Cisco IOS XRv Router Installation and Configuration Guide

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

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

Preface

This Preface contains these sections:

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	January 2014	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

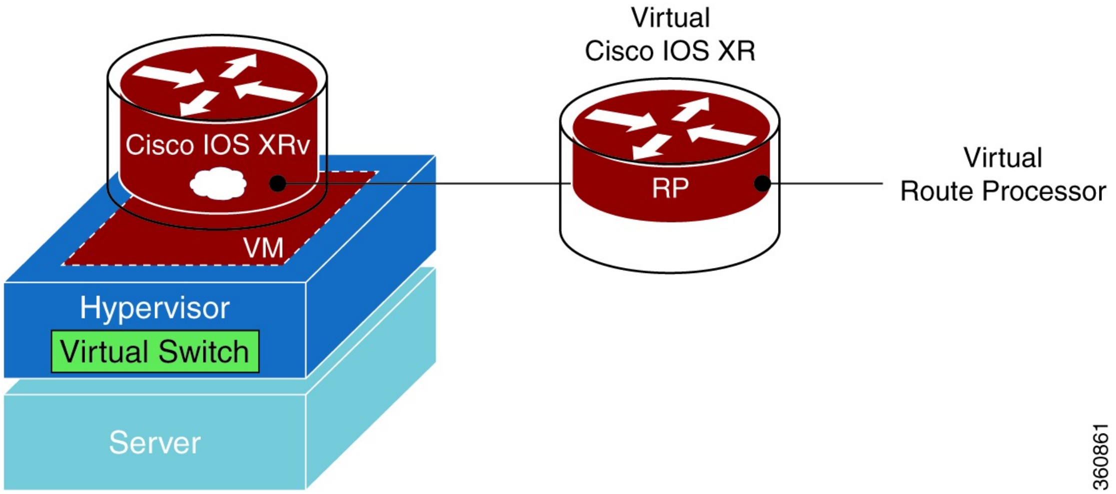
This chapter covers these topics:

Cisco IOS XRv Router

The Cisco IOS XRv Router is a Virtual Machine (VM) based platform running 32-bit IOS XR software with the QNX microkernel. This VM contains a single route processor (RP) with control plane functionality, and line card (LC) network interfaces with their associated functionality. It is a representation of the IOS XR software and operating system, including manageability, control plane features, routing and forwarding functionality. The Cisco IOS XRv Router is not an emulation of any physical router or hardware component.

For more information on installing and configuring the Cisco IOS XRv Router, refer to the *Cisco XR Virtual Router Installation and Configuration Guide*.

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.

Table 1: Cisco IOS XRv Router features and usages

Features	Description
SMUs and PIEs	Provides full support for SMUs and PIEs.
Cisco IOS XR Software Feature Set	Provides support for the 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 in demonstration deployment. Supports single CPU on a single VM in production deployment. The number of CPUs are configured in the hypervisor and automatically detected by the Cisco IOS XRv Router.

Features	Description
Network Drivers	Supports E1000 and VirtIO drivers to pass the traffic to support a wide array of hypervisors. **Note** 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. The Intel E1000 Drive is the only network driver supported for IOS XRv running on ESXi. 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.

Benefits of Virtualization Using the Cisco IOS XRv Router

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

Table 2: Virtualization benefits

Benefits	Description
Hardware independence	The Cisco IOS XRv Router runs on a virtual machine. It can be supported on a x86 hardware that is supported by the virtualization platform.
Resources sharing	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.
Flexibility in deployment	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 manage 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 Secure Shell (SSH) or Telnet to connect to the management Ethernet interface to access the Cisco IOS XRv Router CLI commands.

