# CONTENTS

<table>
<thead>
<tr>
<th>P R E F A C E</th>
<th>Full Cisco Trademarks with Software License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>xi</td>
</tr>
<tr>
<td>Audience</td>
<td>xi</td>
</tr>
<tr>
<td>Related Documentation</td>
<td>xi</td>
</tr>
<tr>
<td>Communications, Services, and Additional Information</td>
<td>xii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P A R T  I</th>
<th>NFVIS 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>C H A P T E R  1</td>
<td>About Cisco Enterprise NFVIS 1</td>
</tr>
<tr>
<td></td>
<td>Benefits of Cisco Enterprise NFVIS 2</td>
</tr>
<tr>
<td></td>
<td>Supported Hardware Platforms 2</td>
</tr>
<tr>
<td></td>
<td>Key Tasks You can Perform Using Cisco Enterprise NFVIS 3</td>
</tr>
</tbody>
</table>

| C H A P T E R  2        | Installing Cisco Enterprise NFVIS Using the KVM Console 5 |
|                        | Installation Prerequisites 5                 |
|                        | Image Signing and Verification 6            |
|                        | RPM Signing 6                               |
|                        | RPM Signature Verification 6               |
|                        | Image Integrity Verification Using sha256sum 6 |
|                        | Entering BIOS Setup 7                      |
|                        | Installing Cisco Enterprise NFVIS on the Cisco UCS C220 M4 Rack Server or Cisco CSP 2100 7 |
|                        | Logging Into the CIMC GUI 7                |
|                        | Activating a Virtual Device 8              |
|                        | Mapping the Cisco Enterprise NFVIS Image 8 |
Configuration Examples for SPAN Session Scenarios 69
  Example: SPAN Session Traffic on a Physical Interface 69
  Example: SPAN Session Traffic on a LAN SRIOV 70
  Example: SPAN Session Traffic on a VLAN 71

CHAPTER 8
  Configuring Packet Capture 73

CHAPTER 9
  VM Image Packaging 75
    VM Image Packaging Utility 75
      Contents 75
      Usage 76
      NFVIS Specific Enhancements 80
      VM Packaging Utility Usage Examples 81
    Standard VM Image Packaging 82
      Generating a VM Package 83
    Appendix 83
      VM Image Package Files 83
      Package Manifest File 84
      Bootstrap Configuration File 84
      VM Image Properties File 85
      Example: Package.mf 89
      Example: Image Properties 90
      Example: Bootstrap Configuration File 91
      Image Properties Template File 91

CHAPTER 10
  Upgrading Cisco Enterprise NFVIS 93

CHAPTER 11
  Configuring vBranch High Availability 95
    Prerequisites for vBranch HA 95
    vBranch HA Design and Topology 96
    Enable Virtual NIC Failure Detection with Track Feature 97
    Isolating LAN and Transit Link Traffic for vBranch HA 99
    Packet Flow for vBranch HA 100
    Configuration Examples for vBranch HA 101
Example: Active Cisco ENCS Configuration with ISRv1 101
Example: Standby Cisco ENCS Configuration with ISRv2 103
Cisco ENCS Failure Points 104

CHAPTER 12  
Cisco ENCS Single WAN IP Deployment Scenarios 109
  Single WAN IP Deployment 109
  Preconfiguring the Cisco ENCS for a Single WAN IP Deployment 110
  Single WAN IP Deployment with Gigabit Ethernet Interface 0/0 111
  Single WAN IP Deployment with the 4G Interface 112

CHAPTER 13  
Resetting to Factory Default 115

CHAPTER 14  
Event Notifications 117
  nfvisEvent 118
  vmleEvent 127

CHAPTER 15  
Syslog Support 145
  Syslog Messages 147

CHAPTER 16  
SNMP Support on NFVIS 153
  Introduction about SNMP 153
  SNMP Operations 153
    SNMP Get 154
    SNMP Notifications 155
  SNMP Versions 155
  SNMP MIB Support 156
  Configuring SNMP Support 158

CHAPTER 17  
TACACS and RADIUS Support on NFVIS 163
  About RADIUS 163
  RADIUS Operation 163
  Configuring a TACACS+ Server 164
  Configuring RADIUS 165
## Specifying TACACS and RADIUS Authentication  

### Chapter 18  
**ENCS Switch Portal Configuration**  
- Switch Settings  
- Configuring Spanning Tree  
- Configuring 802.1x  
- Configuring LACP  
- Configuring VLAN  
- Configuring General Settings  
- Configuring Advanced Settings  
- Configuring Spanning Tree per Interface  

### Chapter 19  
**Configuring Secondary IP and Source Interface**  

### Chapter 20  
**Ports and Port Channels**  
- Configuring Port Channels  
  - Information About Port Channels  
  - Port Channels Bond Mode  
  - Port Channels LACP Mode  
- Creating a Port Channel  
- Adding a Port to a Port Channel  
- Adding a Port Channel to a Bridge  
- Deleting a Port Channel  
- Removing a Port from a Port Channel  
- Removing a Port Channel from a Bridge  
- Configuring LLDP  
- Configuring Admin Status of a Port  
- Tracking Changes for a Port  
- Speed, Duplex and Autonegotiation  

### Chapter 21  
**MSTP for ENCS 5400 8-Port Switch**  

### Chapter 22  
**ENCS 5400 Switch LLDP**  
- Understanding LLDP  

---

*Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.11.1*
Establishing DTLS Tunnel with vManage 217
NFVIS Notifications 219
Stats for Host and VM 219
System CLI 219
NFVIS Local Portal 219
Core Allocation for Host and CCM 219

CHAPTER 30
Enhancements to VM Image Packaging 221
NFVIS Specific Enhancements 221
Cisco SD-WAN Cloud OnRamp for Colocation Packaging Enhancements 222
VM Packaging Parameters 223
VM Packaging Utility Usage Examples 224
Packaging a VM 224
Preface

This guide provides information about how to install and configure Cisco Enterprise Network Function Virtualization Infrastructure Software (Cisco Enterprise NFVIS) on a supported Cisco hardware device. The guide also provides details on virtual machine deployments, configuration of software features, and life cycle management using Representation State Transfer (REST) application programming interface (API).

Audience

This guide is intended for network administrators and operators who are familiar with basic Linux installation and configuration requirements.

Related Documentation

- API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software
- Cisco Enterprise Network Function Virtualization Infrastructure Software Command Reference
- Cisco 5400 Enterprise Network Compute System Hardware Installation Guide
- Cisco 5400 Enterprise Network Compute System Data Sheet
- Getting Started Guide for Cisco UCS E-Series Servers and the Cisco UCS E-Series Network Compute Engine
- Cisco UCS C220 M4 Server Installation and Service Guide
- Configuration Guide for Cisco Network Plug and Play on Cisco APIC-EM
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.
PART I

NFVIS

• About Cisco Enterprise NFVIS, on page 1
• Installing Cisco Enterprise NFVIS Using the KVM Console, on page 5
• Setting Up System Configuration, on page 17
• Cisco Network Plug-n-Play Support, on page 47
• VM Life Cycle Management, on page 53
• VM Deployment Scenarios, on page 61
• SPAN Session or Port Mirroring, on page 67
• Configuring Packet Capture, on page 73
• VM Image Packaging, on page 75
• Upgrading Cisco Enterprise NFVIS, on page 93
• Configuring vBranch High Availability, on page 95
• Cisco ENCS Single WAN IP Deployment Scenarios, on page 109
• Resetting to Factory Default, on page 115
• Event Notifications, on page 117
• Syslog Support, on page 145
• SNMP Support on NFVIS, on page 153
• TACACS and RADIUS Support on NFVIS, on page 163
• ENCS Switch Portal Configuration, on page 167
• Configuring Secondary IP and Source Interface, on page 179
• Ports and Port Channels, on page 181
• MSTP for ENCS 5400 8-Port Switch, on page 187
• ENCS 5400 Switch LLDP, on page 189
• Secure Overlay and Single IP Configuration, on page 191
• Dual WAN Support, on page 195
• Configuring System Shell-Access, on page 199
• Switch Port Security, on page 201
About Cisco Enterprise NFVIS

Explore Content Hub, the all new portal that offers an enhanced product documentation experience. Content Hub offers the following features to personalize your content experience.

