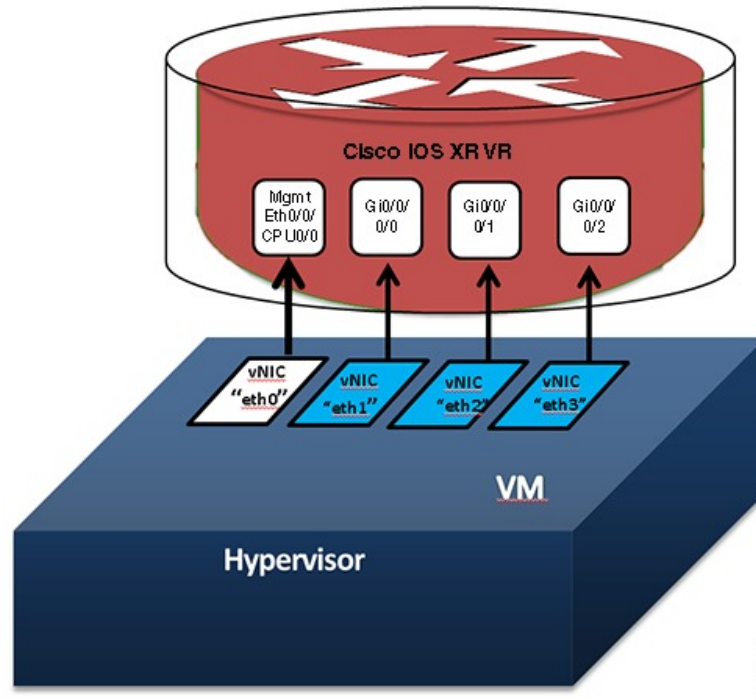
### Mapping the Cisco IOS XRv Router Network Interfaces to Virtual Network Interface Cards (vNICs)

The Cisco IOS XRv Router maps the Ethernet network interfaces to the logical virtual network interface card (vNIC) name assigned by the VM.

When the Cisco IOS XRv Router is booted for the first time, the router interfaces are mapped to the logical vNIC interfaces that were added when the VM was created. Figure 5-1 shows the relationship between the vNICs and the Cisco IOS XRv Router interfaces. The first vNIC added is automatically mapped to the

ManagementEthernet interface port. All subsequent vNICs added are mapped to router interfaces. A maximum of 128 router interfaces (including the ManagementEthernet interface) are supported.

**Figure 3:**



## Adding and Deleting Network Interfaces on the Cisco IOS XRv Router

The Cisco IOS XRv Router maps the router Ethernet interfaces to the logical vNIC name assigned by the VM, which in turn is mapped to a MAC address on the VM host.

To add or delete a vNIC from the VM, you must first power down the VM. If you delete any vNICs, the router must be rebooted. For more information about adding and deleting vNICs, see the [VMware Documentation](#).



