Cisco Nexus 9000v Guide

First Published: 2016-10-31

Last Modified: 2018-02-12

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

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

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/c/en/us/td/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 delivers content directly to your desktop using a reader application. The RSS feeds are a free service.

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000v Guide*.

Note The guide has been renamed to reflect the new product name, *Cisco Nexus 9000v Guide*.

New and Changed Information

This table summarizes the new and changed features for the Cisco Nexus 9000v Guide.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
128 interfaces on KVM Hypervisor	Added support for 128 interfaces on KVM Hypervisor.	7.0(3)I7(3)	KVM-QEMU Support Information, on page 13
Vagrant Process	Added instructions for the new streamlined Vagrant up process.	7.0(3)I7(3)	Deploying Cisco Nexus 9000v on VirtualBox with Vagrant Using a Pre-Packaged Box, on page 23
SNMP Management	Added support for the Cisco Nexus 9000v platform.	7.0(3)I7(1)	Cisco Nexus 9000v Guidelines and Limitations, on page 4
CML	Added support for the Cisco Nexus 9000v platform.	7.0(3)I6(1)	Cisco Nexus 9000v Guidelines and Limitations, on page 4

Feature	Description	Changed in Release	Where Documented
VIRL	Added support for the Cisco Nexus 9000v platform.	7.0(3)I6(1)	Cisco Nexus 9000v Guidelines and Limitations, on page 4
VXLAN BGP EVPN	Added support for the Cisco Nexus 9000v platform.	7.0(3)I6(1)	Cisco Nexus 9000v Guidelines and Limitations, on page 4 Troubleshooting VXLAN BGP EVPN, on page 40
CVAC configuration	Added support for the Cisco Nexus 9000v platform.	7.0(3)I5(2)	Cisco Nexus 9000v Guidelines and Limitations, on page 4
KVM/QEMU hypervisor	Added support for KVM/QEMU environment networking.	7.0(3)I5(2)	KVM or QEMU Environment Networking, on page 18
Cisco Nexus 9000v	Increased support from 9 to 64 data port interfaces.	7.0(3)I5(2)	Cisco Nexus 9000v Guidelines and Limitations, on page 4
Cisco Nexus 9000v	Added vPC support	7.0(3)I5(2)	
Cisco Nexus 9000v	Initial support.	7.0(3)I5(1)	

CHAPTER 2

Cisco Nexus 9000v

This chapter contains the following sections:

About Cisco Nexus 9000v

The Cisco Nexus 9000v is a virtual platform that is designed to simulate the control plane aspects of a network element running Cisco Nexus 9000 software. The Cisco Nexus 9000v shares the same software image running on Cisco Nexus 9000 hardware platform although no specific hardware emulation is implemented. When the software runs as a virtual machine, line card (LC) ASIC provisioning or any interaction from the control plane to hardware ASIC is handled by the Cisco Nexus 9000v software data plane.

The Cisco Nexus 9000v for the Cisco Nexus 9000 Series provides a useful tool to enable the devops model and rapidly test changes to the infrastructure or to infrastructure automation tools. This enables customers to validate configuration changes on a simulated network prior to applying them on a production network. Some users have also expressed interest in using the simulation system for feature test ,verification, and automation tooling development and test simulation prior to deployment. Cisco Nexus 9000v can be used as a programmability vehicle to validate software defined networks (SDNs) and Network Function Virtualization (NFV) based solutions.

Cisco Nexus 9000v Guidelines and Limitations

Cisco Nexus 9000v has the following guidelines and limitations:

- Cisco Nexus 9000v does not support the VGA console. You must provision a serial console on a VM to access the Nexus 9000v switch prompt on initial boot. See Deploying the Cisco Nexus 9000v on VirtualBox, on page 18 for more information.
- When N9000v VMs are created by KVM hypervisor, the following issues may occur due to the default setting on the Linux Bridge:
    - LLDP communication between the VMs: The LLDP communication is not established between N9000v. For the solution, the following Linux Bridge settings should be configured. (In the example, assume vb7af2d7ab777d0 is the Linux Bridge that is used for connecting two VMs.

        1. Stop STP running on the Linux Bridge using the **brctl setageing vb7af2d7ab777d0 0** command.
        2. Allow LLDP to be forwarded on the Linux Bridge using the **echo 0x4000 > /sys/class/net/vb7af2d7ab777d0/bridge/group_fwd_mask** command.
        3. Stop LLDP service running on Linux base host (on which the topology is running) using the **/etc/init.d/lldpd stop** command.
        4. [Optional] Disable multicast snooping using the **echo 0 > /sys/devices/virtual/net/vb7af2d7ab777d0/bridge/multicast_snooping** command.

    - LACP connection between the VMs: The LACP connection is not formed between eNXOSv. For the solution, complete the following steps:
        - The Linux kernel should be patched.
        - Group forward mask should be set up using the **echo 0x4 > /sys/class/net/vb7af2d7ab777d0/bridge/group_fwd_mask** command.
    - The multicast packet may not flow through the Linux Bridge. For the solution, use the **echo 0 > /sys/devices/virtual/net/vb7af2d7ab777d0/bridge/multicast_snooping** command.
    - Some ports may get into STP blocked port by the Linux Bridge. For the solution, disable the STP running on the Linux Bridge using the **brctl setageing vb7af2d7ab777d0 0** command.

- After initial setup of the Cisco Nexus 9000v, you must configure the booting image in your system. Otherwise, the Cisco Nexus 9000v drops to the `loader>` prompt after reload/shut down.

```
switch# configure terminal
switch(config)#  boot nxos n9000-dk9.7.0.3.I2.0.454.bin
switch(config)#  copy running-config startup-config
```

- Cisco Nexus 9000v does not support VGA console. You must provision the serial console on any VM to access the Cisco Nexus 9000v switch prompt on initial boot.

- Beginning with Cisco Nexus Release 7.0(3)I7(1), you can use the Cisco Nexus Network Manager to manage Nexus 9000v chassis node in network management system

  Note The Cisco Nexus 9000v does not support statistics querying.

- The Cisco Nexus 9000v uses vNICs that are entered from the KVM/QEMU command line or from the GUI on ESXi for networking either externally or internally within a hypervisor server. The first NIC is always used as the Cisco Nexus 9000v management interface. The subsequence NICs are used as data ports as e1/1, e1/2, ... e1/9. Ten NICs are allowed with nine NICs for data ports.

  Note Beginning with Cisco NX-OS Release 7.0(3)I5(2), a maximum of 64 data ports (e1/1, e1/2, ... e1/64) are supported.

  Connect only the first NIC for the Cisco Nexus 9000v VM as the management interface to your LAN physical switch or vSwitch (VM Network) connecting directly to a physical switch. Do not connect any data port vNIC to any physical switch that conflicts with your server management connectivity.

- Cisco Nexus 9000v only supports the ESXi standard vSwitch when VMs are interconnected within a hypervisor or an external physical switch.

- The vSwitch mapping to data port interface is required to have Promiscuous Mode as the Accept mode in order to pass traffic between VMs.

- Multicast snooping should be disabled on net-devices configured as bridges on Linux-host machines. These net-devices are used to connect VMs in the Cisco Nexus 9000v setup.

- The Cisco Nexus 9000v operates as a bridge that generates BPDU packets on its Ethernet interfaces as it participates in Spanning Tree Protocol (STP). It also forwards broadcast, unknown unicast, and multicast traffic as expected by classic bridging logic. Do not connect the Cisco Nexus 9000v data plane interfaces to the upstream network in a manner that would create bridging loops or interfere with upstream STP operation.

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), Cisco Nexus 9000v is supported in the Virtual Internet Routing Lab (VIRL) and the Cisco Modeling Lab (CML) environment running as a VM.

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), VXLAN BGP EVPN is supported on Cisco Nexus 9000v. For details on VXLAN configuration, see the Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x.

Benefits of Virtualization Using the Cisco Nexus 9000v

This virtual platform provides these virtualization benefits in a cloud environment and you are not limited to the type of hardware as well as other resources.

Benefits	Description
Hardware Independence	This virtual platform provides these virtualization benefits in a cloud environment and users is not limited to hardware as well as other resources. **Note** The minimum RAM/memory requirement for an Cisco Nexus 9000v based VM is 5GB
Resource Sharing	The resources used by Cisco Nexus 9000v are managed by the hypervisor, and can be shared among VMs. The amount of hardware resources that VM sever 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 Nexus 9000v from a server in one physical location to a server in another physical location without moving any hardware resources.
Dynamic Networking	Users can change network connectivity and configuration in a matter of mins without any physical cabling.

Cisco Nexus 9000v Software Functionality

NX-OS 7.0(3)I6(1) Release and Earlier Releases

Beginning Cisco NX-OS 7.0(3)I5(1), Cisco Nexus 9000v supports emulation and implementation on a subset of hardware dependent features. Enablement of such features in the forwarding plane allows users to simulate a more realistic deployment scenario such as an NX-OS hardware platform.