- Faceted Search to help you find content that is most relevant
- Customized PDFs
- Contextual Recommendations

Cisco Enterprise Network Function Virtualization Infrastructure Software (Cisco Enterprise NFVIS) is a Linux-based infrastructure software designed to help service providers and enterprises to design, deploy and manage network services. Cisco Enterprise NFVIS helps dynamically deploy virtualized network functions, such as a virtual router, firewall, and WAN acceleration, on a supported Cisco device. You do not always require a physical device for every network function. Automated provisioning and centralized management also eliminates costly truck rolls.

Cisco Enterprise NFVIS provides a Linux-based virtualization layer to the Cisco Enterprise Network Function Virtualization (ENFV) solution.

Cisco ENFV Solution Overview

The Cisco ENFV solution helps convert your critical network functions into a software which can deploy network services across dispersed locations in minutes. It provides a fully integrated platform that can run on top of a diverse network of both virtual and physical devices with the following primary components:

- Cisco Enterprise NFVIS
- VNFs
- Unified Computing System (UCS) and Enterprise Network Compute System (ENCS) hardware platforms
- Digital Network Architecture Center (DNAC)

For more details on the Cisco ENFV solution, see the Cisco Enterprise Network Functions Virtualization Solution Overview.

- Benefits of Cisco Enterprise NFVIS, on page 2
- Supported Hardware Platforms, on page 2
Benefits of Cisco Enterprise NFVIS

- Cost effective solution to consolidate multiple physical network appliances into a single server running multiple virtual network functions.
- Flexibility in deploying services quickly and in a timely manner.
- Cloud based VM life cycle management and provisioning.
- In-box life cycle management software to deploy and chain VMs dynamically on the platform.
- Programmable APIs.

Supported Hardware Platforms

Depending on your requirement, you can install Cisco Enterprise NFVIS on the following Cisco hardware platforms:

- Cisco 5100 Series Enterprise Network Compute System (Cisco ENCS)
- Cisco 5400 Series Enterprise Network Compute System (Cisco ENCS)
- Cisco UCS C220 M4 Rack Server
- Cisco Cloud Services Platform 2100 (CSP 2100)
- Cisco ISR4331 with UCS-E140S-M2/K9
- Cisco ISR4351 with UCS-E160D-M2/K9
- Cisco ISR4451-X with UCS-E180D-M2/K9
- Cisco UCS-E160S-M3/K9 Server
- Cisco UCS-E180D-M3/K9
- Cisco UCS-E1120D-M3/K9

Cisco ENCS

The Cisco 5100 and 5400 Series Enterprise Network Compute System combines routing, switching, storage, processing, and a host of other computing and networking activities into a compact one Rack Unit (RU) box. This high-performance unit achieves this goal by providing the infrastructure to deploy virtualized network functions and acting as a server that addresses processing, workload, and storage challenges.

Cisco UCS C220 M4 Rack Server

The Cisco UCS C220 M4 Rack Server is a high-density, general-purpose enterprise infrastructure and application server that delivers world class performance for a wide range of enterprise workloads, including virtualization, collaboration, and bare-metal applications.
Cisco CSP 2100

Cisco Cloud Services Platform 2100 (Cisco CSP 2100) is a software and hardware platform for data center network functions virtualization. This open kernel virtual machine (KVM) platform, with Red Hat Enterprise Linux (RHEL) 7.3 as the base operating system, is designed to host networking virtual services. Cisco CSP 2100 enables network, security, and load balancer teams to quickly deploy any Cisco or third-party network virtual service.

Note

Return Material Authorization (RMA) capability for CSP 2100 is not supported when in use with NFVIS.

Cisco UCS E-Series Server Modules

The Cisco UCS E-Series Servers (E-Series Servers) are the next generation of Cisco UCS Express servers. E-Series Servers are a family of size, weight, and power efficient blade servers that are housed within the Generation 2 Cisco Integrated Services Routers (ISR G2), Cisco 4400, and Cisco 4300 Series Integrated Services Routers. These servers provide a general-purpose compute platform for branch office applications deployed either as bare metal on operating systems, such as Microsoft Windows or Linux; or as virtual machines on hypervisors.

Supported VMs

Currently, the following Cisco supplied VMs and third party VMs are supported:

- Cisco ISRv
- Cisco Adaptive Security Virtual Appliance (ASAv)
- Cisco Virtual Wide Area Application Services (vWAAS)
- Linux Server VM
- Windows Server 2012 VM

Key Tasks You can Perform Using Cisco Enterprise NFVIS

- Perform VM image registration and deployment
- Create new networks and bridges, and assign ports to bridges
- Create custom flavors—a flavor is the customized profile of the VM image
- Perform service chaining of VMs
- Perform VM operations
- Verify system information including CPU, port, memory, and disk statistics

The APIs for performing these tasks are explained in the API Reference for Cisco Enterprise NFVIS.
From a Cisco Enterprise NFVIS command-line interface, you can connect to another server and VMs remotely using the SSH client.
CHAPTER 2

Installing Cisco Enterprise NFVIS Using the KVM Console

Installation Prerequisites, on page 5
Image Signing and Verification, on page 6
Entering BIOS Setup, on page 7
Installing Cisco Enterprise NFVIS on the Cisco UCS C220 M4 Rack Server or Cisco CSP 2100, on page 7
Installing Cisco Enterprise NFVIS on Cisco UCS E-Series Servers, on page 8
Installing Cisco Enterprise NFVIS on a Cisco ENCS 5100 and 5400, on page 12

Installation Prerequisites

Ensure that the following prerequisites are met:

• The IP address is configured for Cisco Integrated Management Controller (CIMC) as well as a login account with administrative privileges.
• The login account is set up with administrative privileges.
• The installation media for Cisco Enterprise NFVIS has an ISO image.
• The IP address of the system (required for remote access) is available.
• Hyper-threading is enabled in BIOS. By default, hyper-threading is enabled in BIOS on the UCS-C, UCS-E and ENCS platforms.

Note

The installation steps are slightly different for Cisco UCS and Cisco ENCS platforms. See the following sections for details:

Installing Cisco Enterprise NFVIS on the Cisco UCS C220 M4 Rack Server or Cisco CSP 2100, on page 7
Installing Cisco Enterprise NFVIS on Cisco UCS E-Series Servers, on page 8
Installing Cisco Enterprise NFVIS on a Cisco ENCS 5100 and 5400, on page 12
Assumptions

- The user is familiar with the supported hardware device, CIMC, Cisco Network Plug and Play, and Cisco Application Policy Infrastructure Controller Enterprise Module (APIC-EM).
- The initial setup of the hardware device is complete, and the device is ready for loading Cisco Enterprise NFVIS.
- The user is familiar with general Linux installation.

For more details on the supported hardware devices, see respective documentation available on Cisco.com.

Image Signing and Verification

Cisco Enterprise NFVIS supports RPM signing and signature verification for all RPM packages in the ISO and upgrade images. You can also verify the integrity of the Cisco Enterprise NFVIS ISO and upgrade images.

RPM Signing

All RPM packages in the Cisco Enterprise NFVIS ISO and upgrade images are signed to ensure cryptographic integrity and authenticity. This guarantees that the RPM packages have not been tampered with and the RPM packages are from Cisco Enterprise NFVIS. The private key, used for signing the RPM packages, is created and securely maintained by Cisco.

RPM Signature Verification

Cisco Enterprise NFVIS verifies all RPM packages during installation or upgrade. The following table describes the Cisco Enterprise NFVIS behavior when the signature verification fails during installation or upgrade.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Enterprise NFVIS 3.7.1 installation</td>
<td>If the signature verification fails while installing Cisco Enterprise NFVIS, the installation is aborted.</td>
</tr>
<tr>
<td>Cisco Enterprise NFVIS upgrade from 3.6.x to</td>
<td>The RPM signatures are verified when the upgrade is being performed. If the signature verification fails, an error is logged but the upgrade is completed.</td>
</tr>
<tr>
<td>Release 3.7.1</td>
<td></td>
</tr>
<tr>
<td>Cisco Enterprise NFVIS upgrade from Release 3.7.1 to later releases</td>
<td>The RPM signatures are verified when the upgrade image is registered. If the signature verification fails, the upgrade is aborted.</td>
</tr>
</tbody>
</table>

Image Integrity Verification Using sha256sum

RPM signing and signature verification can be done only for the RPM packages available in the Cisco NFVIS ISO and upgrade images. To ensure the integrity of all additional non-RPM files available in the Cisco NFVIS ISO image, a hash of the Cisco NFVIS ISO image is published along with the image. Similarly, a hash of the Cisco NFVIS upgrade image is published along with the image. To verify that the hash of Cisco NFVIS ISO image or upgrade image matches the hash published by Cisco, run the following command and compare the hash with the published hash:
Entering BIOS Setup

This section applies only to ENCS 5400 and UCS-E series routers.