### Caution

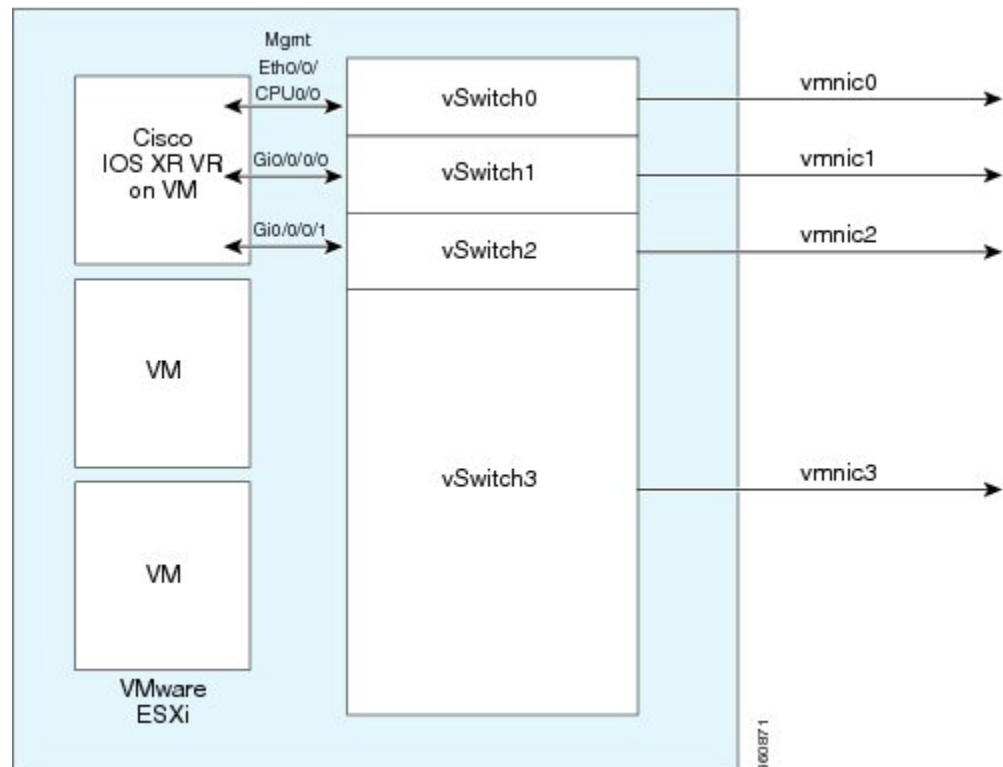
Cisco recommends using caution before removing any existing vNICs on the Cisco IOS XRv Router VM. If you remove a vNIC without first updating the Cisco IOS XRv Router network interface configuration, you risk a configuration mismatch when the router reboots. When the router reboots and a vNIC has been removed, the remaining logical vNIC names could get reassigned to different MAC addresses. As a result, the Ethernet network interfaces on the Cisco IOS XRv Router could get reassigned to different vNICs.



# Mapping Cisco IOS XRv Router Network Interfaces with vSwitch Interfaces

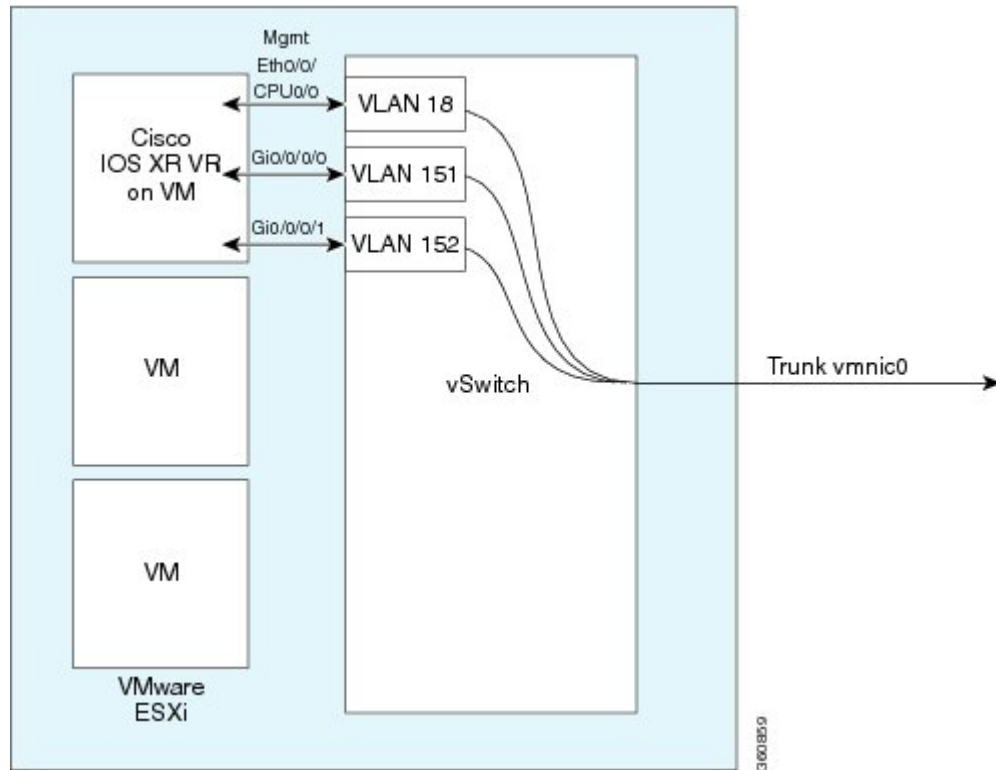
You can configure the network interfaces in ESXi in different ways to accommodate the Cisco IOS XRv Router interfaces. This figure shows an example where each Cisco IOS XRv Router interface is mapped to one host Ethernet interface.