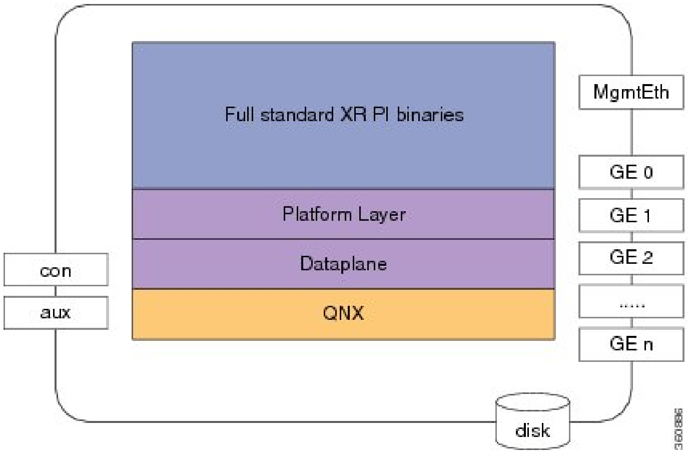
Cisco IOS XRv Router System Architecture

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

- CPU and memory
- Standard PC hardware, such as clock, Interrupt Request Controller (IRQ), and Peripheral Component Interconnect (PCI) bus
- Serial console and auxiliary ports
- Management Ethernet interface
- Network interfaces
- An optional CD-ROM drive that can be mounted by the hypervisor and used to apply bootstrap configuration.

Note The Cisco IOS XRv Router supports disk0 (default, formatted with QNX), harddisk (default, formatted with FAT32), disk1 (optional additional disk, user specified format), boot flash (default, formatted with FAT32) and NVRAM (default, formatted with FAT32).

Figure 2: Cisco IOS XRv Router System Architecture



The Cisco IOS XRv Router platform image (figure above) is made up of these major components:

- Cisco IOS XR PI binaries—The standard platform-independent Cisco IOS XR packages built in the same manner as that for other Cisco IOS 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 Cisco IOS XR features, enabling forwarding and host stack to the router.
- QNX—The standard Cisco IOS XR QNX kernel.

Cisco IOS XRv Router Components

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

Virtual Machine

A 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 VMs.

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 the required 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 VMs is defined as a host machine. Each VM 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 Routeris hypervisor agnostic. The demo image, provided without support, is suitable to deploy on a laptop or on a server that meets the requirements outlined below. For production image, VMWare ESXi 5.0 or later is the only supported hypervisor. For simulation mode, the supported hypervisors are VMWare ESXi 5.0 or later or QEMU 1.0. Additional details regarding VMWare ESXi and QEMU are as follows:

- VMware ESXi 5.0 and later—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. The QEMU 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), same way that VMware Workstation and VirtualBox do. QEMU can also be used 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, which QEMU can use to improve performance. KVM requires a processor with hardware virtualization extension.

Router Interfaces

The Cisco IOS XRv Router interfaces behave the similar way as those on hardware-based Cisco routers. These interfaces function as follows:

- The supported interfaces are Management Ethernet and Gigabit Ethernet 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 virtual network interface card (vNIC) interfaces on the VM.

For more information, see the Mapping the Cisco IOS XRv Router Network Interfaces to vNICs , on page 25

Server Requirements

The Cisco IOS XRv Router can run on the 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 KVM-QEMU 1.0. The server must support at least:

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

Cisco IOS XRv Router Packaging

The Cisco IOS XRv Router packaging includes these images:

- Demo Locked Image (iosxrv-demo-5.1.1.tar)
- Demo Locked Image with Crypto (iosxrv-k9-demo-5.1.1.tar)
- Production Capable Image (iosxrv-5.1.1.tar)
- Production Capable Image with Crypto (iosxrv-k9-5.1.1.tar)

Table 3: Cisco IOS XRv Router images and modes

Images	Modes
Demo Locked Image and Demo Locked Image with Crypto Support	Demo Locked—A portable, downloadable VM that is hamstrung to limit its usefulness, but enables a number of internal and external use cases including Cisco IOS XR training and familiarization, demonstrations, sales tool, and early field trial (EFT) for control plane features. • Available free for users • AAA hardcoded users • Rate limit of 2 Mbps
Production Capable Image and Production Capable Image with Crypto Support	Production—Provides a platform for Cisco IOS XR based virtual appliances, such as a virtualized route reflector (vRR),or as a network positioning system virtual appliance. • No hardcoded AAA users • No rate limit
	Simulation—Provides large-scale, high-fidelity control-plane network simulations. • No hardcoded AAA users • Rate limit of 50 Mbps
	Demo Unlocked—Provides a platform for Cisco IOS XR based virtual appliances, such as a virtualized route reflector (vRR),or as a network positioning system virtual appliance. • No hardcoded AAA users • No rate limit