When you enter the BIOS setup for the first time, ensure that you secure the BIOS by setting up an admin-level and a user-level password. You have to set up the admin password when you access the BIOS menu for the first time. The user password (which only gives access to a small subset of BIOS options) must be set inside the BIOS setup menu.

To set up the admin password, press F2 when the system boots up. You will be prompted to set the password.

To set up the user password, after you log in, go to the ‘Security’ tab and set the password.

Installing Cisco Enterprise NFVIS on the Cisco UCS C220 M4 Rack Server or Cisco CSP 2100

This section provides information about a series of tasks you need to perform to install Cisco Enterprise NFVIS on a Cisco UCS C220 M4 Rack Server or Cisco CSP 2100.

Logging Into the CIMC GUI

Before you begin

- Make sure that you have configured the IP address to access CIMC.
- If not installed, install Adobe Flash Player 10 or later on your local system.

For details on how to configure an IP address for CIMC, see the Set up CIMC for UCS C-Series Server guide on cisco.com.

Step 1
In your web browser, enter the IP address that you configured to access CIMC during initial setup.

Step 2
If a security dialog box displays, do the following:

a) Optional: Select the check box to accept all content from Cisco.

b) Click Yes to accept the certificate and continue.

Step 3
In the log in window, enter your username and password.

When logging in for the first time to an unconfigured system, use admin as the username and password as the password.

Step 4
Click Log In.

The Change Password dialog box only appears the first time you log into CIMC.
**Activating a Virtual Device**

You will have to launch the KVM Console to activate virtual devices.

**Before you begin**

Ensure that you have the Java 1.6.0_14 or a higher version installed on your local system.

---

**Step 1**
Download the Cisco Enterprise NFVIS image from a prescribed location to your local system.

**Step 2**
From CIMC, select the **Server** tab, and click **Launch KVM Console**.

*Note* A JNLP file will be downloaded to your system. You must open the file immediately after it is downloaded to avoid the session timeout.

**Step 3**
Open the renamed `.jnlp` file. When it prompts you to download Cisco Virtual KVM Console, click **Yes**. Ignore all security warnings and continue with the launch.

The KVM Console is displayed.

**Step 4**
From the **Virtual Media** menu on the KVM Console, select **Activate Virtual Devices**.

If prompted with an unencrypted virtual media session message, select **Accept this session**, and click **Apply**. The virtual devices are activated now.

---

**Mapping the Cisco Enterprise NFVIS Image**

**Step 1**
From the **Virtual Media** menu on the KVM Console, select **Map CD/DVD**.

**Step 2**
Browse for the installation file (ISO) on your local system, and select it.

**Step 3**
Click **Map Device**.

The ISO image file is now mapped to the CD/DVD.

---

**Installing Cisco Enterprise NFVIS on Cisco UCS E-Series Servers**

**Before you begin**

- Configure the UCS E interface on the Cisco ISR router.
- Configure the Gigabit Ethernet interface on the Cisco ISR router.
• Ensure that you have the IP address configured for CIMC access as well as a login account with administrative privileges.

• Ensure that the Cisco UCS E server has one of the following supported firmware versions or above:
  - BIOS UCSED.2.5.0.3 or later for UCS-E160D-M2/K9 and UCS-E180D-M2/K9
  - BIOS UCSES.1.5.0.5 or later for UCS-E140S-M2/K9

For more details on how to perform the basic configuration on the Cisco ISR routers, see the following guides:
• Sample Configuration on the Cisco ISR Router to Bring Up a Cisco UCS E Server, on page 10
• Getting Started Guide for Cisco UCS E-Series Servers, Release 1.0(2) Installed in the Cisco ISR 4451-X

For details on how to configure an IP address for CIMC, see the Getting Started Guide for Cisco UCS E-Series Servers, Release 1.0 on cisco.com.

---

Step 1
Log into CIMC.
For details, see Logging Into the CIMC GUI, on page 7

Step 2
From the Server tab, click Launch KVM Console.
The KVM Console opens in a separate window.

Step 3
From the KVM console, click the Virtual Media tab.

Step 4
In the Virtual Media tab, map the virtual media using either of the following methods:
a) Select the Mapped check box for the CD/DVD drive containing the operating system.
b) Click Add Image, browse, and select the Cisco Enterprise NFVIS ISO image, click Open to mount the image, and then select the Mapped check box for the mounted image.

You must keep the Virtual Media tab open during the installation process. Closing the tab unmaps all virtual media.

Step 5
From the Server tab, select BIOS.

Step 6
From the BIOS Actions area, select Configure Boot Order.
The Configure Boot Order dialog box appears.

Step 7
From the Device Types area, select CD/DVD Linux Virtual CD/DVD, and then click Add.

Step 8
Select HDD PCI RAID Adapter, and then click Add.

Step 9
Set the boot order sequence using the Up and Down options. The CD/DVD Linux Virtual CD/DVD boot order option must be the first choice.

Step 10
Click Apply to complete the boot order setup.

Step 11
Reboot the server by selecting the Power Off Server option from the Server Summary page in CIMC.

Step 12
After the server is down, select the Power On Server option in CIMC.

When the server reboots, the KVM console will automatically install Cisco Enterprise NFVIS from the virtual CD/DVD drive. The entire installation might take 30 minutes to one hour to complete.

Step 13
After the installation is complete, the system is automatically rebooted from the hard drive. Log into the system when the command prompt changes from "localhost" to "nfvis" after the reboot.
Wait for some time for the system to automatically change the command prompt. If it does not change automatically, press Enter to manually change the command prompt from "localhost" to "nfvis". Use admin as the login name and Admin123# as the default password.
Note: The system prompts you to change the default password at the first login attempt. You must set a strong password as per the on-screen instructions to proceed with the application. You cannot run API commands or proceed with any tasks unless you change the default password at the first login. API will return 401 unauthorized error if the default password is not reset.

Step 14: You can verify the installation using the System API or by viewing the system information from the Cisco Enterprise NFV portal.

What to do next
You can verify the default configuration, and set up initial IP configuration to launch the Cisco Enterprise NFV portal. For details, see Setting Up System Configuration.

