Configuring NXDB for Cisco Nexus 9000 Series Switches and Communicating with NSXv Controllers (for Layer 2 VXLANs), Release 7.x

First Published: 2018-06-22

Last Modified: 2019-08-09

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.

Documentation 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.
[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.

Convention	Description
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.

Documentation Feedback

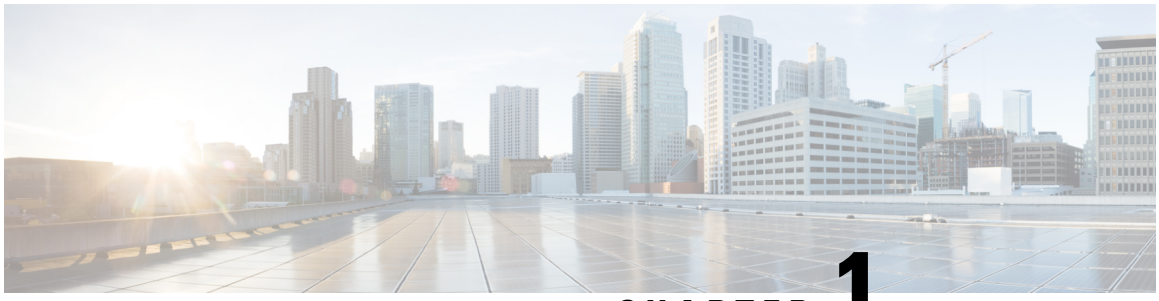
To provide technical feedback on this document, or to report an error or omission, please send your comments to . We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

Overview of NXDB

This chapter includes the following sections:

About NXDB

Nexus Database (NXDB) is a subfeature of NX-API that enables the JavaScript Object Notation Remote Procedure Calls (JSON-RPC) processing functionality of NX-API. It ultimately enables you to configure a Cisco Nexus 9300 or 9300-EX Series switch for Layer 2 VXLAN from an external controller using JSON-RPC NX-API calls. NXDB must be enabled on the switch using Cisco NX-OS in order for the switch to be managed by third-party controllers such as NSXv.

NX-API

Cisco Nexus 9000 Series switches already support NX-API as a programmatic way of interacting with the switch. Traditionally, NX-API improves the accessibility of the Cisco NX-OS command line interface (CLI) by making it available outside of the switch using HTTP. When you enable NX-API on the switch, it starts a web server that enables the switch to start accepting and processing the HTTP requests as they come down from a client (for example, a browser). You can choose to have the output returned in either an XML or JSON format.

NXDB extends the functionality of NX-API. When you enable NXDB, you are enabling NX-API to start accepting and processing JSON-RPC API calls. With this functionality, you can write remote procedure calls (for Layer 2 VXLAN configuration), encapsulate them into HTTP/HTTPs, and send them to the switch. You can write scripts using a programming language such as Python and hence no longer need to have familiarity with the Cisco NX-OS CLI commands to configure Layer 2 VXLAN overlay functionality on the switch.

OVSDB Protocol

The Open vSwitch Database (OVSDB) management protocol is a configuration protocol that allows an external controller and the Cisco Nexus 9300 or 9300-EX Series switch to communicate with one another. External controllers also use the OVSDB Protocol to communicate with hardware VXLAN tunnel endpoints (VTEPs). OVSDB-based communication uses the VXLAN tunnel endpoint (VTEP) Version 5 schema, which defines the data format for the communication between the external controller and the switch.

Cisco NX-OS does not natively support the OVSDB Protocol, so an intermediate agent is needed to accept the OVSDB messages sent from the external controller and make the appropriate JSON-RPC NX-API calls. For this purpose, a separate OVSDB plugin is available that can be run in the Guest Shell container or on an individual Linux server that is connected to the switch.

The OVSDB plugin has OVSDB as its northbound interface (toward the external controller) and the JSON-RPC NX-API as its southbound interface (toward the switch). You must have the OVSDB plugin when trying to establish communication between the external controller, which communicates with the switch using the OVSDB protocol, and the Cisco Nexus switch, which has NXDB enabled.

Object Store

Communication with an external controller not only requires the switch to accept the configuration from the controller but also requires the switch to notify the controller of certain events that take place on the switch. Such events might include port notifications (for example, if the user issues the **shut** or **no shut** command on a port) or MAC notifications (for example, when the switch learns the MAC addresses of hosts that are connected to it). The switch must report these events to the external controller, and the controller may choose to take some action against them (for example, in the case of MAC notifications, the controller might distribute the MAC addresses to other switches in the network).

The NXDB subfeature enables Cisco NX-OS to send the event notifications from the switch. The external controller must register for specific events on the switch. Any time such an event is generated on the switch, Cisco NX-OS notifies the external controller of its occurrence. The OVSDB plugin is registered against events such as port events and MAC events. When such an event occurs, Cisco NX-OS informs the OVSDB plugin about this event in a JSON format. The OVSDB plugin formats this information into an OVSDB-based notification and sends it to the external controller.

In order to allow for this event-notification capability, Cisco NX-OS has introduced the object store. The object store is a shared database that is accessible by Cisco NX-OS components. It contains managed network elements (such as ports, MAC addresses, VLANs, and so on) represented as objects. When an event is generated on the switch, the objects in the object store are updated accordingly. This update triggers an event notification to all registered clients, such as the OVSDB plugin, which can then choose to take some action on the basis of the event.

About VXLAN and NXDB

Virtual Extensible LAN (VXLAN) is an IP overlay technology that allows the creation of virtual IP tunnels between two devices.

With NXDB, the external controller can configure VXLAN on the switch. The external controller maps the VXLAN segments to the traditional VLAN segments on the switch and pushes this configuration to the switch.

In addition, the external controller uses the OVSDB management protocol to configure the VXLAN tunnel endpoints (VTEPs).

VXLAN Overview

Cisco Nexus 9000 switches are designed for hardware-based VXLAN function. It provides Layer 2 connectivity extension across the Layer 3 boundary and integrates between VXLAN and non-VXLAN infrastructures. This can enable virtualized and multitenant data center designs over a shared common physical infrastructure.

VXLAN provides a way to extend Layer 2 networks across Layer 3 infrastructure using MAC-in-UDP encapsulation and tunneling. VXLAN enables flexible workload placements using the Layer 2 extension. It can also be an approach to building a multitenant data center by decoupling tenant Layer 2 segments from the shared transport network.

When deployed as a VXLAN gateway, Cisco Nexus 9000 switches can connect VXLAN and classic VLAN segments to create a common forwarding domain so that tenant devices can reside in both environments.

VXLAN has the following benefits:

- Flexible placement of multitenant segments throughout the data center.

  It provides a way to extend Layer 2 segments over the underlying shared network infrastructure so that tenant workloads can be placed across physical pods in the data center.

- Higher scalability to address more Layer 2 segments.

  VXLAN uses a 24-bit segment ID, the VXLAN network identifier (VNID). This allows a maximum of 16 million VXLAN segments to coexist in the same administrative domain. (In comparison, traditional VLANs use a 12-bit segment ID that can support a maximum of 4096 VLANs.)

- Utilization of available network paths in the underlying infrastructure.

  VXLAN packets are transferred through the underlying network based on its Layer 3 header. It uses equal-cost multipath (ECMP) routing and link aggregation protocols to use all available paths.

VXLAN Tunnel Endpoint

VXLAN uses VXLAN tunnel endpoint (VTEP) devices to map tenants' end devices to VXLAN segments and to perform VXLAN encapsulation and de-encapsulation. Each VTEP has two types of interfaces. One type is a switch interface on the LAN segment to support local endpoint communication through bridging. The other is an IP interface on the northbound segment to encapsulate traffic in VXLAN communications with the VXLAN cloud.

The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. It routes the encapsulated packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address.

BFD over VXLAN

Bidirectional Forwarding Detection (BFD) is an essential feature to provide fast failure detection of a replication node. It can be enabled or disabled on the NSXv GUI on a global basis for all defined replication nodes in the configuration. It provides fast forwarding path failure detection times between the NSXv replication nodes and the hardware VTEP (the Cisco Nexus switch).

Note The terms "replication nodes" and "replication servers" are used interchangeably in this document. However, "replication server" is used in the context of the switch CLI whereas the NSXv terminology is "replication node."

This feature is essentially BFD over VXLAN. The BFD session is hosted on the Cisco switch and on all the replication nodes in the NSXv configuration. The BFD control packets are encapsulated and deencapsulated by the Cisco switch. In vPC mode, the BFD session is hosted only on the vPC primary device, and the vPC secondary device can receive the encapsulated BFD control frames. In such cases, the vPC secondary device deencapsulates the VXLAN packet and sends the deencapsulated packet over the vPC peer link to the vPC primary device (the BFD hosting device).

Explicit TCAM carving is needed on the Cisco switch by way of enabling the redirect-tunnel region. For more information on TCAM carving, see Enabling BFD over VXLAN, on page 15.

On a BFD-enabled setup, the VNIs are hashed to only those replication nodes that have BFD enabled and are in a session UP state. When a BFD session goes down to a replication node, the VNIs are rehashed dynamically to available replication nodes that have BFD enabled and are in the BFD session UP state. This functionality provides dynamic rebalancing of broadcast, unknown unicast, and multicast (BUM) traffic flows upon fast forwarding path failure detection.

Configuring VXLAN on the External Controller

The following VXLAN configurations can be programmed on the external controller and pushed down to the switch:

- VLAN to VXLAN VNI mapping
- Port VLAN mapping
- Peer IP addresses
- Remote MAC addresses
- BFD configuration for replication nodes

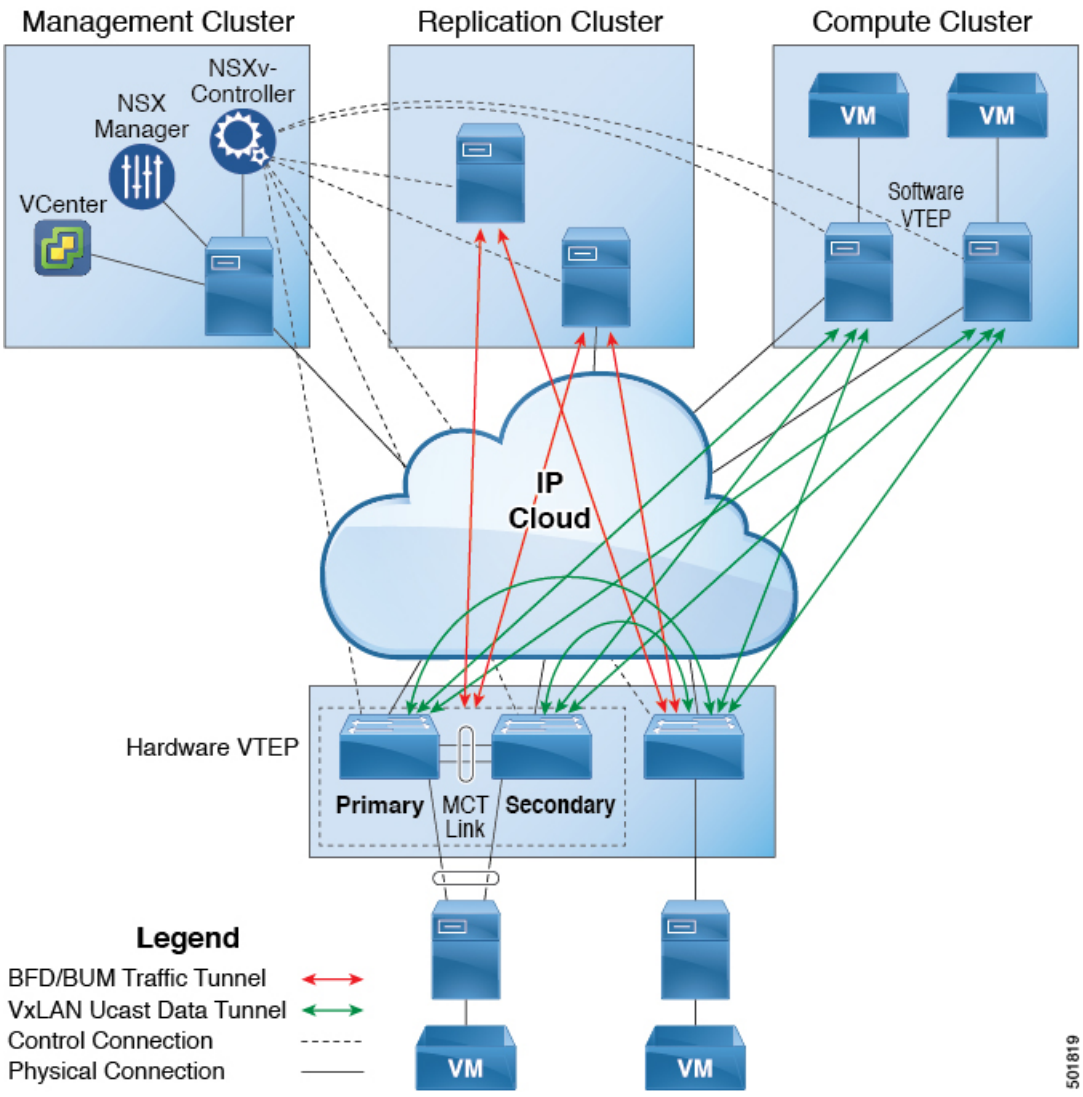
Note To configure the required VXLAN settings on the switch, see Configuring VXLAN on the Switch, on page 13.

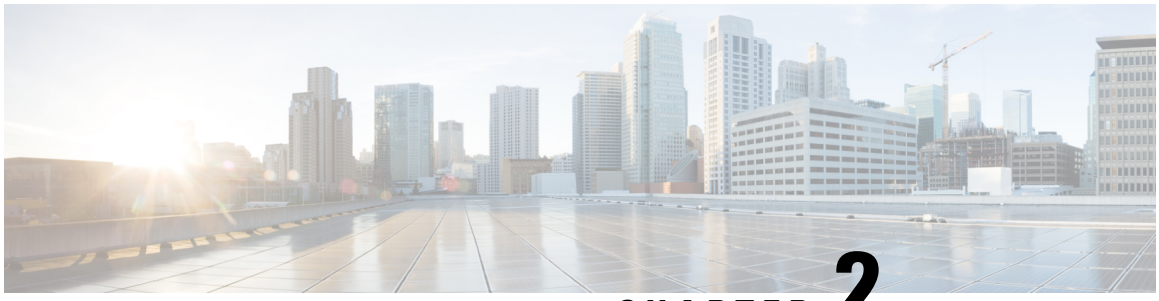
NXDB Topology

The following figure shows an NSXv controller-based topology for Layer 2 VXLAN communication between hardware and software VTEPs. A typical topology consists of the following:

- Management cluster
  - vSphere VCenter VM—The front-end GUI interface for managing the configuration of several components in the setup.
  - NSXv Manager VM—Manages the VXLAN-related configuration.
  - NSXv Controller VMs (three controller VMs)—Pushes the configuration and distributes MAC/peer information to the hardware GW VTEP using OVSDB and uses the Netcpa connection to push configurations to the software VTEP.
- Replication cluster
  - A collection of ESX hosts (called replication nodes) that replicates broadcast, unknown unicast, and multicast (BUM) traffic to other VTEPs in the network.
  - Each replication node hosts BFD sessions to hardware VTEPs.
- Compute cluster
  - A collection of ESX hosts that acts as software VTEPs and encapsulates or de-encapsulates traffic from VMs destined from/to other VTEPs in the network.
- IP core
  - A collection of network devices that forms an IP core responsible for IP routing.
- Hardware VTEP
  - A pair of switches in a vPC or a single switch acting as a VTEP is supported. A VTEP encapsulates traffic from attached hosts and de-encapsulates traffic received from remote VTEPs.
  - Hosts BFD sessions to the replication nodes.
  - Also hosts the OVSDB plugin that connects to the controller. The OVSDB plugin may connect to the controller through either an out-of-band or in-band connection.

Figure 1: Nexus 9000 NXDB NSXv Topology





CHAPTER 2

Enabling NXDB and Installing the OVSDB Plugin

This chapter includes the following sections:

Licensing Requirements for NXDB

This table lists the license package required for using NXDB on the Cisco Nexus 9300 and 9300-EX Series switches.

License	Product ID
TP Services Package (TP_SERVICES_PKG)	N93-TP1K9=(SPARE)

For information on how to apply the license, see the instructions in the *Cisco NX-OS Licensing Guide*.

Prerequisites for NXDB

The following components are required for NXDB:

- A Cisco Nexus 9300 or 9300-EX Series switch running Cisco NX-OS Release 7.0(3)I6(1) or a later release

- Any NSXv controller that follows the OVSDB RFC 7047 and the Open vSwitch Manual
- The OVSDB plugin that is compatible with your Cisco NX-OS release and Cisco device:

Cisco NX-OS Release	OVSDB Plugin Release	JRE	NSXv Release	Cisco Switches
7.0(3)I6(x)	2.1.x	jre-8u112-linux-x64.rpm	6.3.x	9300 and 9300-EX
7.0(3)I7(1) or 7.0(3)I7(2)	2.2.x	jre-8u112-linux-x64.rpm	6.3.x	9300 and 9300-EX
7.0(3)I7(3) or 7.0(3)I7(4)	2.3.x	jre-8u112-linux-x64.rpm	6.3.x	9300 and 9300-EX

Note OVSDB plugin images are backward compatible with prior switch images. For example, 2.2.x will work with the 7.0(3)I6(x) switch image.

Installing the Cisco NX-OS Image on the Switch

Before you can enable NXDB, you must install a version of the Cisco NX-OS image that supports NXDB. For instructions, refer to the *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*.

Note Cisco NX-OS Release 7.0(3)I6(1) and later releases support the NXDB feature.

What to do next

Enable NXDB on the switch.

Enabling NXDB on the Switch

You can enable the switch to be configured via JSON-RPC NX-API calls.

Step 1 **configure terminal**

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **feature nxapi**

```
switch(config)# feature nxapi
```

Enables the NX-API feature.

Step 3 **feature nxdb**

```
switch(config)# feature nxdb
```

Enables NXDB on the switch, which allows the switch to be configured via JSON-RPC NX-API calls.

Step 4 **nxapi use-vrf** {**default** | **management**}

```
switch(config)# nxapi use-vrf management
```

Specifies the VRF to use for NX-API. Choose the management option if the connection to the controller is through the management VRF. The default option specifies the default VRF.

Step 5 (Optional) **copy running-config startup-config**

```
switch(config)# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

What to do next

Assign switch resources for use by the external controller.

Assigning Switch Resources for Use by the External Controller

You can assign both VLANs and interfaces to the external controller.

Guidelines and Limitations for Assigning Switch Resources

VLANs that are assigned to the external controller have the following guidelines and limitations:

- The assigned VLANs must not already exist on the system.
- The assigned VLANs can be configured only as dedicated resources, which means that only the external controller can push down VLAN-related configurations.
- The VLANs are either completely owned by the external controller or completely owned by the switch. If the VLAN is owned by the external controller, the switch cannot configure the port membership for that VLAN. If the VLAN is owned by the switch, any configuration that the controller sends down will be blocked.

Interfaces that are assigned to the external controller have the following guidelines and limitations:

- The Ethernet and port-channel interfaces that are exposed to the external controller must be valid interfaces.
- Virtual port channels (vPCs) are supported.
- vPC domains should be configured with the delay peer-link timer (using the **delay peer-link** *seconds* command). The recommended value is 600 seconds but needs to be adjusted based on the scale.

- The Ethernet and port-channel interfaces can be configured only as shared resources, which means that the configuration for these resources can be driven from both the switch CLI and the external controller.
- If an interface is already assigned, it cannot be changed to an access mode interface.
- OVSDB plugin 2.3.x and later versions support access ports and trunk ports with a native VLAN configuration assigned to the controller. For previous OVSDB plugin versions, an assigned interface is required to be a trunk port without a native VLAN configured.
- FEX interfaces are not supported.
- An assigned interface cannot be configured as a SPAN destination port, and a SPAN destination port cannot be configured as an assigned interface.

Default Behavior of Trunk Interface Commands

The default behavior varies for some trunk interface commands, depending on whether you are entering them on the switch or you are using them to assign interfaces to be used by the external controller.

- **switchport mode trunk**—By default, this command allows all of the VLANs configured in the system to be brought up on this port. However, when a trunk interface is assigned for use by the external controller, only the VLANs that are not assigned are brought up on this port by default. All assigned VLANs are not brought up.
- **switchport trunk allowed vlan** *vlan*—This command can now only accept the non-assigned VLANs. If you try to configure an assigned VLAN, an error message appears.

Assigning VLANs and Interfaces

In order for the switch to accept configurations from an external controller, you must identify the VLANs and interfaces whose configuration can be done from the external controller.

Note You will specify the controller IP address later in Configuring the OVSDB Plugin Using CLI Commands, on page 17.

Step 1 **configure terminal**

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **controller type l2-vxlan identifier** *number*

```
switch(config)# controller type l2-vxlan identifier 1
switch(config-ctrlr-type)#
```

Specifies the assigned interfaces and VLANs for a given controller.

Step 3 **assign interface port-channel** *channel-number* [**shared**]

```
switch(config-ctrlr-type)# assign interface port-channel 100 shared
```

Assigns a vPC interface to the controller.

Note In vPC setups, you must assign an MCT port channel to the controller. However, the MCT port channel will not be visible in the NSX controller GUI.

Step 4 [**no**] **assign vlan** *vlan-range* **dedicated**

```
switch(config-ctrlr-type)# assign vlan 501 dedicated
```

Assigns the VLANs that can be managed by the external controller.

Step 5 [**no**] **assign interface** {**ethernet** *port/slot* | **port-channel** *port-channel-list*} **shared**

```
switch(config-ctrlr-type)# assign interface ethernet 1/4, ethernet 1/17 shared
```

Assigns the Ethernet or port-channel interfaces that can be managed by the external controller.

Assigned interfaces in a vPC pair are appended with the serial number of the device on the controller. For example, if eth1/1 and eth1/2 are assigned to a vPC pair with serial numbers SAL1951VJF5 and SAL1952VJF5, the ports are visible in the NSX controller GUI as follows: eth1/1_SAL1951VJF5 and eth1/2_SAL1952VJF5. However, vPC port-channel interfaces that are assigned to the controller will not have the serial number suffixes.

Note After running **no assign** then assigning back the VLANs in the controller context, the ovsdb-plugin will need to be restarted using the following command: **guestshell run sudo ovsdb-plugin service restart** .

Step 6 Required: **controller description** *string*

```
switch(config-ctrlr-type)# controller description externalcontroller
```

Describes or specifies a name for the external controller.

Step 7 (Optional) **copy running-config startup-config**

```
switch(config-ctrlr-type)# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

What to do next

Install the OVSDB plugin.

Installing the OVSDB Plugin

Once the switch has been configured to enable NXDB, you can install the OVSDB plugin.

Note If this is not a new installation and you are upgrading the OVSDB plugin, see Upgrading the Cisco NX-OS Image and the OVSDB Plugin, on page 45.

Before you begin

Locate the OVSDB plugin and digital signature key (GPG key) on Cisco.com. To do so, go to https://software.cisco.com/download/home, search for and select your Cisco switch in Select a Product (for example, 9372), and click **NX-OS Other Software**. Then download the public GPG key and the OVSDB plugin for NSXv controller integration files.

Step 1 Install the digital signature key.

```
switch# run guestshell sudo rpm --import /bootflash/arm-Nexus9k-rel.gpg
```

Step 2 Destroy and create the Guest Shell with the following recommended configuration for CPU, memory, and hard disk size.

```
switch# guestshell destroy
switch# guestshell resize cpu 18
switch# guestshell resize mem 2500
switch# guestshell resize rootfs 1200
switch# guestshell enable
```

Caution The **guestshell destroy** command removes the entire contents of the Guest Shell, so the switch admin should be aware that all previously installed packages will be removed and must be reinstalled (if needed) after enabling the Guest Shell later.

Note On vPC setups, the above set of commands must be entered for both the primary and secondary vPC.

Step 3 Copy the JRE RPM file and OVSDB plugin RPM file to the bootflash.

```
switch# copy scp://user@scpserver.cisco.com//download/JRE-RPM-file bootflash:JRE-RPM-file

switch# copy scp://user@scpserver.cisco.com//download/OVSDB-plugin-RPM-file
bootflash:OVSDB-plugin-RPM-file
```

Step 4 Access the Guest Shell prompt.

```
switch# run guestshell
[guestshell@guestshell ~]$
```

Step 5 Required: Install the JRE RPM file in the Linux prompt.

```
[guestshell@guestshell ~]$ sudo rpm -i /bootflash/JRE-RPM-file
```

Step 6 Install the OVSDB plugin RPM file.

```
[guestshell@guestshell ~]$ sudo rpm -i /bootflash/OVSDB-plugin-RPM-file
```

Step 7 Verify the installation of the JRE RPM file and OVSDB plugin RPM file.

```
[guestshell@guestshell ~]$ sudo rpm -qa | grep jre
[guestshell@guestshell ~]$ sudo rpm -qa | grep ovsdb
```

What to do next

Configure the switch for VXLAN.