Note

- You must extract the .ova and .vmdk files from .tar image file and deploy the Cisco IOS XRv Router.
- Use .ova file to deploy the Cisco IOS XRv Router on VMware ESXi.
- Use .vmdk file to deploy the Cisco IOS XRv Router on QEMU.
- After deployment the Cisco IOS XRv Router will default to demo mode. To move from demo mode to production or simulation mode, configure this command with the iosxrv-5.1.1.vmdk or iosxrv-5.1.1.ova image:

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

- To return to demo, configure this command:

  ```
  no platform mode
  ```

Supported Cisco IOS XRv Router Features

Cisco IOS XR Software running on the Cisco IOS XRv Router provides the following features:

- IP features—Supports a wide range of IPv4 and IPv6 services and routing protocols such as IPv4 unicast services, IPv6 unicast services, IPv4 Multicast services, IPv4 and IPv6 equal-cost multipathing (ECMP), IPv4 and IPv6 load balancing, Cisco Discovery Protocol, IPv4 and IPv6 addressing, and Internet Control Message Protocol (ICMP).
- Layer 3 routing protocols—Supports routing protocols such as Border Gateway Protocol Version 4 (BGPv4), Open Shortest Path First Version 2 (OSPFv2) and Version 3 (OSPFv3), and Intermediate System-to-Intermediate System (IS-IS) Protocol.
- Multiprotocol Label Switching (MPLS) features—Supports MPLS features such as MPLS Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP), Diffserv Aware Traffic Engineering (TE), MPLS Traffic Engineering control plane (RFCs 2702 and 2430), MPLS forwarding and MPLS load balancing.
- Network Management—Supports features like Enhanced CLI, XML interface and Simple Network Management Protocol (SNMP) support.
- Software Maintenance Update (SMU)— Supports applying fixes for software defects between releases.

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

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

Table 4: VM parameters

Parameter	Minimum	Maximum
Memory (RAM)	3 GB	8 GB
Hard Disk	1 disk, 3 GB	Primary disk must be 3 GB, secondary disk of arbitrary size can be added.
CPUs	1 CPU	8 CPUs in demonstration deployment. 1 CPU in production deployment.
Serial Ports	1 serial port (IOS XR console)	4 serial ports (XR console, XR aux port, 2 debugging ports)

Parameter	Minimum	Maximum
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
- E1000 NIC

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.

Quick EMUlator (QEMU) 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 the Cisco IOS XRv Router reference platform is version 1.0.

Supported KVM-QEMU Features and Operations

This table lists KVM-QEMU features and operations that are supported and non-supported on the Cisco IOS XRv Router.

Table 5: Supported KVM-QEMU features and operations

Supported	Non-supported
Power on/off	Suspend/Resume
E1000 NICs	Snapshots
VirtIO NICs	Cloning

Hypervisor Limitations

The following are hypervisor limitations for the Cisco IOS XRv Router:

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

CHAPTER 3

Preparing for Installation

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

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 software installed to disk. The VMDK image is recommended for deploying Cisco IOS XRv Router on the QEMU hypervisor.
- ova—Open Virtualization Appliance (OVA) containing a compressed .vmdk and virtual machine hardware definition (CPUs, RAM, NICs, etc). The OVA image is recommended for deploying Cisco IOS XRv Router on the VMware hypervisor.