Sample Configuration on the Cisco ISR Router to Bring Up a Cisco UCS E Server

The following sample configuration shows the basic configuration performed on the Cisco ISR 4451 router with DHCP enabled.

```plaintext
Last configuration change at 02:36:37 UTC Thu Feb 18 2016
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
!
hostname NFVIS-ISR4451
!
boot-start-marker
boot system bootflash:isr4300-universalk9.03.16.01a.S.155-3.S1a-ext.SPA.bin
boot-end-marker
!
!
vr definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
!
no aaa new-model
!
!
ip domain name cisco.com
!
!
subbere templating
!
multilink bundle-name authenticated
!
!)```
license udi pid ISR4331/K9 sn FDO192207MN
!
ucse subslot 1/0
  ucse access-port shared-lom console
  ucse ip address 172.19.183.172 255.255.255.0 default-gateway 172.19.183.1
  spanning-tree extend system-id
  !
redundancy
  mode none
  !
  !
  vlan internal allocation policy ascending
  !
  !
  interface GigabitEthernet0/0/0
    ip address 172.19.183.171 255.255.255.0
    media-type rj45
    negotiation auto
    !
  interface GigabitEthernet0/0/1
    no ip address
    shutdown
    negotiation auto
    !
  interface GigabitEthernet0/0/2
    no ip address
    shutdown
    negotiation auto
    !
  interface ucse1/0/0
    ip unnumbered GigabitEthernet0/0/0
    negotiation auto
    switchport mode trunk
    no mop enabled
    no mop sysid
    !
  interface ucse1/0/1
    no ip address
    no negotiation auto
    switchport mode trunk
    no mop enabled
    no mop sysid
    !
  interface GigabitEthernet0
    vrf forwarding Mgmt-intf
    no ip address
    shutdown
    negotiation auto
    !
  interface Vlan1
    no ip address
    shutdown
    !
    ip default-gateway 172.19.183.1
    ip forward-protocol nd
    no ip http server
    no ip http secure-server
    ip tftp source-interface GigabitEthernet0
    ip route 0.0.0.0 0.0.0.0 172.19.183.1
Installing Cisco Enterprise NFVIS on a Cisco ENCS 5100 and 5400

Note
Software or hardware RAID controller setup is not supported with Cisco ENCS in Cisco Enterprise NFVIS Release 3.5.1.

Before you begin

- Make sure that you have configured the IP address to access CIMC.
- If not installed, install Adobe Flash Player 10 or later on your local machine.

For details on how to configure an IP address for CIMC, see the Set up CIMC for UCS C-Series Server and Getting Started Guide for Cisco UCS E-Series Servers and the Cisco UCS E-Series Network Compute Engine on cisco.com.

Step 1
In your web browser, enter the IP address that you configured to access CIMC during initial setup.

Step 2
If a security dialog box displays, do the following:

a) Optional: Select the check box to accept all content from Cisco.
b) Click Yes to accept the certificate and continue.

Step 3
In the Log in window, enter your username and password.
When logging in for the first time to an unconfigured system, use admin as the username and password as the password.

Step 4
Click Log In.
The Change Password dialog box only appears the first time you log into CIMC.
Step 5  Change the password as appropriate and save. The CIMC home page is displayed.

Step 6  From the CIMC Server tab, select Summary, and click Launch KVM Console. The KVM Console opens in a separate window.

Step 7  From the Virtual Media menu on the KVM Console, select Activate Virtual Devices. If prompted with an unencrypted virtual media session message, select Accept this session, and click Apply. The virtual devices are activated now.

Step 8  From the Virtual Media menu on the KVM Console, select Map CD/DVD.

Step 9  Browse for the installation file (ISO) on your local system, and select it.

Step 10 Click Map Device. The ISO image file is now mapped to the CD/DVD.

Step 11 From the CIMC Server tab, select BIOS.

Step 12 From the BIOS Actions area, select Configure Boot Order. The Configure Boot Order dialog box appears.

Step 13 From the Device Types area, select CD/DVD Linux Virtual CD/DVD, and then click Add.

Step 14 Select HDD, and then click Add.

Step 15 Set the boot order sequence using the Up and Down options. The CD/DVD Linux Virtual CD/DVD boot order option must be the first choice.

Step 16 Click Apply to complete the boot order setup.

Step 17 Reboot the server by selecting the Power Off Server option from the Server Summary page in CIMC.

Step 18 After the server is down, select the Power On Server option in CIMC.

When the server reboots, the KVM console will automatically install Cisco Enterprise NFVIS from the virtual CD/DVD drive. The entire installation might take 30 minutes to one hour to complete.

Step 19 After the installation is complete, the system is automatically rebooted from the hard drive. Log into the system when the command prompt changes from "localhost" to "nfvis" after the reboot.

Wait for some time for the system to automatically change the command prompt. If it does not change automatically, press Enter to manually change the command prompt from "localhost" to "nfvis". Use admin as the login name and Admin123# as the default password.

Note  The system prompts you to change the default password at the first login. You must set a strong password as per the on-screen instructions to proceed with the application. You cannot run API commands or proceed with any tasks unless you change the default password at the first login. API will return 401 unauthorized error if the default password is not reset.

Step 20 You can verify the installation using the System API or by viewing the system information from the Cisco Enterprise NFVIS portal.

---

**Installing Cisco Enterprise NFVIS on a Cisco ENCS 5104**

Step 1  Create bootable usb with NFVIS image.
In this example, we used Rufus utility in Windows environment. Rufus utility can be downloaded [https://rufus.akeo.ie/](https://rufus.akeo.ie/).

For this example, following parameters were used to burn bootable NFVIS USB device:

- **Device:** USB stick
- **Partition scheme:** MBR
- **Filesystem:** FAT32
- **Cluster size:** use default
- **Volume label:** use default
- **Quick format:** checked
- **Create bootable:** select "ISO Image" and click next icon then choose NFVIS image.
- **Create extended label:** checked

Press **Start** and wait for completion.

Eject USB thumb drive

---

**Step 2**  
Insert USB device in one of USB slot in ENCS5104.

**Step 3**  
Power on system.

**Step 4**  
During system boot up, press F6 key.

Press **<DEL>** or **<F2>** to enter setup, **<F6>** Boot Menu, **<F12>** Network Boot in 5 seconds or press any key to continue.

**Step 5**  
Once you press F6, you will see the following screenshot to select which device you want to boot from. Select your USB device.

In the following screenshot example, there is STEC USB being used. That display will vary depending on your usb device vendor. Use the arrow key to select that device.
Step 6  Wait until installation is completed. System will be rebooted once installation is done.

Step 7  Log into the system with username admin and Admin123# as a default password

Step 8  You will be prompted and asked to change password at the first login. You must set a strong password per the on-screen instruction to proceed.

Step 9  You can verify the installation status using the System API or command line interface per the NFVIS user guide.

What to do next

You can verify the default configuration, and set up initial IP configuration to launch the Cisco Enterprise NFV portal. For details, see Setting Up System Configuration.
Setting Up System Configuration

- Default System Configuration on the Cisco ENCS, on page 17
- Default System Configuration on the Cisco UCS C220 M4 Server and Cisco CSP 2100, on page 19
- Default System Configuration on the Cisco UCS E-Series Servers, on page 20
- Setting Up Initial Configuration, on page 20
- User Roles and Authentication, on page 28
- Configuring the IP Receive ACL, on page 32
- Configuring Your Banner and Message of the Day, on page 33
- Setting the System Time Manually or With NTP, on page 34
- Enabling or Disabling the Portal Access, on page 35
- Configuring System Logs, on page 36
- Network File System Support, on page 37
- Secure Boot of host, on page 38
- Secure Boot of VNF, on page 39
- CIMC Control, on page 39
- DPDK Support, on page 42
- Backup and Restore NFVIS and VM Configurations, on page 44
- Grub Edit Protection, on page 44
- Route Distribution, on page 45

Default System Configuration on the Cisco ENCS

The diagram below illustrates the default network configuration of Cisco Enterprise NFVIS with the Cisco ENCS.
• LAN ports—Eight physical Gigabit Ethernet ports for inbound and outbound traffic.

• WAN port—You can use one of the dual media Ethernet ports (wan-br and wan2-br) for DHCP connection.

• Bridges—They form a Layer 2 domain between virtual network interface controllers (vNICs) of VMs. A vNIC is used by a virtual machine to provide virtual network interfaces by defining a range of MAC addresses. The default management IP address (192.168.1.1) for the NFVIS host is configured on the management port. Multiple VMs can use the same LAN port for local connectivity.
• Network—It is a segment Layer 2 bridge domain where only the specific VLAN traffic is allowed.

• Reserved VLANs in the LAN network on the ENCS 5400 platform—The VLAN range 2350-2449 is reserved for internal use and should not be used on the external switch ports and for virtual machines in the LAN ports”. Note that this limitation doesn't apply to the WAN ports.

• Internal 192.168.10.00/24 and 192.168.50.0/24 networks—The IP subnet 192.168.10.0/24 and 192.168.50.0/24 are used for the ENCS-5400 internal networks. A user should not use this IP subnet on the NFVIS management network. In the future NFVIS releases, this internal subnet will be isolated so that users can use this for NFVIS management.

The following networks and bridges are automatically configured. You can configure more as required.

• A LAN network (lan-net) and a LAN bridge (lan-br)
• A WAN network (wan-net) and a WAN bridge (wan-br)

wan2-net and wan2-br are the default configurations for ENCS 5400 and ENCS 5100.

The default networks and bridges cannot be deleted.

Default System Configuration on the Cisco UCS C220 M4 Server and Cisco CSP 2100

Configuring the networks in Cisco Enterprise NFVIS allows inbound and outbound traffic and VMs to be service chained. The following diagram illustrates the default network configuration:

Figure 3: Default Network Configuration with Cisco UCS C220 M4 and Cisco CSP 2100

The following networks and bridges are created by default, and cannot be deleted. You can configure more as required.

• A LAN network (lan-net) and a LAN bridge (lan-br)—The default static management IP address (192.168.1.1) for the NFVIS host is configured on the LAN bridge. All other ports for inbound and outbound traffic are associated with the LAN bridge. Any LAN port can be used to access the default static IP address. By default, the hostname is set to "nfvis".
A WAN network (wan-net) and a WAN bridge (wan-br)—This is created with the "eth0" port, and is configured to enable the DHCP connection.

By default, the first port on the device is associated with the WAN bridge. All the other ports on the device are associated with the LAN bridge.

For more details about the initial setup, see the Installing the Server chapter in the Cisco UCS C220 M4 Server Installation and Service Guide or Cisco Cloud Services Platform 2100 Hardware Installation Guide.

Default System Configuration on the Cisco UCS E-Series Servers

The following networks and bridges are created by default, and cannot be deleted. You can configure more as required.

- A LAN network (lan-net) and a LAN bridge (lan-br)—The default static management IP address (192.168.1.1) for the NFVIS host is configured on the LAN bridge. All other ports for inbound and outbound traffic are associated with the LAN bridge. By default, the hostname is set to "nfvis".
- A WAN network (wan-net) and a WAN bridge (wan-br)—The physical WAN ports are on the Cisco ISR module. They are not externally available on the Cisco UCS E server. The WAN traffic comes from the ISR WAN ports, and goes through the backplane to the Cisco UCS-E server. The backplane has one internal WAN interface (GE0) to establish connection with the Cisco UCS-E server. By default, the "GE0" interface is enabled for the DHCP connection.

For more details on the initial setup, see the Getting Started Guide for Cisco UCS E-Series Servers and the Cisco UCS E-Series Network Compute Engine.

Setting Up Initial Configuration

For initial login, use admin as the default user name, and Admin123# as the default password. Immediately after the initial login, the system prompts you to change the default password. You must set a strong password as per the on-screen instructions to proceed with the application. All other operations are blocked until default password is changed. API will return 401 unauthorized error if the default password is not reset.
If wan-br and wan2-br has not obtained IP addresses through DHCP, the zero touch deployment is terminated. To manually apply the IP configurations answer 'y' and the system proceeds with dhclient on wan-br until the configurations are changed. For dhclient to continue to request IP address for PnP flow on both WAN interfaces answer 'n'.

You must follow the rules to create a strong password:

- Must contain at least one upper case and one lower case letter.
- Must contain at least one number and one special character (# - _ * ?).
- Must contain seven characters or greater. Length should be between 7 and 128 characters.

You can change the default password in three ways:

- Using the Cisco Enterprise NFVIS portal.
- Using the CLI—When you first log into Cisco Enterprise NFVIS through SSH, the system will prompt you to change the password.
- Using PnP (for details, see the Cisco Network Plug-n-Play Support, on page 47).
- Using console - After the initial login using the default password, you are prompted to change the default password.

NFVIS Version: 3.10.0-9

Copyright (c) 2015-2018 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

The copyrights to certain works contained in this software are owned by other third parties and used and distributed under third party license agreements. Certain components of this software are licensed under the GNU GPL 2.0, GPL 3.0, LGPL 2.1, LGPL 3.0 and AGPL 3.0.

nfvis login: console (automatic login)

login:
login:
login:
login:
login: admin

Cisco Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS Version: 3.10.0-9

Copyright (c) 2015-2018 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

The copyrights to certain works contained in this software are owned by other third parties and used and distributed under third party license agreements. Certain components of this software are licensed under the GNU GPL 2.0, GPL 3.0, LGPL 2.1, LGPL 3.0 and AGPL 3.0.

admin@localhost's password:

admin connected from ::1 using ssh on nfvis

nfvis# show version
To commit the target configuration to the active (running) configuration, use the `commit` command in any configuration mode in Cisco Enterprise NFVIS Release 3.5.1 and later. Changes made during a configuration session are inactive until the `commit` command is entered. By default, the commit operation is pseudo-atomic, meaning that all changes must succeed for the entire commit operation to succeed.

**Connecting to the System**

**Using IPv4**

The three interfaces that connect the user to the system are the WAN and WAN2 interfaces and the management interface. By default, the WAN interface has the DHCP configuration and the management interface is configured with the static IP address 192.168.1.1. If the system has a DHCP server connected to the WAN interface, the WAN interface will receive the IP address from this server. You can use this IP address to connect to the system.

You can connect to the server locally (with an Ethernet cable) using the static management IP address; to connect to the box remotely using a static IP address, the default gateway needs to be configured.

You can connect to the system in the following three ways:

- Using the local portal—After the initial login, you are prompted to change the default password.
- Using the KVM console—After the initial login using the default password, you are prompted to change the default password.
- Using PnP—After the initial provisioning through PnP, the configuration file pushed by the PNP server must include the new password for the default user (admin).

**Using IPv6**

IPv6 can be configured in static, DHCP stateful and Stateless Autoconfiguration (SLAAC) mode. By default, DHCP IPv6 stateful is configured on the WAN interface. If DHCP stateful is not enabled on the network, the router advertisement (RA) flag decides which state the network stays in. If the RA shows Managed (M) flag, then the network stays in DHCP mode, even if there is no DHCP server in the network. If the RA shows Other (O) flag, then the network switches from DHCP server to SLAAC mode.

SLAAC provides ipv6 address and default gateway. Stateless dhcp is enabled in the SLAAC mode. If the server has dns and domain configured, then SLAAC also provides those values via stateless dhcp.

**Performing Static Configuration without DHCP**

Starting from NFVIS 3.10.1 release, for ENCS 5400 and ENCS 5100, wan2-br obtains an IP address from DHCP. To configure default gateway, first use `no bridges bridge wan2-br dhcp` command.

If you want to disable DHCP and use static configuration, initial configuration is done by setting the WAN IP address and/or management IP address, and the default gateway. You can also configure a static IP on a created bridge.

To perform initial configuration on the system without using DHCP:

```
configure terminal
```
system settings mgmt ip address 192.168.1.2 255.255.255.0
bridges bridge wan-br ip address 209.165.201.22 255.255.255.0
system settings default-gw 209.165.201.1
commit

Note
When an interface is configured with a static IP address, DHCP is automatically disabled on that interface.

Now you can either use the management IP or WAN IP to access the portal.

To configure static IPv6 on the WAN interface:

configure terminal
system settings mgmt ipv6 address 2001:DB8:1:1::72/64
bridges bridge wan-br ipv6 address 2001:DB8:1:1::75/64
system settings default-gw-ipv6 2001:DB8:1:1::76
commit

Note
When an interface is configured with a static IPv6 address, DHCP IPv6 is automatically disabled on that interface. There are three options for IPv6 - static, DHCP and SLAAC, out of which only one can be enabled at a time.

Configuring DHCP on the WAN or Management Interface

Note
Starting from NFVIS 3.10.1, you can configure DHCP on any bridge. You can only have one DHCP bridge or management interface active at a time, and cannot have DHCP and default gateway configured at the same time.

You can configure DHCP either on the WAN interface or the management interface; you cannot configure DHCP on both the interfaces simultaneously.

To configure DHCP on any one of the interfaces (WAN or management), delete the default gateway.

To configure DHCP on the management interface:

configure terminal
no system settings default-gw
system settings mgmt dhcp
commit
exit
hostaction mgmt-dhcp-renew

To configure DHCP IPv6 on the management interface:

configure terminal
no system settings default-gw-ipv6
system settings mgmt dhcp-ipv6
commit
exit
hostaction mgmt-dhcp-renew
To configure DHCP on the WAN interface:

```bash
cfg
no system settings default-gw
system settings wan dhcp
commit
exit
hostaction wan-dhcp-renew
```

**Note**
Starting from NFVIS 3.10.1, you can configure DHCP IPv6 on any bridge. You can only have one DHCP IPv6 bridge or management interface active at a time, and cannot have DHCP IPv6 and default gateway IPv6 or SLAAC IPv6 configured at the same time.

To configure DHCP IPv6 on the WAN interface:

```bash
cfg
no system settings default-gw-ipv6
system settings wan dhcp-ipv6
commit
exit
hostaction wan-dhcp-renew
```

### Configuring SLAAC on the WAN or Management Interface

**Note**
Starting from NFVIS 3.10.1, you can configure SLAAC IPv6 on any bridge. You can only have one SLAAC IPv6 bridge or management interface active at a time, and cannot have SLAAC IPv6 and default gateway IPv6 or DHCP IPv6 configured at the same time.

To configure SLAAC IPv6 on the WAN interface:

```bash
cfg
system settings wan slaac-ipv6
commit
```

To configure SLAAC IPv6 on the management interface:

```bash
cfg
system settings mgmt slaac-ipv6
commit
```

### Verifying Initial Configuration

The `show system settings-native` command is used to verify initial configuration. Use `show bridge-settings` and `show bridge-settings bridge_name` commands to verify the configuration for any bridge on the system.

Extract from the output of the `show system settings-native` command when both WAN and management interfaces have a static configuration:

```bash
system settings-native mgmt ip-info interface lan-br
system settings-native mgmt ip-info ipv4_address 192.168.1.2
system settings-native mgmt ip-info netmask 255.255.255.0
```
Excerpt from the output of the `show system settings-native` command when the management interface has a DHCP configuration and the WAN interface has a static configuration:

```
system settings-native mgmt ip-info interface MGMT
system settings-native mgmt ip-info ipv4_address 192.168.1.2
system settings-native mgmt ip-info netmask 255.255.255.0
!  
!  
```

Extract from the output of the `show system settings-native` command when the WAN interface has a DHCP configuration and the management interface has a static configuration:

```
system settings-native mgmt ip-info interface lan-br
system settings-native mgmt ip-info ipv4_address 209.165.201.2
system settings-native mgmt ip-info netmask 255.255.255.0
!  
!  
```

Extract from the output of the `show system settings-native` command when the management interface has a DHCP configuration and the WAN interface has a static configuration:

```
system settings-native mgmt dhcp enabled
system settings-native wan ip-info interface wan-br
system settings-native wan ip-info ipv4_address 209.165.201.22
system settings-native wan ip-info netmask 255.255.255.0
!  
!  
```

Extract from the output of the `show system settings-native` command when the WAN interface has a DHCP configuration and the management interface has a static configuration:
## Configuring VLAN for NFVIS Management Traffic

A VLAN is a method of creating independent logical networks within a physical network. VLAN tagging is the practice of inserting a VLAN ID into a packet header in order to identify which VLAN the packet belongs to.

You can configure a VLAN tag on the WAN bridge (wan-br) interface to isolate Cisco Enterprise NFVIS management traffic from VM traffic. You can also configure VLAN on any bridge on the system (wan2-br for ENCS5400 or ENCS 5100, and user-br for all systems)

### Note
You cannot have the same VLAN configured for the NFVIS management and VM traffic.

For more details on the VLAN configuration, see the Understanding and Configuring VLANs module in the *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide.*
To configure a VLAN:

```
configure terminal
bridges bridge wan-br vlan 120
commit
```

Verifying VLAN Configuration

Run the `show bridge-settings wan-br vlan` command to verify the VLAN configuration as shown below:

```
nfvis# show bridge-settings wan-br vlan
bridges bridge wan-br vlan 120
```

**VLAN APIs and Commands**

<table>
<thead>
<tr>
<th>VLAN APIs</th>
<th>VLAN Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/bridges/bridge/wan-br/vlan</td>
<td>• bridges bridge wan2-br vlan</td>
</tr>
<tr>
<td>• /api/config/bridges/bridge/wan2-br/vlan</td>
<td>• bridges bridge user-br vlan</td>
</tr>
<tr>
<td>• /api/config/bridges/bridge/user-br/vlan</td>
<td>• show bridge-settings wan-br vlan</td>
</tr>
<tr>
<td>• /api/operational/bridge-settings/bridge/wan-br/vlan</td>
<td>• show bridge-settings wan2-br vlan</td>
</tr>
<tr>
<td>• /api/operational/bridge-settings/bridge/wan2-br/vlan</td>
<td>• show bridge-settings user-br vlan</td>
</tr>
<tr>
<td>• /api/operational/bridge-settings/bridge/user-br/vlan</td>
<td>• show bridge-settings vlan</td>
</tr>
</tbody>
</table>

**Configuring System Routes**

In addition to the default routes in the system, you can configure additional system routes. This configuration is specifically useful when certain destinations are not reachable through the default routes.

While you can create a route just by providing the destination and prefix length, a valid route requires that you specify either a device or a gateway or both.

To configure additional system routes:

```
configure terminal
system routes route 209.165.201.1 dev lan-br
commit
```

Verifying the System Routes Configuration

To verify the system routes configuration, use the `show system routes` command as shown below:

```
nfvis# show system routes
DESTINATION PREFIXLEN STATUS
-----------------------------
209.165.201.1    12  -
209.165.201.2   12   -
209.165.201.3  24   -
```
System Routes APIs and Commands

<table>
<thead>
<tr>
<th>System Routes APIs</th>
<th>System Routes Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/system/routes</td>
<td>• system routes route</td>
</tr>
<tr>
<td>• /api/config/system/routes/route/&lt;host</td>
<td>• show system routes</td>
</tr>
<tr>
<td>destination,netmask&gt;</td>
<td></td>
</tr>
</tbody>
</table>

User Roles and Authentication

Role based access enables the administrator to manage different levels of access to the system's compute, storage, database, and application services. It uses the access control concepts such as users, groups, and rules, which you can apply to individual API calls. You can also keep a log of all user activities.

Table 1: Supported User Roles and Privileges

<table>
<thead>
<tr>
<th>User Role</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Owns everything, can perform all tasks including changing of user roles, but cannot delete basic infrastructure. Admin's role cannot be changed; it is always &quot;administrators&quot;.</td>
</tr>
<tr>
<td>Operators</td>
<td>Start and stop a VM, and view all information</td>
</tr>
<tr>
<td>Auditors</td>
<td>Read-only permission</td>
</tr>
</tbody>
</table>

Rules for User Passwords

The user passwords must meet the following requirements:

- Must have at least seven characters length or the minimum required length configured by the admin user.
- Must not have more than 128 characters.
- Must contain a digit.
- Must contain one of the following special characters: hash (#), underscore (_), hyphen (-), asterisk (*), and question mark (?).
- Must contain an uppercase character and a lowercase character.
- Must not be same as last five passwords.

Creating Users and Assigning Roles

The administrator can create users and define user roles as required. You can assign a user to a particular user group. For example, the user "test1" can be added to the user group "administrators".
All user groups are created by the system. You cannot create or modify a user group.

To create a user:

```bash
configure terminal
rbac authentication users create-user name test1 password Test1_pass role administrators
commit
```

To delete a user:

```bash
configure terminal
rbac authentication users delete-user name test1
commit
```

To change the password, use the `rbac authentication users user change-password` command in global configuration mode. To change the user role, use the `rbac authentication users user change-role` command in global configuration mode.

### User Management APIs and Commands

<table>
<thead>
<tr>
<th>User Management APIs</th>
<th>User Management Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/rbac/authentication/users</td>
<td>rbac authentication users</td>
</tr>
<tr>
<td>/api/operations/rbac/authentication/users</td>
<td>rbac authentication users user change-password</td>
</tr>
<tr>
<td>/user/&lt;user-name&gt;/change-password</td>
<td>rbac authentication users user change-role</td>
</tr>
<tr>
<td>/api/operations/rbac/authentication/users/user</td>
<td></td>
</tr>
<tr>
<td>/operate/change-role</td>
<td></td>
</tr>
<tr>
<td>/api/config/rbac/authentication/users/user?deep</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Minimum Length for Passwords

The admin user can configure the minimum length required for passwords of all users. The minimum length must be between 7 to 128 characters. By default, the minimum length required for passwords is set to 7 characters.

```bash
configure terminal
rbac authentication min-pwd-length 10
commit
```

### Minimum Password Length APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/rbac/authentication/</td>
<td>rbac authentication min-pwd-length</td>
</tr>
</tbody>
</table>
Configuring Password Lifetime

The admin user can configure minimum and maximum lifetime values for passwords of all users and enforce a rule to check these values. The default minimum lifetime value is set to 1 day and the default maximum lifetime value is set to 60 days.

When a minimum lifetime value is configured, the user cannot change the password until the specified number of days have passed. Similarly, when a maximum lifetime value is configured, a user must change the password before the specified number of days pass. If a user does not change the password and the specified number of days have passed, a notification is sent to the user.

Note

The minimum and maximum lifetime values and the rule to check for these values are not applied to the admin user.

```
configure terminal
rbac authentication password-lifetime enforce true min-days 2 max-days 30
commit
```

Password Lifetime APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/rbac/authentication/password-lifetime/</td>
<td>rbac authentication password-lifetime</td>
</tr>
</tbody>
</table>

Deactivating Inactive User Accounts

The admin user can configure the number of days after which an unused user account is marked as inactive and enforce a rule to check the configured inactivity period. When marked as inactive, the user cannot login to the system. To allow the user to login to the system, the admin user can activate the user account by using the `rbac authentication users user username activate` command.

Note

The inactivity period and the rule to check the inactivity period are not applied to the admin user.