The following are restrictions and host environment differences for Cisco Nexus 9000v:

- Interface throughputs are currently rate-limited to prevent any production usage. The interface rate-limit is set as 4M per device.
- For the Oracle VM VirtualBox hypervisor, the SATA block device must be used due to the image size increase from the 7.0(3)I5(1) branch which impacts the legacy bios IDE controller size limit. SATA block devices also significantly improve disk performance, especially for initial boot up time. The IDE controller can be used in the VMware ESXi and the KVM/QEMU hypervisor, but it is significantly slower than the SATA controller.
- A SATA block device can also be used in KVM/QEMU hyerpvisor, but it requires QEMU 2.2.0 or later version. Similar to the Oracle VM VirtualBox hypervisor, a performance improvement can be obtained in the KVM/QEMU environment.

Supported Features

The following table displays specific Layer 2 and Layer 3 software feature support based on branch/lineup.

Table 2: Supported Layer 2 and Layer 3 Features (Software)

Technology Branch/Lineup	Nexus Feature Name	Support Statement Cisco NX-OS 7.0(3)I5(1), 7.0(3)I5(2), and 7.0(3)I6(1) Releases
OS Infra	Bash Shell	Supported
	Guest Shell	Supported
	SSH	Supported
	RPM Installation	Supported
	POAP	Supported for the management interface in Cisco NX-OS Release 7.0(3)I5(1) and for all interfaces in Cisco NX-OS Release 7.0(3)I5(2) and Cisco NX-OS Release 7.0(3)I6(1)
Programmability	NXAPI	Supported
	Puppet Integration (Native)	Supported
	Puppet Integration (Guest Shell)	Supported
	Chef Integration (Native)	Supported
	Chef Integration (Guest Shell)	Supported
L3 Features	CDP	Supported
	LLDP	Supported
	BGP v4	Supported (No BFD, EVPN) **Note** VXLAN EVPN is supported in Release7.0(3)I6(1) and later.
	BGP v6	Supported (No BFD, EVPN)
	OSPFv2	Supported (No BFD, EVPN)
	OSPFv3	Supported (No BFD, EVPN)
	EIGRP	Supported
	RIP	Supported

Technology Branch/Lineup	Nexus Feature Name	Support Statement Cisco NX-OS 7.0(3)I5(1), 7.0(3)I5(2), and 7.0(3)I6(1) Releases
L2 Features	L2 Switching Unicast	Supported
	L2 Switching Broadcast	Supported
	L2 Switching Multicast	Supported as Broadcast (not explicit Mcast) , No PIM or Mcast Group support
	ARP Suppression	Supported in Cisco NX-OS Release 7.0(3)I7(1) and later.
	MAC learning	Supported
	Static/Router MAC	Supported
	Switchport	Supported
	802.1q VLAN Trunk/Access	Supported
	STP	Supported
	L3 SVI	Supported
	Subinterfaces	Supported
	VXLAN and VXLAN EVPN	Supported (Flood and Learn); BGP EVPN without ARP suppression in Cisco NX-OS Release 7.0(3)I6(1) and later. BGP EVPN with ARP suppression in Cisco NX-OS Release 7.0(3)I7(1) and later.
	vPC	Not supported in Cisco NX-OS Release 7.0(3)I5(1), but supported in Cisco NX-OS Release 7.0(3)I5(2) and Cisco NX-OS Release 7.0(3)I6(1)
	Port channel	Supported

Note The Cisco Nexus 9000v features in this table have been verified to operate only with the Cisco devices mentioned in this document.

If a networking or system feature is not identified as a supported feature in this document, it should be considered as unsupported despite that it may seem to work correctly. Unsupported features did not have any level of regression testing on Cisco Nexus 9000v.

Table 3: NX-OS Features Not Supported (Not Tested)

NX-OS Features	Limitations
QoS	Not supported on Cisco Nexus 9000v.
BFD	Not supported on Cisco Nexus 9000v.
ACL	Not supported on Cisco Nexus 9000v.
Policy maps	Not supported on Cisco Nexus 9000v.
SPAN	Not supported on Cisco Nexus 9000v.
IGMP Snooping	Not supported on Cisco Nexus 9000v.
AMT	Not supported on Cisco Nexus 9000v.
LISP	Not supported on Cisco Nexus 9000v.
OTV	Not supported on Cisco Nexus 9000v.
EVPN Multi-Site	Not supported on Cisco Nexus 9000v.

The following list (not comprehensive) contains known system limitations.

Table 4: NX-OS System Limitations

System Capabilities	Limitations
MAC Address	Cisco Nexus 9000v does not integrate the L2FM module and L2FDWR data plane. It maintains its own MAC Table. Therefore the behavior of the MAC address related CLIs will be different from the physical platform.
Statistics	Cisco Nexus 9000v re-uses a software data plane that was created for L2FDWR. With this data plane, Cisco Nexus 9000v will not be able to account for and display traffic related statistics (such as interface statistics).
Consistency Checker	The consistency checker has a hardware dependency and hence is not supported on Cisco Nexus 9000v. All ‘show’ and ‘exec’ commands will result with appropriate error/warnings.
Network Throughput	Low data plane performance. Additional rate limiter is in place to limit the total amount of traffic received by Cisco Nexus 9000v to 4M.
TOR-ISSU	TOR-ISSU is not supported.

System Capabilities	Limitations
Link Status	Cisco Nexus 9000v virtual interfaces serve as the ‘Ethernet Ports’. The link status of these links within the NX-OS is dependent on the Hypervisor’s capability.
Link-down	Connectivity between the two ends of the interface link is simulated, hence it is important that you shut the interface in both the ends, followed by no shut at both the ends of the interface link.

Cisco Nexus 9000v Feature UI/CLI Difference From Hardware Platform

Feature enablement in the Cisco Nexus 9000v virtual platform is the same as Cisco Nexus 9000 hardware platform.

For example, the following features can be enabled:

- **feature telnet**
- **feature bash-shell**
- **feature ospf**
- **feature bgp**
- **feature interface-vlan**
- **feature nv overlay**

However, not all commands are available for Cisco Nexus 9000v, such as hardware data plane specific commands. Some of these commands exist in the command parse chain, but these commands might not display correct output information. It is not possible for the virtual platform to verify all commands on Cisco Nexus 9000v that exist for the Cisco Nexus 9000 hardware platform.

A few commands are critical for Cisco Nexus 9000v to display Layer 2/Layer 3 information, but are not provided for the Cisco Nexus 9000v platform. The following displays substitute commands:

NX-OS Hardware Platform Commands	Substitute for Cisco Nexus 9000v
show mac address-table	**show system internal l2fwder mac**
clear mac address-table	**clear mac address-table datapath static|dynamic**

Cisco Nexus 9000v System Management Configuration

Beginning Cisco NX-OS 7.0(3)I7(1), Cisco Nexus 9000v runs the same software as Nexus 9000 Series TOR hardware platform in aspect of control plane. All applicable CLIs should be the same as hardware platform. The Simple Network Management Protocol (SNMP) for Nexus 9000v chassis management is added in this release. Nexus 9000v SNMP software inherent basic SNMP infrastructure from Nexus 9000 Series hardware platform. System management configuration should follow Cisco Nexus 9000 series documentation. However, management entity is subject to Nexus 9000v platform specific limitation. For example, interfaces statistics will not be available for any management request because Nexus 9000v platform does not have such data available. For details about supported features, see Cisco Nexus 9000v Software Functionality, on page 6.

Cisco Nexus 9000v SNMP chassis management supports the following entity MIBs. However, only applicable and meaningful attributes can be retrieved from this platform.

- CISCO entity Asset MIB
- ceEXTEntityLEDTable
- ciscoEntityExtMIB
- ciscoRFMIB
- ciscoTSMIB
- ciscoEntityFRUControlMIB
- ciscoSyslogMIB

Cisco Nexus 9000v Resource Requirements

The Cisco Nexus 9000v uses the Cisco Nexus 9000 Series hardware software image. It requires the minimum resources as shown in the following list. These resources are generally not oversubscribed on any server.

- 8G memory
- Minimum 4G. We recommend a 8G VM configuration for complex topology and enabling features.
- 1-4 vCPUs
- 4G hard disk
- 1 serial port
- 1 network interface card (NIC)

Server Software Requirements

The Cisco Nexus 9000v can run on Cisco Unified Computing System (UCS) servers or servers from leading vendors that support VMware ESXi 5.1 (Post Build 1065491/ ESXi 5.5) or the combination of Ubuntu Linux 14.04LTS or later version and KVM-QEMU 2.0.

if you only need a standalone Cisco Nexus 9000v node, the Cisco Nexus 9000v can also be deployed on a laptop or and Apple Mac Pro with a virtual box hypervisor as long as your laptop meets basic resource requirements.

VMware ESXi Support Information

The virtual machine (VM) runs on the VMware vSphere Hypervisor. You can use the same VMware vSphere hypervisor to run serial VMs. Use the VMware vSphere Client GUI to create and manager VMs.

The VMware vSphere Client is an application for creating, configuring, and managing VMs on the VMware vCenter Server. The Cisco Nexus 9000v can boot from a virtual disk located on the data store. You can perform basic administration tasks such as starting and stopping the Cisco Nexus 9000v, 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 https://www.vmware.com/partners/global-alliances/cisco.html.

For more information about VMware features and operations, see the https://www.vmware.com/support/pubs/ .