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 The Cisco IOS XRv Router does not include ROMMON; instead, 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
```

CHAPTER 4

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

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 6: 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 7: 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: *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 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 8: Cisco IOS XRv Router deployment examples

VM 1	VM 2
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	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

CHAPTER 5

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

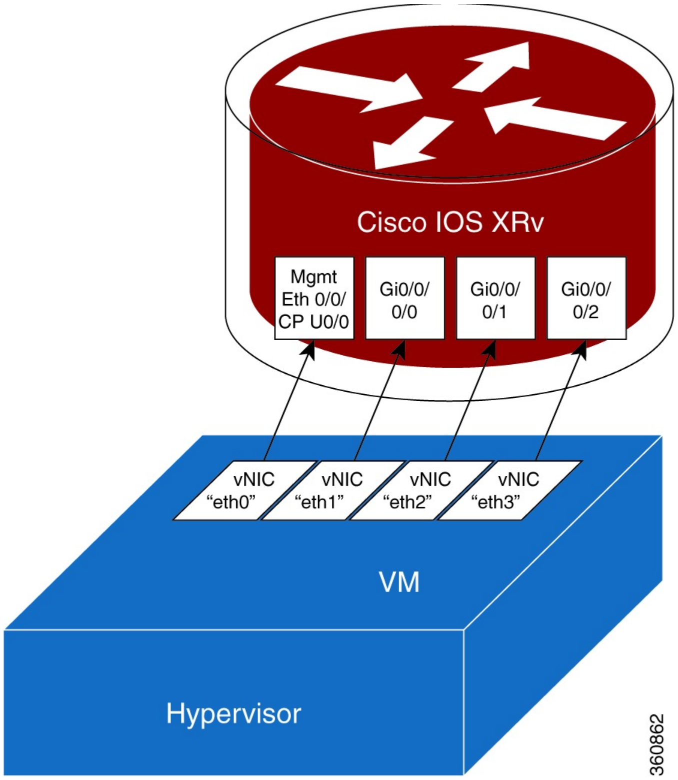
This chapter provides information about mapping the Cisco IOS XRv Router network interfaces to virtual network interface cards (vNICs), adding and deleting network interfaces, and mapping them to vSwitch interfaces.

Mapping the Cisco IOS XRv Router Network Interfaces to vNICs

The Cisco IOS XRv Router maps the Ethernet network interfaces to the logical 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. The following figure shows the relationship between the vNICs and the Cisco IOS XRv Router interfaces. The first vNIC that is added is automatically mapped to the Management Ethernet interface port. All the subsequent vNICs added are mapped to the router interfaces. A maximum of 128 router interfaces (including the Management Ethernet interface) are supported.

Figure 3: Cisco IOS XRv Router interfaces and vNICs