Configuring VXLAN on the Switch

In order for the switch to accept Layer 2 VXLAN configurations from an external controller, you must configure some VXLAN settings on the switch.

Step 1 **configure terminal**

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **feature vn-segment-vlan-based**

```
switch(config)# feature vn-segment-vlan-based
```

Configures the global mode for all VXLAN bridge domains.

Step 3 **feature nv overlay**

```
switch(config)# feature nv overlay
```

Enables the VXLAN feature.

Step 4 **interface loopback** *number* and **ip address** *ip-address*

Non-vPC Standalone Switch

```
switch(config)# loopback interface 0
switch(config-if)# ip address 1.1.101.1/32
```

vPC Primary Switch

```
switch(config)# loopback interface 0
switch(config-if)# ip address 1.1.101.1/32
switch(config-if)# ip address 1.1.106.1/32 secondary
```

vPC Secondary Switch

```
switch(config)# loopback interface 0
switch(config-if)# ip address 1.1.102.1/32
switch(config-if)# ip address 1.1.106.1/32 secondary
```

Configures the loopback interface and the VTEP IP address of the switch on the loopback interface.

A non-vPC standalone switch requires a loopback interface with only one IP address. In a vPC-based system, a primary IP address and a secondary IP address are required on the loopback address of both vPC switches (as shown in this example). The primary IP address must be unique on the two vPC switches, and the secondary IP address must be common to both. The secondary IP address becomes the VTEP IP address of the vPC VTEP.

Step 5 Required: **exit**

```
switch(config-if)# exit
switch(config)#
```

Exits interface configuration mode.

Step 6 Required: **interface nve** *number*

```
switch(config)# interface nve 1
switch(config-if-nve)#
```

Creates a VXLAN overlay interface that terminates VXLAN tunnels. The network virtualization endpoint (NVE) interface serves as a single logical interface for the VXLAN network ports.

Note The switch supports only one NVE interface.

Step 7 Required: **source-interface loopback** *number*

```
switch(config-if-nve)# source-interface loopback 0
```

Configures a loopback interface with a valid /32 IP address as the source interface on the switch. This /32 IP address must be known by the transient devices in the transport network and the remote VXLAN tunnel endpoints (VTEPs). This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network.

Step 8 Required: **auto-remap-replication-servers**

```
switch(config-if-nve)# auto-remap-replication-servers
```

Automatically remaps traffic to a different replication node when a replication node is added or goes down.

Step 9 Required: **host-reachability protocol controller** *controller-name*

```
switch(config-if-nve)# host-reachability protocol controller 1
```

Specifies that the external controller will distribute the host reachability information (such as the MAC addresses and IP addresses of the host) in the network.

Step 10 Required: **config-source controller**

```
switch(config-if-nve)# config-source controller
```

Enables the switch to receive configurations from the controller.

Step 11 **show nve peers**

```
switch(config-if-nve)# show nve peers
Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.0.133.1       Up    CP        4d00h    n/a
nve1      10.0.134.1       Up    CP        4d00h    n/a
nve1      10.0.140.1       Up    CP        4d00h    n/a
nve1      10.0.142.142     Up    CP        4d00h    n/a
nve1      10.0.143.143     Up    CP        2d23h    n/a
nve1      10.0.151.1       Up    CP        4d00h    n/a
nve1      10.0.152.1       Up    CP        4d00h    n/a
nve1      10.0.153.1       Up    CP        4d00h    n/a
nve1      10.0.154.1       Up    CP        4d00h    n/a
nve1      10.0.155.1       Up    CP        4d00h    n/a
nve1      10.0.156.1       Up    CP        4d00h    n/a
nve1      10.0.157.1       Up    CP        4d00h    n/a
```

```
nve1      10.0.158.1       Up    CP        4d00h    n/a
nve1      10.0.159.1       Up    CP        4d00h    n/a
nve1      10.0.160.1       Up    CP        4d00h    n/a
nve1      10.0.161.1       Up    CP        4d00h    n/a
nve1      10.0.162.1       Up    CP        4d00h    n/a
nve1      10.0.163.1       Up    CP        4d00h    n/a
nve1      10.0.164.1       Up    CP        4d00h    n/a
nve1      10.0.165.1       Up    CP        4d00h    n/a
nve1      10.0.166.1       Up    CP        4d00h    n/a
```

Displays the status of the switch's NVE peers.

Step 12 **show nve vni ingress-replication**

```
switch(config-if-nve)# show nve vni ingress-replication

Interface VNI      Replication List  Source      Up Time
--------- -------- ----------------- -------     -------

nve1      10001    10.0.140.1        CONTROLLER  03:31:56

nve1      10002    10.0.134.1        CONTROLLER  03:31:58

nve1      10003    10.0.133.1        CONTROLLER  03:31:56

nve1      10004    10.0.140.1        CONTROLLER  03:31:58
```

Displays the mapping of VNI to ingress-replication peer list and uptime for each peer.

The ingress-replication peer list is a list of the VTEP IP addresses to which BUM packets need to be unicast-replicated. For NSXv, the list always contains a single entry, which maps to one of the replication nodes in the topology. This replication node is chosen on the basis of hashing the available VNIs to the available replication nodes.

What to do next

Enable BFD over VXLAN.

Enabling BFD over VXLAN

Follow these steps to enable BFD over VXLAN.

Step 1 Enable the BFD feature.

```
switch# configure terminal
switch(config)# feature bfd
```

Step 2 Enable TCAM carving of the redirect-tunnel region.

```
switch(config)# hardware access-list tcam region redirect-tunnel 256
```

On the Cisco Nexus 9300 switch, access control lists (ACLs) are implemented in TCAM regions in the hardware. To redirect BFD packets received over VXLAN, you need to install ACL rules in the hardware to redirect these packets to the supervisor module. Certain TCAM regions are enabled by default, and certain regions are not. Because the

redirect-tunnel TCAM region is not enabled by default and is responsible for implementing the redirect ACL for BFD packets, it is required for this configuration. For TCAM changes to take effect in the hardware, a reboot is required after enabling this configuration.

Note This step is required only for Cisco Nexus 9300 Series switches. Cisco Nexus 9300-EX Series switches do not require TCAM carving.

Step 3 Define a BFD control VLAN on the switch and map it to control VNI 0. The BFD control frames are encapsulated with VNI 0.

```
switch(config)# vlan 3000
switch(config-vlan)# vn-segment 0
```

Note You must explicitly specify the BFD VLAN, and it cannot be one of the already assigned VLANs.

Step 4 Define a BFD control SVI with IP forwarding. This command is required to punt the BFD packet to the Supervisor upon receive.

```
switch(config)# interface Vlan3000
switch(config-if)# no shutdown
switch(config-if)# ip forward
```

What to do next

Assign NXDB roles.

Assigning NXDB Roles

Network-admin users can assign roles that limit access to NXDB operations on the switch.

Cisco NX-OS supports two NXDB roles for users who are configured for remote use through TACACS+:

- nxdb-admin—Allowed to execute **get** and **set** JSON-RPC NX-API calls from the external controller.
- nxdb-operator—Allowed to execute only **get** JSON-RPC NX-API calls from the external controller.

When NXDB is enabled, the nxdb-admin role is automatically assigned to the permanent user (admin).

Network-admin users can assign the nxdb-admin or nxdb-operator role to other users as necessary.

Note Representational State Transfer (REST) requests using credentials received from TACACS+ behave as expected.

Step 1 **configure terminal**

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **username** *user-id* [**password** [**0** | **5**] *password*] **role** {**nxdb-admin** | **nxdb-operator**}

```
switch(config)# username NewUser password 5 4Ty18Rnt role nxdb-operator
```

Configures a user account with the specified NXDB role. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hypen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.

The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is 0 (clear text).

Step 3 (Optional) **show user-account**

```
switch(config)# show user-account
```

Displays the NXDB role configuration for remote users who log in directly to the switch.

Step 4 (Optional) **copy running-config startup-config**

```
switch(config)# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

What to do next

Configure the OVSDB plugin.

Configuring the OVSDB Plugin

The OVSDB plugin is installed on both the primary and secondary switches in a vPC setup. The software remains dormant on the secondary switch until that switch becomes the operational primary switch. The primary switch establishes a connection to the external controller.

You must configure the OVSDB plugin on both the primary and secondary vPC switches, by entering Cisco NX-OS CLI commands or running a configuration script.

Configuring the OVSDB Plugin Using CLI Commands

You can use CLI commands to configure the OVSDB plugin.

Step 1 Make sure that the controller IP address is reachable before you configure the OVSDB plugin.

Step 2 View information on how to configure the OVSDB plugin.

```
switch# guestshell run sudo ovsdb-plugin config set -h
usage: ovsdb-plugin config set [-h]
                                        [--switch-description SWITCH_DESCRIPTION]
                                        [--keep-test-config]
                                        [--log-level {error,warn,info,debug,trace}]
```

```
                                        [--log-server ADDRESS]
                                        [--log-type {file,udp}]
                                        [--max-json-peers INT]
                                        [--run-in-switch]
                                        [--schema-version {1.3.99,1.3.0}]
                                        [--vrf NAME] [--xms INT] [--xmx INT]
                                        [-v]
                                        ct_addr1[,ct_addr2]
                                        sw_addr1[,sw_addr2] user1[,user2]
                                        pswd1[,pswd2] name

positional arguments:
  ct_addr1[,ct_addr2]   Controller cluster address in IP:PORT format. Port
                        defaults to 6632 if not included. To specify two or
                        more controllers, separate them with a comma (no
                        spaces). E.g. 10.21.1.10:6632,10.21.1.11:6632
  sw_addr1[,sw_addr2]   Switch address in IP:PORT format. Port defaults to 443
                        if not includes. In VPC mode, specify two switches by
                        separating them with a comma (no spaces). E.g.
                        10.21.1.10:443,10.21.1.11:443
  user1[,user2]         Switch username(s). To specify a different username
                        for a second switch separate with a comma (no spaces).
                        E.g. user1,user2. If no username is given for the
                        second switch, the first one will be used
  pswd1[,pswd2]         Switch password(s). To specify a different password
                        for a second switch, separate with a comma (no
                        spaces). E.g. pswd1,pswd2. If no password is given for
                        the second switch, the first one will be used
  name                  Switch name.

optional arguments:
  -h, --help            show this help message and exit
  --switch-description SWITCH_DESCRIPTION
                        Switch description.
  --keep-test-config    Keep the test config
  --log-level {error,warn,info,debug,trace}
                        Log level to use. Defaults to info
  --log-server ADDRESS  When --log-type is set to UDP, use this to specify the
                        remote where the logs will be sent. The address must
                        be in IP:PORT format. PORT defaults to 514 if not
                        included
  --log-type {file,udp}
                        The type of logging to use. When set to file, the path
                        is always PLUGIN_ROOT/log/ovsdb-plugin.log. Defaults
                        to file
  --max-json-peers INT  Maximum number of JSON peers. Defaults to 6 if not
                        included. Set to -1 to have the plugin auto-compute
                        the value based on the number of controllers
  --run-in-switch       Configures the plugin for running in the switch
  --schema-version {1.3.99,1.3.0}
                        Set the schema version to use. Defaults to 1.3.0
  --vrf NAME            Used only when --run-in-switch is set. This configures
                        the plugin to use the given VRF name when
                        communicating with the controller. Defaults to
                        management
  --xms INT             Set initial Java heap size in MB. Defaults to 2048
  --xmx INT             Set maximum Java heap size in MB. Defaults to 2048
  -v, --verbose         Show extended information
```

Step 3 Configure the OVSDB plugin to establish connectivity to the switch and the controller.

Generic Syntax: Configuration on Non-vPC Switches

```
switch# guestshell run sudo ovsdb-plugin config set --run-in-switch --vrf vrf-name --log-level debug
--log-type file controller-ip-address:controller-port switch-mgmt-ip-address
switch-username switch-password switch-name
```

Note The VRF name configured here should match the VRF configured in the **nxapi use-vrf** command on the switch.

Example: Configuration on Non-vPC Switches

```
switch# guestshell run sudo ovsdb-plugin config set --run-in-switch --vrf default --log-level debug
 --log-type file :1.1.128.252:6640
1.1.103.1 admin pswd1234 N9K-TOR1
```

Generic Syntax: Configuration on vPC Switches

On a vPC system, the configuration command must be run on both the vPC primary and secondary switches.

```
switch# guestshell run sudo ovsdb-plugin config set --run-in-switch --vrf vrf-name --log-level debug
--log-type file controller-ip-address:controller-port local-switch-ip,remote-switch-ip
local-switch-username,remote-switch-username local-switch-password,remote-switch-password,
common-switch-name --switch-description common-switch-description
```

Example: vPC Primary Configuration

```
switch# guestshell run sudo ovsdb-plugin config set --run-in-switch --vrf default --log-level debug
 --log-type file 1.1.128.252:6640
1.1.104.1,1.1.105.1 admin,admin pswd1234,pswd1234 TOR1-TOR2 --switch-description TOR1-TOR2
```

Example: vPC Secondary Configuration

```
switch# guestshell run sudo ovsdb-plugin config set --run-in-switch --vrf default --log-level debug
 --log-type file 1.1.128.252:6640
1.1.105.1,1.1.104.1 admin,admin pswd1234,pswd1234 TOR1-TOR2 --switch-description TOR1-TOR2
```

Note For NSXv, the controller port must be 6640. For vPC setups, the *switch-name* must be identical on both the vPC primary and secondary switches.

By default, the schema version used by OVSDB is 1.3.0.

Step 4 Verify the OVSDB plugin configuration.

Example: Non-vPC Switch

```
switch# guestshell run sudo ovsdb-plugin config show

Controllers:
  #1 addr           : 2.2.128.252:6640
VPC                 : No
In switch           : Yes
VRF                 : default
Switches:
  #1 addr           : 1.1.24.2:443
  type              : STANDALONE
  user              : admin
  name              : SAL184333WH
  description       :
Log type            : file
Log level           : trace
Log server          : -
```

```
TTY log path        : -
Maxi JSON peers     : 6
Min heap size       : 2048 MB
Max heap size       : 2048 MB
Schema              : 1.3.0
```

Example: vPC Primary Switch

```
switch# guestshell run sudo ovsdb-plugin config show

Controllers:
  #1 addr           : 2.2.128.252:6640
VPC                 : Yes
In switch           : Yes
VRF                 : default
Switches:
  #1 addr           : 1.1.24.1:443
  type              : LOCAL
  user              : admin
  name              : ovsdb-plugin
  description       :
  #2 addr           : 1.1.24.4:443
    type            : REMOTE
    user            : admin
    name            : ovsdb-plugin
    description     :
Log type            : file
Log level           : trace
Log server          : -
TTY log path        : -
Maxi JSON peers     : 6
Min heap size       : 2048 MB
Max heap size       : 2048 MB
Schema              : 1.3.0
```

Example: vPC Secondary Switch

```
switch# guestshell run sudo ovsdb-plugin config show

Controllers:
  #1 addr           : 2.2.128.252:6640
VPC                 : Yes
In switch           : Yes
VRF                 : default
Switches:
  #1 addr           : 1.1.24.4:443
  type              : LOCAL
  user              : admin
  name              : ovsdb-plugin
  description       :
  #2 addr           : 1.1.24.1:443
    type            : REMOTE
    user            : admin
    name            : ovsdb-plugin
    description     :
Log type            : file
Log level           : trace
Log server          : -
TTY log path        : -
Maxi JSON peers     : 6
Min heap size       : 2048 MB
Max heap size       : 2048 MB
```

```
Schema              : 1.3.0
```

If you want to fine tune the configuration of the OVSDB plugin, see Extended OVSDB Plugin Configuration, on page 53.

Step 5 Make sure that the configured switch IP address and the controller IP address are pingable before moving to the next procedure.

What to do next

Create a custom certificate, for Cisco NX-OS 7.0(3)I7(5) and later releases. See Creating a Custom Certificate, on page 22.

Obtain the necessary certificates and run the OVSDB plugin, for releases prior to Cisco NX-OS Release 7.0(3)I7(5). See Running the OVSDB Plugin, on page 23.

Configuring the OVSDB Plugin Using a Configuration Script

The following example shows how to use a configuration script to configure the OVSDB plugin (running inside a guestshell container on the switch) in a non-vPC system:

```
cd to directory: /usr/local/ovsdb/bin/
[admin@guestshell bin]$ sudo ./ovsdb-plugin user-input
Running in switch? (yes/no) [y] yes
Would you like to configure OVSDB plugin in Dual Switch mode? (yes/no) [n]no
Enter the switch address in IP:PORT format. Port defaults to 443 if not includes
>>10.23.237.22
Enter the Switch username. >> admin
Enter the switch password. >>
Password:
Enter the Controller cluster address in IP:PORT format. Port defaults to 6632 if not included.
 To specify two or more controllers, separate them with a comma (no spaces). E.g.
10.21.1.10:6632,10.21.1.11:6632 >> 10.23.237.33:6640
Enter the switch name >> ovsdb-plugin
Enter the schema version to use. Defaults to 1.3.0.
Choices are:
1. 1.5.1
2. 1.3.0
Enter the option >>2
Would you like to set the parameter --vrf? [default] management
Would you like to configure optional parameters? (yes/no) [n]yes
Enter the switch description >> node02
Configure default log level to use? (error/warn/info/debug/trace) [debug] debug
Would you like to configure the type of logging to use ? Options are  (file/udp) [file]
Configuration saved
```

The following example shows how to use a configuration script to configure the OVSDB plugin (running inside a guestshell container on the switch) in a vPC system:

```
cd to /usr/local/ovsdb/bin/
[admin@guestshell bin]$ sudo ./ovsdb-plugin user-input
Running in switch? (yes/no) [y] yes
Would you like to configure OVSDB plugin in Dual Switch mode? (yes/no) [n]yes
Enter the local switch address in IP:PORT format. Port defaults to 443 if not includes
>>10.23.237.21
Enter the remote-peer switch address in IP:PORT format. Port defaults to 443 if not includes
 >> 10.23.237.24
```

```
Enter the Switch username. >> admin
Enter the second switch username in the VPC setup >> admin
Enter the switch password. >>
Password:
Enter the second switch password. >>
Password:
Enter the Controller cluster address in IP:PORT format. Port defaults to 6632 if not included.
 To specify two or more controllers, separate them with a comma (no spaces). E.g.
10.21.1.10:6632,10.21.1.11:6632 >> 10.23.237.33:6640
Enter the switch name >> ovsdb-plugin
Enter the schema version to use. Defaults to 1.3.0.
Choices are:
1. 1.5.1
2. 1.3.0
Enter the option >>2
Would you like to set the parameter --vrf? [default] management
Would you like to configure optional parameters? (yes/no) [n]yes
Enter the switch description >> node02
Configure default log level to use? (error/warn/info/debug/trace) [debug]
Would you like to configure the type of logging to use ? Options are  (file/udp) [file]
Configuration saved
```

Creating a Custom Certificate

For Cisco NX-OS Release 7.0(3)I7(5) and later releases, you must create and apply a custom certificate to run the OVSDB plugin. Although earlier releases can use a default certificate, we recommend that custom certificates be used.

Step 1 Enable the Bash shell.

```
switch# feature bash
```

Step 2 Create OpenSSL custom SSL key files.

```
switch# run bash openssl req -nodes -x509 -new -keyout /bootflash/ssl.key -out /bootflash/ssl.crt
-days 1000 -subj /C=US/ST=CA/L='San Jose'/O='Cisco Systems Inc.'/OU=dcnxos/CN=nxos/
switch# run bash openssl rsa -in /bootflash/ssl.key -out /bootflash/ssl.key
```

The default certification keys are not supported.

Step 3 Configure the switch NX-API certificate settings with the custom certificate keys.

```
switch# nxapi certificate httpscrt certfile ssl.crt
switch# nxapi certificate httpskey keyfile ssl.key
switch# nxapi certificate enable
```

Step 4 Verify that the custom certificate is applied in the switch.

```
switch# run bash cat /nginx_1_fe/conf/nginx.conf | grep ssl
        ssl                  on;
        ssl_certificate      /var/nginx/cert/server.crt_vdc_1;
        ssl_certificate_key  /var/nginx/cert/server.key_vdc_1;
        ssl_session_timeout  5m;
        ssl_prefer_server_ciphers on;
        ssl_protocols TLSv1.1 TLSv1.2;
        ssl_ciphers …
```

The certificate lines must include "_vdc_1" (or "_restore" if the switch has been reloaded). If "_default" is displayed, the custom certificate is not applied.

What to do next

Obtain the certificates and run the OVSDB plugin. See Running the OVSDB Plugin, on page 23.

Running the OVSDB Plugin

You need to obtain a certificate to run the OVSDB plugin.

Step 1 Perform one of the following to generate a certificate for the switch:

- If you are not using vPCs, enter this command:

```
switch# guestshell run sudo ovsdb-plugin cert bootstrap
```

- If you are using vPCs, do the following:

1. On the secondary vPC, run the following command to generate the certificate:

```
switch# guestshell run sudo ovsdb-plugin cert bootstrap --receive
```

2. On the primary vPC, run the following command, paste the certificate that is shown as part of Step a, and press Enter.

```
switch# guestshell run sudo ovsdb-plugin cert bootstrap --send secondary-ip-address
```

Note The **bootstrap** option needs to be used only for the first certificate download on the plugin. For subsequent certificate downloads, the **cert reset** option needs to be used.

Step 2 View the switch's certificate.

```
switch# guestshell run sudo ovsdb-plugin cert show
```

Step 3 Use the information in the output of the previous command to configure the switch's certificate on the controller GUI.

Step 4 Start the OVSDB plugin so that it connects to the switch and controller.