```
configure terminal
rbac authentication account-inactivity enforce true inactivity-days 2
commit
```

Deactivate Inactive User Accounts APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/rbac/authentication/account-inactivity/</td>
<td>rbac authentication account-inactivity</td>
</tr>
</tbody>
</table>

Activating an Inactive User Account

The admin user can activate the account of an inactive user.
configure terminal
rbac authentication users user guest_user activate
commit

**Activate Inactive User Account APIs and Commands**

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/operations/rbac/authentication/users/user/username/activate</td>
<td>rbac authentication users user activate</td>
</tr>
</tbody>
</table>

**Certification**

**Generate Sign-Request**

```
nfvis(config)# system certificate signing-request ?
```

Possible completions:

- common-name
- country-code
- locality
- organization
- organization-unit-name
- state

The .csr file will be saved in /data/intdatastore/download/nfvvis.csr

Use the scp command to download the file.

**Install CA Sign Certificate**

After CA sign in, the user needs to use the scp command to upload the file into nfvis.

```
nfvis(config)# system certificate install-cert path file:///<full path of the file>
```

The path needs to start with "file://"

**Switch Certificate**

```
nfvis(config)# system certificate use-cert cert-type ca-signed
```

nginx process restarts after the switch.

The users cannot access the log files. The log files are added to all the user actions and the user can download and view some of the logs from portal. A notification is generated when the log files reach 75% capacity.

**Secure Copy Command**

The secure copy (scp) command allows only the admin user to secure copy a file from the Cisco NFVIS to an external system or from an external system to Cisco NFVIS. The scp command is:

```
scp source destination
```
For detailed information about how to use the `scp` command to copy to or from supported locations, see the `scp` section in *Cisco Enterprise Network Function Virtualization Infrastructure Software Command Reference*.

**Examples**

The following example copies the sample.txt file from intdatastore to an external system.

```
nfvis# scp intdatastore:sample.txt user@203.0.113.2:/Users/user/Desktop/sample.txt
```

The following example copies the test.txt file from an external system to intdatastore.

```
nfvis# scp user@203.0.113.2:/Users/user/Desktop/test.txt intdatastore:test_file.txt
```

The following example copies the test.txt file from an external system to USB.

```
nfvis# scp user@203.0.113.2:/user/Desktop/my_test.txt usb:usb1/test.txt
```

The following example copies the sample.txt file to an NFS location.

```
nfvis# scp user@203.0.113.2:/user/Desktop/sample.txt nfs:nfs_test/sample.txt
```

The following example copies the sample.txt file from an external system with IPv6 address.

```
nfvis# scp user@[2001:DB8:0:ABCD::1]:/user/Desktop/sample.txt intdatastore:sample.txt
```

The following example copies the nfvis_scp.log file to an external system.

```
nfvis# scp logs:nfvis_scp.log user@203.0.113.2:/Users/user/Desktop/copied_nfvis_scp.log
```

### Configuring the IP Receive ACL

To filter out unwanted traffic, you can configure `ip-receive-acl` to block or allow certain traffic based on the IP address and service ports.

To configure the source network for Access Control List (ACL) access to the management interface:

```
configure terminal
system setting ip-receive-acl 198.0.2.0/24
commit
```

**Verifying the Trusted IP Connection**

Use the `show running-config system settings ip-receive-ac` command to display the configured source network for ACL access to the management interface.

```
nfvis# show running-config system settings ip-receive-ac
system settings ip-receive-acl 198.51.100.11/24
service
 [ ssh https scpd]
action accept
priority 100
```
Port 22222 and Management Interface ACL

Management interface ACL provides the Access Control List (ACL) to restrict the traffic through the management interface for setting up different ACL of subnet inside a big subnet. From 3.7.1 release, port 22222 is closed by default on an NFVIS system.

To open port 22222:

```
configure terminal
system settings ip-receive-acl 0.0.0.0/0 service scpd priority 2 action accept
commit
```

Priority can be set to any number, as long as there is no other ACL that drops packets from same IP with lower priority number.

Use `no system settings ip-receive-acl` to close port 22222. When an entry is deleted from `ip-receive-acl`, all configurations to that source are deleted since the source IP address is the key. To delete one service, configure other services again.

From 3.8.1 release, only an admin user can use the scp command on this port to upload or download only from restricted folders like `/data/intdatastore/`.

Use the `show running-config system settings ip-receive-acl` command to verify the interface configuration:

```
nfvis# show running-config system settings ip-receive-acl
system settings ip-receive-acl 10.156.0.0/16
  service [ ssh https scpd ]
  action    accept
  priority 100
!```

Configuring Your Banner and Message of the Day

Cisco Enterprise NFVIS supports two types of banners: system-defined and user-defined banners. You cannot edit or delete the system-defined banner, which provides copyright information about the application. Banners are displayed on the login page of the portal.

You can post messages using the Message of the Day option. The message is displayed on the portal's home page when you log into the portal.

To configure your banner and message:

```
configure terminal
banner-motd banner "This is a banner" motd "This is the message of the day"
commit
```
Currently, you can create banners and messages in English only. You can view the system-defined banner using the `show banner-motd` command. This command does not display the user-defined banner or message.

### Banner and Message APIs and Commands

<table>
<thead>
<tr>
<th>Banner and Message APIs</th>
<th>Banner and Message Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/banner-motd</td>
<td></td>
</tr>
<tr>
<td>• /api/operational/banner-motd</td>
<td>• banner-motd</td>
</tr>
<tr>
<td>• /api/operational/banner-motd</td>
<td>• show banner-motd</td>
</tr>
</tbody>
</table>

### Setting the System Time Manually or With NTP

You can configure the Cisco Enterprise NFVIS system time manually or synchronise with an external time server using Network Time Protocol (NTP).

To set the system time manually:

```bash
configure terminal
system set-manual-time 2017-01-01T00:00:00
commit
```

NTP is automatically disabled when the time clock is set manually.

To set the system time using NTP IPv4:

```bash
configure terminal
system time ntp preferred_server 209.165.201.20 backup_server 1.ntp.esl.cisco.com
commit
```

To set the system time using NTP IPv6:

```bash
configure terminal
commit
```

### Verifying the System Time Configuration

To verify all system time configuration details, use the `show system time` command in privileged EXEC mode as shown below:

```bash
nfvis# show system time

system time current-time 2017-01-01T17:35:39+00:00
system time current-timezone "UTC (UTC, +0000)"
```

```
REMOTE   REFID  ST  T  WHEN  POLL  REACH  DELAY
```
OFFSET JITTER

<table>
<thead>
<tr>
<th>*calo-timeserver .GPS.</th>
<th>1</th>
<th>u</th>
<th>4</th>
<th>64</th>
<th>1</th>
<th>69.423</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2749736</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* sys.peer and synced, o pps.peer, # selected, + candidate, - outlyer, . excess, x falskicker, space reject

If the NTP server is invalid, it will not be displayed in the table. Also, when an NTP server is queried, if a response is not received before the timeout, the NTP server will also not be displayed in the table.

System Time APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th></th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/system/set-manual-time</td>
<td></td>
<td>• system time</td>
</tr>
<tr>
<td>• /api/config/system/time/ntp/preferred_server</td>
<td></td>
<td>• show system time</td>
</tr>
<tr>
<td>• /api/config/system/time/ntp/backup_server</td>
<td></td>
<td>• system set-manual-time</td>
</tr>
<tr>
<td>• /api/config/system/time/timezone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• /api/operational/system/time?deep</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enabling or Disabling the Portal Access

The Cisco Enterprise NFVIS portal access is enabled by default. You can disable the access if required.

To disable the portal access:

```
configure terminal
system portal access disabled
commit
```

You can enable the portal access using the `enable` keyword with the `system portal access` command.

Verifying the Portal Access

Use the `show system portal status` command to verify the portal access status as shown below:

```
nfv# show system portal status
system portal status "access disabled"
```
Portal Access APIs and Commands

<table>
<thead>
<tr>
<th>Portal Access APIs</th>
<th>Portal Access Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/system/portal</td>
<td>• system portal access</td>
</tr>
<tr>
<td>• /api/operational/system/portal/status</td>
<td>• show system portal status</td>
</tr>
</tbody>
</table>

Configuring System Logs

You can view system logs for troubleshooting purposes. There are two log types and five log levels. The two log types are configuration and operational.

The INFO and WARNING log levels are set by default respectively for the configuration and operational log types. You can change them as required. However, the change to the log level is not persisted across a reboot. After a reboot, the default log levels are used.