Nexus 9000v on ESXi 6.5 Deployment Notes

If you are deploying Cisco Nexus 9000v on VMware ESXi 6.5, please ensure that you have checked the following:

- Check that the deployment environment has the correct ESXi 6.5 server and host licenses. Invalid licenses may cause instability in your deployment environment. The instability issues are VM related, such as, no access to a VM serial console,inability to access the Cisco nexus 9000v switch prompt, or incorrect error messages.
- We recommend using the Opera browser, if you are deploying in a Mac environment: http://www.opera.com.
- EFI default firmware option: Nexus 9000v requires EFI firmware boot. Download our distributed ova file from http://software.cisco.com. Select EFI from the Edit Virtual Machine setting menu before powering the VM on.

  Note You do not need to perform this if you are deploying the Cisco Nexus 9000v using the previous vSphere client in Windows.

- The distributed vmdk file downloaded from http://software.cisco.com is not compatible with the ESXi 6.5 release format. To use the old vmdk file, see Using an Old vmdk File with ESXi 6.5, on page 12.

Using an Old vmdk File with ESXi 6.5

Step 1 Convert the distributed vmdk format to ESXi native disk format and use the SATA controller.

```
nexus9000v-user@fe-ucs-dt13:vmfkstools -i nxosv-final.7.0.3.I6.1vmdk nxos-final.7.0.3.I6.1.esx.vmdk
```

Step 2 Choose ESXi 5.5 and later from the Select compatibility step during VM creation.

Step 3 Add SATA Controller.

Step 4 Add Existing Hard Diskand select the nxos-final.7.0.3.I6.1.esx.vmdk you created in 1.

Step 5 Choose New SATA Controller, instead of IDE.

KVM-QEMU Support Information

The 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, ivm-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 Nexus 9000v reference platform is version 2.2.0 or later.

Starting with Cisco NX-OS Release 7.0(3)I7(3), 128 interfaces are suppoted for Cisco Nexus 9000v switches only on KVM hypervisor. This support is applicable for Ubuntu 14.04.4 LTS and 16.04.3 LTS environments and Qemu distort qemu-2.10.0-rc3.tar.xz.

Cisco Nexus 9000v now supports interfaces up to 128+1 (128 data ports, for example, e1/1, e1/2.., e1/128, and management interface). If you do not need 128 interfaces, there is no any negative impact. Only number of vNICs users entered are displayed in the proper interface state. All other interfaces that do not have any associated vNICs display **link not connected** state.

For the interfaces to work smoothly, make sure that the following criteria is met:

- Total 129 vNICs (128 data + 1 for management) from KVM hypervisor command line are required in order to have 128 interfaces available.
- VM resources should be sufficient in terms of memory and vCPUs based on the enabled features and interfaces.
- Extra 3+ minutes are required to boot up the system due to significant PCI scan time on kernel boot-up. The Qemu 2.9.93 (the tested version) is recommended to reduce the VM boot up time. Typical VM boot up time is proximately 5-6 minutes for all 128 data port interfaces to be able to pass the traffic in a large topology system. Any previous released Qemu version could take longer time to boot up the VM.
- 8G+ memory footprint is required in order to have 128 connected interfaces.

VirtualBox Support Information

VirtualBox is a powerful x86 and AMD64/Intel 64 virtualization product for enterprise as well as for the home user. It is free software available as Open Source Software under the terms of the GNU General Public License (GPL) version 2 and you can obtain more information and download from https://www.virtualbox.org/ web site.

VMware Fusion Support Information

VMware Fusion is also a powerful virtualization product for enterprise as well as PC user.

Cisco Nexus 9000v Installation and Deployment

Cisco Nexus 9000v currently does not support virtio block disk. To optimize performance, specific virtual artifact formats are recommended to be used in particular hypervisor.

Hypervisor	Virtual Artifact Format
EXSi	Virtual Machine Disk Image (vmdk), Open Virtualization Appliance (ova)
KVM/Qemu	QEMU Copy On Write (qcow2),Open Virtualization Appliance (ova)
Virtual Box	Virtual Machine Disk Image (vmdk), or packaged box
VMware Fusion	Open Virtualization Appliance (ova)

Cisco Nexus 9000v Software Upgrade and Downgrade

The software upgrade and downgrade of Cisco Nexus 9000v does not follow normal hardware platform procedures. A common upgrade method for Cisco Nexus 9000v is to tftp or scp a new image into the bootflash, then boot the new image from the loader> prompt or set the boot image in "config t; boot nxos bootflash:new_image.bin". A similar approach is used for downgrade.

Note This approach requires sufficient bootflash disk space to hold another image. As such, the nxos.7.0.3.I2.2a image is not upgradable to a new release. In this case, you can create a new VM based on the nxosv-final.7.0.3.I2.2d release; and then upgrade to a new release.

Cisco Nexus 9000v Configuration

Beginning with Cisco NX-OS Release 7.0(3)I5(2), Cisco Cisco Nexus 9000v supports the Cisco Virtual Appliance Configuration (CVAC). This out-of-band configuration mechanism is similar to the PowerOn Auto Provisioning (POAP) autoconfiguration, but instead of downloading the configuration across the network as POAP does, CVAC receives the configuration injected into the Cisco Cisco Nexus 9000v environment on a CD-ROM. The configuration is detected and applied at startup time.

CVAC can be used for a bootstrap configuration (supplying just enough configuration to bring the switch into a reachable state suitable for subsequent configuration using Telnet, RESTful APIs, or other standard mechanisms) or a full configuration (taking the entire configuration of another router and replicating it into a newly launched platform VM). The configuration should be in a plain-text file called nxos_config.txt. You can package the configuration file onto the CD-ROM using the following command:

```
mkisofs -output nxosconfig.iso -l --relaxed-filenames --iso-level 2 <file(s) to add>
```

If the system does not detect a CVAC configuration, the POAP process begins, and the POAP interface prompts you for the initial installation. See the *NX-OS Fundamentals Configuration Guide* for information about POAP for a newly installed switch.

The Cisco Cisco Nexus 9000v supports the same control plane features and configuration that are supported on the Cisco Nexus 9000 Series hardware platforms. The configuration commands for the control plane features follow the same syntax as the Cisco Nexus 9000 Series switches.

Cisco Nexus 9000v Deployment

Provisioning the Cisco Nexus 9000v in the ESXi Hypervisor

Before You Begin

You must have already installed your ESXi hypervisor.

Step 1 Log into the VMware vSphere Client.

Step 2 Select your server's IP address and click the **Create a new virtual machine** link.

Step 3 In the **Configuration** screen, click the **Custom** button and click **Next**.

Step 4 In the **Name and Location** screen, enter your virtual machine's name and click **Next**.

Step 5 In the **Storage** screen, select the datastore and click **Next**.

Step 6 In the **Virtual Machine Version** screen, choose the **Virtual Machine Version** button and click **Next**.

Step 7 In the **Guest Operating System** screen, choose Other (64-bit), Version, and click **Next**.

Step 8 In the **CPUs** screen, choose Number of cores per virtual socket value and click **Next**.

Step 9 In the **Memory** screen, enter the memory size and click **Next**.

Note The minimum size is 4 GB, but the preferable size is 8 GB.

Step 10 In the **Network** screen, select the number of NICs you want to connect.

Note The first NIC must be the management interface. Do not connect all dataports to the management interface.

Step 11 In the **SCSI Controller** screen, choose your SCSI controller, and click **Next**.

Step 12 In the **Select a disk** screen, choose the **Use an existing virtual disk** button, and click **Next**.

Step 13 In the **Select Existing Disk** screen:

a) Browse to the location of the cisconexus9000v-dk9-final.7.0.3.I2.1.vmdk file and click **OK**.
b) Click **Next**.

Step 14 In the **Advanced Options** screen, confirm that the Virtual Device Node is IDE (0:0), and click **Next**.

Step 15 In the **Ready to Complete** screen, chose the **Edit the virtual machine settings before completion** check box, and click **Continue**.

Step 16 In the **Virtual Machine Properites** screen, click the **Add** button to add a Serial Port, confirm that the Virtual Device Node is IDE (0:0), and click **Next**.

Step 17 In the **Device Type** screen, choose Serial Port and click **Next**.

Step 18 In the **Serial Port Type** screen, choose the **Connect via Network** button, and click **Next**.

Step 19 In the **Network Serial Port Settings** screen:

a) In the **Network Backing** frame, choose the **Server** button.

b) Enter telnet and the server IP address in the **Port URI** field, and click **Next**.

Step 20 In the **Ready to Complete** screen, click **Finish**.

Step 21 In the **Getting Started** tab, click the **Edit virtual machine settings** link.

Step 22 In the vSphere Client, select the name of the virtual machine and click the **Options** tab.

Step 23 In the **Virtual Machine Properties** screen, choose Boot Options in the Setting column.

Step 24 In the **Firmware** panel, choose the **EFI** button and click **OK.**

Step 25 Power on the virtual machine.

Deploying a Cisco Nexus 9000v on a KVM or QEMU in a Hypervisor

The Cisco Nexus 9000v can be brought up in the KVM or QEMU hypervisor. The following table lists the parameters that are supported for the Cisco Nexus 9000v deployment on KVM or QEMU.