```
switch# guestshell run sudo ovsdb-plugin service start
```

What to do next

Verify the connection status for the OVSDB plugin.

Verifying the Connection Status for the OVSDB Plugin

The OVSDB plugin establishes a connection toward the external controller on one side and toward the switch running Cisco NX-OS on the other. To verify the success of these connections, enter the **guestshell run sudo ovsdb-plugin service status** command.

Example Output on vPC Primary Switch:

```
switch# guestshell run sudo ovsdb-plugin service status --more

Status: Running

Connections

  Switches:

      #1 addr         : 2.2.14.1
         type         : Local
         vpc          : Enabled/Primary
         state        : Up
         config ready : Up
         nxapi        : Up
         websocket    : Up

      #2 addr         : 2.2.14.4
         type         : Remote
         vpc          : Enabled/Secondary
         state        : Up
         config ready : Up
         nxapi        : Up
         websocket    : Up

  Controllers:

      #1 addr         : 2.2.128.254
         state        : Up

      #2 addr         : 2.2.128.253
         state        : Up

      #3 addr         : 2.2.128.252
         state        : Up
```

Example Output on vPC Secondary Switch:

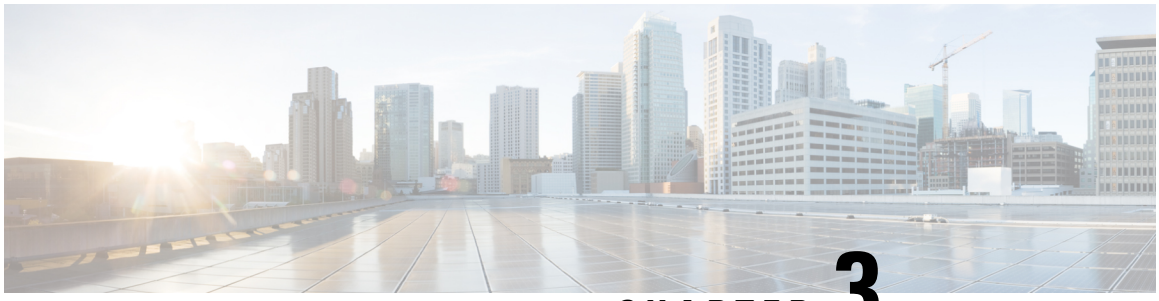
```
switch# guestshell run sudo ovsdb-plugin service status --more

Status: Running

Connections

  Switches:

      #1 addr         : 2.2.14.4
         type         : Local
         vpc          : Enabled/Secondary
         state        : Up
         config ready : Up
         nxapi        : Up
         websocket    : Up
```

```
      #2 addr         : 2.2.14.1
         type         : Remote
         vpc          : Enabled/Primary
         state        : Up
         config ready : Up
         nxapi        : Up
         websocket    : Up

  Controllers:

      #1 addr         : 2.2.128.252
         state        : Down
```

Note The controller connection state on the vPC secondary switch is shown as Down because only the plugin running on the vPC primary switch connects to the controller.

Example Output on a Non-vPC Switch:

```
switch# guestshell run sudo ovsdb-plugin service status --more

Status: Running

Connections

  Switches:

      #1 addr         : 1.1.24.2
         type         : Local
         vpc          : Disabled
         state        : Up
         config ready : Up
         nxapi        : Up
         websocket    : Up

  Controllers:

      #1 addr         : 2.2.128.253
         state        : Up

      #2 addr         : 2.2.128.254
         state        : Up

      #3 addr         : 2.2.128.252
         state        : Up
```



CHAPTER 3

Using the Cisco Nexus 9000 Switch as the Default Gateway

This chapter includes the following sections:

Routing for VNIs That Have a HW-VTEP Binding

When an NSX logical switch is connected to a HW-VTEP using OVSDB, it cannot be attached to the Distributed Logical Router (DLR) at the same time. This limitation exists with all NSX implementations of this feature regardless of the hardware vendor providing the HW-VTEP functionality. Traditionally this meant that the default gateway for the VMs and bare-metal devices attached to the VNI/VLAN combination had be an external device. This device could be an Edge Services Gateway (ESG) VM attached to the VNI or a traditional router connected to the VLAN (or a physical firewall or another service device). These legacy options are depicted in the following figures.

Figure 2: Legacy Option 1: Using an ESG VM as the Default Gateway

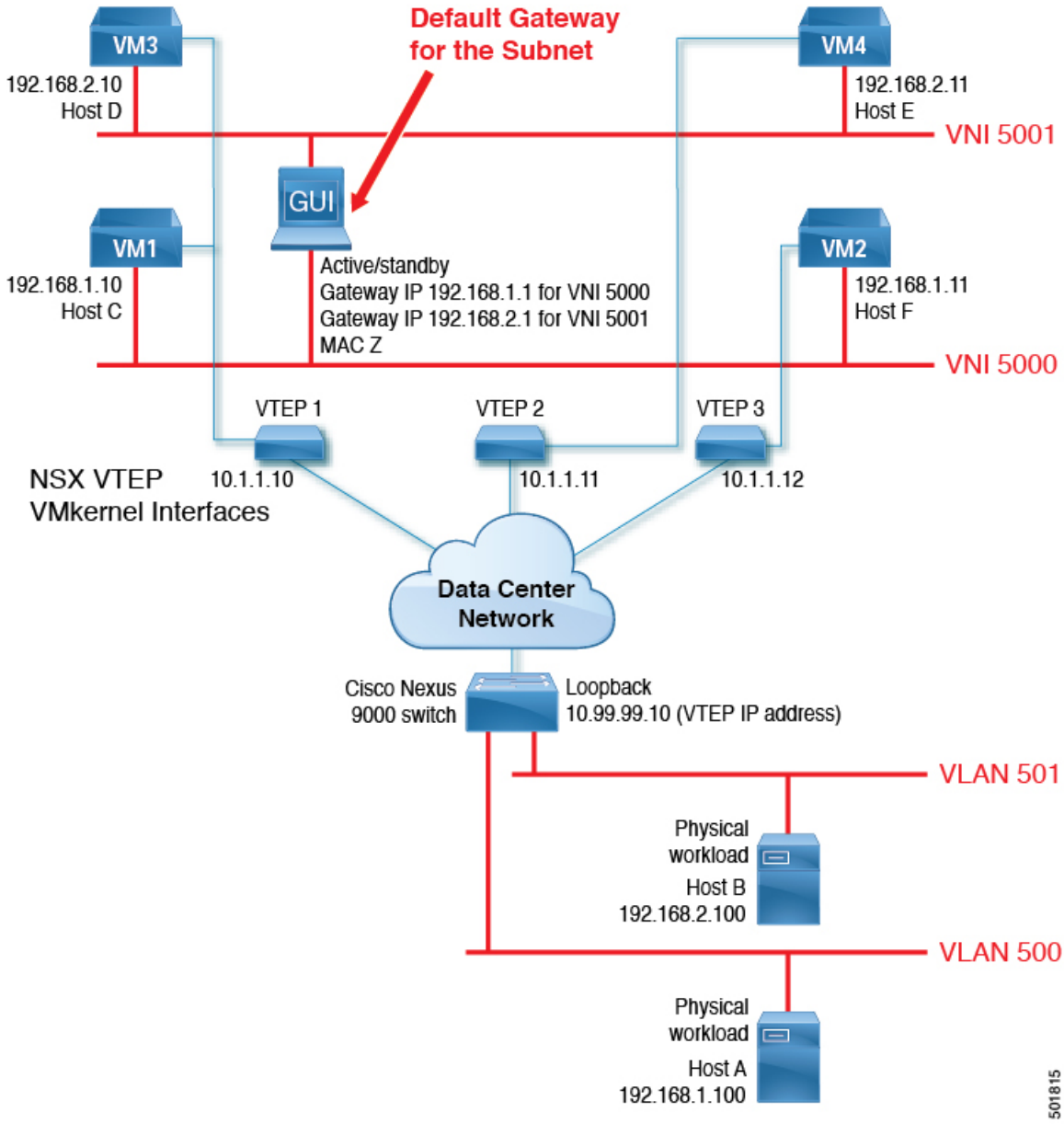
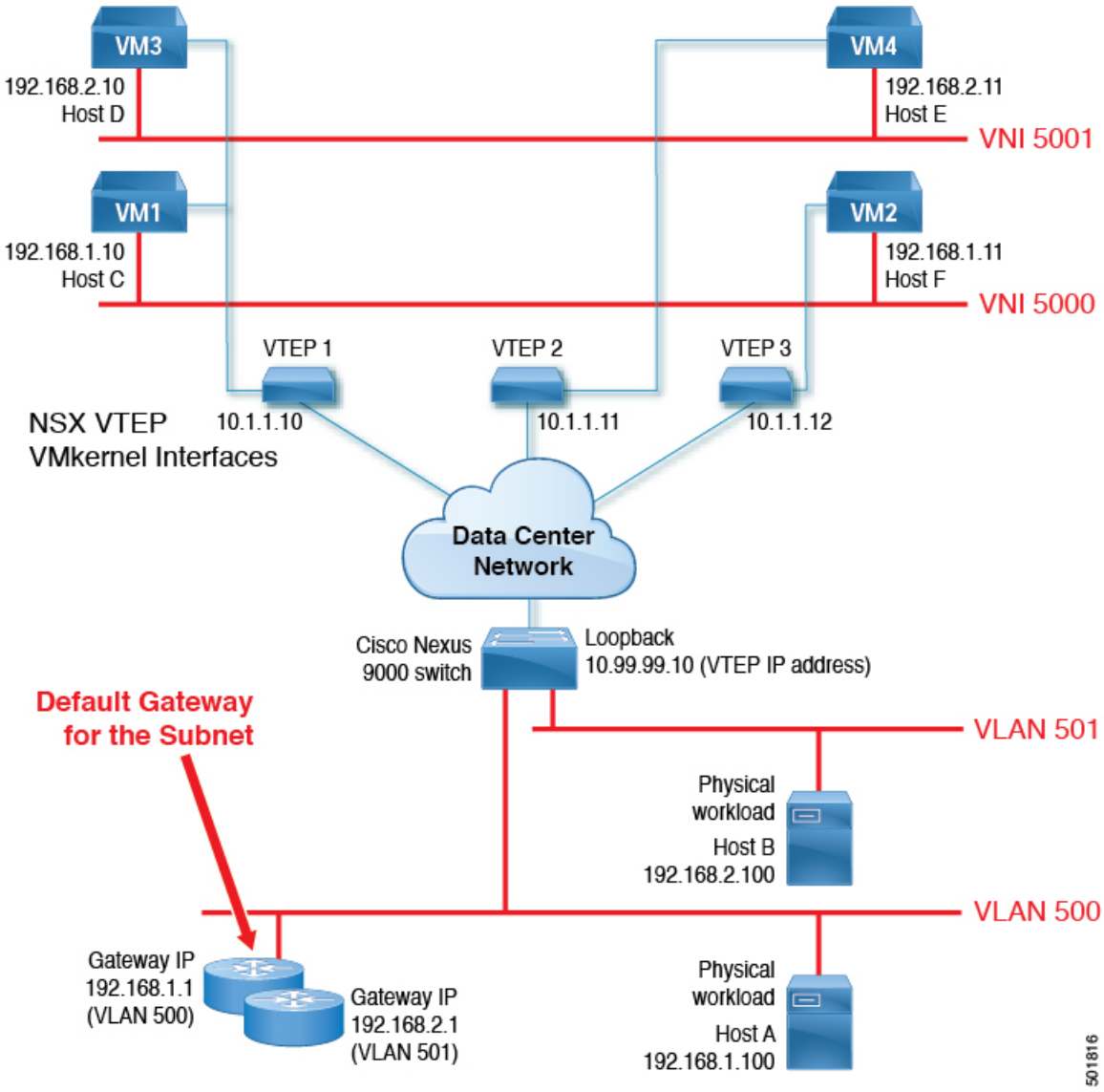
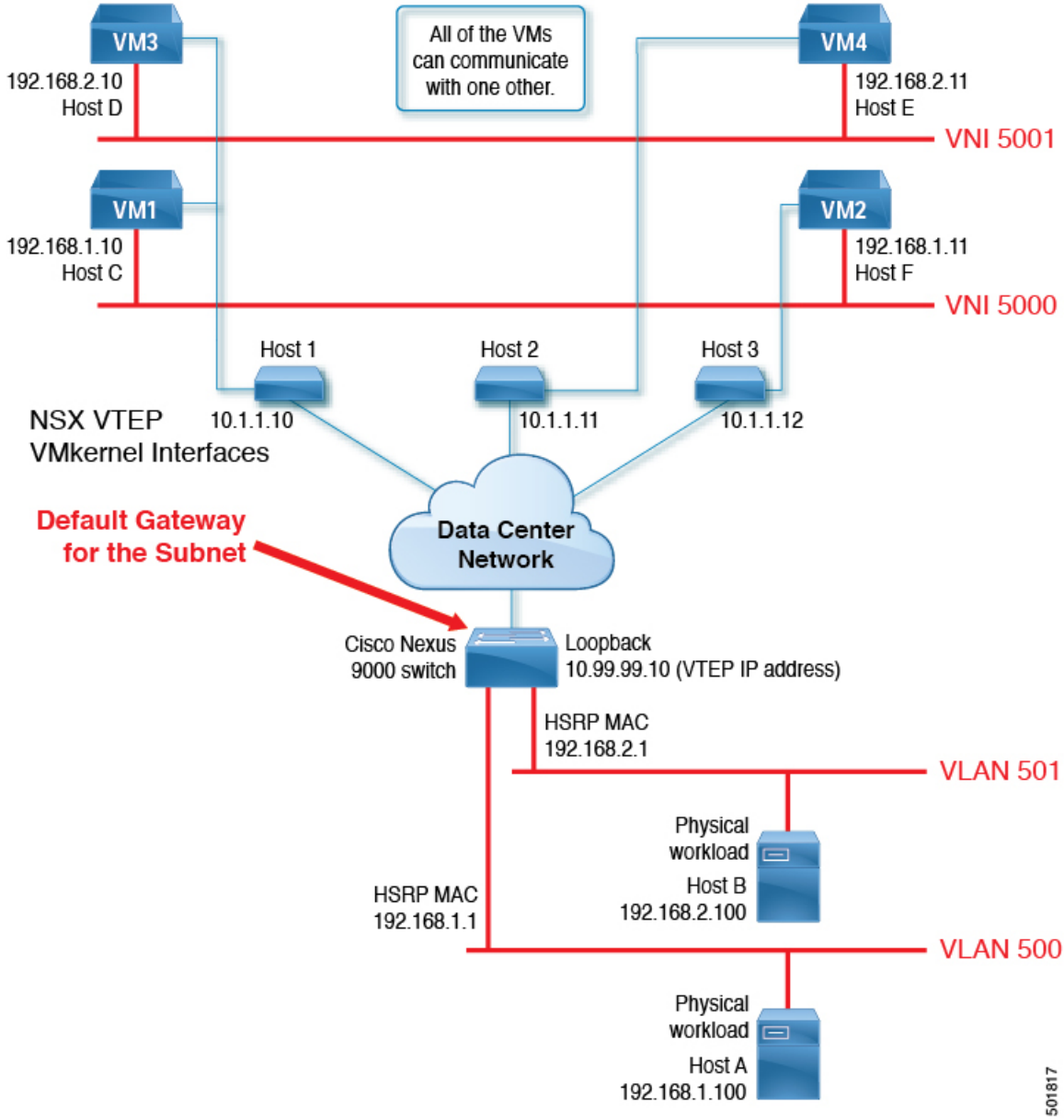


Figure 3: Legacy Option 2: Using an External Router as the Default Gateway



With the Cloud-Scale ASICs available in the Cisco Nexus 9300-EX switches, it is now possible for the Cisco Nexus 9000 switches doing the OVSDB integration to also be the default gateway for the subnet being extended. This capability allows for CAPEX savings because an external physical router is no longer necessary. By providing the default gateway and routing capabilities using the switched virtual interface (SVI) feature, the Cisco Nexus 9000 switch can perform routing in the hardware while also providing OPEX savings. Redundancy can be achieved by using a first-hop redundancy protocol such as the Hot Standby Router Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP). This new capability is depicted in the following figure.

Figure 4: Using the Cisco Nexus 9000 Switch as the Default Gateway



Prerequisites for Default Gateway Integration

The following components are required for default gateway integration:

- One of the following switches running Cisco NX-OS Release 7.0(3)I7(1) or a later release:
  - Cisco Nexus 93108TC-EX switch
  - Cisco Nexus 93180LC-EX switch

- Cisco Nexus 93180YC-EX switch

Note Switches that meet the minimum requirements for NSX OVSDB integration but do not meet the requirements for the gateway feature cannot use both features at the same time.

Configuring a Redundant Default Gateway on Two vPC Switches Acting as HW-VTEPs Using HSRP

You can take advantage of HSRP when configuring a default gateway on a pair of switches running vPC for redundant HW-VTEP connectivity. HSRP is a first-hop redundancy protocol that allows a transparent failover of the first-hop IP router. For more information on HSRP, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

Step 1 Assign a VLAN to the controller.

Example:

```
switch# configure terminal
switch(config)# controller type l2-vxlan identifier 1
switch(config)# assign vlan 600-602 dedicated <<VLANs assigned to the controller
```

We recommend that the SVIs are created after the VLANs are assigned to the controller.

Note A mapping of the logical switch/VNI to the VLAN does not have to be done in the NSX Manager before the SVI for the VLAN is created.

Step 2 Enable HSRP on both vPC switches.

Example:

```
switch# configure terminal
switch(config)# feature hsrp
```

Using a basic HSRP configuration on the two vPC switches provides default gateway redundancy.

Step 3 Create an SVI and assign it a unique IP address on each switch.

Example:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 600
switch(config-if)# no shut
```

Step 4 Configure a matching unique HSRP group number and virtual IP address (IPv4 and/or IPv6) under the SVI on both switches.

Example:

```
switch(config-if)# ip address 10.10.10.2/24
switch(config-if)# hsrp version 2
switch(config-if)# hsrp 1
switch(config-if-hsrp)# ip 10.10.10.1
```

When you configure HSRP on a network segment, you provide a virtual MAC address and a virtual IP address for the HSRP group. The virtual MAC address is advertised to the NSX controllers using OVSDB. You configure the same virtual address on each HSRP-enabled interface in the group. You also configure a unique IP address on each interface that acts as the real address.

Note We recommend using a unique HSRP group address for each SVI because the group address is used to derive the MAC address for the virtual IP address. Using a unique group number (from 0 to 4095) for each HSRP group creates a unique MAC address for each virtual IP address.

Example

The following example configuration is for vPC switch 1:

```
switch# configure terminal
switch(config)# feature hsrp <<Enable the HSRP feature
switch(config)# feature interface-vlan
switch(config)# interface vlan 600 <<Create the SVI for VLAN 600
switch(config-if)# no shut
switch(config-if)# ip address 10.10.10.2/24 <<Assign a unique IP address to the SVI
switch(config-if)# hsrp version 2 <<Set the HSRP version
switch(config-if)# hsrp 1 <<Enter a unique HSRP group configuration
switch(config-if-hsrp)# ip 10.10.10.1 <<Enter the virtual IP to be used as the subnet gateway
switch(config-if-hsrp)# end
```

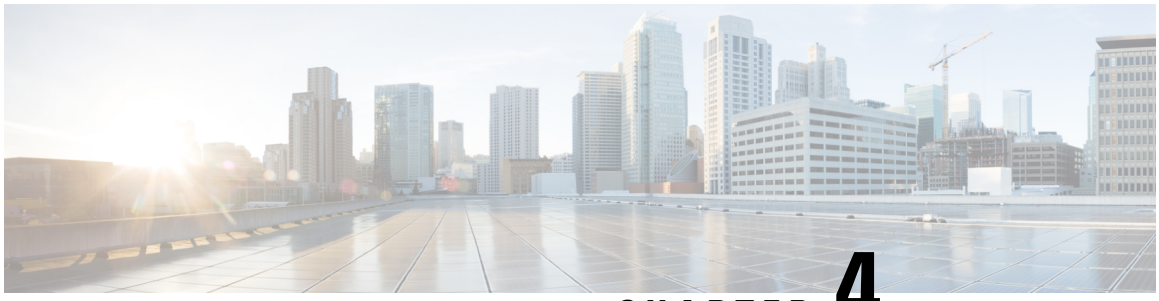
The following example configuration is for vPC switch 2:

```
switch# configure terminal
switch(config)# feature hsrp
switch(config)# feature interface-vlan
switch(config)# interface vlan 600
switch(config-if)# no shut
switch(config-if)# ip address 10.10.10.3/24 <<Assign a unique IP address to the SVI
switch(config-if)# hsrp version 2
switch(config-if)# hsrp 1
switch(config-if-hsrp)# ip 10.10.10.1
switch(config-if-hsrp)# end
```

In these examples, IP address 10.10.10.1 is used as the default gateway for VLAN 600 and whichever logical switch/VNI is mapped to it in the NSX Manager. Both IPv4 and IPv6 addressing are supported for this feature.

A separate SVI and HSRP configuration is created for each VNI/VLAN pair for which the switches will be the default gateway. If the vPC pair of switches is to be used as the default gateway for a logical switch/VNI but the pair does not have a physical workload attached to the vPCs to be mapped to that VNI, a spare vPC and VLAN can be created and assigned to the controller. The vPC must be set up as a trunk in both switches using the **switchport mode trunk** command, but it is not required to have any physical interfaces assigned to it. This vPC and VLAN can then be connected to the logical switch using the hardware binding configuration in the NSX Manager. This association allows

the vPC pair of switches to be a redundant external default gateway for the VNI. The same spare vPC can be connected to several VNIs as long as a different VLAN is used for each VNI.



CHAPTER 4

Configuration Checklist

This chapter includes the following sections:

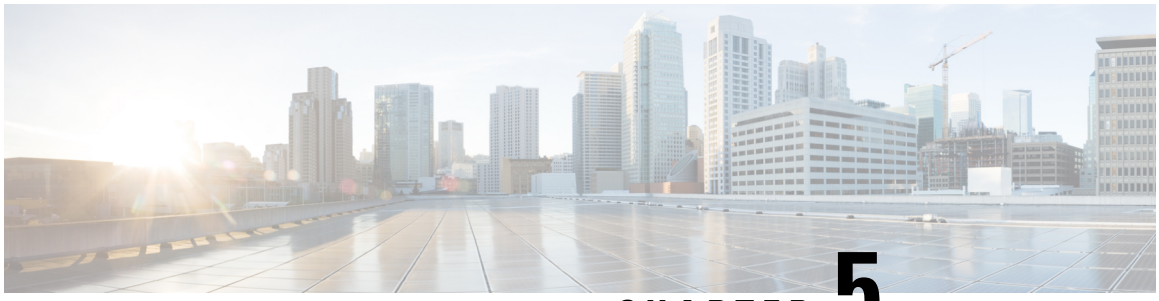
Configuration Checklist

The following configurations (described in the previous chapters) must be completed before the switch can connect to the external controller:

- At least one interface is assigned
- At least one VLAN is assigned
- The **host-reachability protocol controller** command is configured under the NVE interface
- The **config-source controller** command is configured

The switch will not connect to the external controller if any of these conditions is not met.

Configuration Checklist



CHAPTER 5

Configuring Replication Servers

This chapter includes the following sections:

About Replication Servers

Replication servers are dedicated VXLAN tunnel endpoints (VTEPs) that are responsible for replicating Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic. The VTEPs must send the BUM traffic to only one of the replication servers, which will then replicate it to all VTEPs in that virtual network identifier (VNID).

In Cisco NX-OS, replication servers are picked on a per-VNID basis. Therefore, all BUM traffic from a VNID is sent to this replication server.

Rebalancing Replication Servers

Replication servers can arrive at the switch at any time. To help manage the load, you can rebalance the VNIDs across the replication servers.

The first time that the total number of replication servers in the system goes from 0 to 1, the switch waits for 1 minute before programming this information into the hardware. Doing so prevents continuous rebalancing of the VNIDs as the replication servers come down one by one.

Step 1 **nve interface nve 1** {**remap-replication-servers** | **replication-server**}

```
switch#  nve interface nve 1 remap-replication-servers
```

The following options are available:

- **remap-replication-servers** —Immediately rebalances the VNIDs across all available replication servers. If another replication server is sent down after you enter this command, you must re-enter this command to rebalance the replication servers again. This behavior is the default behavior of the switch.
- **replication-server** —Marks the replication server as Up or Down.

Step 2 (Optional) **copy running-config startup-config**

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Setting the Operational State of a Replication Server

If a replication server's operational state is not sent down from the external controller, the switch continues to replicate the BUM traffic to that replication server, which can result in some traffic being lost. To prevent this behavior, you can mark a replication server as being down or up.

Step 1 **show nve replication-servers**

```
switch# show nve replication-servers
Interface Replication Servers  State Ready
--------- -------------------  ----- -----
nve1      10.0.134.1           Up    Yes
          10.0.133.1           Up    Yes
          10.0.140.1           Up    Yes
```

Displays the list of replication servers in the system.

Step 2 **nve interface nve 1 replication-server** *ip-address* [**up** | **down**]

```
switch#  nve interface nve 1 replication-server 10.10.10.1 up
```

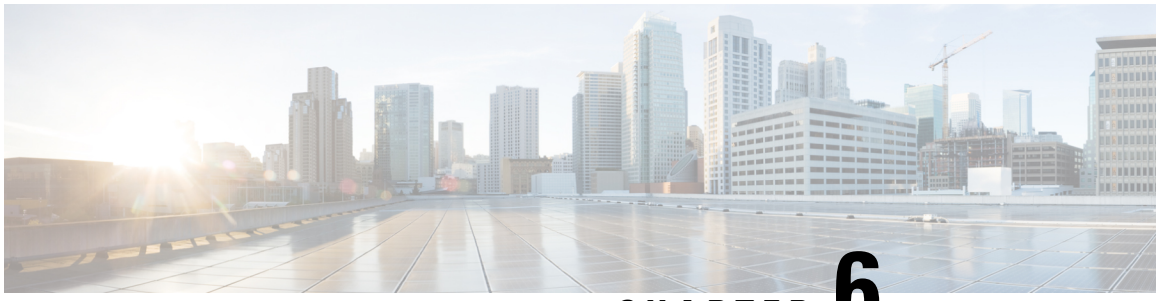
Sets the operational state of the replication server:

- **up**—Marks the replication server as being up. The resulting behavior depends on the replication server rebalancing configuration (either automatic rebalancing or rebalancing once upon CLI execution).
- **down**—Marks the replication server as being down. The VNIDs are immediately rehashed to the replication servers that are up.

Step 3 (Optional) **copy running-config startup-config**

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.



CHAPTER 6

Changing the Scope

This chapter includes the following sections:

Supported Scopes

NXDB supports two scopes:

- Base scope—You can enter configuration and **show** commands in this scope in order to configure the switch and view the configurations. This scope is the default scope and the normal operating mode of the switch. When you enter **show** commands in this scope, you are viewing configurations that are owned by the switch.
- External controller scope—You can enter **show** commands in this scope in order to see the configurations that have been pushed down from the external controller. You cannot enter configuration commands in this scope. When you enter **show** commands in this scope, you are viewing configurations that are owned by the external controller.