**Figure 4: Cisco IOS XRv Router Interfaces Mapped to Individual ESXi Host Ethernet Interfaces**



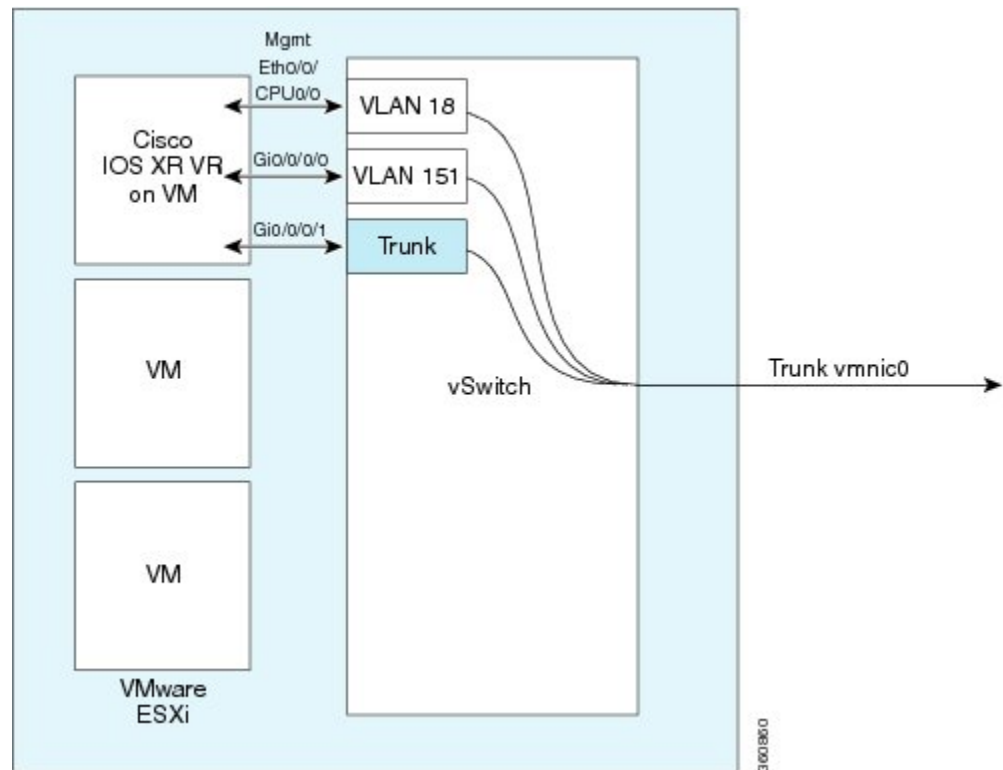
shows an example with multiple Cisco IOS XRv Router interfaces sharing one host ESXi Ethernet interface.

**Figure 5: Cisco IOS XRv Router Interfaces Sharing One ESXi Host Ethernet Interface**



shows one Cisco IOS XRv Router interfaces mapped directly to a trunk interface on the vSwitch.

**Figure 6: Cisco IOS XRv Router Interfaces Directly Mapped to vSwitch Trunk**







## Troubleshooting the Cisco IOS XRv Router

---

This chapter covers information about troubleshooting the Cisco IOS XRv Router.

- [Verifying the Cisco IOS XRv Router Hardware and VM Requirements, page 31](#)
- [Troubleshooting Network Connectivity Issues, page 32](#)
- [Troubleshooting VM Performance Issues, page 32](#)
- [Troubleshooting Crashes and Stack Traces, page 33](#)

### Verifying the Cisco IOS XRv Router Hardware and VM Requirements

To help troubleshoot issues with the Cisco IOS XRv Router, make sure that the router is installed on supported hardware and that the VM requirements are being met:

- Verify that the server hardware is supported by the hypervisor vendor.  
If using VMware, verify that the server is listed on the VMware Hardware Compatibility List. See the VMware documentation for more information.
- Verify that the I/O devices (for example, FC, iSCSI, SAS) being used are supported by the VM vendor.
- Verify that sufficient RAM is allocated on the server for the VMs and the hypervisor host.  
If using VMware, make sure the server has enough RAM to support both the VMs and ESXi.
- Verify the hypervisor version is supported by the Cisco IOS XRv Router.
- Verify that the correct VM settings for the amount of memory, number of CPUs, and disk size are configured.
- Verify that the vNICs are configured using a supported network driver.
- Verify that Cisco IOS XRv Router was configured correctly.