Parameter	Example	Description
/path_to/qemu	/usr/bin/qemu-system-x86_64	Path to QEMU executable. (The QEMU software can be downloaded from http://wiki.qemu.org/download for different versions.)
-nographic	-nographic	Recommended, as the Cisco Nexus 9000v does not support VGA.
-bios file	-bios bios.bin	Required. The Cisco Nexus 9000v uses EFI boot and requires a compatible BIOS image to operate. We recommend using the latest OVMF BIOS file with the SATA controller for better performance in terms of disk operation. QEMU 2.6 is recommended with the SATA controller. For more information, see http://www.linux-kvm.org/page/OVMF.
-smp	-smp 4	The Cisco Nexus 9000v supports one to four vCPUs, but two to four are recommended.

Parameter	Example	Description
-m memory	-m 8096	Minimum memory is required.
-serial telnet:host:port,server,nowait	-serial telnet:localhost:8888,server,nowait or -serial telnet:server_ip:8888,server,nowait	Requires at least one.
-net ... -net ... or -netdev ... -device ...	-net socket,vlan=x,name=nl_s0,listen=localhost:12000 -net nic, vlan=x,model=e1000,macaddr=aaaa.bbbb.cccc -netdev socket,listen=localhost:12000,id=eth_s_f -device e1000,addr=s.f,netdev=eth_s_f, mac=aaaa.bbbb.cccc,multifunction=on,romfile= or -netdev tap,ifname=tap_s_f,script=no, downscript=no,id=eth_s_f -device e1000,addr=s.f,netdev=eth_s_f, mac=aaaa.bbbb.ccc,multifunction=on,romfile=	The net/net or netdev/device pairs are for networking a virtual network interface card (vNIC). The _s_f represents the PCI slot number and function number. QEMU 2.0 or above has the capability to plug in at least 20 PCI slots and four functions, which accommodates about 80 vNICs in total. The slot range is from 3 to 19, and the function number range is from 0 to 3. The mac= option passes the MAC address of each vNIC MAC address to the VM interfaces. The first -netdev is automatically mapped to the mgmt0 interface on the VM. The second -netdev is mapped to the e1/1 interface and so on up to the sixty-fifth on e1/64. Make sure these MAC addresses are unique for each network device.
-enable-kvm	-enable-kvm	This flag is required for the Cisco Nexus 9000v.

Parameter	Example	Description
-drive ... -device ... (for the SATA controller)	-device ahci,id=ahci0,bus=pci.0 -drive file=img.qcow2,if=none,id=drive-sata-disk0,format=qcow2 -device ide-drive,bus=ahci0.0,drive=drive-sata-disk0,id=drive-sata-disk0	Format to use the SATA controller. We recommend using the SATA controller with QEMU 2.6.0 because this controller offers better performance than the IDE controller. However, you can use the IDE controller if you have an early QEMU version that does not support the SATA controller.
-drive ... -device .... (for the IDE controller)	-drive file=img.qcow2,if=none,id=drive-ide0-0-0,format=qcow2 -device ide-hd,bus=ide.0,unit=1,drive=drive-ide0-0-0,id=ide0-0-0	Format to use the IDE controller.
-drive ... media=cdrom	-drive file=cfg.iso,media=cdrom	CD-ROM disk containing a switch configuration file that will be applied after the Cisco Nexus 9000v comes up. 1. Name a text file (nxos_config.txt). 2. Use Linux commands to make cfg.iso, mkisofs -o cfg.iso -l --iso-level 2 nxos_config.txt.

KVM or QEMU Environment Networking

Deploying the Cisco Nexus 9000v on VirtualBox

Cisco Nexus 9000v deployment on VirtualBox uses one of two forms of distributed virtual artifacts: VMDK or Pre-packaged Box. The simplest method is to use Pre-packaged Box along with Vagrant software. However, the box is created for simple standalone VM deployment with very minimal configuration. This procedure is covered in Deploying Cisco Nexus 9000v on VirtualBox with Vagrant Using a Pre-Packaged Box, on page 23.

Some basic steps and concepts are shown here to create a virtual machine similar to other kinds of VM guests. These instructions are generally for Mac users, but slight differences are highlighted for Window users.

Creating a VM in a VirtualBox Environment Using VMDK

Before You Begin

Hardware

- More than 12 GB of memory.
- More than 4 cores of CPU.

Software

- Mac OS X
  - Install VirtualBox
  - Install Vagrant
  - Install socat for serial console:

```
Homebrew:
  ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
socat:
  brew install socat
```

- Windows
  - Install VirtualBox
  - Install Vagrant
  - Install PuTTY for serial console. (For more information, see www.putty.org.)
  - Install Git to access tools, such as SSH. (For more information, see www.git-for-windows.github.io.)

DETAILED STEPS

	Command or Action	Purpose
Step 1	Copy VMDK to your local hard drive. Launch VirtualBox and click the New icon.	
Step 2	Enter the Name as n9kv, select the Type as Linux, and select the Version as Other Linux (64-bit).	
Step 3	Enter the Memory size (RAM) as 8192 MB.	
Step 4	Select **Use an existing virtual hard drive file**. Click the folder icon (bottom right side) and select the VMDK created in step 1.	
Step 5	Select your VM and click **Settings**.	

	Command or Action	Purpose
Step 6	In the Attributes of Settings, highlight the IDE controller. Select **Add SATA controller** and add the VMDK file. Click **OK** to save the settings.	**Note** Adding the SATA controller enables VirtualBox to boot when the system image size exceeds 512 MB.
Step 7	Click the System icon. • On the Motherboard tab: ◦ Check the Enable EFI checkbox. ◦ Ensure the Modify Boot order so that hard disk is first. • On the Processor tab: ◦ Ensure the Processor is 2. • On the Accelerator tab: ◦ Ensure that VT-x/AMD-V is checked.	
Step 8	Click the Audio icon. Uncheck the Enable Audio checkbox.	
Step 9	Click the Network icon. • On the Adapter 1 tab: ◦ Check the Enable Network Adapter checkbox. ◦ Ensure the Port Forwarding configuration is: ◦ Name = ssh ◦ Host Port = 2222 ◦ Guest Port = 22	
Step 10	Click the Ports icon. • On the Serial Ports tab/Port 1 tab: ◦ Check the Enable Serial Port checkbox. ◦ Ensure that Port Mode is Host Pipe. ◦ Check Create Pipe (or uncheck Connect to existing pipe/socket). ◦ Ensure Port/File Path is /tmp/test.	**Note** For Windows, on the Serial Ports tab/Port 1 tab: • Ensure Path/Address is \\.\pipe\COM1.
Step 11	Click the Ports icon. • On the USB tab:	

	Command or Action	Purpose
	◦ Uncheck the Enable USB 2.0 (EHCI) controller checkbox.	
Step 12	Click the Start button.	Starts the VM. **Note** Ensure that you have the proper serial console setup in your Mac or Windows environment. For more information, see Setup Serial Console on VirtualBox Environment, on page 21.

Setup Serial Console on VirtualBox Environment

Mac OS X

1 Use the 'socat' utility to connect to the serial port (console access) of the VM.

a Install Homebrew and then use homebrew to install 'socat'.

```
Homebrew:
  ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
socat:
  brew install socat
```

b Run the socat command from an xterm.

```
socat unix-connect:/tmp/test stdin
```

2 Go through the NXOS boot process.

a Answer 'no' for the secure password.

b Set admin password to 'admin'.

3 Configure the router.

```
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface mgmt0
switch(config-if)# ip address 10.0.2.15/24  <--- NOTE: can use "ip address dhcp" here
instead
switch(config-if)# no shut
switch(config-if)# end
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# username vagrant password vagrant role network-admin
switch(config)# username vagrant shell bash
switch(config)# boot nxos bootflash:nxos.7.0.3.I2.2d.bin  <--- Note: use correct image
name from "dir" command output
switch(config)# copy r s
[########################################] 100%
```

```
Copy complete.
switch(config)#
```

Windows

1 Open PuTTY.

2 Open "Basic options for your PuTTY session".

   a Specify "\\.\pipe\COM1" for Serial line path.

   b Select "Serial" for "Connection type".

   c Select "9600" for "Speed".

3 Go through the NXOS boot process.

   a Answer 'no' for the secure password.

   b Set admin password to 'admin'.