Changing the Scope

You can change the scope to view configurations that have been pushed down from the external controller.

Step 1 **switch-scope controller l2-vxlan** *number*

```
switch# switch-scope controller l2-vxlan 1
switch%%ctrlr-1#
```

Changes from the base scope to the external controller scope.

Step 2 (Optional) **end**

```
switch%%ctrlr-1# end
switch#
```

Exits the external controller scope and returns to the base scope.

Determining the Scope

You can determine whether you are in the base scope or the external controller scope by viewing the switch prompt or by entering the **where** command.

- When you enter the **switch-scope controller** command to change to the controller scope, the switch prompt changes to "switch%%ctrlr-1" to indicate the scope in which the **show** commands are being executed.

  ```
  switch# switch-scope controller l2-vxlan 1
  switch%%ctrlr-1#
  ```

- You can enter the **where** command to see the scope in which the **show** commands are being executed. The output varies depending on whether you enter the command in the base scope or the external controller scope.

  This example shows the **where** command entered in the base scope:

  ```
  switch# where
      admin@switch%default
  switch#
  ```

  This example shows the **where** command entered in the external controller scope:

  ```
  switch# switch-scope controller l2-vxlan 1
  switch%%ctrlr-1# where
      admin@switch%default%l2-vxlan-1
  switch%%ctrlr-1#
  ```

Viewing Configurations from the External Controller

You can enter **show** commands in the external controller scope to view the configurations pushed down from the external controller.

Note The **show run** command shows all of the configurations that have been configured using the CLI. The **show run controller** command shows all of the configurations that have been pushed down from the controller.

This example shows how the output of the **show port-channel summary** command varies depending on whether you are in the base scope or the external controller scope, if you have assigned the switch resources for the external controller as follows:

```
controller type l2-vxlan identifier 1
assign vlan 1001-1256 dedicated
assign interface ethernet 1/2-3 shared
```

```
assign interface port-channel 1-10 shared
controller description ctrlr-1
```

The following output displays when you enter the **show port-channel summary** command in the base scope:

```
switch# show port-channel summary
Flags: D - Down P - Up in port-channel (members)
I - Individual  H - Hot-standby (LACP only)
s - Suspended   r - Module-removed
S - Switched    R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-      Type Protocol Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)    Eth  LACP     Eth1/5(P)  Eth1/6(P)
2     Po2(SU)    Eth  LACP     Eth1/7(P)  Eth1/8(P)
3     Po3(SU)    Eth  LACP     Eth1/9(P)  Eth1/10(P)
4     Po4(SU)    Eth  LACP     Eth1/11(P) Eth1/12(P)
5     Po5(SU)    Eth  LACP     Eth1/13(P) Eth1/14(P)
6     Po6(SU)    Eth  LACP     Eth1/15(P) Eth1/16(P)
7     Po7(SU)    Eth  LACP     Eth1/17(P) Eth1/18(P)
8     Po8(SU)    Eth  LACP     Eth1/19(P) Eth1/20(P)
9     Po9(SU)    Eth  LACP     Eth1/21(P) Eth1/22(P)
10    Po10(SU)   Eth  LACP     Eth1/23(P) Eth1/24(P)
101   Po101(SD)  Eth  NONE     --
102   Po102(SD)  Eth  NONE     --
103   Po103(SD)  Eth  NONE     --
104   Po104(SD)  Eth  NONE     --
1001  Po1001(SU) Eth  LACP     Eth1/35(P) Eth1/36(P)
```

The following output displays when you enter the **show port-channel summary** command in the external controller scope:

```
switch%%ctrlr-1# show port-channel summary
Flags: D - Down P - Up in port-channel (members)
I - Individual  H - Hot-standby (LACP only)
s - Suspended   r - Module-removed
S - Switched    R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-     Type Protocol Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)   Eth  LACP     Eth1/5(P)  Eth1/6(P)
2     Po2(SU)   Eth  LACP     Eth1/7(P)  Eth1/8(P)
3     Po3(SU)   Eth  LACP     Eth1/9(P)  Eth1/10(P)
4     Po4(SU)   Eth  LACP     Eth1/11(P) Eth1/12(P)
5     Po5(SU)   Eth  LACP     Eth1/13(P) Eth1/14(P)
6     Po6(SU)   Eth  LACP     Eth1/15(P) Eth1/16(P)
7     Po7(SU)   Eth  LACP     Eth1/17(P) Eth1/18(P)
8     Po8(SU)   Eth  LACP     Eth1/19(P) Eth1/20(P)
9     Po9(SU)   Eth  LACP     Eth1/21(P) Eth1/22(P)
10    Po10(SU)  Eth  LACP     Eth1/23(P) Eth1/24(P)
```

In the external controller scope, the **show port-channel summary** command displays only the external controller-exposed port-channel interfaces.

After you assign the interfaces to the external controller, no VLANs are assigned to them until the external controller pushes some associations down.

This example shows the output of the shared interfaces in different scopes if VLANs 10-1000 are configured on the switch but not assigned to the external controller:

Base scope:

```
interface ethernet 1/2
    switchport mode trunk
    switchport trunk allowed vlan 10-1000
```

External controller scope:

```
interface ethernet 1/2
    switchport mode trunk
    switchport trunk allowed vlan none
```

In the above example, no assigned VLANs are configured on the shared interface.

This example shows the output of the initial state (when the controller is not pushing the port VLAN association):

Base scope:

```
show running-config interface Ethernet 1/1
!Command: show running-config interface Ethernet1/1
!Time: Tue Mar  8 15:11:38 2017

version 7.0(3)I6(1)

interface Ethernet1/1
  switchport mode trunk
  switchport trunk allowed vlan 100-1000
  !controller type l2-vxlan identifier 1
```

External controller scope:

```
show running-config interface Ethernet 1/1
!Executed in context controller l2-vxlan 1

!Command: show running-config interface Ethernet1/1
!Time: Tue Mar  8 15:12:41 2017

version 7.0(3)I6(1)

interface Ethernet1/1
  switchport mode trunk
  switchport trunk allowed vlan none
  !controller type l2-vxlan identifier 1
```

This example shows the output when the controller is pushing the port VLAN association (Ethernet 1/1, 503):

Base scope:

```
show running-config interface Ethernet 1/1
!Command: show running-config interface Ethernet1/1
!Time: Tue Mar  8 15:11:38 2017

version 7.0(3)I6(1)

interface Ethernet1/1
  switchport mode trunk
  switchport trunk allowed vlan 100-1000
```

```
  !controller type l2-vxlan identifier 1
```

External controller scope:

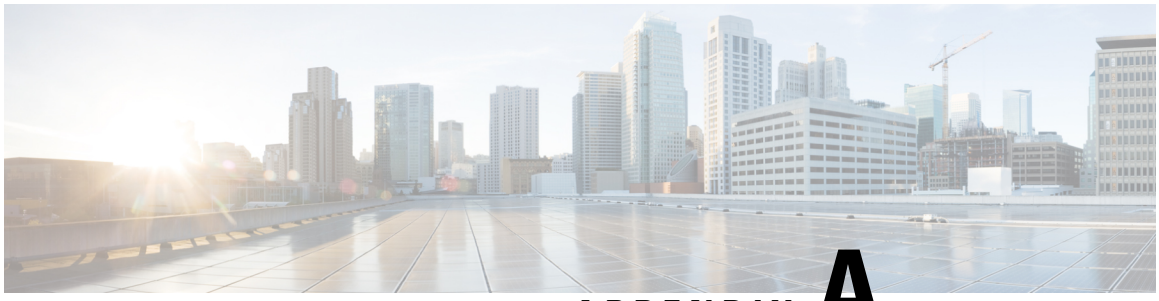
```
show running-config interface Ethernet 1/1
!Executed in context controller l2-vxlan 1

!Command: show running-config interface Ethernet1/1
!Time: Tue Mar  8 15:12:41 2017

version 7.0(3)I6(1)

interface Ethernet1/1
  switchport mode trunk
  switchport trunk allowed vlan 503
  !controller type l2-vxlan identifier 1
```

Note For a list of the **show** commands whose output varies by scope, see Show Command Outputs, on page 69.

Saving the Configuration

When the configuration is pushed down from the external controller, it is not saved into the startup configuration. The external controller is responsible for restoring the configuration as soon as the switch comes up again. You can issue the **copy running-config startup-config** command in either the base scope or the external controller scope, but only the configurations that appear in the **show running-config** command in the base scope will be copied. The external controller scope configurations will not be copied to the startup configuration.

When the configuration is pushed from the switch CLI, Cisco NX-OS is responsible for restoring it. You can issue the **copy running-config startup-config** command to save the configuration that has been pushed from the switch CLI to the startup configuration.



APPENDIX A

Upgrading the Cisco NX-OS Image and the OVSDB Plugin

This appendix includes the following sections:

Upgrading Both the Cisco NX-OS Image and the OVSDB Plugin

You can upgrade both the Cisco NX-OS image and the OVSDB plugin.

Step 1 Copy the Cisco NX-OS image and the OVSDB plugin to the switch.

Step 2 Stop the OVSDB plugin.

```
switch# guestshell run sudo ovsdb-plugin service stop
```

Step 3 Upgrade the Cisco NX-OS image. For instructions, refer to the *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*.

Step 4 To upgrade the OVSDB plugin, follow these steps:

a) Copy the new OVSDB plugin file to the bootflash.

```
switch# copy scp://user@scpserver.cisco.com//download/ovsdb-plugin-2.3.1.rpm
bootflash:ovsdb-plugin-2.3.1.rpm
```

b) Access the Guest Shell prompt.

```
switch# run guestshell
[guestshell@guestshell ~]$
```

c) Find the name of the previously installed OVSDB plugin file.

```
[guestshell@guestshell ~]$ sudo rpm -qa | grep ovsdb
```

d) Remove the previously installed OVSDB plugin file, using the name in the output of Step 4c.

```
[guestshell@guestshell ~]$ sudo rpm -e ovsdb-plugin-2.2.1.rpm
```

e) Install the new OVSDB plugin file.

```
[guestshell@guestshell ~]$ sudo rpm -i /bootflash/ovsdb-plugin-2.3.1.rpm
```

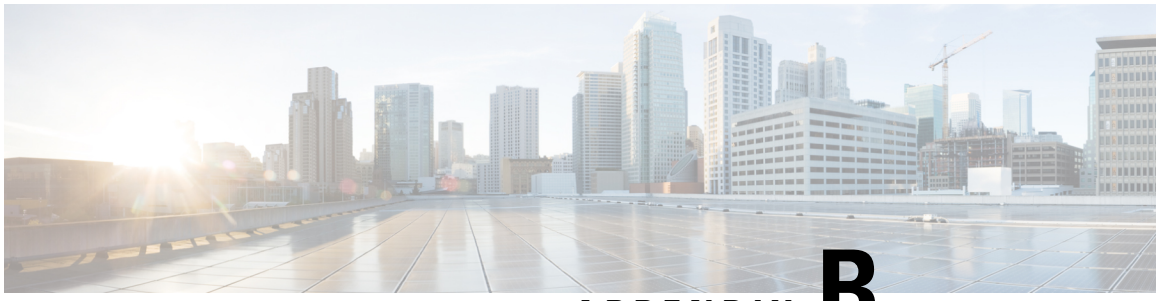
Note With this new installation, the binary file is at /usr/local/ovsdb/bin/, and the logs are at /usr/local/ovsdb/log/.

f) Exit the Guest Shell.

```
[guestshell@guestshell ~]$ exit
```

g) Start the OVSDB plugin.

```
switch# guestshell run sudo ovsdb-plugin service start
```



APPENDIX B

Upgrading the Cisco NX-OS Image and the OVSDB Plugin for vPC Setups

This appendix provides an example of the recommended upgrade sequence for vPC setups. In this example, the switches in the vPC are S1 (primary) and S2 (secondary). The OVSDB plugin on the S1 switch is A1, and the OVSDB plugin on the S2 switch is A2, and the controller-cluster is C.

This example uses the following initial versions and new versions of the components that will be upgraded.

- Initial versions:
  - The controller is running 6.3.x.
  - The OVSDB plugin is 2.1.x.
  - The switches are running Cisco NX-OS Release 7.0(3)I6(x).
- New versions:
  - The controller is running 6.3.x.
  - The OVSDB plugin is 2.2.0.x.
  - The switches are running Cisco NX-OS Release 7.0(3)I7(x).

This appendix includes the following sections. The components must be upgraded in this order:

Prerequisites

- On switch S1, use the **guestshell run sudo ovsdb-plugin service status --more** command to verify that OVSDB plugin A1 is connected to controller C and switches S1 and S2.
- On switch S2, use the **guestshell run sudo ovsdb-plugin service status --more** command to verify that OVSDB plugin A2 is connected to switches S1 and S2, and the connection to C is down.

- Make sure that the new OVSDB plugin RPM and JRE8 RPM images are copied to the /bootflash on both vPC switches.
- Make sure that the new switch image and the new EPLD image are also copied to both vPC switches.

Upgrading the OVSDB Plugin on the vPC Secondary Switch

Step 1 Access the Guest Shell prompt.

```
run guestshell
```

Step 2 Stop the OVSDB plugin.

```
sudo ovsdb-plugin service stop
```

Step 3 Display the currently installed OVSDB plugin.

```
sudo rpm -qa | grep ovsdb
```

Step 4 Remove the OVSDB plugin.

```
sudo rpm -e ovsdb-plugin-
filename
```

Step 5 Display the currently installed JRE.

```
sudo rpm -qa | grep re
```

Step 6 Remove JRE7.

```
sudo rpm -e jre-1.7.0_80-fcs.x86_64
```

Step 7 Install JRE8.

```
sudo rpm -i  /bootflash/jre-8u112-linux-x64.rpm
```

Step 8 Install the new OVSDB plugin.

```
sudo rpm -i /bootflash/ovsdb-plugin-2.3.1.rpm
```

Step 9 Make sure the OVSDB plugin version shows the new version.

```
sudo ovsdb-plugin service version
```

Step 10 Ping the controller IP addresses and switch IP addresses to make sure the connectivity is good.

Step 11 Reconfigure the OVSDB plugin using the appropriate **config set** command.

```
sudo ovsdb-plugin config set --run-in-switch --vrf management
--log-level debug --log-type file 172.31.148.197:6640 172.31.145.141,172.31.145.144
admin,admin
```

```
insieme
,
insieme
 ovsdb-plugin --switch-description ovsdb-plugin
```

Note If there are no changes made to the controller cluster, then the previous certificates should be good and a certificate reset is not required. If the controller has new credentials or is a new install, perform the certificate bootstrap/certificate reset steps.

Step 12 Start the OVSDB plugin.

```
sudo ovsdb-plugin service start
```

Step 13 Verify that the OVSDB plugin connects to switches S1 and S2.

```
sudo ovsdb-plugin service status --more
```

Upgrading the OVSDB Plugin on the vPC Primary Switch

Step 1 Access the Guest Shell prompt.

```
run guestshell
```

Step 2 Stop the OVSDB plugin.

```
sudo ovsdb-plugin service stop
```

Step 3 Display the currently installed OVSDB plugin.

```
sudo rpm -qa | grep ovsdb
```

Step 4 Remove the OVSDB plugin.

```
sudo rpm -e ovsdb-plugin-
filename
```

Step 5 Display the currently installed JRE.

```
sudo rpm -qa | grep re
```

Step 6 Remove JRE7.

```
sudo rpm -e jre-1.7.0_80-fcs.x86_64
```

Step 7 Install JRE8.

```
sudo rpm -i  /bootflash/jre-8u112-linux-x64.rpm
```

Step 8 Install the new OVSDB plugin.

```
sudo rpm -i /bootflash/ovsdb-plugin-2.3.1.rpm
```

Step 9 Make sure the OVSDB plugin version shows the new version.

```
sudo ovsdb-plugin service version
```

Step 10 Ping the controller IP addresses and switch IP addresses to make sure the connectivity is good.

Step 11 Reconfigure the OVSDB plugin using the appropriate **config set** command.

```
sudo ovsdb-plugin config set --run-in-switch --vrf management
--log-level debug --log-type file 172.31.148.197:6640 172.31.145.144,172.31.145.141
admin,admin
password
,
password
 ovsdb-plugin --switch-description ovsdb-plugin
```

Note If there are no changes made to the controller cluster, then the previous certificates should be good and a certificate reset is not required. If the controller has new credentials or is a new install, perform the certificate bootstrap/certificate reset steps.

Step 12 Start the OVSDB plugin.

```
sudo ovsdb-plugin service start
```

Step 13 Verify that the OVSDB plugin connects to switches S1 and S2 and the three controllers.

```
sudo ovsdb-plugin service status --more
```

Upgrading the vPC Secondary Switch

Step 1 Upgrade the Cisco NX-OS image.

```
install all nxos bootflash:
new-switch-image
.bin
```

Step 2 After the upgrade, make sure that the switch goes through the reboot cycle and loads the new image and that all controller configurations are downloaded to the switch.

```
show running controller
```

Step 3 Access the Guest Shell prompt on the S2 switch.

```
run guestshell
```

Step 4 Verify that the OVSDB plugin is running after the upgrade.

```
sudo ovsdb-plugin service status --more
```

Step 5 Exit the Guest Shell.

```
exit
```

Step 6 Upgrade the EPLD image.

```
install epld bootflash:
epld-filename.img
```

Upgrading the vPC Primary Switch

Step 1 Upgrade the Cisco NX-OS image.

```
install all nxos bootflash:
new-switch-image
.bin
```

This step results in a vPC failover. Switch S2 becomes the new primary, and after the reload, switch S1 becomes the new secondary.

Step 2 After switch S2 reboots and loads the new image, make sure it gets all of the configuration. OVSDB plugin A2 pushes the configuration to switch S2.

```
show running controller
```

Step 3 Access the Guest Shell prompt on the S2 switch.

```
run guestshell
```

Step 4 Verify that the OVSDB plugin connects to switch S2.

```
sudo ovsdb-plugin service status --more
```

Step 5 Access the Guest Shell prompt on the S1 switch.

```
run guestshell
```

Step 6 Verify that the OVSDB plugin connects to switch S1.

```
sudo ovsdb-plugin service status --more
```

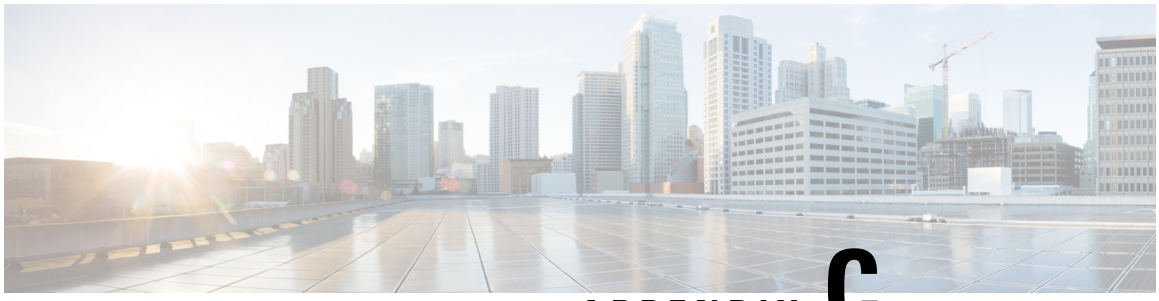
Step 7 Upgrade the EPLD image.

```
install epld bootflash:
epld-filename.img
```

Upgrading the Cisco NX-OS Image and the OVSDB Plugin for vPC Setups

What to do next

If necessary, perform any controller-side upgrades. For information, see the documentation from VMWare.



APPENDIX C

Extended OVSDB Plugin Configuration

If desired, you can fine tune the configuration of the OVSDB plugin using the information in this appendix.

This appendix includes the following sections:

Performing Basic OVSDB Plugin Operations

The OVSDB plugin supports two modes of operation:

- Single switch mode—The OVSDB plugin connects to only one switch and one controller or controller cluster.
- Dual switch mode—The OVSDB plugin connects to a pair of switches and one controller or controller cluster.

To perform basic OVSDB plugin operations, use these commands:

Command	Purpose
guestshell run sudo ovsdb-plugin service start	Starts the OVSDB plugin. **Note** Check for startup errors using **cat nohup.out** or tail the logs using **tail -f log/ovsdb-plugin.log**.
guestshell run sudo ovsdb-plugin service stop	Stops the OVSDB plugin.
guestshell run sudo ovsdb-plugin service restart	Restarts the OVSDB plugin.
guestshell run sudo ovsdb-plugin service version	Displays the version of the OVSDB plugin.
guestshell run sudo ovsdb-plugin config set -h	Displays information on how to configure the OVSDB plugin.

Command	Purpose
service enable	Auto-starts the OVSDB plugin on switch bootup.
service disable	Disables auto-start of the OVSDB plugin on switch bootup.

This example shows the OVSDB configuration help:

```
switch# guestshell run sudo ovsdb-plugin config set -h
usage: ovsdb-plugin config set [-h]
                                        [--switch-description SWITCH_DESCRIPTION]
                                        [--keep-test-config]
                                        [--log-level {error,warn,info,debug,trace}]
                                        [--log-server ADDRESS]
                                        [--log-type {file,udp}]
                                        [--max-json-peers INT]
                                        [--run-in-switch]
                                        [--schema-version {1.3.99,1.3.0}]
                                        [--vrf NAME] [--xms INT] [--xmx INT]
                                        [-v]
                                        ct_addr1[,ct_addr2]
                                        sw_addr1[,sw_addr2] user1[,user2]
                                        pswd1[,pswd2] name

positional arguments:
  ct_addr1[,ct_addr2]   Controller cluster address in IP:PORT format. Port
                        defaults to 6632 if not included. To specify two or
                        more controllers, separate them with a comma (no
                        spaces). E.g. 10.21.1.10:6632,10.21.1.11:6632
  sw_addr1[,sw_addr2]   Switch address in IP:PORT format. Port defaults to 443
                        if not includes. In VPC mode, specify two switches by
                        separating them with a comma (no spaces). E.g.
                        10.21.1.10:443,10.21.1.11:443
  user1[,user2]         Switch username(s). To specify a different username
                        for a second switch separate with a comma (no spaces).
                        E.g. user1,user2. If no username is given for the
                        second switch, the first one will be used
  pswd1[,pswd2]         Switch password(s). To specify a different password
                        for a second switch, separate with a comma (no
                        spaces). E.g. pswd1,pswd2. If no password is given for
                        the second switch, the first one will be used
  name                  Switch name.

optional arguments:
  -h, --help            show this help message and exit
  --switch-description SWITCH_DESCRIPTION
                        Switch description.
  --keep-test-config    Keep the test config
  --log-level {error,warn,info,debug,trace}
                        Log level to use. Defaults to info
  --log-server ADDRESS  When --log-type is set to UDP, use this to specify the
                        remote where the logs will be sent. The address must
                        be in IP:PORT format. PORT defaults to 514 if not
                        included
  --log-type {file,udp}
                        The type of logging to use. When set to file, the path
                        is always PLUGIN_ROOT/log/ovsdb-plugin.log. Defaults
                        to file
  --max-json-peers INT  Maximum number of JSON peers. Defaults to 6 if not
                        included. Set to -1 to have the plugin auto-compute
                        the value based on the number of controllers
```

```
--run-in-switch       Configures the plugin for running in the switch
--schema-version {1.3.99,1.3.0}
                      Set the schema version to use. Defaults to 1.3.0
--vrf NAME            Used only when --run-in-switch is set. This configures
                      the plugin to use the given VRF name when
                      communicating with the controller. Defaults to
                      management
--xms INT             Set initial Java heap size in MB. Defaults to 2048
--xmx INT             Set maximum Java heap size in MB. Defaults to 2048
-v, --verbose         Show extended information
```

Generating the SSL Keys

Step 1 Choose one of these methods to generate the SSL keys:

- In non-vPC mode without Guest Shell or remote vPC mode, enter the **sudo ovsdb-plugin cert bootstrap** command and skip the remaining steps.
- In non-vPC mode with Guest Shell, enter the **guestshell run sudo ovsdb-plugin cert bootstrap** command and skip the remaining steps.
- In local vPC mode, enter the **guestshell run sudo ovsdb-plugin cert bootstrap --receive** command on the first instance, and go to the next step.

Step 2 Copy the temporary certificate displayed on the terminal.

Step 3 Enter the **guestshell run sudo ovsdb-plugin cert bootstrap --send IP_OF_FIRST** command on the second instance.

Step 4 Paste the temporary certificate when prompted.

Note You can also pipe in the temporary certificate to the second instance (for example, **cat tempcert.txt | guestshell run sudo ovsdb-plugin cert bootstrap --send**).