## Troubleshooting Network Connectivity Issues

To troubleshoot network connectivity issues for the Cisco IOS XRv Router, do the following:

- Verify that the vNIC for the VMs are connected to the correct physical NIC, or to the proper vSwitch.
- If using virtual LANS (VLANs), make sure the vSwitch is configured with the correct VLAN.
- If using static MAC addresses, or VMs that are cloned, make sure there are no duplicate MAC addresses.

## Troubleshooting VM Performance Issues

The Cisco IOS XRv Router operates within a set of supported VM parameters and settings to provide certain levels of performance that have been tested by Cisco.

Use vSphere Client to view data to troubleshoot VM performance. If you're using vCenter, you can view historical data. If you're not using vCenter, you can view live data from the host.

To troubleshoot performance issues, perform these steps:

- Verify that the router is configured for the correct MTU setting.

By default, the maximum MTU setting on the router is set to 1500. To support jumbo frames, you need to edit the default VMware vSwitch settings. For more information, see the VMware vSwitch documentation.




---

**Note** ESXi 5.0 supports a maximum MTU size of 9000, even if jumbo frames are enabled on the router.

---

- The Cisco IOS XRv Router does not support memory sharing between VMs. On the ESXi host, check the memory counters to find out how much used memory and shared memory is on the VM. Verify that the balloon and swap used counters are zero.
- If a given VM does not have enough memory to support the Cisco IOS XRv Router, increase the size of the VM's memory. Insufficient memory on the VM or the host can cause the Cisco IOS XRv Router console to hang and be non-responsive.




---

**Note** When troubleshooting performance issues, note that other VMs on the same host as the Cisco IOS XRv Router can impact the performance of the Cisco IOS XRv Router. Verify that other VMs on the host are not causing memory issues that are impacting the Cisco IOS XRv Router VM.

---

- Verify that no network packets are being dropped. On the ESXi host, check the network performance and view the counters to measure the number of receive packets and transmit packets dropped.

For more information about verifying the VM performance indicators, see the [VMware Documentation](#).

# Troubleshooting Crashes and Stack Traces

To troubleshoot crashes and stack traces you may need to contact your Cisco customer support representative for assistance. This section outlines the steps that you should perform prior contacting your next level of support, as this will reduce the amount of time spent resolving the issue.

To gather information before contacting your customer support representative, follow these steps:

## SUMMARY STEPS

1. Save local copies of the crash dump and core files:
2. Present these files to Cisco customer support representative along with a description of the issue.
3. Capture and share these screen shots:
4. If possible, then share a copy of Cisco IOS XRv Router .vmdk file, as this can be used for additional debugging.

## DETAILED STEPS

---

**Step 1** Save local copies of the crash dump and core files:

- a. Access Aux Port through a console.
- b. Enter username and password.
- c. Look for 'dumper' directory, and search for 'first'.

```
# ls /dumper | grep first
first.qnet.20130827-140537.node0_0_CPU0.x86.Z
first.qnet.20130827-140537.node0_0_CPU0.x86.cpu_info.Z
first.qnet.20130827-140537.node0_0_CPU0.x86.txt
```

- d. Copy these files locally.

Either use standard tftpboot or guestfish CLI commands to copy.

```
launch
mount /dev/sda1 /
copy-out /dumper/<filename>
```

**Step 2** Present these files to Cisco customer support representative along with a description of the issue.

**Step 3** Capture and share these screen shots:

- Entire console log window.
- Output of show dll.
- Output of show run.
- Output of show log.

**Step 4** If possible, then share a copy of Cisco IOS XRv Router .vmdk file, as this can be used for additional debugging.

---