4 Configure the router.

```
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# interface mgmt0
switch(config-if)# ip address 10.0.2.15/24  <--- NOTE: can use "ip address dhcp" here
instead
switch(config-if)# no shut
switch(config-if)# end
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# username vagrant password vagrant role network-admin
switch(config)# username vagrant shell bash
switch(config)# boot nxos bootflash:nxos.7.0.3.I2.2d.bin  <--- Note: use correct image
name from "dir" command output
switch(config)# copy r s
[########################################] 100%
Copy complete.
switch(config)#
```

Set Up SSH Passwordless Connection to VM

Navigate to login screen and login as vagrant (password: vagrant). Place the vagrant insecure key into the 'authorized_keys' file.

```
switch(config)# exit
switch# exit

User Access Verification
switch login: vagrant
Password:
-bash-4.2$
-bash-4.2$ pwd
/var/home/vagrant
-bash-4.2$ cd .ssh
-bash-4.2$ pwd
/var/home/vagrant/.ssh
-bash-4.2$ echo
"ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4Tj
GYe7gHzIw+niNltGEFHzD8+v1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0
aYet2PkEDo3MlTBckFXPITAMzF8dJSIFo9D8HfdOV0IAdx4O7PtixWKn5y2hMNG0zQPyUecp4pzC
6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGjyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+
R4LOzFUGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHlnVYCzRdK8jlqm8tehUc9c9WhQ
== vagrant insecure public key" > authorized_keys
```

```
-bash-4.2$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4TjG
Ye7gHzIw+niNltGEFHzD8+v1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0a
Yet2PkEDo3MlTBckFXPITAMzF8dJSIFo9D8HfdOV0IAdx4O7PtixWKn5y2hMNG0zQPyUecp4pzC6k
ivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGjyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4
LOzFUGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHlnVYCzRdK8jlqm8tehUc9c9WhQ
== vagrant insecure public key
-bash-4.2$
```

Deploying Cisco Nexus 9000v on VirtualBox with Vagrant Using a Pre-Packaged Box

Starting with Cisco NX-OS Release 7.0(3)I7(3), the Cisco Nexus 9000v Vagrant startup process is streamlined.

See the following customization guidelines and caveats for using Vagrant/vbox:

- The users' customization in Vagrant file is no longer needed.
- There is no need to change the named pipe for Windows users. The serial console can be accessed using port 2023. Now all users can use the **telnet localhost 2023** command to access the serial console using port 2023.
- Now the standard box process is used as any other VM distribution. You can simply bring-up a VM using the base box name.
- The box name can be changed into a different name other than **base** using the **config.vm.box** field.
- The bootstrap configuration is still possible if you want to apply a different configuration on the switch other than pre-baked configuration in **.box** from the release image file. In this case, **vb.customize pre-boot** should be used, for example:

  ```
  vb.customize "pre-boot", [
                                  "storage attach", :id,
                                  "--storagectl", "SATA",
                                  "--port", "1",
                                  "--device", "0",
                                  "--type", "dvddrive",
                                  "--medium", "./nxosv_config.iso", ]
  ```

- The VM interface MAC address can be customized using the **config.vm.base_mac** field, but this modification must be done prior to entering the **vagrant up** CLI command and after entering the **vagrant init** CLI command. If you want to modify the MAC address after entering the **vagrant up** CLI command or after the VM is created, the box commands should be used to modify the VM.

  For example, enter the **vboxmanage list vms** CLI command to find out the VM that is created by the **vagrant up** CLI command:

  ```
  vboxmanage list vms
  ```

  Use the VM listed from the earlier command output, for example, test_default_1513628849309_59058 is found from the **vboxmanage list vms** command as displayed in the following example:

  ```
  vboxmanage modifyvm test_default_1513628849309_59058 --macaddress1 080B206CEEAC
  ```

Complete the following steps to deploy Cisco Nexus 9000v on VirtualBox with Vagrant using a pre-packaged box:

Step 1 Open a terminal in your Mac or PC (GitBash) and make a directory.

Step 2 Download a released image to this directory (for example, nexus9000v-final.7.0.3.I7.3d.box).

Step 3 Execute **vagrant init**.

Step 4 Execute **vagrant box add base nxosv-final.7.0.3.IGP7.0.365.box**.

Step 5 Bring up the VM using the **vagrant up** command in the current directory.

Step 6 Wait for a few minutes to let the bootup finish. Then proceed to the next step.

Step 7 Execute **vagrant ssh** to access the Nexus 9000v bash shell and enter **vagrant** for the password.

Step 8 You can monitor the boot up process from the serial console using **telnet localhost 2023**.

Deleting the VM

Step 1 Shut down the VM.

```
nexus9000v-user@fe-ucs-dt13:~/n9kv/box-test$ vagrant halt --force box-test ==> box-test:
Forcing shutdown of VM...
nexus9000v-user@fe-ucs-dt13:~/n9kv/box-test$
```

Step 2 Delete the VM from the system.

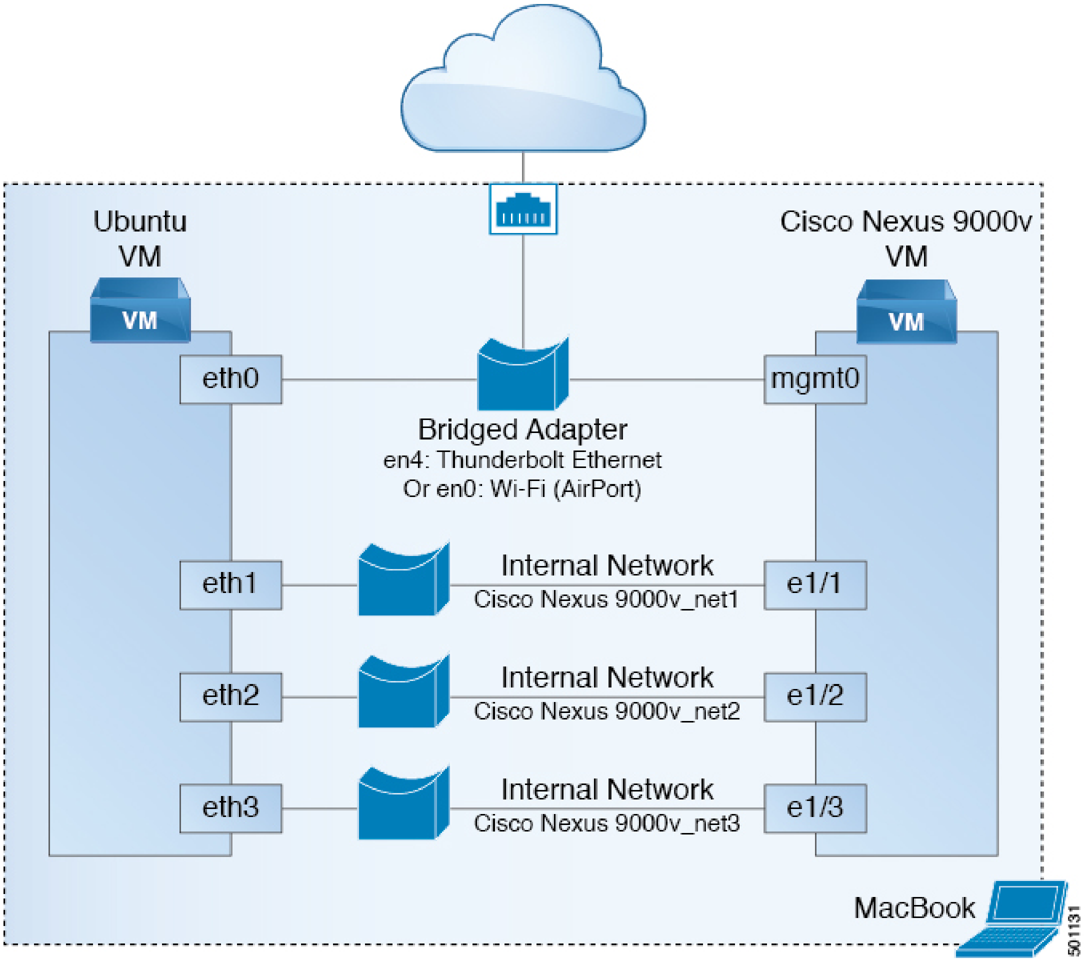
```
nexus9000v-user@fe-ucs-dt13:~/n9kv/box-test$ vagrant destroy box-test
    box-test: Are you sure you want to destroy the 'box-test' VM? [y/N] y
==> box-test: Destroying VM and associated drives...
nexus9000v-user@fe-ucs-dt13:~/n9kv/box-test$
```

Network Topology Examples

A key advantage of Cisco Nexus 9000v is that you can set up a quick networking topology without hardware or complicated cabling tasks to obtain a look and feel about a Cisco Nexus 9000 switch platform.

For example, you can quickly set up a two node system with a server connecting to a Cisco Nexus 9000 virtual machine on laptop. A more complex system can also be setup with a large resource server to do a multiple node simulation. With the topology, you can do tooling and automation in a simulated network that could be applied in a real customer network environment. The following examples show how to interconnect VMs on a laptop or UCS servers.