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.

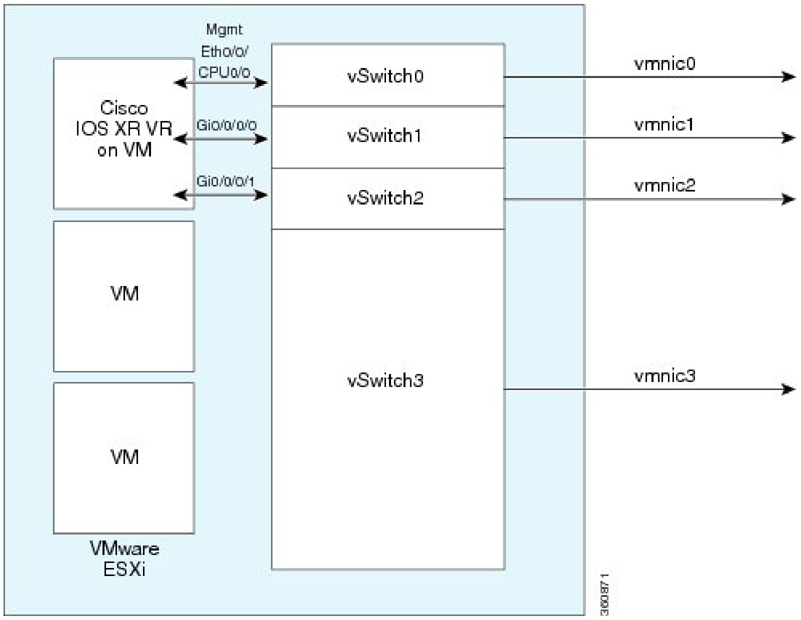
To add or delete a vNIC from the VM, you must first power down the VM. If you delete any vNICs, the router must be rebooted. For more information about adding and deleting vNICs, see the VMware Documentation.

Caution We recommend that you exercise 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 is removed, the remaining logical vNIC names may get reassigned to different MAC addresses. As a result, the Ethernet network interfaces on the Cisco IOS XRv Router may get reassigned to different vNICs.

Mapping Cisco IOS XRv Router Network Interfaces with vSwitch Interfaces

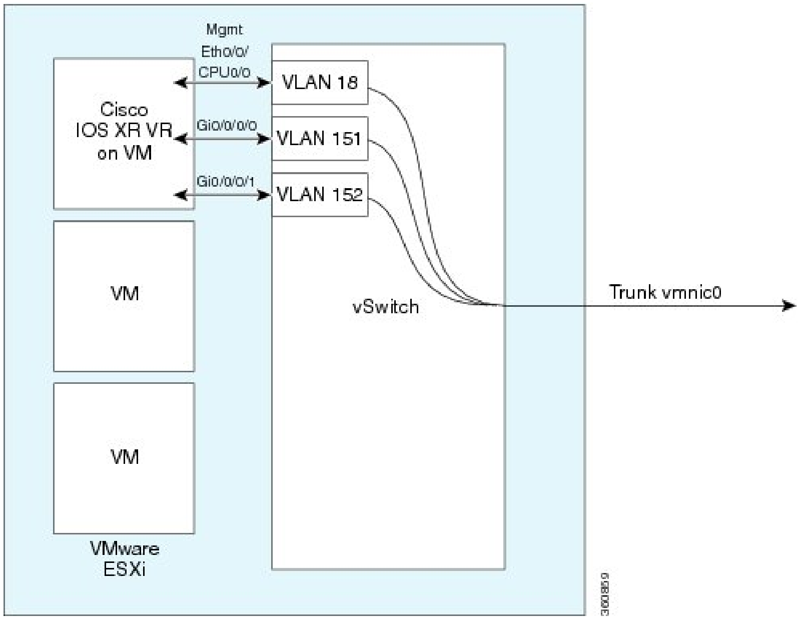
You can configure the network interfaces in the VMware ESXi in different ways to accommodate the Cisco IOS XRv Router interfaces. This figure shows a Cisco IOS XRv Router interface being mapped to a host Ethernet interface.

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



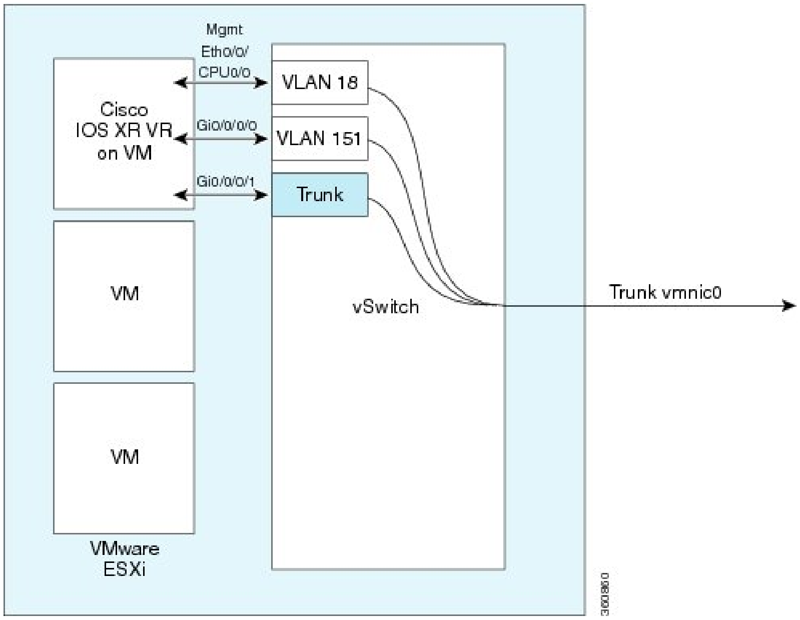
This figure shows multiple Cisco IOS XRv Router interfaces sharing one host VMware ESXi Ethernet interface.

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



The figure below shows an interface of a Cisco IOS XRv Router being mapped directly to a trunk interface on the vSwitch trunk.

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



CHAPTER 6

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

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 for VMware EXSi

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 tftp 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.