What to do next

If you ever need to reset the SSL keys, follow the previous steps but enter the **guestshell run sudo ovsdb-plugin cert reset** command in place of the **guestshell run sudo ovsdb-plugin cert bootstrap** command.

Configuring the Startup config.json File

Normally, the startup config.json file is created and edited by the CLI code (using information provided by the CLI user). The following sections document many configuration file settings that are not directly supported by the CLI.

Caution Pay serious attention to the syntax when directly editing the configuration file. For example, an extra space can cause the application to be unable to run.

On startup, the application validates the configuration file. If the configuration file is invalid, the application sends an alert to the console, and the output is captured in the nohup.out file. If the application fails to start

after directly editing the file, review the nohup.out file. It should provide suggestions as to what part of the configuration file has changed.

Always make a backup of the config.json file before directly editing the file. For nearly every value, a restart of the application is required for the application to fully act upon the changed values. The exceptions are the default log level and the switch username and password. To have the running application see a change to the default log level or the switch usernames and passwords, you must run the **ovsdb-plugin service reloadconfig** command.

The startup config.json file supports the following sections:

- loggingInfo—Specifies whether and how to log.
- tlsInfo—Specifies SSL keys and certificates.
- systemInfo—Specifies db schema and running-in-switch flags.
- supportedSwitchType—Specifies the supported switch types.
- switchInfo—Contains information on the single switch to be used.
- managerInfo—Contains information on the inbound and outbound connections that the OVSDB plugin will support.
- otherInfo—Specifies optional flags and settings.

loggingInfo

Several different styles and configurations can be configured. This application uses the Log4j 1.2logging library. Many of the settings describe values that the application will use to auto-build a Log4j configuration file, which is used to create the applications logging. Logging can be turned off, written to disk, sent remotely (via SysLog), or configured in a separate Log4j-styled configuration file.

To turn off all logging:

```
disableAllLogging: true
        - Disables all logging.
        - No other settings are needed, validated, or used.
```

To write log files to disk:

```
loggingStyle: 1
        baseDir: "/path/to/base/dir/no/ending/slash"    <<-- note: on Windows, use / not \
        maxFileSizeInMb: 5
        maxNumFiles: 10
```

Details on each setting:

```
disableAllLogging: values are true or false.  Default = false.
        - Setting this disables all logging.
        - This setting will override ALL the other logging settings.

    loggingStyle: this is an integer value between 1 and 31. Default = 1
        - the value can represent multiple settings
        - the value represents the AND-ing of one ore more bit values
        - however, some settings are EXCLUSIVE (i.e., cannot be combined)
        - The standard settings are:
            1 - simple logging to local file
        * location is specified in baseDir [see below]
```

```
            4 - Syslog Logging
          * Host location is specified in sysLogHost [see below]
          * Facility is specified in sysLogFacility [see below]
             16 - Log4J Logging
          * User-specified log4J file is specified in loggingConfigFile [see below]
         - Note that Log4j Logging is "exclusive" in that it cannot be combined with another

           logging style.  However the other settings can be "mixed and matched".
            - For example, a setting of 5 would mean Syslog logging AND Simple Local Logging.

              - Note also that 17 would NOT mean simple local logging AND Log4j logging
                because Log4j logging is exclusive.

    The ovsdb plugin uses an asynchronous logging facility:
    - Logging will asynchronously be written to the log location
    - Note that async also means that if the logging device is full then the application
          will continue even though it cannot write to the log.
        - Note that the asyncBufferSize setting controls how many messages can be buffered
        - Scenario:
        - Application is running.
        - Logging device becomes full
        - A count of AsyncBufferSize more logging statements are made
        - The buffer is now full; this FIFO buffer will now beging dropping logging
statements.
       - The running application continues on; it is not affected by the device full issue.

        - The current contents of the async buffer will be flushed to the logging device
        - Logging will continue.
        - The running application continues on; however, now it is successfully logging.

    asyncBufferSize: number of messages that can be buffered IF async = True
    - Once this number of messages are buffered, the system will begin to drop logging
statements
    - Successful logging will resume once the logging device is no longer full.

    baseDir: "/path/to/base/dir/no/ending/slash"
        - REQUIRED value if loggingStyle AND 1 = 1 (i.e., simple local logging is set)
        - On windows, use / and not \
        - Should NOT end with a trailing slash
       - Note that although this is called "local" logging, it just have to be a path that
 is accessible
         to the application; so the actual device can be "remote" yet locally accesible.

    defaultLogLevel: logLevel
        - REQUIRED value
        - logLevel must be one of FATAL, ERROR, WARN, INFO, DEBUG, TRACE
        - Capitalization is not significant
        - This sets the DEFAULT log level for all application logging.
      - Logging on a case-by-case basis can be overridden by usage of the overrideLogLevels
 section [see below].

    loggingConfigFile: path to the "override" log4j config file
        - Most of the rest of the settings in the LoggingInfo section describe
          settings that will be used to "auto generate" a Log4j-style config file.
        - However, the user can decide to hand-carve their own config file.
        - To do this:
        - Set logging style = 16 (aka Log4j Logging)
        - provide a path to your own config file via the loggingConfigFile setting
        - The application will only validate that a file exists at this location.
          It will NOT do ANY validation of the contents of the file.  It will simply
          tell the Log4j logging system where your config file is.

    overrideLogLevels: this section allows the DEFAULT log level to be overridden
        - This section can consist of 1 or more overrides.
```

```
        - You can override the default by going higher or by going lower.
        - You can override the log level by referencing a java class or family of classes.
        - For example, there is a class called:
com.nexus.andromeda.ovsdb.jsonrpc.util.JJsonPeer
        - You could set an override directly on that class, or you could set an override
          in the com.nexus.andromeda.ovsdb.jsonrpc.util package or at the
          com.nexus.andromeda.ovsdb.jsonrpc.util level.
        - Therefore, the label "CLASSNAME" is kind of mis-named.
        - As with everything in this file, exact spacing is CRITICAL. (Never use tabs).
        - For example, here is a default override:
           {
                "className": "org.apache.commons.beanutils",
                "level": "FATAL"
            }
        - This shows that the class or package "tree" (you can't really tell here) named
          org.apache.commons.beanutils should be set to FATAL level.
        - In this case, the reason is that package is incredibly "noisy" and we only ever
          care about this package if it logs something at the FATAL level.

    maxFileSizeInMb: integer value for max file size before roll over
        - REQUIRED value if loggingStyle is localDebug or localBuffered

    maxNumFiles: integer max number of files to roll over
        - REQUIRED value if loggingStyle is localDebug or localBuffered
        - NOTE: Max size of all log files will be
          (maxFileSizeInMb x maxNumFiles) in MB.

    messageFiltering: this section provides the ability to filter out specific messages
        - There are certain logging statements that occur fairly frequently.
        - These messages are specially programmed to read this messageFiltering section.
        - The theClass field describes the class generating the logging statement.
        - The regex describes the line to be "filtered out".
        - The count indicates how often to let the statement show up in the logs.
        - For example, a count of 10 means only write this statement to the log every 10th
          time it occurs.

    sysLogHost - the Host IP of the computer that is support syslog logging
        - Only applicable if loggingStyle AND 4 = 4 (i.e., syslog logging is set)

    sysLogFacility - the facility that is supporting syslog logging
        - Only applicable if loggingStyle AND 4 = 4 (i.e., syslog logging is set)
```

tlsInfo

```
tlsInfo:
    privateKeyFile:      "C:/ovsdb-plugin/config/jay-server.p12"
    certificateFile:     "C:/ovsdb-plugin/config/ContrllerCaCert.der"

    privateKeyPassword:  "ovsdb-plugin"
    certificatePassword: "mypassword"

    # optional flag
    turnOnSSLDebugging:   true
```

systemInfo

```
systemInfo:
    schemaVersion: value
    runningInSwitch: boolean-flag
```

```
# Current legal schema versions are "1.3.0" and 1.3.99"

# Running in switch command indicates whether or not we are
configured to run in the switch.  This affects how certain
decisions are made in the system.
```

supportedSwitchType

Defines an array of all the legal types of switches to be used.

```
supportedSwitchType:
    - switchType: "N9K-C9396PX"
    - switchType: "N9K-C93128TX"
    - switchType: "N9K-C9372TX"
    - switchType: "N9K-C9332PQ"
    - switchType: "N9K-C9372PX"
   etc.
```

Note You can use a wildcard in the switchType definition, which would replace all of the previous definitions.

```
supportedSwitchType:
    - switchType: "N9K-C93*"
```

This definition matches all of the switches that start with "N9K-C93". Alternatively, the SwitchType could be set to just *, which would match all switches. However, the wildcard character can occur only as the last character in the string, so "N9K-C93*X" would not be supported.

switchInfo

Defines an array of the single switch to run against.

```
- host: 10.21.111.17
          user: admin
          password: mySwitchPassword
          certificateFile: path-to-CA-certificate-file
```

managerInfo

Defines an array of the connections that the OVSDB plugin will support.

```
managerInfo:
    - name: "controller"
      target: ptcp
      port: 6640
      max_backoff: 1000
      inactivity_probe: 0
```

Details on each setting:

```
name: theName
        - Required value defining the name of this connection.
```

```
    - Both configured and learned controllers must have the word "Controller" in their
      name.

target: theTarget
    - Required value describing the connection.
    - One of PTCP, TCP, PSSL or SSL.
       - TCP and PTCP are TCP connections.
       - SSL and PSSL are SSL connections.
       - The leading p indicates that the OVSDB plugin is to act as the server.
       - The lack of p indicates that the OVSDB plugin is to act as the client.

port: integer portNumber
    - Required port number value.

max_backoff: integer backValue in Milliseconds
    - Required value.
    - Indicates how long the OVSDB plugin should wait after a broken connection before
      it establishes a new connection.

inactivity_probe: integer inactivity_probe in Milliseconds
    - Required value.
    - Indicates how long the OVSDB plugin should wait when not receiving any data on a
      connection before it sends out an echo request.
    - Once an echo request is sent, if this same amount of time expires without a
      response, the OVSDB plugin will close the socket connection.
   - For example, if inactivity_probe is set to 60000, the OVSDB plugin, after waiting

      for 120 seconds with no data received, will close the socket.
  - A value of 0 indicates no inactivity probes should be sent on the given interface.
```

otherInfo

This class allows last-second configuration without having to generate or update class files to support the new fields. It is both YAML and Json compatible, and it allows you to treat non-YAML-standard embedded strings as key-value pairs.

Note The flags must include double quotes (for example, flag: "*VariableName*: 123" and flag: "*AnotherVariableName*: x45").

Currently defined values (flags in the current config.json file):

```
- flag: "StartUp_PopulateDb: True"
       * This is used by the ovsdb-plugin startup code.  If not present, the value defaults
 to False.
         This value tells the plugin to attempt to build its own OVSDB based on existing
switch configuration.
         Sometimes it might be useful to toggle to False to allow the plugin to take a
fresh
         configuration from the controller.
       * The default value is True.
```

Other available configuration flags that are supported but are not currently being set (that is, the default values are being used):

```
- flag: "DB_minElapsedTimeBeforeLogging: NNN"
       * At one point in time, we were concerned about database performance.
         This flag allowed us to track every database call that took MORE than
```

```
          a certain time.  For example, setting this value to 400 would cause any
          db call that took MORE than 400 ms to be logged (so it could be tracked).
        * The default value is 150 aka 150 msecs.

   - flag: "JP_HowOftenToLogSocketConnectFailuresInMs: NNNNN"
        * For each connection (or ManagerInfo) in the config.json file, certain code
          will attempt to make a socket connection to each of those records.
         For some connections (like the utility connection used to support the ovsdb cli),

          it is VERY VERY likely NOT to make a connection.  However, each connection
          failure causes logging.  This setting HOW OFTEN (in ms) we should SHOW an error.
       Use of this feature will reduce the amount of uninformative logging data generated.

        * Note that the current defalut value is grossly = Show Errors every 20th time.

   - flag: "JP_OffsetFromInactivyProbeTimeInMs: NNNNN"
        * This is a millisecond value used by the JSON Peers.
        * The ManagerInfo record in the config file (as well as the data conveyed to us by
 a controller)
          contains an InactivityProbe value.
       * This value describes how long (in milliseconds) it will take of inactivity before
 the
          specified connection will send us a probe (aka echo) message "to keep the socket
 warm"
          (i.e., to maintain the connection".
        * The JSON Peer uses this same value.  If it sees this InactivityProbe amount of
time without
          any activity, it will decide to send an echo.
        * So, back to this flag.  This flag allows us to control who will send the echo.
          (because, basically, both sides are "agreeing" to send it at, say, 60 seconds;
so it is
        INCREDIBLY arbitrary which "end" will end up sending the actual echo/probe message).

        * This configured value ends up being SUBTRACTED from the configured (or provided)
 inactivity
          probe value and it is this calculated value that ends up being used in the JSON
Peers
          calculations.
        * (Assume here that the configured Inactivity Probe Time = 60 seconds).
        * So, if you set this value to 2000, you are telling the JSON Peer to use, say,
60-2 or 58
          seconds as its calculated value.  That means, most likely, that the JSON Peer
will end up
          sending the echo to the controller.
        * Alternatively, you could set this value to -2000.  This means that the JSON Peer
 will use
         ( 60 - (-2) ) or 62 seconds as its probe time.  This would mean that, most likely,
 the
          controller will end up sending echoes to the JSON Peer.
        * This value defaults -2000 aka -2 seconds aka the controller will probably send
us echoes.

   - flag: "JP_ManageJsonPeersLoopTimeInMs: NNNNN"
        * This millisecond value determines what the delay (in milliseconds) between
subsequent loops
          for the ManagerJJsonPeers code should be.
        * This value defaults to 1500 (aka 1.5 seconds).

   - flag: "JP_ReportManageJJsonPeerStatusTimeInSeconds: 20000"
        * This SECONDS value determines approximately how often the "Manager Json Peer
logging report"
          should be generated.
        * This should be a value more than 45 seconds, since the "report" is not really
that useful.
```

Extended OVSDB Plugin Configuration

```
       * Also, this thread is constantly running; setting this to N seconds means that you
 will get a
         1 line log of ManagerJJsonPeer status information FOR THE ENTIRE ANDROMEDA RUN.
       * This value defaults to 120 which means you would get a report every 2 minutes.
       * Setting this value to 0 means generate no reports.

   - flag: "JP_ReportThreadCountTimeInMs: NNNNN"
       * The system has an ability to occasionally generate a log statement which
         indicates how many Threads are currently in use and what the peak usage has been.

         This flag defaults = -1 = do not display.
       * Note that this is not a "fine tuned" time value; I would suggest not to expect
         any more accuracy (or granularity) than about a second (aka 1000).
       * I have typically set this to 10000 or 15000 (aka 10 or 15 seconds).
       * The log statment would look like:
            (timestamp) [ManageJJsonPeers] INFO  jsonrpc.util.ManageJJsonPeers | Java
Thread Count:
         Current = 30, Peak = 33

    - flag: "JP_ReportThreadInfoTimeInMs: NNNNN"
       * The system has an ability to occasionally generate a log statement which
         provides *detailed* information on *every* thread that is currently running.
      * This option takes a bit of time to generate and can easily generate over 100 lines
 of logging output.
         Therefore, it is felt that this should not be set to any value less than about
20000.
       * This flag defaults = -1 = do not display.
       * Note that this is not a "fine tuned" time value; I would suggest not to expect
         any more accuracy (or granularity) than about a second (aka 1000).
       * I have typically set this to 20000 or 30000 (aka 20 or 30 seconds).

    - flag: "JP_SocketTimeoutValue: 60000"
       * This is used by the JJsonPeer to set the SocketTimeout value used by the JSON
Peers.
       * This value defaults to 0.
       * 0 means NO TIMEOUT (i.e., block forever)

    - flag: "JPSR_SingleSocketReadSize: 10000"
       * This value determines the size for each read attempt on the JSON Peer socket".
       * A too large value will slow the average read times down; a too small value will
cause too many
         reads to be performed.
       * There is no "absolultely correct" value.
       * However, values < about 2000 or > about 20000 probably are not very efficient.
       * If not set, this value defaults to 5000.

    - flag: "JpSupport_ThreadPoolSize: 10"
       * This value determines how many threads are used to support JSON Peer support
threads.
       * If not set, this value defaults the "maximumNumJJsonPeerThreads" value + 2.

    - flag: "JpSupportJP_WaitPoolTermination: 5"
       * This weirdly-named value determines how long the Jp Support shutdown code waits
for the thread
         pool to terminate.
       * If not set, this value defaults to 3.

    - flag: "JP_TimeBetweenLongRunningCmdsLoggingInMs: NNNNN"
       * When the JJSonPeer receives a command, it passes it to the "SA" to determine a
response.
         Another thread (specifically, ManageJJsonPeers) watches over the JJsonPeers.
         If this value is set to NNN, any time we are waiting for LONGER than NNN
milliseconds
         for the SAL to generate a response, a log statement would be generated.
```

```
        * This defaults to 10000 aka 10 seconds.
        * This log statement will look something like:
         (timestamp) [(who)] WARN  jsonrpc.util.JJsonPeerStatus | JP=(connection descriptor):
 has been
          building a Response to a rcvd command for (amountOfTime)  ms. !!!!!!

    - flag: "LoadDbFromSwitchMaxWaitTimeInMs: NNNN"
        * This value is only to be used in a Worst Case Scenario.
        * The value indicates the Maximum Time that the "Load DB" code should take under
ANY circumstances.
        * If the code EVER takes this long,
            - the code waiting for this process to complete will be returned to
            - a failure will be returned.
        * This integer number of milliseconds value defaults to 5*60*1000 (aka 5 minutes).

    - flag: "LoadDbFromSwitchSingleSleepTimeInMs: NNNN"
        * This value indicates how long each sleep time should be while waiting for the
"Load DB" code to complete.
        * If the value is configured too small the code waiting will spend too much CPU
waiting and looping.
        * If the value is configured too large then it is likely that (on average) half of
 the configured time
          will be "wasted" because the code is done but the waiting code is unneccessarily
 sleeping too long.
        * This integer number of milliseconds value defaults to 500.

    - flag: "MAIN_DelayBeforeRunMatrixInSecs: NN"
        * This value is used by the main routine (AndromedaMain) to determine how many
seconds to delay
          (after everything else is done) before calling The Matrix.
        * This integer number of seconds value defaults to 0.

    - flag: "maximumNumJJsonPeerThreads: NNN”
        * This is used by the JJsonPeerThreadBoss to determine the maximum
          number of JJsonPeer threads that should exist at any one time.
        * Note, without this flag present, (or if it has a value of -1) the value is auto
calculated.

    - flag: "NOTIF_ThreadPoolSize: 10"
        * This value determines how many threads are used to support the internal event
Notifiation system.
        * If not set, this value defaults to 6.

    - flag: "NOTIF_WaitPoolTermination: 5"
       * This value determines how long the notification shutdown code waits for the thread
 pool to terminate.
        * If not set, this value defaults to 3.

    - flag: "NX_AaaRefreshCycleInSecs: NNNNN"
       * This value determines how often the "AAA Refresh" (to the switch) should be done.

        * This value defaults to 540 seconds.

    - flag: "NX_AaaRefreshStartDelayInMsecs: NNNNN"
        * This value determines a start delay the "AAA Refresh" (to the switch) should be
done.
        * This value defaults to 5500 seconds.

    - flag: "NX_CheckConnectivityTimeout: NNNNN"
       * This millisecond value determines how long to "sleep" between subsequent attempts
 to connect
          to the switch.
        * This value defaults to 10000 (aka 10 seconds).
```

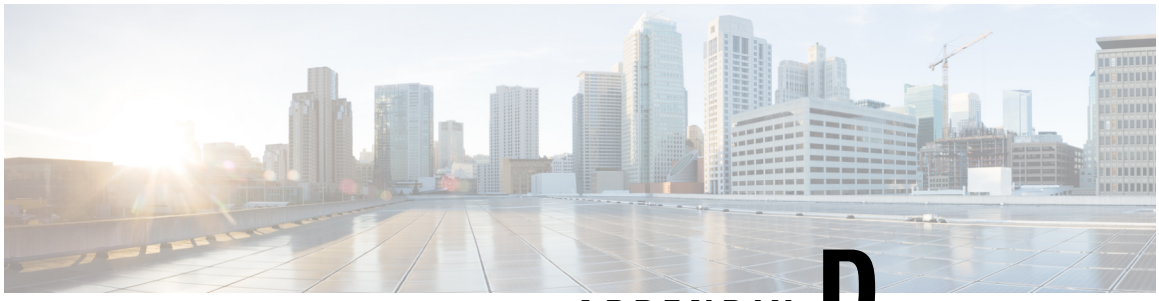
```
    - flag: "NX_ConnectionRetryCount: N"
       * This value determines how many times several low-level NXAPI calls are made BEFORE
 an error is logged.
        * The affected routines (in the NXAPITLServiceImpl class) are:
           + attemptToReconnectAsNeeded
           + ensureConnectedToWebsocket
           + checkSocketConnectivity
        * This value defaults to 2.

    - flag: "NX_ConnectTimeout: NNNNN"
        * This millisecond value determines what the connect timeout value for the socket
to the switch should be.
        * This value defaults to 20000 (aka 20 seconds).

    - flag: "NX_ReadTimeout: NNNNN"
        * This millisecond value determines what the read timeout value for the socket to
the switch should be.
        * This value defaults to 20000 (aka 20 seconds).

    - flag: "Port_Vlan_Batching_Config: NNN"
        * This value determines how many port-vlan pairs are gathered together in a single
 batched NXAPI call.
       * The code sends the SMALLER of (the num avail and this configured size) in a single
 NXAPI call.
        * This value defaults to 100.

    - flag: "Vnid_Vlan_Batching_Config: NNN"
        * This value determines how many vlan-vnid pairs are gathered together in a single
 batched NXAPI call.
       * The code sends the SMALLER of (the num avail and this configured size) in a single
 NXAPI call.
        * This value defaults to 100.

    - flag: "SHUT_OkToStopLoggingAtShutdown: False"
      * This TRUE/FALSE value determines whether or to suppress all logging once a shutdown
 occurs.
        * If not set, this value defaults to TRUE.

    - flag: "StartUp_PopulateDb: True"
        * This flag is used by the "Load DB" code to determine whether or not the code
should run at all.
        * If not set, this value defaults to TRUE.
        * If set to False, "Load DB" simply returns without doing anything.

    - flag: "StartUp_SystemReadyDelayInMs: 60000"
        * This is used by the ovsdb-plugin code.  It used to be used at startup, but now
it is used
          whenever we need to wait for a given switch to return a good ConfigReady value.
        * If not present, the value defaults to 10000.
        * Note that there is no range-checking on this value.

    - flag: "SUB_GoodSubscriptionRefreshRateInMsecs: NNNNN"
       * This millisecond value determines approximately how often the Subscription Refresh

          threads run "when things are going well".
        * This value defaults to 40000 (aka 40 seconds).

    - flag: "SUB_BadSubscriptionRefreshRateInMsecs: NNNNN"
       * This millisecond value determines approximately how often the Subscription Refresh

          threads run "when things are going poorly".
        * It seems like this value should be smaller than the "good" value
          of SUB_GoodSubscriptionRefreshRateInMsecs.
        * This value defaults to 15000 (aka 15 seconds).
```

```
    - flag: "SUB_NumErrsBeforeDeclareBad: N"
        * This counter value determines how many consecutive errors a Subscription Refresh
          Thread should see before it declares the NXAPI connection as down.
       * Note that crossing this threshhold puts the Subscription Refresh thread in "probe
 mode",
          where it is in a loop, probing the nxapi connection and waiting for it to come
up.
        * See NumSuccessesToEscapeProbeMode for related info.
        * This value defaults to 4.

    - flag: "SUB_NumSuccessesToEscapeProbeMode: 5"
       * This counter value determines how many consecutive successes a Subscription Refresh

          Thread should see (once it has declared the connection down and is in "probe
mode").
        * See SUB_NumFailuresToDeclareBad for related info.
        * This value defaults to 5.

    - flag: "SUB_SocketConnectTimeoutInMsecs: 60000"
        * This is used by the SubscriptionRefreshThread as its "connection timeout" value.
       * This value is only used by the Single Subscription code in the Subscription Refresh
 Thread.
        * This value defaults to 20000 aka 20 seconds.

    - flag: "SUB_SocketReadTimeoutInMsecs: NNNNN"
        * This millisecond value determines what the read timeout value for the socket to
the switch should be.
       * This value is only used by the Single Subscription code in the Subscription Refresh
 Thread.
        * This value defaults to 20000 (aka 20 seconds).

    - flag: "SUB_ProbeModeDelayInMsecs: NNNNN"
       * This millisecond value determines how often the "probe the nxapi connection until
 it comes back up"
          code will loop.
        * This value defaults to 10000 (aka 10 seconds).

    - flag: "TL_ConnectionDownSleepInMs: NNNNN"
       * This millisecond value is used by the AsyncTaskWork (which sends batch information
 to the switch).
        * The AsyncTaskWorker, once it finds the NXAPI connection to be down, it sits in a
 loop and keeps waiting
          for an UP.
        * This value is the sleep time (in milliseconds) between subsequent connection
checks.
        * This value defaults to 20000 aka 20 seconds.
```

The following flags are not supported:

```
- flag: "DoNewAndromedaStatusFormat: True"
    - flag: "JP_NumBadMsgCounterThreshHold: 10"
    - flag: "JP_NumEndOfStreamReadsThreshHold: 10000"
    - flag: "JP_NumGenSocketExceptThreshHold: 10"
    - flag: "JP_NumSocketTimeoutExceptThreshHold: 10"
    - flag: "NX_MaxDebugLogOutput: NNNNN"
    - flag: "Startup_SystemReadyIgnoreResults: False"
    - flag: "StartUp_SystemReadyNumTries: 3"
    - flag: "VPC_BeanRequestDelayInSecs: NNN"
    - flag: "VPC_BeanRequestTimeoutInSecs: NNNN"
    - flag: "VPC_DoWeAllowNonVpcSwitches: True"
    - flag: "VPC_HowManyBeanErrsBeforeDeclarePrimary: NNN"
```

Extended OVSDB Plugin Configuration



APPENDIX D

Best Practices for NXDB

This appendix includes the following sections:

Configuring CoPP

You can classify communication between NXDB and the external controller in a Control Plane Policing (CoPP) management class. Doing so avoids congestion of the default CoPP class that affects the control communication.