VirtualBox Topology on a Laptop

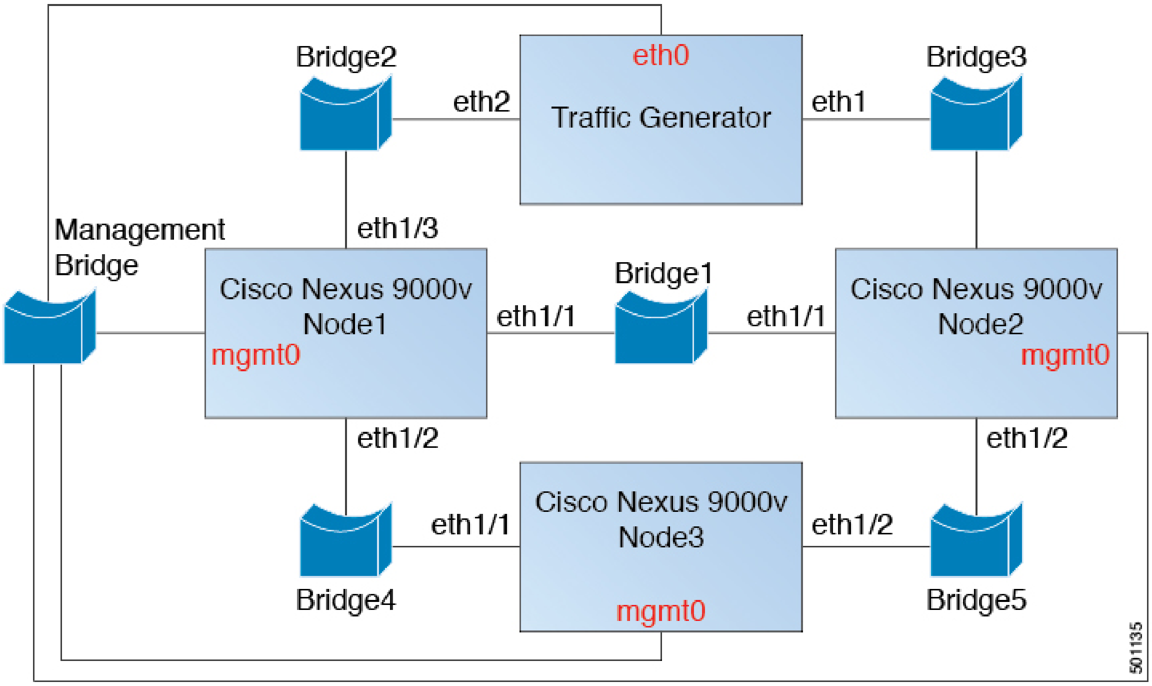


An example diagram above is a typical configuration with Cisco Nexus 9000v and Ubuntu VM two node system. In this case, Both Ubuntu VM and Cisco Nexus 9000v would obtain IPs statically or dynamically visa DHCP protocol reachable from cloud. Similarly, both Ubuntu and Cisco Nexus 9000v can be managed through management network. Ubuntu VM can send/receive packets to Cisco Nexus 9000v through Cisco Nexus 9000v data ports, eth1/1, eth1/2, and eth1/3, or ... e1/9.

Key to Setup:

- Bridge or NAT to Laptop physical ethernet port for management connectivity
- Internal Network for data ports between VMs, change "Promiscuous Mode" to "Allow All"

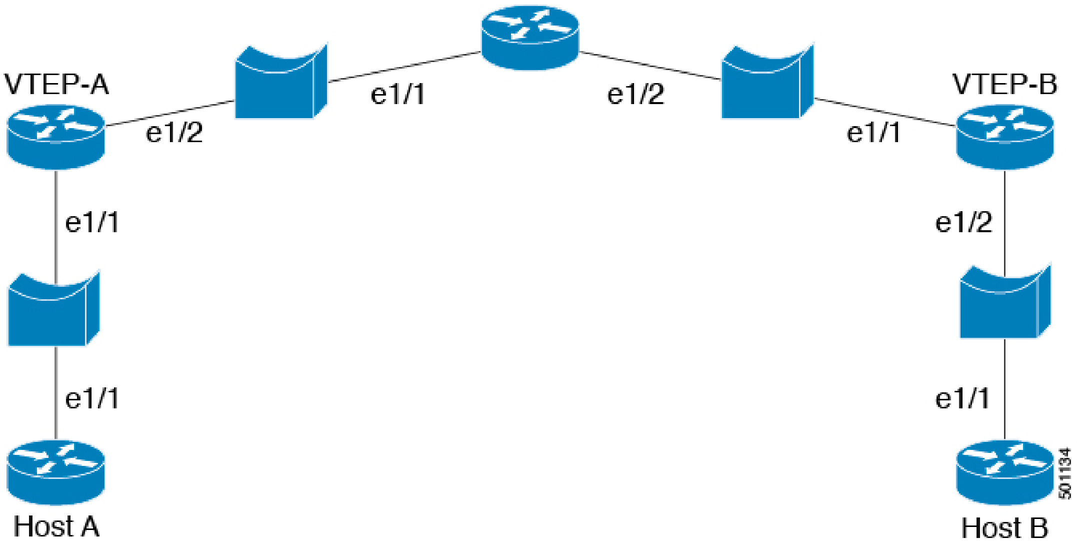
Three Node Topology with Traffic Generator



The nodes in the above diagram are instantiated using the hypervisor specific machine definitions. For networking, each data port interface pair needs to be connected to unique bridge/vSwitch. All the management ports of the Cisco Nexus 9000v (mgmt0) need to be connected to the management bridge and provided a unique IP address, which will enable access to these devices from an external network.

Each data port interface pair that needs to be interconnected should be mapped to the same Bridge/vSwitch. Similar to VirtualBox topology, vSwitch/Bridge must have "Promiscuous Mode" set to "Accept" and "Vlan ID" to "All" for networking to work between Cisco Nexus 9000v nodes. Please read "Troubleshooting" section for hypervisor specific handling for data port communication.

Five Nodes VXLAN Topology



This topology can simulate basic vxlan functionality on Cisco Nexus 9000v platform. Similar bridge/vSwitch setup should be done as shown in other topology examples.

CHAPTER 3

Troubleshooting the Cisco Nexus 9000v

This chapter contains the following sections:

Common Issues For All Hypervisors

How to boot when VM falls into "loader >" prompt

Generally, the first time boot is successful. However, the system boot could fail and drops to the "loader >" prompt on the VGA console or serial console depending on how the VM is provisioned.

Example:

```
loader > dir
Setting listing for bootflash:
Number of devices detected by BIOS is 1
Number of devices detected by BIOS is 1
Number of devices detected by BIOS is 1
Going to print files for device bootflash:
 .rpmstore
 nxos.7.9.3.15.9.66. bin
Number of devices detected by BIOS is 1
Number of devices detected by BIOS is 1
Number of devices detected by BIOS is 1
Clearing listing for bootflash:

loader >
```

To continue the boot, enter the **boot nxos.7.0.3.I5.0.66.bin** command at the "loader >" prompt

How to prevent VM from dropping into "loader >" prompt

As soon as you set up your Cisco Nexus 9000v (following set up of POAP interface), you need to configure the boot image in your system to avoid dropping to the "loader >" prompt after reload/shut down.

Example:

```
config t

  boot nxos n9000-dk9.7.0.3.I2.0.454.bin

copy running starting
```

ESXi Hypervisor

How to use SATA controller to speed up Cisco Nexus 9000v booting process

Cisco Nexus 9000v uses the same hardware platform image boot on hypervisors. ESXi 5.5 and later versions support a SATA controller on an ESXi server that you can use to speed up Cisco Nexus 9000v boot time. To create a VM with a SATA controller, the regular ESXi VM creation steps are applicable except the following are required for a successful VM boot:

- The VMware vSphere Web Client is needed to access this support.
- Download the vmdk image into the ESXi server.

  Convert this monolith vmdk into a VMware native disk type using vmkfstools (command line tool available with the ESXI server)

  Example:

  ```
  vmkfstools -i nexus9000v-final.7.0.3.I5.0.66.vmdk
  nexus9000v-final.7.0.3.I5.0.66-esx.vmdk)
  ```
- Create a VM that is compatible with ESXi 5.5 (or later) and VM version 10.
- Add the SATA controller.
- Add the existing disk with the SATA controller selected.
- Continue the VM booting process from the ESXi VM creation instruction.

How to access the "loader >" prompt from the serial console

EFI BIOS defaults all input/output to the VM console. When a VM drops to "loader >" prompt, you must go to the vSphere client to access "loader >" to boot another image. You can change this behavior by adding an extra configuration in the ESXi VM editing mode.

You can use one of the following methods:

- In the vSphere client Configuration Parameters window, you can add one row in the configuration (Edit Settings > VM Options > Advanced > Edit Configuration).
- You can add **efi.serialconsole.enabled = "TRUE"** to the .vmx file once the VM is created.

How to connect to the switch on ESXi if the EFI serial console is not enabled

On ESXi when you are monitoring the VM console, you might see "Leaving grub land". After this, even though it appears that nothing is happening, the communication has transferred to the serial port you had configured.

```
Read length 646737920
Hd5 for size 646737920
    [Initrd, addr-Ox59236000, size=0x268c70000]

segment header
length: 4, vendor: 16 flags: 4, loadaddr: 2500000, image len: 600 memory length
: 600
Reading data for vendor seg . Length 1536

Image length: 651842048 bytes

image hash: d411d638 b48101f6 2e5e7fOb f0130b67
Leaving grub land
```

To connect to the switch you need to open a terminal and enter the **telnet** <*esxi host*> <*port number*> command.

```
rahushen@rtp-ads-15Ø->
rahushen@rtp-ads-15Ø->telnet fe-ucs-dt7 7ØØØ
Trying 1Ø.122.84.213...
Connected to fe-ucs-dt7.
Escape character is '^]'.

User Access Verification
switch login: admin
Password :
Cisco NX-OS Software
Copyright (c) 2ØØ2-2Ø15, Cisco Systems, Inc. All rights reserved.
Cisco Nexus 9000v software ("Cisco Nexus 9000v") and related documentation,
files or other reference materials ("Documentation") are
the proprietary property and confidential information of Cisco
Systems, Inc. ("Cisco") and are protected, without limitation,
pursuant to United States and International copyright and trademark
laws in the applicable jurisdiction which provide civil and criminal
penalties for copying or distribution without Cisco's authorization.

Any use or disclosure, in whole or in part, of the Cisco Nexus 9000v Software
or Documentation to any third party for any purposes is expressly
prohibited except as otherwise authorized by Cisco in writing.
The copyrights to certain works contained herein are owned by other
third parties and are used and distributed under license. Some parts
of this software may be covered under the GNU Public License or the
GNU Lesser General Public License. A copy of each such license is
available at
http://www.gnu.org/licenses/gpl.html and
http://www.gnu.org/Iicenses/lgpl.html
**************************************************************************
* Cisco Nexus 9000v is strictly limited to use for evaluation, demonstration       *
* and NX-OS education. Cisco Nexus 9000v is provided as-is and is not supported    *
* by Cisco's Technical Advisory Center. Any use or disclosure, in whole  *
* or in part of the Cisco Nexus 9000v Software or Documentation to any third       *
* party for any purposes is expressly prohibited except as otherwise     *
* authorized by Cisco in writing.                                        *
**************************************************************************

switch#
```

The vCenter or UCS server connectivity is lost as soon as Cisco Nexus 9000v is up

Caution When connecting a vNIC into a vSwitch or bridge, an incorrect network connection might result in losing the connectivity to your hypervisor server or vCenter on ESXi.

Cisco Nexus 9000v uses vNICs users entered from the KVM/QMEU command line or from a graphical representation on ESXi for networking, either externally or internally within a hypervisor server. The first NIC is always used as the Cisco Nexus 9000v management interface. The subsequent NICs are used as a data port, such as e1/1, e1/2, and up to e1/9. Total number of ten NICs are allowed and only nine NICs for data ports.

Connect only the first NIC for the Cisco Nexus 9000v VM as the management interface to your lab LAN physical switch or vSwitch (VM Network) connecting directly to physical switch in the lab (or do not connect any data port vNIC to any physical switch conflicting with your server management connectivity).

Cisco Nexus 9000v data port is not passing traffic in ESXi server

To ensure a smooth operation, specific configuration settings on vSwitch must be enabled:

1. Ensure all instances of vSwitch connecting to Cisco Nexus 9000v be in "Promiscuous Mode" = "Accept", pointing to the UCS server. You can access this option through "Configuration > Properties > Edit" from the vSphere Client.
2. Ensure all instances of vSwitch pass through all VLANs. You can access this option through "Configuration > Properties > Edit" from the vSphere Client.

KVM or QEMU Hypervisor

Multicast on KVM or QEMU Hypervisor

The Cisco Nexus 9000v multicast feature is supported as broadcast. To get this feature work properly, the IGMP multicast snooping must be disabled in this environment on all bridge interfaces.

The following example shows how to disable vxlan_br1, vxlan_br2, vxlan_br3, and vxlan_br4 from the linux prompt.

```
echo 0 > /sys/devices/virtual/net/vxlan_br1/bridge/multicast_snooping

echo 0 > /sys/devices/virtual/net/vxlan_br2/bridge/multicast_snooping

echo 0 > /sys/devices/virtual/net/vxlan_br3/bridge/multicast_snooping

echo 0 > /sys/devices/virtual/net/vxlan_br4/bridge/multicast_snooping
```

How to use SATA controller to speed up Cisco Nexus 9000v booting process

Cisco Nexus 9000v uses the same hardware platform image boot on hypervisors. ESXi 5.5 and later versions support a SATA controller on an ESXi server that you can use to speed up Cisco Nexus 9000v boot time. To create a VM with a SATA controller, the regular ESXi VM creation steps are applicable except the following are required for a successful VM boot:

- The VMware vSphere Web Client is needed to access this support.
- Download the vmdk image into the ESXi server.

  Convert this monolith vmdk into a VMware native disk type using vmkfstools (command line tool available with the ESXI server)

  Example:

  ```
  vmkfstools -i nexus9000v-final.7.0.3.I5.0.66.vmdk
  nexus9000v-final.7.0.3.I5.0.66-esx.vmdk)
  ```
- Create a VM that is compatible with ESXi 5.5 (or later) and VM version 10.
- Add the SATA controller.
- Add the existing disk with the SATA controller selected.
- Continue the VM booting process from the ESXi VM creation instruction.

VirtualBox

Networking on VirtualBox or Vagrant

To use the dataplane interfaces on VirtualBox or Vagrant, ensure the following:

- The interfaces must be in 'promiscuous' mode.

  In the VirtualBox network settings, select "Allow All" for the Promiscuous mode.
- Ensure all instances of Cisco Nexus 9000v in your topology have unique MAC addresses by using the **show interface mac** command.

VM Fails to Boot up on VirtualBox/Vagrant

Check the following:

- Ensure that enough resources, such as memory or vCPU, are available. Close all applications that consume a significant amount of memory in your PC or server. Check the available free memory.
- Go to the VirtualBox GUI and power down the corresponding VM created from the Vagrant software (long name with tag specified in Vagrant configuration file) or VM created manually from vmdk.
- Make sure that the "serial console" is correctly provisioned.
- Check block disk type and make ensure it is using the SATA controller.

- PowerOn the VM again. The VGA console should appear with the "loader >" prompt. Follow "How to Boot If VM Fails to loader > prompt" troubleshooting topic, and monitor the booting up process through the serial console.

L2FWDER Troubleshooting

Overview

L2fwder is a centralized forwarding component in Cisco Nexus 9000v which performs the following:

- Rx and Tx packets from or to the vmnics
- L2 switching orbridging
    - MAC learning
        - Dynamic MAC learned in packet path
        - Static MACs learned from L2FM via MTS notifications
            - VMACS
            - GW-MAC
- Switching
    - Maintains an array of potential bridge domains
        - Each Bridge domain keeps track of interfaces
            - In forwarding state
            - In Blocked state as an STP state
    - Switching of packets based on the destination MAC in bridge domain based MAC tables
        - Unicast traffic
        - BUM traffic
    - VXLAN Decapsulation
- Punting packets for Layer 3 processing to kstack and netstack
- VXLAN Decap
    - NVE peer-learning by punting the first packet to kstack/netstack for NVE processing.
    - Learning of remote MACs against the remote VTEP interface.
    - Punting ARP packets in case of Layer 3-gateway to kstack/netstack for ARP to learn the remote host routes.

- VXLAN Encap
  - Performed by netstack and packet manger. (Similar to process in hardware, Nexus 9000 platform, for sup-generated packets.)
- VXLAN BGP EVPN
  - In Cisco Nexus 9000v, MAC routes are produced by L2FWDER into L2RIB directly by replacing L2FM, while HMM continues to produce the MAC IP routes into L2RIB similarly as it occurs in Cisco Nexus 9000v.

Commands for L2FWDER

Common Commands	**debug l2fwder** ?	
	err	Control and data path errors.
	fdb	Events over fdb.
	ha	Events from sysmgr.
	ipc	Events over ipc.
	packet	Packet forwarding information.
	pkttrace	Packet trace.
	vxlan	VXLAN plugin.
Clear Commands	**clear mac address-table datapath dynamic**	
	clear mac address-table datapath static	

Troubleshooting RX/TX Path

- Rx-Path

  The logs to monitor for successful pickup from vmnics and sending it to kstack/netstack.

```
l2fwder_get_data_with_wrr(515):Packet received over Driver type 0
l2fwder_input(67):In   0x0800 78   0 5254.005b.cf97 -> 5254.004c.4e42 Eth1/4
l2fwder_ethernet_output(196):Driver TUN
```

```
l2fwder_action_send_to_stack(865):l2fwder_action_send_to_stack: tx to ifindex 0 iod 8

l2fwder_ethernet_output(304):l2fwder_ethernet_output: driver_type[2] pktQ count[1]
```

- Tx-Path

  The logs to monitor for successful pickup from tuntap and sending it to kstack/netstack.

```
l2fwder_get_data_with_wrr(515):Packet received over Driver type 2

l2fwder_ethernet_output(199):Driver ETH

l2fwder_ethernet_output(251):Out  0x0800 78   0 5254.004c.4e42 -> 5254.005b.cf97 Eth1/4

l2fwder_ethernet_output(304):l2fwder_ethernet_output: driver_type[0] pktQ count[1]
```

- Known Unicast MAC forwarding

```
l2fwder_action_process(934):l2fwder_action_process: process action 1

l2fwder_action_tx_unicast(796):l2fwder_action_tx_unicast: tx to ifindex 1a000600 iod
8  h_type 0

l2fwder_ethernet_output(199):Driver ETH
```

- MAC database (FDB) lookup related logs for a success lookup (Other than BUM traffic)

```
l2fwder_get_mac_lookup_fwd_info(857):Lookup Result is * 0xPo200(1) ret is 1
l2fwder_get_mac_lookup_fwd_info(897):action ucast
```