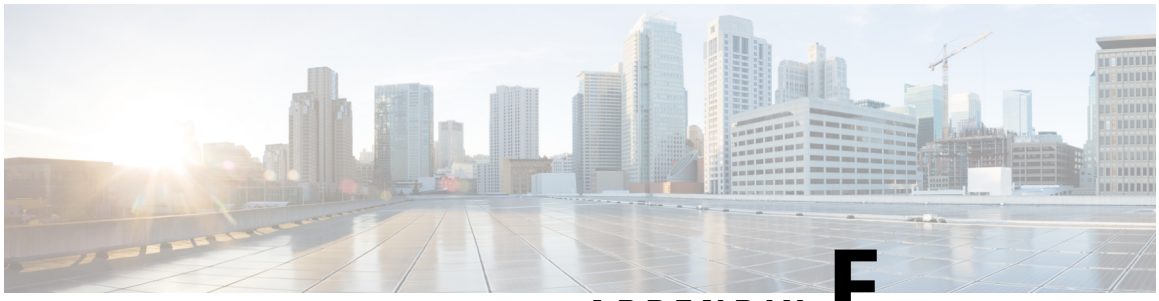
```
copp copy profile strict prefix controller
ip access-list controller-copp-acl
  10 permit tcp any any eq 6640
  20 permit tcp any eq 6640 any
  30 permit tcp any any eq www
  40 permit tcp any eq www any
  50 permit tcp any any eq 443
  60 permit tcp any eq 443 any

class-map type control-plane match-any controller-copp-class-management
  match access-group name controller-copp-acl
control-plane
  service-policy input controller-copp-policy-strict
```

Clearing MAC Addresses

All MAC addresses, whether local or remote, are installed as dynamic MAC addresses. The local MAC addresses are learned from the data plane and have aging enabled on them. The remote MAC addresses are learned from the external controller and have aging disabled on them. MAC addresses learned from the external controllers can be deleted from the system (from both Cisco NX-OS and the Cisco Nexus hardware) only when the external controller sends a delete message.

To delete the local and remote MAC addresses from the hardware and then redownload the remote MAC addresses to the hardware from the object store database maintained in Cisco NX-OS, enter the **clear mac address-table dynamic** command from the exec configuration mode on the switch.

To delete the remote MAC addresses from NXDB, the object store, and the hardware, enter the **guestshell run sudo ovsdb-plugin db del ucast-macs-remote --mac 00:00:00:00:00:01 --vlan 1001** command in NXDB. The OVSDB RFC causes the external controller to redownload the MAC addresses to NXDB.



APPENDIX E

Show Command Outputs

This appendix includes the following sections:

Show Command Outputs

The output of these **show** commands varies depending on whether you are in the base scope or the controller scope:

- **show controller** *controller-number* **accounting log**
- **show interface**
- **show interface brief**
- **show interface ethernet switchport**
- **show interface ethernet trunk**
- **show interface port-channel switchport**
- **show interface port-channel trunk**
- **show interface switchport**
- **show interface trunk**
- **show mac address table**
- **show port-channel summary**
- **show running-config**
- **show vlan**
- **show vlan brief**
- **show vpc**
- **show vpc consistency-parameters global**

This appendix provides sample outputs for these commands in both the base scope and the controller scope so that you can see the differences.

show controller accounting log

Base scope and controller scope:

```
switch# show controller 1 accounting log

2018-04-23 23:07:39,763 uWSGIWorker1Core1 - 10.23.237.22 setBdAssociations(
2018-04-23 23:07:39,770 uWSGIWorker1Core1 - 10.23.237.22 vlan-201,vxlan-10002
2018-04-23 23:07:39,770 uWSGIWorker1Core1 - 10.23.237.22 vlan-203,vxlan-10004
2018-04-23 23:07:39,770 uWSGIWorker1Core1 - 10.23.237.22 vlan-202,vxlan-10003
2018-04-23 23:07:39,771 uWSGIWorker1Core1 - 10.23.237.22 )
2018-04-23 23:07:39,945 uWSGIWorker1Core1 - 10.23.237.22
setBdProperties(vlan-201,replicationServer,0.0.0.0,0,0,None)
2018-04-23 23:07:39,946 uWSGIWorker1Core1 - setBdProperties data= { "aggregateBdEntry": {
"attributes": { "vlan": "vlan-201","replicationMode": "replicationServer","gipo":
"0.0.0.0","suppressArp": "0","isL3": "0","rn": "vlan-[vlan-201]","status": "" }   }   }
2018-04-23 23:07:40,057 uWSGIWorker1Core1 - 10.23.237.22
setBdProperties(vlan-203,replicationServer,0.0.0.0,0,0,None)
2018-04-23 23:07:40,057 uWSGIWorker1Core1 - setBdProperties data= { "aggregateBdEntry": {
"attributes": { "vlan": "vlan-203","replicationMode": "replicationServer","gipo":
"0.0.0.0","suppressArp": "0","isL3": "0","rn": "vlan-[vlan-203]","status": "" }   }   }
2018-04-23 23:07:40,183 MainThread - 10.23.237.22
setBdProperties(vlan-202,replicationServer,0.0.0.0,0,0,None)
2018-04-23 23:07:40,184 MainThread - setBdProperties data= { "aggregateBdEntry": {
"attributes": { "vlan": "vlan-202","replicationMode": "replicationServer","gipo": "0.0.0.
0","suppressArp": "0","isL3": "0","rn": "vlan-[vlan-202]","status": "" }   }   }
2018-04-23 23:07:40,302 uWSGIWorker1Core1 - 10.23.237.22 setIntfAssociations(
2018-04-23 23:07:40,303 uWSGIWorker1Core1 - 10.23.237.22 eth1/10/1,vlan-202
2018-04-23 23:07:40,303 uWSGIWorker1Core1 - 10.23.237.22 eth1/10/1,vlan-203
2018-04-23 23:07:40,304 uWSGIWorker1Core1 - 10.23.237.22 eth1/10/1,vlan-201
2018-04-23 23:07:40,304 uWSGIWorker1Core1 - 10.23.237.22 )
2018-04-23 23:07:40,872 uWSGIWorker1Core1 - 10.23.237.22 setTunnelIntfEntries(
2018-04-23 23:07:40,872 uWSGIWorker1Core1 - 10.23.237.22 vxlanipv4,1.1.11.4,default,multicast
2018-04-23 23:07:40,872 uWSGIWorker1Core1 - 10.23.237.22 vxlanipv4,1.1.11.3,default,multicast
2018-04-23 23:07:40,873 uWSGIWorker1Core1 - 10.23.237.22 )
2018-04-23 23:07:42,452 uWSGIWorker1Core1 -  setRemoteBfdEntry(1.1.11.3,1.1.11.3,
00:50:56:65:a7:b7, 100, 300)
2018-04-23 23:07:42,453 uWSGIWorker1Core1 -  dn=sys/tunnelIntfTable/intf-[1.1.11.3]/remoteBfd

2018-04-23 23:07:42,570 uWSGIWorker1Core1 -  setRemoteBfdEntry(1.1.11.4,1.1.11.4,
00:50:56:68:56:3b, 100, 300)
2018-04-23 23:07:42,570 uWSGIWorker1Core1 -  dn=sys/tunnelIntfTable/intf-[1.1.11.4]/remoteBfd

2018-04-23 23:07:43,108 MainThread - 10.23.237.22 setMacEntries(
2018-04-23 23:07:43,108 MainThread - 10.23.237.22 00:00:aa:02:00:02,vlan-202,1.1.101.5
2018-04-23 23:07:43,109 MainThread - 10.23.237.22 )
```

show interface

Note The output of this command in the base scope is very lengthy. Ellipses (...) are used to indicate information that appears in the output but is not displayed in this example.

Base scope:

```
switch# show interface
mgmt0 is up
admin state is up,
  Hardware: GigabitEthernet, address: f8c2.8823.2d88 (bia f8c2.8823.2d88)
  Internet Address is 172.31.205.11/21
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  full-duplex, 1000 Mb/s
  Auto-Negotiation is turned on
  Auto-mdix is turned off
  EtherType is 0x0000
  1 minute input rate 29688 bits/sec, 51 packets/sec
  1 minute output rate 376 bits/sec, 0 packets/sec
  Rx
    3605590 input packets 3061 unicast packets 1083467 multicast packets
    2519062 broadcast packets 299578144 bytes
  Tx
    82771 output packets 79418 unicast packets 1726 multicast packets
    1627 broadcast packets 6306308 bytes

Ethernet1/1 is down (Administratively down)
admin state is down, Dedicated Interface
  Hardware: 1000/10000 Ethernet, address: f8c2.8823.2d90 (bia f8c2.8823.2d90)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is access
  auto-duplex, auto-speed, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
  Last link flapped never
  Last clearing of "show interface" counters never
  0 interface resets
  30 seconds input rate 0 bits/sec, 0 packets/sec
  30 seconds output rate 0 bits/sec, 0 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  RX
    220 unicast packets  0 multicast packets  0 broadcast packets
    220 input packets  309760 bytes
    0 jumbo packets  0 storm suppression packets
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
  TX
    220 unicast packets  0 multicast packets  0 broadcast packets
    220 output packets  309760 bytes
    0 jumbo packets
    0 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause
...
Ethernet1/23 is up
...
Ethernet1/54 is down (XCVR not inserted)
```

Show Command Outputs

```
...
port-channel107 is up
admin state is up,
  Hardware: Port-Channel, address: f8c2.8823.2da4 (bia f8c2.8823.2da4)
  MTU 1500 bytes, BW 20000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Switchport monitor is off
  EtherType is 0x8100
  Members in this channel: Eth1/21, Eth1/22
  Last clearing of "show interface" counters never
  1 interface resets
  30 seconds input rate 15939960 bits/sec, 1996 packets/sec
  30 seconds output rate 20722240 bits/sec, 2580 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 15.94 Mbps, 1.99 Kpps; output rate 20.72 Mbps, 2.55 Kpps
  RX
    206843476 unicast packets  13820 multicast packets  0 broadcast packets
    206857296 input packets  206019253084 bytes
    0 jumbo packets  0 storm suppression packets
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
  TX
    229324224 unicast packets  62096 multicast packets  41152079 broadcast packe
ts
    270538399 output packets  269402543354 bytes
    0 jumbo packets
    0 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause

loopback0 is up
admin state is up,
  Hardware: Loopback
  Internet Address is 99.1.1.2/32
  MTU 1500 bytes, BW 8000000 Kbit, DLY 5000 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, medium is broadcast
  Auto-mdix is turned off
    35492 packets input 36343808 bytes
    0 multicast frames 0 compressed
    0 input errors 0 frame 0 overrun 0 fifo
    0 packets output 0 bytes 0 underruns
    0 output errors 0 collisions 0 fifo
    0 out_carrier_errors

loopback1 is up
admin state is up,
  Hardware: Loopback
  Internet Address is 100.1.1.3/32
  MTU 1500 bytes, BW 8000000 Kbit, DLY 5000 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, medium is broadcast
  Auto-mdix is turned off
    0 packets input 0 bytes
    0 multicast frames 0 compressed
    0 input errors 0 frame 0 overrun 0 fifo
```

```
    0 packets output 0 bytes 0 underruns
    0 output errors 0 collisions 0 fifo
    0 out_carrier_errors

Vlan1 is down (Administratively down), line protocol is down, autostate enabled
  Hardware is EtherSVI, address is  f8c2.8823.2d8f
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  ARP type: ARPA
  Last clearing of "show interface" counters never
  L3 in Switched:
    ucast: 0 pkts, 0 bytes

nve1 is up
admin state is up,  Hardware: NVE
  MTU 9216 bytes
  Encapsulation VXLAN
  Auto-mdix is turned off
  RX
    ucast: 565400905 pkts, 589147743010 bytes - mcast: 41162020 pkts, 42890824840 bytes
  TX
    ucast: 587275760 pkts, 611941198124 bytes - mcast: 0 pkts, 0 bytes
```

Controller scope:

```
switch%%ctrlr-1# show interface
Ethernet1/21 is up
admin state is up, Dedicated Interface
  Belongs to Po107
  Hardware: 1000/10000 Ethernet, address: f8c2.8823.2da4 (bia f8c2.8823.2da4)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
  Last link flapped 1d04h
  Last clearing of "show interface" counters never
  1 interface resets
  30 seconds input rate 3983264 bits/sec, 497 packets/sec
  30 seconds output rate 10428352 bits/sec, 1298 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 3.98 Mbps, 493 pps; output rate 10.43 Mbps, 1.29 Kpps
  RX
    51840455 unicast packets  6928 multicast packets  0 broadcast packets
    51847383 input packets  51634672778 bytes
    0 jumbo packets  0 storm suppression packets
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
  TX
    116047939 unicast packets  57054 multicast packets  20586016 broadcast packets
    136691009 output packets  136094691284 bytes
```

```
    0 jumbo packets
    0 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause
...
Ethernet1/22 is up
...
Ethernet1/23 is up
...
Ethernet1/24 is up
...
port-channel107 is up
admin state is up,
  Hardware: Port-Channel, address: f8c2.8823.2da4 (bia f8c2.8823.2da4)
  MTU 1500 bytes, BW 20000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Switchport monitor is off
  EtherType is 0x8100
  Members in this channel: Eth1/21, Eth1/22
  Last clearing of "show interface" counters never
  1 interface resets
  30 seconds input rate 15932608 bits/sec, 1992 packets/sec
  30 seconds output rate 20713512 bits/sec, 2581 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 15.94 Mbps, 1.98 Kpps; output rate 20.72 Mbps, 2.55 Kpps
  RX
    207364096 unicast packets  13856 multicast packets  0 broadcast packets
    207377952 input packets  206537798812 bytes
    0 jumbo packets  0 storm suppression packets
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
  TX
    229896908 unicast packets  62250 multicast packets  41256210 broadcast packets
    271215368 output packets  270076671212 bytes
    0 jumbo packets
    0 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause
```

show interface brief

Base scope:

```
switch# show interface brief
--------------------------------------------------------------------------------
Port   VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0  --           up     172.31.205.11                           1000     1500

--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/1        1       eth  access down    Administratively down      auto(D) --
```

```
Eth1/2        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/3        1       eth  access down    Administratively down      auto(D) --
Eth1/4        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/5        1       eth  access down    Administratively down      auto(D) --
Eth1/6        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/7        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/8        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/9        1       eth  access down    XCVR not inserted          auto(D) --
Eth1/10       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/11       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/12       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/13       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/14       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/15       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/16       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/17       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/18       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/19       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/20       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/21       1       eth  trunk  up      none                        10G(D) 107
Eth1/22       1       eth  trunk  up      none                        10G(D) 107
Eth1/23       1       eth  trunk  up      none                        10G(D) --
Eth1/24       1       eth  trunk  up      none                       1000(D) --
Eth1/25       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/26       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/27       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/28       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/29       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/30       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/31       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/32       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/33       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/34       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/35       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/36       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/37       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/38       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/39       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/40       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/41       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/42       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/43       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/44       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/45       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/46       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/47       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/48       1       eth  access down    XCVR not inserted          auto(D) --
Eth1/49       --      eth  routed up      none                        40G(D) 1
Eth1/50       --      eth  routed up      none                        40G(D) 1
Eth1/51       --      eth  routed up      none                        40G(D) 2
Eth1/52       --      eth  routed up      none                        40G(D) 2
Eth1/53       1       eth  access down    Link not connected         auto(D) --
Eth1/54       1       eth  access down    XCVR not inserted          auto(D) --

--------------------------------------------------------------------------------
Port-channel VLAN    Type Mode   Status  Reason                   Speed   Protocol
Interface
--------------------------------------------------------------------------------
Po1           --      eth  routed up      none                     a-40G(D)  lacp
Po2           --      eth  routed up      none                     a-40G(D)  lacp
Po107         1       eth  trunk  up      none                     a-10G(D)  lacp

--------------------------------------------------------------------------------
Interface     Status     Description
```

```
--------------------------------------------------------------------------------
Lo0           up         --
Lo1           up         --

--------------------------------------------------------------------------------
Interface Secondary VLAN(Type)                    Status Reason
--------------------------------------------------------------------------------
Vlan1     --                                      down   Administratively down

--------------------------------------------------------------------------------
Port           Status Reason          MTU
--------------------------------------------------------------------------------
nve1           up     none            9216
```

Controller scope:

```
switch%%ctrlr-1# show interface brief
--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/21       1       eth  trunk  up      none                       10G(D) 107
Eth1/22       1       eth  trunk  up      none                       10G(D) 107
Eth1/23       1       eth  trunk  up      none                       10G(D) --
Eth1/24       1       eth  trunk  up      none                      1000(D) --

--------------------------------------------------------------------------------
Port-channel VLAN    Type Mode   Status  Reason                   Speed   Protocol
Interface
--------------------------------------------------------------------------------
Po107        1       eth  trunk  up      none                    a-10G(D)  lacp
```

show interface ethernet switchport

Base scope:

```
switch# show interface ethernet 1/24 switchport
Name: Ethernet1/24
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
```

Controller scope:

```
switch%%ctrlr-1# show interface ethernet 1/24 switchport
Name: Ethernet1/24
  Switchport: Enabled
```

```
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: - (Vlan not created)
  Trunking Native Mode VLAN: - (Vlan not created)
  Trunking VLANs Allowed: 1001-2000
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
```

show interface ethernet trunk

Base scope:

```
switch# show interface ethernet 1/24 trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Eth1/24       1       trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
Eth1/24       1-4094

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Eth1/24       none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Eth1/24       1,101-150,1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
Feature VTP is not enabled
Eth1/24       1,101-150,1001-2000
```

Controller scope:

```
switch%%ctrlr-1# show interface ethernet 1/24 trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Eth1/24       -      trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
```

```
Eth1/24       1001-2000

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Eth1/24       none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Eth1/24       1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
Eth1/24       1001-2000
```

show interface port-channel switchport

Base scope:

```
switch# show interface port-channel 107 switchport
Name: port-channel107
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
```

Controller scope:

```
switch%%ctrlr-1# show interface port-channel 107 switchport
Name: port-channel107
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: - (Vlan not created)
  Trunking Native Mode VLAN: - (Vlan not created)
  Trunking VLANs Allowed: 1001-2000
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
```

```
  Operational private-vlan: none
```

show interface port-channel trunk

Base scope:

```
switch# show interface port-channel 107 trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Po107         1       trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
Po107         1-4094

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Po107         none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Po107         1,101-150,1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
Feature VTP is not enabled
Po107         1,101-150,1001-2000
```

Controller scope:

```
switch%%ctrlr-1# show interface port-channel 107 trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Po107         -      trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
Po107         1001-2000

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Po107         none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Po107         1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
```

```
Po107         1001-2000
```

show interface switchport

Note The output of this command in the base scope is very lengthy. Ellipses (...) are used to indicate information that appears in the output but is not displayed in this example.

Base scope:

```
switch# show interface switchport
Name: Ethernet1/1
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: access
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
Name: Ethernet1/2
...
Name: Ethernet1/24
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
...
Name: Ethernet1/54
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: access
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
```

```
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
Name: port-channel107
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Allowed: 1-4094
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
```

Controller scope:

```
switch%%ctrlr-1# show interface switchport
 Name: Ethernet1/21
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: - (Vlan not created)
  Trunking Native Mode VLAN: - (Vlan not created)
  Trunking VLANs Allowed: 1001-2000
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
Name: Ethernet1/22
...
Name: Ethernet1/24
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: - (Vlan not created)
  Trunking Native Mode VLAN: - (Vlan not created)
  Trunking VLANs Allowed: 1001-2000
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
```

```
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
Name: port-channel107
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Switchport Block Multicast: Not enabled
  Switchport Block Unicast: Not enabled
  Operational Mode: trunk
  Access Mode VLAN: - (Vlan not created)
  Trunking Native Mode VLAN: - (Vlan not created)
  Trunking VLANs Allowed: 1001-2000
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: none
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: none
  Administrative private-vlan trunk private VLANs: none
  Operational private-vlan: none
```

show interface trunk

Base scope:

```
switch# show interface trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Eth1/21       1       trnk-bndl     Po107
Eth1/22       1       trnk-bndl     Po107
Eth1/23       1       trunking      --
Eth1/24       1       trunking      --
Po107         1       trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
Eth1/21       1-4094
Eth1/22       1-4094
Eth1/23       1-4094
Eth1/24       1-4094
Po107         1-4094

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Eth1/21       none
Eth1/22       none
Eth1/23       none
Eth1/24       none
Po107         none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Eth1/21       none
Eth1/22       none
```

```
Eth1/23       1,101-150,1001-2000
Eth1/24       1,101-150,1001-2000
Po107         1,101-150,1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
Feature VTP is not enabled
Eth1/21       none
Feature VTP is not enabled
Eth1/22       none
Feature VTP is not enabled
Eth1/23       1,101-150,1001-2000
Feature VTP is not enabled
Eth1/24       1,101-150,1001-2000
Feature VTP is not enabled
Po107         1,101-150,1001-2000
```

Controller scope:

```
switch%%ctrlr-1# show interface trunk
--------------------------------------------------------------------------------
Port          Native  Status        Port
              Vlan                  Channel
--------------------------------------------------------------------------------
Eth1/21       -      trnk-bndl     Po107
Eth1/22       -      trnk-bndl     Po107
Eth1/23       -      trunking      --
Eth1/24       -      trunking      --
Po107         -      trunking      --

--------------------------------------------------------------------------------
Port          Vlans Allowed on Trunk
--------------------------------------------------------------------------------
Eth1/21       1001-2000
Eth1/22       1001-2000
Eth1/23       1001-2000
Eth1/24       1001-2000
Po107         1001-2000

--------------------------------------------------------------------------------
Port          Vlans Err-disabled on Trunk
--------------------------------------------------------------------------------
Eth1/21       none
Eth1/22       none
Eth1/23       none
Eth1/24       none
Po107         none

--------------------------------------------------------------------------------
Port          STP Forwarding
--------------------------------------------------------------------------------
Eth1/21       1001-2000
Eth1/22       1001-2000
Eth1/23       1001-2000
Eth1/24       1001-2000
Po107         1001-2000

--------------------------------------------------------------------------------
Port          Vlans in spanning tree forwarding state and not pruned
--------------------------------------------------------------------------------
Eth1/21       1001-2000
Eth1/22       1001-2000
Eth1/23       1001-2000
```

```
Eth1/24       1001-2000
Po107         1001-2000
```

show mac address table

This command displays the MAC address table with remote MAC addresses pushed from the controller.