- MAC database (FDB) lookup for BUM traffic

Troubleshooting MAC Learning

- Command to check the MAC database in L2FWDER:

```
switch# show system internal l2fwder mac

Legend:

        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC

        age - seconds since last seen,+ - primary entry using vPC Peer-Link,

        (T) - True, (F) - False, C - ControlPlane MAC

   VLAN     MAC Address      Type      age     Secure NTFY Ports

---------+-----------------+--------+---------+------+----+------------------

G   100    5254.004c.4e42    static   -          F     F   sup-eth1(R)

G   200    5254.004c.4e42    static   -          F     F   sup-eth1(R)

*   200    5254.00c5.9daf   dynamic   00:07:45   F     F      Po200
```

- Event history command to check for static MAC learning:

```
Event:E_DEBUG, length:73, at 930108 usecs after Wed Sep 14 04:13:14 2016

    [117] [23935]: Learning SUCCESS for static 1 mac 52:54:00:c5:9d:af bd 200
```

- Debug log check for dynamic MAC learning:

```
l2fwder_fdb_insert_entry(231):FDB insert for MAC 52:54:00:c5:9d:af bd 200 total entries
 1
```

Troubleshooting Packet Drops in l2fwder/pktmgr/netstack for layer 2/Layer 3 Traffic

- L2FWDER Global Counters:

```
switch(config)# show l2fwder statistics
 Decap stats:
                              RX    DROP
                DCE_CORE       0       0
           2 dot1q decap       0       0
           Sub-interface       0       0
              Switchport  140940       0
               Undefined  210758       0
                   Stack  635671       0
           1 dot1q decap       0       0
                   VXLAN       0       0
            PORT_CHANNEL  105986       0

Encap stats:
                              TX    DROP
                DCE_CORE       0       0
           2 dot1q decap       0       0
           Sub-interface       0       0
              Switchport  482493       0
               Undefined  211186       0
                   Stack       0       0
           1 dot1q decap       0       0
                   VXLAN       0       0
            PORT_CHANNEL       0       0

Switching stats:
          Unicast     860
            Flood   29372
```

```
        Multicast       0
             Punt   29615
             Drop       0
 LTL Packet Count       0

Punt stats:
      Packets punted 351004

SMM stats:
MAC                 Eth-type   Hit-count
========================================
    0180.c200.0014    0x0000           0
    0180.c200.0015    0x0000           0
    0100.0cdf.dfdf    0x0000           0
    ffff.ffff.ffff    0x0806       29078
    0180.c200.0041    0x22f4           0
    0100.0ccc.cccc    0x0000       13963
    0180.c200.0002    0x0000           0
    0180.c200.0003    0x0000           0
    0180.c200.000e    0x0000           0
    0180.c200.0000    0x0000        1652
    0100.0ccc.cccd    0x0000       97087
    0001.0203.0405    0x0000        1604
    0000.0000.0000    0x0000           0

  Dropped          31
  Consumed     115690
  No Action     29070
  lookup fail  206781

RMM stats:
  Dropped         0
  Consumed   205699
  Rate Limit Dropped      0

VACL stats:
sw-bd        VACL           Hit-count
```

```
========================================

  Dropped         0
  Consumed        0
  Copy+Fwd        0
  No Action       0

Port-Channel stats:
 VSL Drop Packets       0

MAC Learning Disabled stats:
 Packets recieved on Peer-Link:MAC Learning Disabled     313

Action Flood Stats:
 Port-Channel Split-Horizon Packets      48
 VSL Drop Packets                         0

    Forwarding state of ports in bridge domains

switch# show system internal l2fwder bd
Following is the BD State:-
BD_ID  State  Enh_Fwd  Mode
-----  -----  -------  -----
    1      1        0     0
List of all IODs: 9
List of BLK IODs: 8
----------------------------
BD_ID  State  Enh_Fwd  Mode
-----  -----  -------  -----
  100      0        0     0
List of all IODs: 5 7 16
List of BLK IODs: 16
```

Troubleshooting VXLAN BGP EVPN

In the Cisco Nexus 9000v, L2FWDER is the emulated data plane and is responsible for the MAC learning of the connected hosts through source MAC learning.

Note For more information about BGP EVPN, see the *Cisco 9000 Series NX-OS VXLAN Configuration Guide*.

The example in this section considers the following two VTEP end points:

- Leaf0 (VTEP 1) which has hosts with MAC addresses 2222.3333.4444, 000c.2980.d40a in VLAN 1001 and 1002 respectively.
- Leaf1(VTEP 2) which has hosts with MAC addresses 000c.29b9.1375, 000c.29b9.1375 in VLAN 1001 and 1002 respectively.

The following examples shows the MAC and MAC IP route exchange between the two VTEP end points:

- **Local MAC and MAC IP routes in Leaf0**
  - Command to view the source MAC learning:

```
leaf0# show sys int l2fwder mac | inc dynamic
*  1002    000c.2980.d40a   dynamic   01:13:40   F     F     Eth1/2
*  1001    2222.3333.4444   dynamic   00:58:38   F     F     Eth1/2
```

  - L2FWDER produces the learnt end host MACs as MAC routes in the L2RIB table. The command to display the learnt MAC routes in L2RIB:

```
leaf0# show l2route mac all | inc Local

Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
1001        2222.3333.4444 Local  L,             0          Eth1/2
1002        000c.2980.d40a Local  L,             0          Eth1/2
```

  - While L2FWDER is responsible for producing the mac routes, the MAC IP route information is produced by Host Mobility Manager(HMM) in L2RIB. The command to display the MAC IP route information in L2RIB is:

```
switch# sh l2route mac-ip all | inc Local
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
1001        2222.3333.4444 HMM    --            0          5.1.1.1        Local
1002        000c.2980.d40a HMM    --            0          5.2.1.1        Local
```

  - The MAC IP route information is produced by the Host Mobility Manager (HMM) in L2RIB. The command to display the MAC IP route information is:

```
leaf0# show l2route mac-ip all | inc Local

Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
1001        2222.3333.4444 HMM    --            0          5.1.1.1        Local
1002        000c.2980.d40a HMM    --            0          5.2.1.1        Local
```

  - The command to display the BGP learnt local MAC and MAC IP routes per VNI is:

```
leaft1# show bgp l2vpn evpn vni-id 5001
BGP routing table information for VRF default, address family L2VPN EVPN

BGP table version is 79, local router ID is 6.1.1.1

Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best

Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist,
I-injected
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup
*>l[2]:[0]:[0]:[48]:[2222.3333.4444]:[0]:[0.0.0.0]/216
                      6.1.1.1
               100      32768 i
*>l[2]:[0]:[0]:[48]:[2222.3333.4444]:[32]:[5.1.1.1]/272
                      6.1.1.1
        100      32768 i
```

- **Remote MAC and MAC IP routes in Leaf1**
  - In the remote VTEP, the MAC and the MAC IP route information flows through BGP into the L2RIB, and finally L2FWDER receives the end host MAC reachability information.

```
leaft1# show bgp l2vpn evpn vni-id 5001
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 53, local router ID is 6.2.2.2
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-i
njected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf        Weight
Path
*>i[2]:[0]:[0]:[48]:[2222.3333.4444]:[0]:[0.0.0.0]/216
                      6.1.1.1
                 100             0 i
*>i[2]:[0]:[0]:[48]:[2222.3333.4444]:[32]:[5.1.1.1]/272
                      6.1.1.1
                 100             0 i

leaf1# show l2route mac all | inc BGP
1001        2222.3333.4444 BGP    SplRcv         0          6.1.1.1
1002        000c.2980.d40a BGP    SplRcv         0          6.1.1.1

eaf1# show l2route mac-ip all | inc BGP
1001        2222.3333.4444 BGP    --             0          5.1.1.1        6.1.1.1
1002        000c.2980.d40a BGP    --             0          5.2.1.1        6.1.1.1

leaf1# show system internal l2fwder mac  | inc nve-peer
*  1002    000c.2980.d40a    static   -          F     F  (0x47000001) nve-peer1
6.1.1.1
*  1001    2222.3333.4444    static   -          F     F  (0x47000001) nve-peer1
6.1.1.1
```

Troubleshooting VXLAN Encap/Decap

The following is in addition to the normal datapath debugging described in other sections:

NVE manager commands to check the provisioning and learning of NVE peers.	**show nve vni**
	show nve peers all
	show ip overlay-traffic

Commands

Counter gauging commands.	**show l2fwder statistics**
	show system internal pktmgr stats
	show ip traffic
Debug commands to capture packet in datapath.	**debug l2fwder** [**packet** | **pktrace** | **error**]
	debug pktmgr [**frame** | **pkt-errors** | **data** | **tunnel**]
	debug ip packet
	tcpdump **Note** (Debug on the vmnic.)

Collecting VM Logs

The Cisco Nexus 9000v uses all code from the physical hardware platform. Therefore, all logging and core files collected from the hardware platform apply to the Cisco Nexus 9000v system. If any issues arise, we recommend that you take a snapshot of the VM or make a copy of the .vmdk or .qcow2 file for further analysis.