Base scope and controller scope:

```
switch# show mac address table
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, ~ - vsan
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C 1001     0000.0c07.ac0a   dynamic  0         F      F    nve1(1.1.101.5)
+ 1001     0017.0100.0001   dynamic  0         F      F    Po110
C 1001     0050.5601.0001   dynamic  0         F      F    nve1(2.2.129.10)
C 1001     f40f.1b6f.e64f   dynamic  0         F      F    nve1(1.1.101.5)
C 1001     f80b.cb9f.04d1   dynamic  0         F      F    nve1(2.2.201.2)
+ 1002     0017.0100.0002   dynamic  0         F      F    Po110
C 1002     0050.5601.0002   dynamic  0         F      F    nve1(2.2.129.10)
C 1002     f40f.1b6f.e64f   dynamic  0         F      F    nve1(1.1.101.5)
C 1003     0013.0100.0003   dynamic  0         F      F    nve1(2.2.201.2)
+ 1003     0017.0100.0003   dynamic  0         F      F    Po110
C 1003     0050.5601.0003   dynamic  0         F      F    nve1(2.2.129.10)
C 1004     0013.0100.0004   dynamic  0         F      F    nve1(2.2.201.2)
+ 1004     0017.0100.0004   dynamic  0         F      F    Po110
```

show port-channel summary

In the controller scope, this command displays only the controller-exposed port-channel interfaces.

Base scope:

```
switch# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(RU)     Eth      LACP      Eth1/49(P)   Eth1/50(P)
2     Po2(RU)     Eth      LACP      Eth1/51(P)   Eth1/52(P)
107   Po107(SU)   Eth      LACP      Eth1/21(P)   Eth1/22(P)
```

Controller scope:

```
switch%%ctrlr-1# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
```

```
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
107   Po107(SU)   Eth      LACP      Eth1/21(P)   Eth1/22(P)
```

show running-config

In the base scope, this command displays the configurations for all interfaces. In the controller scope, this command displays the configurations only for the VLANs and interfaces that are assigned for use by the external controller.

Note You can enter the **show running-config controller** command in the base scope as an alternative to entering the **show running-config** command in the controller scope.

Note The output of this command in the base scope is very lengthy. Ellipses (...) are used to indicate information that appears in the output but is not displayed in this example.

Base scope:

```
switch# show running-config
!Time: Mon Apr 18 11:32:24 2017
version 7.0(3)I6(1)
power redundancy-mode combined force

switchname priv04-tor1
vdc priv04-tor1 id 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 58 maximum 58
  limit-resource m6route-mem minimum 8 maximum 8

feature telnet
feature nxapi
feature nxdb
feature bash-shell
feature scp-server
cfs eth distribute
feature ospf
feature interface-vlan
feature vn-segment-vlan-based
feature lacp
feature vpc
feature bfd
clock protocol none vdc 1
feature nv overlay

no password strength-check
```

Show Command Outputs

```
username admin password 5 $5$6XbR9SSQ$dLZ4Kv46u0I2zqtzKXGRiQUDqDf5.yCR6cDq6Mn74M5  role
network-admin
nxapi http port 80
nxapi https port 443
nxapi use-vrf management
ip domain-lookup
system default switchport shutdown
ip access-list ovs-copp-acl-andro
  10 permit tcp any any eq 6632
  20 permit tcp any eq 6632 any
  30 permit tcp any any eq www
  40 permit tcp any eq www any
  50 permit tcp any any eq 443
  60 permit tcp any eq 443 any
ip access-list ovs-copp-acl-auto-rp
  10 permit ip any 224.0.1.39/32
  20 permit ip any 224.0.1.40/32
ip access-list ovs-copp-acl-bgp
  10 permit tcp any gt 1024 any eq bgp
  20 permit tcp any eq bgp any gt 1024
ipv6 access-list ovs-copp-acl-bgp6
  10 permit tcp any gt 1024 any eq bgp
  20 permit tcp any eq bgp any gt 1024
ip access-list ovs-copp-acl-dhcp
  10 permit udp any eq bootpc any
  20 permit udp any neq bootps any eq bootps
ip access-list ovs-copp-acl-dhcp-relay-response
  10 permit udp any eq bootps any
  20 permit udp any any eq bootpc
ipv6 access-list ovs-copp-acl-dhcp6
  10 permit udp any eq 546 any
  20 permit udp any any eq 547
ipv6 access-list ovs-copp-acl-dhcp6-relay-response
  10 permit udp any eq 547 any
  20 permit udp any any eq 546
ip access-list ovs-copp-acl-eigrp
  10 permit eigrp any any
ipv6 access-list ovs-copp-acl-eigrp6
  10 permit eigrp any any
ip access-list ovs-copp-acl-ftp
  10 permit tcp any any eq ftp-data
  20 permit tcp any any eq ftp
  30 permit tcp any eq ftp-data any
  40 permit tcp any eq ftp any
ip access-list ovs-copp-acl-hsrp
  10 permit udp any 224.0.0.0/24 eq 1985
ipv6 access-list ovs-copp-acl-hsrp6
  10 permit udp any ff02::66/128 eq 2029
ip access-list ovs-copp-acl-icmp
  10 permit icmp any any echo
  20 permit icmp any any echo-reply
ipv6 access-list ovs-copp-acl-icmp6
  10 permit icmp any any echo-request
  20 permit icmp any any echo-reply
ipv6 access-list ovs-copp-acl-icmp6-msgs
  10 permit icmp any any router-advertisement
  20 permit icmp any any router-solicitation
  30 permit icmp any any nd-na
  40 permit icmp any any nd-ns
  50 permit icmp any any mld-query
  60 permit icmp any any mld-report
  70 permit icmp any any mld-reduction
  80 permit icmp any any mldv2
ip access-list ovs-copp-acl-igmp
```

```
  10 permit igmp any 224.0.0.0/3
mac access-list ovs-copp-acl-mac-cdp-udld-vtp
  10 permit any 0100.0ccc.cccc 0000.0000.0000
mac access-list ovs-copp-acl-mac-cfsoe
  10 permit any 0180.c200.000e 0000.0000.0000 0x8843
  20 permit any 0180.c200.000e 0000.0000.0000
mac access-list ovs-copp-acl-mac-dot1x
  10 permit any 0180.c200.0003 0000.0000.0000 0x888e
mac access-list ovs-copp-acl-mac-l2-tunnel
  10 permit any any 0x8840
mac access-list ovs-copp-acl-mac-l3-isis
  10 permit any 0180.c200.0015 0000.0000.0000
  20 permit any 0180.c200.0014 0000.0000.0000
  30 permit any 0900.2b00.0005 0000.0000.0000
mac access-list ovs-copp-acl-mac-lacp
  10 permit any 0180.c200.0002 0000.0000.0000 0x8809
mac access-list ovs-copp-acl-mac-lldp
  10 permit any 0180.c200.000e 0000.0000.0000 0x88cc
mac access-list ovs-copp-acl-mac-sdp-srp
  10 permit any 0180.c200.000e 0000.0000.0000 0x3401
mac access-list ovs-copp-acl-mac-stp
  10 permit any 0100.0ccc.cccd 0000.0000.0000
  20 permit any 0180.c200.0000 0000.0000.0000
mac access-list ovs-copp-acl-mac-undesirable
  10 permit any any
ip access-list ovs-copp-acl-msdp
  10 permit tcp any gt 1024 any eq 639
  20 permit tcp any eq 639 any gt 1024
ip access-list ovs-copp-acl-ntp
  10 permit udp any any eq ntp
  20 permit udp any eq ntp any
ipv6 access-list ovs-copp-acl-ntp6
  10 permit udp any any eq ntp
  20 permit udp any eq ntp any
ip access-list ovs-copp-acl-ospf
  10 permit ospf any any
ipv6 access-list ovs-copp-acl-ospf6
  10 permit 89 any any
ip access-list ovs-copp-acl-pim
  10 permit pim any 224.0.0.0/24
  20 permit udp any any eq pim-auto-rp
  30 permit ip any 224.0.0.13/32
ip access-list ovs-copp-acl-pim-mdt-join
  10 permit udp any 224.0.0.13/32
ip access-list ovs-copp-acl-pim-reg
  10 permit pim any any
ipv6 access-list ovs-copp-acl-pim6
  10 permit pim any ff02::d/128
  20 permit udp any any eq pim-auto-rp
ipv6 access-list ovs-copp-acl-pim6-reg
  10 permit pim any any
ip access-list ovs-copp-acl-ptp
  10 permit udp any 224.0.1.129/32 eq 319
  20 permit udp any 224.0.1.129/32 eq 320
ip access-list ovs-copp-acl-radius
  10 permit udp any any eq 1812
  20 permit udp any any eq 1813
  30 permit udp any any eq 1645
  40 permit udp any any eq 1646
  50 permit udp any eq 1812 any
  60 permit udp any eq 1813 any
  70 permit udp any eq 1645 any
  80 permit udp any eq 1646 any
ipv6 access-list ovs-copp-acl-radius6
```

```
  10 permit udp any any eq 1812
  20 permit udp any any eq 1813
  30 permit udp any any eq 1645
  40 permit udp any any eq 1646
  50 permit udp any eq 1812 any
  60 permit udp any eq 1813 any
  70 permit udp any eq 1645 any
  80 permit udp any eq 1646 any
ip access-list ovs-copp-acl-rip
  10 permit udp any 224.0.0.0/24 eq rip
ipv6 access-list ovs-copp-acl-rip6
  10 permit udp any ff02::9/64 eq 521
ip access-list ovs-copp-acl-sftp
  10 permit tcp any any eq 115
  20 permit tcp any eq 115 any
ip access-list ovs-copp-acl-snmp
  10 permit udp any any eq snmp
  20 permit udp any any eq snmptrap
  30 permit tcp any any eq 161
  40 permit tcp any any eq 162
ipv6 access-list ovs-copp-acl-snmp6
  10 permit udp any any eq snmp
  20 permit udp any any eq snmptrap
  30 permit tcp any any eq 161
  40 permit tcp any any eq 162
ip access-list ovs-copp-acl-ssh
  10 permit tcp any any eq 22
  20 permit tcp any eq 22 any
ipv6 access-list ovs-copp-acl-ssh6
  10 permit tcp any any eq 22
  20 permit tcp any eq 22 any
ip access-list ovs-copp-acl-tacacs
  10 permit tcp any any eq tacacs
  20 permit tcp any eq tacacs any
ipv6 access-list ovs-copp-acl-tacacs6
  10 permit tcp any any eq tacacs
  20 permit tcp any eq tacacs any
ip access-list ovs-copp-acl-telnet
  10 permit tcp any any eq telnet
  20 permit tcp any any eq 107
  30 permit tcp any eq telnet any
  40 permit tcp any eq 107 any
ipv6 access-list ovs-copp-acl-telnet6
  10 permit tcp any any eq telnet
  20 permit tcp any any eq 107
  30 permit tcp any eq telnet any
  40 permit tcp any eq 107 any
ip access-list ovs-copp-acl-tftp
  10 permit udp any any eq tftp
  20 permit udp any any eq 1758
  30 permit udp any eq tftp any
  40 permit udp any eq 1758 any
ipv6 access-list ovs-copp-acl-tftp6
  10 permit udp any any eq tftp
  20 permit udp any any eq 1758
  30 permit udp any eq tftp any
  40 permit udp any eq 1758 any
ip access-list ovs-copp-acl-traceroute
  10 permit icmp any any ttl-exceeded
  20 permit icmp any any port-unreachable
  30 permit udp any any range 33434 33534
ip access-list ovs-copp-acl-undesirable
  10 permit udp any any eq 1434
ip access-list ovs-copp-acl-vpc
```

```
  10 permit udp any any eq 3200
ip access-list ovs-copp-acl-vrrp
  10 permit ip any 224.0.0.18/32
ipv6 access-list ovs-copp-acl-vrrp6
  10 permit ipv6 any ff02::12/128
class-map type control-plane match-any ovs-copp-class-critical
  match access-group name ovs-copp-acl-bgp
  match access-group name ovs-copp-acl-rip
  match access-group name ovs-copp-acl-vpc
  match access-group name ovs-copp-acl-bgp6
  match access-group name ovs-copp-acl-ospf
  match access-group name ovs-copp-acl-rip6
  match access-group name ovs-copp-acl-eigrp
  match access-group name ovs-copp-acl-ospf6
  match access-group name ovs-copp-acl-eigrp6
  match access-group name ovs-copp-acl-auto-rp
  match access-group name ovs-copp-acl-mac-l3-isis
class-map type control-plane match-any ovs-copp-class-exception
  match exception ip option
  match exception ip icmp unreachable
  match exception ipv6 option
  match exception ipv6 icmp unreachable
class-map type control-plane match-any ovs-copp-class-exception-diag
  match exception ttl-failure
  match exception mtu-failure
class-map type control-plane match-any ovs-copp-class-important
  match access-group name ovs-copp-acl-hsrp
  match access-group name ovs-copp-acl-vrrp
  match access-group name ovs-copp-acl-hsrp6
  match access-group name ovs-copp-acl-vrrp6
  match access-group name ovs-copp-acl-mac-lldp
  match access-group name ovs-copp-acl-icmp6-msgs
class-map type control-plane match-any ovs-copp-class-l2-default
  match access-group name ovs-copp-acl-mac-undesirable
class-map type control-plane match-any ovs-copp-class-l2-unpoliced
  match access-group name ovs-copp-acl-mac-stp
  match access-group name ovs-copp-acl-mac-lacp
  match access-group name ovs-copp-acl-mac-cfsoe
  match access-group name ovs-copp-acl-mac-sdp-srp
  match access-group name ovs-copp-acl-mac-l2-tunnel
  match access-group name ovs-copp-acl-mac-cdp-udld-vtp
class-map type control-plane match-any ovs-copp-class-l3mc-data
  match exception multicast rpf-failure
  match exception multicast dest-miss
class-map type control-plane match-any ovs-copp-class-l3uc-data
  match exception glean
class-map type control-plane match-any ovs-copp-class-management
  match access-group name ovs-copp-acl-ftp
  match access-group name ovs-copp-acl-ntp
  match access-group name ovs-copp-acl-ssh
  match access-group name ovs-copp-acl-ntp6
  match access-group name ovs-copp-acl-sftp
  match access-group name ovs-copp-acl-snmp
  match access-group name ovs-copp-acl-ssh6
  match access-group name ovs-copp-acl-tftp
  match access-group name ovs-copp-acl-andro
  match access-group name ovs-copp-acl-snmp6
  match access-group name ovs-copp-acl-tftp6
  match access-group name ovs-copp-acl-radius
  match access-group name ovs-copp-acl-tacacs
  match access-group name ovs-copp-acl-telnet
  match access-group name ovs-copp-acl-radius6
  match access-group name ovs-copp-acl-tacacs6
  match access-group name ovs-copp-acl-telnet6
```

Show Command Outputs

```
class-map type control-plane match-any ovs-copp-class-monitoring
  match access-group name ovs-copp-acl-icmp
  match access-group name ovs-copp-acl-icmp6
  match access-group name ovs-copp-acl-traceroute
class-map type control-plane match-any ovs-copp-class-multicast-router
  match access-group name ovs-copp-acl-pim
  match access-group name ovs-copp-acl-msdp
  match access-group name ovs-copp-acl-pim6
  match access-group name ovs-copp-acl-pim-reg
  match access-group name ovs-copp-acl-pim6-reg
  match access-group name ovs-copp-acl-pim-mdt-join
class-map type control-plane match-any ovs-copp-class-nat-flow
  match exception multicast rpf-failure
class-map type control-plane match-any ovs-copp-class-normal
  match access-group name ovs-copp-acl-mac-dot1x
  match protocol arp
class-map type control-plane match-any ovs-copp-class-normal-dhcp
  match access-group name ovs-copp-acl-dhcp
  match access-group name ovs-copp-acl-dhcp6
class-map type control-plane match-any ovs-copp-class-normal-dhcp-relay-response
  match access-group name ovs-copp-acl-dhcp-relay-response
  match access-group name ovs-copp-acl-dhcp6-relay-response
class-map type control-plane match-any ovs-copp-class-normal-igmp
  match access-group name ovs-copp-acl-igmp
class-map type control-plane match-any ovs-copp-class-redirect
  match access-group name ovs-copp-acl-ptp
class-map type control-plane match-any ovs-copp-class-undesirable
  match access-group name ovs-copp-acl-undesirable
  match exception multicast sg-rpf-failure
policy-map type control-plane ovs-copp-policy-strict
  class ovs-copp-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-critical
    set cos 7
    police cir 19000 pps bc 128 packets conform transmit violate drop
  class ovs-copp-class-important
    set cos 6
    police cir 3000 pps bc 128 packets conform transmit violate drop
  class ovs-copp-class-multicast-router
    set cos 6
    police cir 3000 pps bc 128 packets conform transmit violate drop
  class ovs-copp-class-management
    set cos 2
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-normal
    set cos 1
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 64 packets conform transmit violate drop
  class ovs-copp-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class ovs-copp-class-redirect
    set cos 1
    police cir 1500 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-exception
```

```
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-exception-diag
    set cos 1
    police cir 50 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-monitoring
    set cos 1
    police cir 75 pps bc 128 packets conform transmit violate drop
  class ovs-copp-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class ovs-copp-class-undesirable
    set cos 0
    police cir 15 pps bc 32 packets conform transmit violate drop
  class ovs-copp-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class ovs-copp-class-l2-default
    set cos 0
    police cir 50 pps bc 32 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 32 packets conform transmit violate drop
copp profile strict
snmp-server user admin network-admin auth md5 0xb346ad8d706187f507e01fba4e7c7acb priv
0xb346ad8d706187f507e01fba4e7c7acb localizedkey
rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO

vlan 1,100,3000
vlan 3000
  vn-segment 0

vrf context management
  ip route 0.0.0.0/0 172.31.144.1
hardware access-list tcam region ifacl 0
hardware access-list tcam region vacl 0
hardware access-list tcam region copp 512
hardware access-list tcam region redirect-tunnel 256
hardware qos ns-buffer-profile mesh
vpc domain 200
  peer-switch
  role priority 100
  peer-keepalive destination 172.31.145.144
  peer-gateway
  ipv6 nd synchronize
  ip arp synchronize


interface Vlan1

interface Vlan100
  no shutdown
  mtu 9216
  ip address 1.1.14.1/24
  ip router ospf p1 area 0.0.0.0

interface Vlan3000
  no shutdown
  ip forward
```

Show Command Outputs

```
interface port-channel1
  description uplink
  no switchport
  mtu 9216
  ip address 1.1.13.1/24
  ip ospf network point-to-point
  ip router ospf p1 area 0.0.0.0

interface port-channel2
  description uplink
  no switchport
  mtu 9216
  ip address 1.1.15.1/24
  ip ospf network point-to-point
  ip router ospf p1 area 0.0.0.0

interface port-channel10
  switchport mode trunk
  !controller type l2-vxlan identifier 1
  speed 1000
  vpc 10

interface port-channel100
  switchport mode trunk
  !controller type l2-vxlan identifier 1
  spanning-tree port type network
  vpc peer-link

interface nve1
  no shutdown
  source-interface loopback0
  auto-remap-replication-servers
  host-reachability protocol controller 1
  source-interface hold-down-time 30
  config-source controller

interface Ethernet1/1

interface Ethernet1/2

interface Ethernet1/3

interface Ethernet1/4

interface Ethernet1/5

interface Ethernet1/6

interface Ethernet1/7

interface Ethernet1/8
  description edge
  switchport mode trunk
  switchport trunk allowed vlan 1-200,204-4094
  spanning-tree port type edge trunk
  spanning-tree bpdufilter enable
  no shutdown

interface Ethernet1/9

interface Ethernet1/10

interface Ethernet1/11
```

```
interface Ethernet1/12

interface Ethernet1/13

interface Ethernet1/14

interface Ethernet1/15

interface Ethernet1/16

interface Ethernet1/17

interface Ethernet1/18

interface Ethernet1/19

interface Ethernet1/20

interface Ethernet1/21
  switchport mode trunk
  speed 1000
  channel-group 10 mode active
  no shutdown

interface Ethernet1/22

interface Ethernet1/23
  description edge
  switchport mode trunk
  switchport trunk allowed vlan 1-200,204-4094
  spanning-tree port type edge trunk
  spanning-tree bpdufilter enable
  no shutdown

interface Ethernet1/24
  description edge
  switchport mode trunk
  !controller type l2-vxlan identifier 1
  spanning-tree port type edge trunk
  spanning-tree bpdufilter enable
  no shutdown

interface Ethernet1/25

interface Ethernet1/26

interface Ethernet1/27

interface Ethernet1/28

interface Ethernet1/29

interface Ethernet1/30

interface Ethernet1/31

interface Ethernet1/32

interface Ethernet1/33
  switchport mode trunk
  !controller type l2-vxlan identifier 1
  spanning-tree port type edge trunk
  no shutdown
```

Show Command Outputs

```
interface Ethernet1/34
interface Ethernet1/35
interface Ethernet1/36
interface Ethernet1/37
interface Ethernet1/38
interface Ethernet1/39
interface Ethernet1/40
interface Ethernet1/41
interface Ethernet1/42
interface Ethernet1/43
interface Ethernet1/44
interface Ethernet1/45
interface Ethernet1/46
interface Ethernet1/47
interface Ethernet1/48
interface Ethernet1/49
interface Ethernet1/50
interface Ethernet1/51
interface Ethernet1/52
interface Ethernet1/53
interface Ethernet1/54
interface Ethernet1/55
interface Ethernet1/56
interface Ethernet1/57
interface Ethernet1/58
interface Ethernet1/59
interface Ethernet1/60
interface Ethernet1/61
interface Ethernet1/62
interface Ethernet1/63
interface Ethernet1/64
interface Ethernet1/65
```

```
interface Ethernet1/66

interface Ethernet1/67

interface Ethernet1/68

interface Ethernet1/69

interface Ethernet1/70

interface Ethernet1/71

interface Ethernet1/72

interface Ethernet1/73

interface Ethernet1/74

interface Ethernet1/75

interface Ethernet1/76

interface Ethernet1/77

interface Ethernet1/78

interface Ethernet1/79

interface Ethernet1/80

interface Ethernet1/81

interface Ethernet1/82

interface Ethernet1/83

interface Ethernet1/84

interface Ethernet1/85

interface Ethernet1/86

interface Ethernet1/87

interface Ethernet1/88

interface Ethernet1/89

interface Ethernet1/90

interface Ethernet1/91

interface Ethernet1/92

interface Ethernet1/93

interface Ethernet1/94

interface Ethernet1/95

interface Ethernet1/96

interface Ethernet2/1
  description uplink
```

```
  no switchport
  mtu 9216
  channel-group 1 mode active
  no shutdown

interface Ethernet2/2
  description uplink
  no switchport
  mtu 9216
  channel-group 1 mode active
  no shutdown

interface Ethernet2/3
  switchport mode trunk
  channel-group 100 mode active
  no shutdown

interface Ethernet2/4
  switchport mode trunk
  channel-group 100 mode active
  no shutdown

interface Ethernet2/5
  no switchport
  mtu 9216
  channel-group 2 mode active
  no shutdown

interface Ethernet2/6
  no switchport
  mtu 9216
  channel-group 2 mode active
  no shutdown

interface mgmt0
  vrf member management
  ip address 172.31.145.141/21

interface loopback0
  description vtep_loopback_interface
  ip address 1.1.101.1/32
  ip address 1.1.101.5/32 secondary
  ip router ospf p1 area 0.0.0.0

interface loopback1
  description andromeda_connecting_interface
  ip address 1.1.24.1/32
  ip router ospf p1 area 0.0.0.0
terminal log-all
line console
  exec-timeout 0
  speed 115200
line vty
  exec-timeout 0
boot nxos bootflash:/sanity.image
router ospf p1
  router-id 100.100.100.1
  log-adjacency-changes detail
ip arp timeout 28800
no logging console

controller type l2-vxlan identifier 1
  controller description NODE01
  assign vlan 201-203 dedicated
```

```
  assign interface port-channel10, port-channel100 shared
  assign interface Ethernet1/24, Ethernet1/33 shared
```

Controller scope:

```
switch# show running-config
!Time: Mon Apr 18 13:24:38 2017

version 7.0(3)I6(1)
feature vn-segment-vlan-based
feature nv overlay

vlan 201-203
vlan 201
  vn-segment 10002
vlan 202
  vn-segment 10003
vlan 203
  vn-segment 10004


interface port-channel10
  switchport mode trunk
  switchport trunk allowed vlan 201-203
  !controller type l2-vxlan identifier 1
  speed 1000

interface port-channel100
  switchport mode trunk
  switchport trunk allowed vlan 201-203
  !controller type l2-vxlan identifier 1

interface nve1
  source-interface loopback0
  auto-remap-replication-servers
  host-reachability protocol controller 1
  source-interface hold-down-time 30
  config-source controller
  member vni 10002
    ingress-replication protocol static
      peer-ip 1.1.11.3
  member vni 10003
    ingress-replication protocol static
      peer-ip 1.1.11.4
  member vni 10004
    ingress-replication protocol static
      peer-ip 1.1.11.3
  bfd-neighbor 1.1.11.3 1.1.11.3 0023.2000.0002
  bfd-neighbor 1.1.11.4 1.1.11.4 0023.2000.0001

interface Ethernet1/21
  switchport mode trunk
  switchport trunk allowed vlan 201-203
  speed 1000

interface Ethernet1/34
  switchport mode trunk
  switchport trunk allowed vlan 201-203
  !controller type l2-vxlan identifier 1

interface Ethernet1/35
  switchport mode trunk
  switchport trunk allowed vlan 201-203
  !controller type l2-vxlan identifier 1
```

```
interface Ethernet2/5
  switchport mode trunk
  switchport trunk allowed vlan 201-203

interface Ethernet2/6
  switchport mode trunk
  switchport trunk allowed vlan 201-203

controller type l2-vxlan identifier 1
  controller description NODE04
  assign vlan 201-203 dedicated
  assign interface port-channel10, port-channel100 shared
  assign interface Ethernet1/34-35 shared
```

show vlan

Base scope:

```
switch# show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Po107, Eth1/1, Eth1/2, Eth1/3
                                                Eth1/4, Eth1/5, Eth1/6, Eth1/7
                                                Eth1/8, Eth1/9, Eth1/10, Eth1/11
                                                Eth1/12, Eth1/13, Eth1/14
                                                Eth1/15, Eth1/16, Eth1/17
                                                Eth1/18, Eth1/19, Eth1/20
                                                Eth1/21, Eth1/22, Eth1/23
                                                Eth1/24, Eth1/25, Eth1/26
                                                Eth1/27, Eth1/28, Eth1/29
                                                Eth1/30, Eth1/31, Eth1/32
                                                Eth1/33, Eth1/34, Eth1/35
                                                Eth1/36, Eth1/37, Eth1/38
                                                Eth1/39, Eth1/40, Eth1/41
                                                Eth1/42, Eth1/43, Eth1/44
                                                Eth1/45, Eth1/46, Eth1/47
                                                Eth1/48, Eth1/53, Eth1/54
101  VLAN0101                         active    Po107, Eth1/21, Eth1/22
                                                Eth1/23, Eth1/24
102  VLAN0102                         active    Po107, Eth1/21, Eth1/22
                                                Eth1/23, Eth1/24
...
150  VLAN0150                         active    Po107, Eth1/21, Eth1/22
                                                Eth1/23, Eth1/24

VLAN Type         Vlan-mode
---- -----        ----------
1    enet         CE
101  enet         CE
102  enet         CE
...
150  enet         CE

Remote SPAN VLANs
------------------------------------------------------------------------------

Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
```

Controller scope:

```
switch%%ctrlr-1# show vlan
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1001 VLAN1001                          active    Po107, Eth1/23, Eth1/24
1002 VLAN1002                          active    Po107, Eth1/23, Eth1/24
1003 VLAN1003                          active    Po107, Eth1/23, Eth1/24
...
1999 VLAN1999                          active    Po107, Eth1/23, Eth1/24
2000 VLAN2000                          active    Po107, Eth1/23, Eth1/24

VLAN Type         Vlan-mode
---- -----        ----------
1001 enet         CE
1002 enet         CE
...
1999 enet         CE
2000 enet         CE

Remote SPAN VLANs
--------------------------------------------------------------------------------

Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
```

show vlan brief

Base scope:

```
switch# show vlan brief
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Po107, Eth1/1, Eth1/2, Eth1/3
                                                 Eth1/4, Eth1/5, Eth1/6, Eth1/7
                                                 Eth1/8, Eth1/9, Eth1/10, Eth1/11
                                                 Eth1/12, Eth1/13, Eth1/14
                                                 Eth1/15, Eth1/16, Eth1/17
                                                 Eth1/18, Eth1/19, Eth1/20
                                                 Eth1/21, Eth1/22, Eth1/23
                                                 Eth1/24, Eth1/25, Eth1/26
                                                 Eth1/27, Eth1/28, Eth1/29
                                                 Eth1/30, Eth1/31, Eth1/32
                                                 Eth1/33, Eth1/34, Eth1/35
                                                 Eth1/36, Eth1/37, Eth1/38
                                                 Eth1/39, Eth1/40, Eth1/41
                                                 Eth1/42, Eth1/43, Eth1/44
                                                 Eth1/45, Eth1/46, Eth1/47
                                                 Eth1/48, Eth1/53, Eth1/54
101  VLAN0101                          active    Po107, Eth1/21, Eth1/22
                                                 Eth1/23, Eth1/24
102  VLAN0102                          active    Po107, Eth1/21, Eth1/22
                                                 Eth1/23, Eth1/24
...
149  VLAN0149                          active    Po107, Eth1/21, Eth1/22
                                                 Eth1/23, Eth1/24
150  VLAN0150                          active    Po107, Eth1/21, Eth1/22
                                                 Eth1/23, Eth1/24
```

Controller scope:

```
switch%%ctrlr-1# show vlan brief
VLAN Name                              Status    Ports
```

```
---- -------------------------------- --------- -------------------------------
1001 VLAN1001                         active    Po107, Eth1/23, Eth1/24
1002 VLAN1002                         active    Po107, Eth1/23, Eth1/24
1003 VLAN1003                         active    Po107, Eth1/23, Eth1/24
...
1999 VLAN1999                         active    Po107, Eth1/23, Eth1/24
2000 VLAN2000                         active    Po107, Eth1/23, Eth1/24
```

show vpc

In the base scope, the output of this command shows the non-assigned VLANs on the vPC ports. In the controller scope, the output of this command shows the assigned VLANs on the vPC ports.

Base scope:

```
switch# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 1
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 180s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Peer-link delay status            : Timer is off.(timeout = 300s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po10   up     1,100,3000

vPC status
----------------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
20   Po20   up     success     success                    1,100,3000
```

In the above output, VLANs 1, 100, and 3000 are existing VLANs on vPC port po20 that are not assigned VLANs.

Controller scope:

```
switch# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
```

```
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 1
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 180s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Peer-link delay status            : Timer is off.(timeout = 300s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po10   up     1001-1160

vPC status
----------------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
20   Po20   up     success     success                    1001-1160
```

In the above output, the active VLANs 1001-1160 are the assigned VLANs.

show vpc consistency-parameters global

This command displays the assigned vPC interfaces and assigned VLANs for the primary and secondary switches. The assigned vPC interfaces and assigned VLANs should match on both vPC peers.

Base scope and controller scope:

```
switch# show vpc consistency-parameters global

Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
STP MST Simulate PVST       1     Enabled                Enabled
STP Port Type, Edge         1     Normal, Disabled,      Normal, Disabled,
BPDUFilter, Edge BPDUGuard        Disabled               Disabled
STP MST Region Name         1     ""                     ""
STP Disabled                1     None                   None
STP Mode                    1     Rapid-PVST             Rapid-PVST
STP Bridge Assurance        1     Enabled                Enabled
STP Loopguard               1     Disabled               Disabled
STP MST Region Instance to  1
 VLAN Mapping
STP MST Region Revision     1     0                      0
Interface-vlan admin up     2     100,3000               100,3000
Interface-vlan routing      2     1,100,3000             1,100,3000
capability
Nve1 Admin State, Src       1     Up, Up, 1.1.101.5, CP, Up, Up, 1.1.101.5, CP,
Admin State, Secondary IP,         FALSE                  FALSE
 Host Reach Mode, VMAC
Advertisement
Assigned VRF's              1     #0                     #0
Assigned VPC Interfaces     1     10                     10
Assigned Vlans              1     201-203                201-203
QoS (Cos)                   2     ([0-7], [], [], [],    ([0-7], [], [], [],
                                  [], [])                [], [])
```

```
Network QoS (MTU)           2     (1500, 1500, 1500,     (1500, 1500, 1500,
                                  1500, 0, 0)            1500, 0, 0)
Network Qos (Pause:         2     (F, F, F, F, F, F)     (F, F, F, F, F, F)
T->Enabled, F->Disabled)
Input Queuing (Bandwidth)   2     (0, 0, 0, 0, 0, 0)     (0, 0, 0, 0, 0, 0)
Input Queuing (Absolute     2     (F, F, F, F, F, F)     (F, F, F, F, F, F)
Priority: T->Enabled,
F->Disabled)
Output Queuing (Bandwidth   2     (100, 0, 0, 0, 0, 0)   (100, 0, 0, 0, 0, 0)
Remaining)
Output Queuing (Absolute    2     (F, F, F, T, F, F)     (F, F, F, T, F, F)
Priority: T->Enabled,
F->Disabled)
Allowed VLANs               -     1,100,3000             1,100,3000
Local suspended VLANs       -     -                      -
```

Show BFD Command Outputs

This section provides sample outputs for the **show** BFD commands.

show bfd neighbors

On a non-vPC device:

```
switch# show bfd neighbors
OurAddr          NeighAddr       LD/RD                 RH/RS           Holdown(mult)
State       Int                    Vrf
1.1.102.1        1.1.11.3        1090519041/94501819   Up              800(3)            Up
           nve1
1.1.102.1        1.1.11.4        1090519042/895389263  Up              705(3)            Up
           nve1
```

Note If this command is entered on a vPC secondary device, it should be displayed as Down because the session is hosted only on the vPC primary device.

show nve bfd neighbors

On a non-vPC device:

```
switch# show nve bfd neighbors
Interface Neighbor VTEP IP Inner-IP       Inner-MAC      vPC CC state
--------- ---------------- -------------- ---------      ------------
nve1      1.1.11.3         1.1.11.3       0023.2000.0001      N/A
          1.1.11.4         1.1.11.4       0023.2000.0002      N/A
```

On a vPC node:

```
switch# show nve bfd neighbors
Interface Neighbor VTEP IP Inner-IP       Inner-MAC      vPC CC state
--------- ---------------- -------------- ---------      ------------
nve1      1.1.11.3         1.1.11.3       0023.2000.0001      Pass
```

```
1.1.11.4        1.1.11.4       0023.2000.0002      Pass
```

Note In vPC modes, the vPC consistency checker (CC) is run to make sure that the controller-pushed BFD configuration is consistent on both the vPC primary and secondary devices.

show running controller

```
switch# show running controller
interface nve1
  source-interface loopback0
  auto-remap-replication-servers
  host-reachability protocol controller 1
  config-source controller
  member vni 10002
    ingress-replication protocol static
      peer-ip 1.1.11.3
  member vni 10003
    ingress-replication protocol static
      peer-ip 1.1.11.4
  member vni 10004
    ingress-replication protocol static
      peer-ip 1.1.11.3
  bfd-neighbor 1.1.11.3 1.1.11.3 0023.2000.0001
  bfd-neighbor 1.1.11.4 1.1.11.4 0023.2000.0002
```

The BFD configuration that is pushed from the controller is highlighted in the output above.

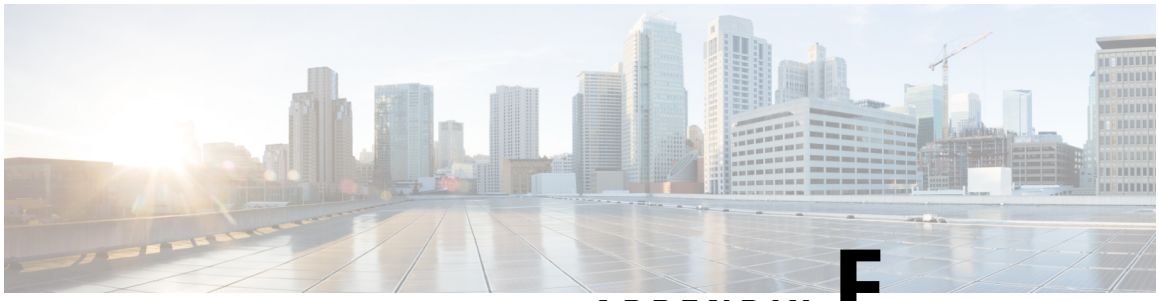
show vpc consistency-parameters global

BFD-related parameters are part of the vPC consistency checker.

```
switch# show vpc consistency-parameters global
Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
Vlan to Vn-segment Map      1     4 Relevant Map(s)      4 Relevant Map(s)
STP Mode                    1     Rapid-PVST             Rapid-PVST
STP Disabled                1     None                   None
STP MST Region Name         1     ""                     ""
STP MST Region Revision     1     0                      0
STP MST Region Instance to  1
 VLAN Mapping
STP Loopguard               1     Disabled               Disabled
STP Bridge Assurance        1     Enabled                Enabled
STP Port Type, Edge         1     Normal, Disabled,      Normal, Disabled,
BPDUFilter, Edge BPDUGuard        Disabled               Disabled
STP MST Simulate PVST       1     Enabled                Enabled
Nve Admin State, Src Admin  1     Up, Up, 1.1.101.5, CP  Up, Up, 1.1.101.5, CP
 State, Secondary IP, Host
 Reach Mode
BFD Neighbor:               1     Outer/Inner IP/MAC     Outer/Inner IP/MAC
                                  1.1.11.3/1.1.11.3/0023 1.1.11.3/1.1.11.3/0023
                                  .2000.0001             .2000.0001
```

Show Command Outputs

BFD Neighbor:	**1**	**Outer/Inner IP/MAC 1.1.11.4/1.1.11.4/0023.2000.0002**	**Outer/Inner IP/MAC 1.1.11.4/1.1.11.4/0023.2000.0002**
Nve Vni Configuration	1	10002-10004,16777215	10002-10004,16777215
Nve encap Configuration	1	vxlan	vxlan
Assigned Vlans	1	201-203	201-203
Assigned VPC Interfaces	1	10	10
Interface-vlan admin up	2	100,3000	100,3000
Interface-vlan routing capability	2	1,100,3000	1,100,3000
Allowed VLANs	-	1,100,201-203,3000	1,100,201-203,3000
Local suspended VLANs	-	-	-



APPENDIX F

Debugging NXDB

This appendix includes the following sections:

Collecting Logs and Debugging Information

To collect logs and debugging information for the external controller, enter the **show tech-support controller** command. This command also creates another tar.gz file in bootflash bootflash:Tech-NXOS_DME_logs_20160809_134358.tar.gz, which also needs to be copied as part of the tech-support collection. This file is displayed as part of the a severity 2 syslog message at the time of command execution.

```
switch(config)# terminal monitor
switch(config)# show tech-support controller > sh_tech_ctrl
2016 Aug 9 13:43:58 switch %USER-2-SYSTEM_MSG: Open NxOS Logs stored as
bootflash:Tech-NXOS_DME_logs_20160809_134358.tar.gz, please collect this
file as part of show tech support -vsh
```

To collect logs and debugging information for the switches, enter the **show tech-support vxlan** and **show tech-support vxlan platform** commands.

To collect logs and debugging information for the object store, enter the **show tech-support object-store user admin** command. The object store log is stored in a file, and the command output displays the name and location of the log file (for example, /logflash/debug_logs/objstore_20160408_101450.log).

To collect logs and debugging information for the OVSDB plugin, enter the **guestshell run sudo ovsdb-plugin tech-support** command. As the OVSDB plugin logs are collected, you can store them in a different server to retrieve them. To do so, use the **run guestshell** command to connect to the Guest Shell, copy logs/tech-support to the bootflash, and SCP from the bootflash to the location where this log can be retrieved. The log is located at /usr/local/ovsdb/ovsdbplugin_20160809-142608.tar.gz and is displayed as part of the tech-support command execution.

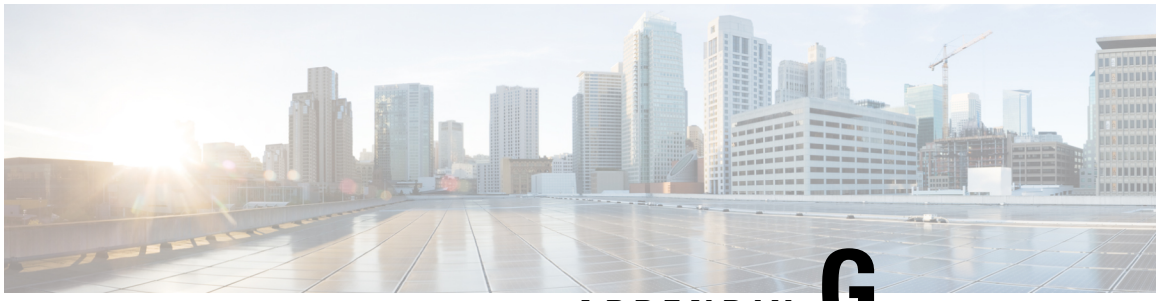
```
switch# guestshell run sudo ovsdb-plugin tech-support
nohup: ignoring input and appending output to 'nohup.out'
nohup: ignoring input and redirecting stderr to stdout
Data archive written to /usr/local/ovsdb/ovsdbplugin_20160809-142742.tar.gz
```

You can also collect logs and debugging information for the OVSDB plugin using SCP.

```
switch# guestshell run sudo ovsdb-plugin tech-support scp admin@172.31.145.141:/
--vrf management
```

```
nohup: ignoring input and appending output to 'nohup.out'
nohup: ignoring input and redirecting stderr to stdout
Data archive written to /usr/local/ovsdb/ovsdbplugin_20160809-142504.tar.gz
Sending archive to admin@172.31.145.141:/...
The authenticity of of host '172.31.145.141 (172.31.145.141)' can't be established.
RSA key fingerprint is bc:43:88:61:b5:27:ee:90:3b:fb:4a:b5:8c:38:ae:f9.
Are you sure you want to continue connecting(yes/no)? yes
Warning: Permanently added '172.31.145.141' (RSA) to the list of known hosts.
User Access Verification Password:
ovsdbplugin_20160809-142504.tar.gz 100%  20MB 20.0MB/s 00:01
```

To collect the OVSDB client dump from the OVSDB plugin, enter the **ovsdb-client -v dump tcp:1.1.24.1 hardware_vtep** command. The switch interface in this command should be reachable from the OVS client and should be in the same VRF on which the OVSDB plugin is running.



APPENDIX G

IP Addresses Used in vPC Systems

The following IP addresses are used in a vPC-based VTEP pair using in-band connectivity to the controller:

NVE Bound Loopback Interface – Secondary IP Addresses as VTEP IP Addresses

In this example, the NVE interface is bound to the loopback0 interface. This loopback interface must have two IP addresses, a primary IP address and a secondary IP address. In a vPC pair, the vPC switches must have a unique primary IP address and a common secondary IP address (1.1.106.1). This secondary IP address is the VTEP IP address of the vPC pair and is used in the data path.

vPC Primary Configuration

```
NSXv-TOR1# sh ru int nve 1

interface nve1
  no shutdown
  source-interface loopback0
  auto-remap-replication-servers
  host-reachability protocol controller 1
  source-interface hold-down-time 360
  config-source controller

NSXv-TOR1# sh ru int loopback 0

interface loopback0
  description vtep_loopback_interface
  ip address 1.1.101.1/32
  ip address 1.1.106.1/32 secondary
  ip router ospf 1 area 0.0.0.0
```

vPC Secondary Configuration

```
NSXv-TOR2# sh ru int nve 1

interface nve1
  no shutdown
  description L2
  source-interface loopback0
```

```
  auto-remap-replication-servers
  host-reachability protocol controller 1
  source-interface hold-down-time 360
  config-source controller

NSXv-TOR2# sh ru int loopback 0

interface loopback0
  ip address 1.1.102.1/32
  ip address 1.1.106.1/32 secondary
  ip router ospf 1 area 0.0.0.0
```

Cisco NX-OS does not support a dynamic change of the NVE loopback secondary IP address (the tunnel or VTEP IP address). Follow this procedure to change the VTEP IP address.

On a vPC pair, follow these steps:

1. Shut the NVE interface on both the primary and secondary vPC switches.
2. Remove the existing VTEP IP address (using the **no ip address** command) on the NVE loopback interface on both the primary and secondary vPC switches.
3. Add the new secondary IP address on both the primary and secondary vPC switches.
4. Enter the **no shut** command on the NVE interface on both the primary and secondary vPC switches.

On a standalone switch, follow these steps:

1. Shut the NVE interface.
2. Remove the existing primary VTEP IP address (using the **no ip address** command) on the NVE loopback interface.
3. Add the new primary IP address.
4. Enter the **no shut** command on the NVE interface.

Plugin Connectivity Loopback IP Addresses

For inband connections to the controller, Cisco recommends having a separate loopback address (for example, loopback1) configured on each switch in the vPC.

vPC Primary switch

```
NSXv-TOR1# sh ru int loopback 1

interface loopback1
  ip address 1.1.104.1/32
  ip router ospf 1 area 0.0.0.0
```

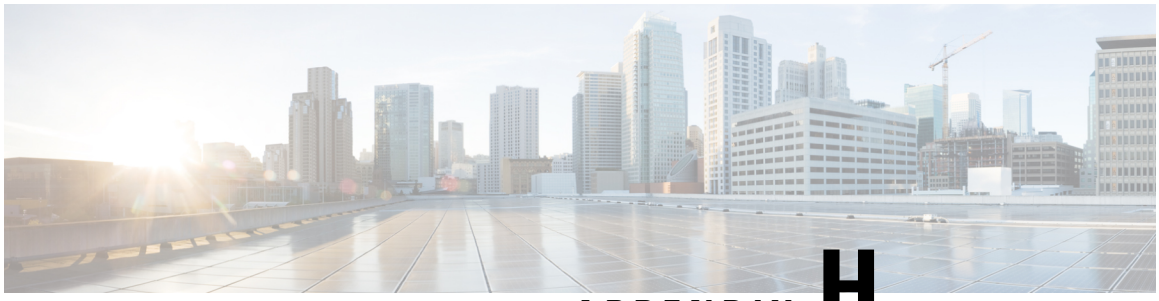
vPC Secondary switch

```
NSXv-TOR2# sh ru int loopback 1

interface loopback1
  ip address 1.1.105.1/32
  ip router ospf 1 area 0.0.0.0
```

The OVSDB plugin uses these IP addresses to connect to the switch:

```
NSXv-TOR2# guestshell run sudo ovsdb-plugin config show
  Controllers       :
    #1 addr         : 1.1.128.252:6640
  VPC               : Yes
  In switch         : Yes
  VRF               : default
  Switches          :
    #1 addr         : 1.1.105.1:443
       type         : LOCAL
       user         : admin
       name         : TOR1-TOR2
       description  : TOR1-TOR2

    #2 addr         : 1.1.104.1:443
       type         : REMOTE
       user         : admin
       name         : TOR1-TOR2
       description  : TOR1-TOR2
  Log type          : file
  Log level         : debug
  Log server        : -
  TTY log path      : -
  Max JSON peers    : 6
  Min heap size     : 2048 MB
  Max heap size     : 2048 MB
  Schema            : 1.3.0
```

IP Addresses Used in vPC Systems



APPENDIX H

Configuring the OVSDB Plugin When Deploying a New Controller

This appendix includes the following sections:

Configuring the OVSDB Plugin When Deploying a New Controller

Follow these steps to configure the OVSDB plugin when deploying a new controller (with a new IP address). These steps are recommended on an already installed pre-existing system to migrate to a new set of controllers.

Step 1 Enter the **guestshell run sudo ovsdb-plugin service stop** command on the vPC primary switch.

Step 2 Enter the **guestshell run sudo ovsdb-plugin service stop** command on the vPC secondary switch.

Step 3 Enter the **guestshell run sudo ovsdb-plugin config set** commands to reconfigure the OVSDB plugin on both the vPC primary and secondary switches. Make sure to use the new controller IP address in these commands.

Step 4 Enter the **guestshell run sudo ovsdb-plugin cert reset** command on the vPC primary switch. Check if the files in the config/ssl/ directory have new updated timestamps.

Step 5 Enter the **guestshell run sudo ovsdb-plugin cert reset** command on the vPC secondary switch. Check if the files in the config/ssl/ directory have new updated timestamps.

Step 6 Enter the **guestshell run sudo ovsdb-plugin cert reset --receive -v** command on the vPC primary switch.

Step 7 Enter the **guestshell run sudo ovsdb-plugin cert reset --send** *remote-vpc-primary-ip* **:6640 -v** command on the vPC secondary switch. Copy the receiver side certificate on the sender side.

Step 8 Enter the **guestshell run sudo ovsdb-plugin cert show** command on the vPC primary switch.

Step 9 Enter the **guestshell run sudo ovsdb-plugin cert show** command on the vPC secondary switch.

Step 10 Check if the output in Steps 8 and 9 are showing the same content.

Step 11 Paste the **cert show** output in vCenter as the TOR vPC pair's certificate.

Step 12 Enter the **guestshell run sudo ovsdb-plugin service start** command to start the OVSDB plugin on both the vPC primary and secondary switches.

Configuring the OVSDB Plugin When Deploying a New Controller