The following table explains the log levels:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>Information, typically of interest only when diagnosing problems.</td>
</tr>
<tr>
<td>INFO</td>
<td>Confirmation that things are working as expected.</td>
</tr>
<tr>
<td>WARNING</td>
<td>An indication that something unexpected happened, or indicative of some problem in the near future (for example, ‘disk space low’). The software application is still working as expected.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Due to a serious problem, the software application is not able to perform some function.</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>A serious error, indicating that the program itself may not be able to continue running.</td>
</tr>
</tbody>
</table>

You can configure system logs using the `system set-log` command in global configuration or privileged EXEC mode:

```
system set-log level error logtype configuration
```

Verifying the System Log Configuration

To verify the system log configuration, use the `show system logging-level` command as shown below:

```
nfvis# show system logging-level
system logging-level configuration error
system logging-level operational warning
```
System Log APIs and Commands

<table>
<thead>
<tr>
<th>System Log APIs</th>
<th>System Log Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/operations/system/set-log</td>
<td>• system set-log logtype</td>
</tr>
<tr>
<td>/api/operational/system/logging-level</td>
<td>[all/configuration/operational] level</td>
</tr>
<tr>
<td></td>
<td>[critical/debug/error/info/warning]</td>
</tr>
<tr>
<td></td>
<td>• show system logging-level</td>
</tr>
</tbody>
</table>

Network File System Support

The Network File System (NFS) is an application where the user can view, store and update the files on a remote device. NFS allows the user to mount all or a part of a file system on a server. NFS uses Remote Procedure Calls (RPC) to route requests between the users and servers.

NFS Mount and Unmount

To mount NFS:

```conf
configure terminal
system storage nfs_storage
nfs
100
10.29.173.131
/export/vm/amol
commit
```

To unmount NFS use `no system storage nfs_storage` command.

Image Registration on NFS

Images in tar.gz, ISO and qcow2 format, remote images and images on mounted NFS can be registered on NFS.

To register tar.gz images on NFS:

```conf
configure terminal
vm_lifecycle images image myas10 src file:///data/mount/nfs_storage/repository/asav961.tar.gz properties property placement value nfs_storage
commit
```

Similar configuration can be used for the various images formats.

To unregister an image from NFS use `no vm_lifecycle images` command.

Deploy VM on NFS

To deploy a VM on NFS, under deployment vm group use `placement type zone_host host nfs_storage` command.
Secure Boot of host

This feature is supported only on ENCS 5400. Upgrade BIOS to version 2.6 for this feature. The secure boot feature is only available for release 3.9.1 fresh install.

The secure boot feature prevents malicious software applications and unauthorized operating systems from loading into the system during the system start up process. If secure boot feature is enabled, only the authorized software applications boots up from the device. Each device has keys that allow software with the correct signature to boot up on the device.

This feature ensures that the software applications that boot up on the device are certified by Cisco. The NFVIS 3.9.1 image is signed with Cisco key. If secure boot is enabled the signature is verified during the device boot up. If the verification fails, the image does not boot up.

Secure boot is disabled by default and to enable it you must change firmware configurations from CIMC. Secure boot needs to boot from a separate UEFI partition.

To enable secure boot:

1. Get into CIMC and use **show bios detail** command to view the BIOS version.

   ENCS# scope bios
   ENCS/bios # show detail
   BIOS:
   
   BIOS Version: "ENCS54_2.6 (Build Date: 07/12/2018)"
   Boot Order: EFI
   FW Update/Recovery Status: Done, OK
   Active BIOS on next reboot: main
   UEFI Secure Boot: disabled
   ENCS/bios #

2. **enable secure boot**.

   ENCS/bios # set secure-boot enable
   Setting Value : enable
   Commit Pending.
   ENCS/bios *# commit
   ENCS/bios # show detail
   BIOS:
   
   BIOS Version: "ENCS54_2.6 (Build Date: 07/12/2018)"
   Boot Order: EFI
   FW Update/Recovery Status: None, OK
   Active BIOS on next reboot: main
   UEFI Secure Boot: enabled
   ENCS/bios #

Legacy boot, UEFI boot and UEFI secure boot are the three boot modes. Secure boot can only be used on a disk that has UEFI partition.

You can configure boot order from CIMC command or portal or from BIOS setup menu. With CIMC you can only configure legacy boot order. BootOrderRules are by default set to Strict and so boot order follows the CIMC configuration. Since CIMC cannot be used to configure UEFI boot order, the BootOrderRules setting must be changed to Loose when secure boot is enabled.
If BootOrderRules is set to Loose, the boot order will follow the BIOS setup menu. When an operating system is installed in secure boot mode, the new UEFI boot option for the OS automatically appears at the top of the BIOS menu boot order list, to boot the installed operating system.

To set BootOrderRule to Loose:

ENCS/bios # scope advanced
ENCS/bios/advanced # set BootOrderRules Loose
ENCS/bios/advanced *# commit
Changes to BIOS set-up parameters will require a reboot.
Do you want to reboot the system?[y|N]y

Secure Boot of VNF

Starting from NFVIS 3.11.1 release, support for UEFI secure boot of secure boot capable VNFs is added. A secure compute system ensures that the intended software on the system runs without malware or tampered software. This protection begins as soon as the system is powered-on. The UEFI (Unified Extensible Firmware Interface) specification defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature.

NFVIS already supports UEFI secure boot for the host to ensure that the NFVIS software boot up is genuine. This feature is now extended to UEFI secure boot to the VNF. VNF secure boot requires an environment to support UEFI secure boot and modification of the VNF to support secure boot. This release adds the infrastructure in NFVIS to support UEFI secure boot of secure-boot capable VNFs.

VNFs can indicate their secure boot capability using properties in the image_properties.xml file in the tar.gz package for the VNF. VNFs can boot in both BIOS and UEFI secure firmware modes.

The VNF shim is signed by Microsoft which is specified as an additional image property shim_signature with the value as microsoft or N/A

The supported secure boot combinations of VNFs are:

boot_mode: efi-secure-boot,
shim_signature: microsoft

boot_mode: bios,
shim_signature: N/A

Combinations that do not match the above will default to the BIOS mode. The Image Repository page on the NFVIS portal shows if the image is capable of secure boot.

CIMC Control

On ENCS 5400, NFVIS administrators have authoritative control of the device. This includes capability to change the IP address used to reach the CIMC and modifying the CIMC and BIOS passwords.
CIMC Access using NFVIS

**Note**
CIMC access using NFVIS is supported only on ENCS 5400.

To access CIMC using NFVIS WAN or management interface IP address, use the `system settings cimc-access enable` command. Once you configure CIMC access on NFVIS, the stand alone CIMC access using CIMC IP address is disabled and you will be able to access CIMC using NFVIS management interface IP address. The configurations remain on the device even after the device reboot.

When the CIMC access is configured, it enables a few ports to access services like SSH, SNMP, HTTP and HTTPS into the CIMC.

The following port numbers are being used for forwarding services to CIMC:

- 20226 for SNMP
- 20227 for SSH
- 20228 for HTTP
- 20229 for HTTPS

If you are unable to access CIMC using NFVIS, check the `show log nfvis_config.log` file.

Use `system settings cimc-access disable` to disable this feature.

**BIOS-CIMC Update**

Starting from 3.8.1 release, for ENCS 5400 router, if existing BIOS/CIMC version is lower than the bundled image in 3.8.1 NFVIS package, it is updated automatically during the NFVIS upgrade or installation. Also the CPU microcode is upgraded. The upgrade time takes longer than the previous releases and the upgrade will be done automatically, and you cannot stop the process once it is initiated.

For ENCS 5100 router, BIOS will be upgraded automatically to a new version but you need to boot up the server manually after the upgrade.

**BIOS and CIMC Password**

To change the BIOS and CIMC password for ENCS 5400 use `hostaction change-bios-password newpassword` or `hostaction change-cimc-password newpassword` commands. The change in the password will take effect immediately after the commands are executed. For both CIMC and BIOS passwords any alphanumeric character along with some special characters (@ # ) are allowed.

For CIMC, the password must contain a minimum of eight characters.

For BIOS, the password must contain a minimum of seven characters and the first letter cannot be #.
**BIOS and CIMC Password APIs and Commands**

<table>
<thead>
<tr>
<th>BIOS and CIMC Password APIs</th>
<th>BIOS and CIMC Password Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/hostaction/</td>
<td>• change-cimc-password</td>
</tr>
<tr>
<td></td>
<td>• change-bios-password</td>
</tr>
</tbody>
</table>

**NFVIS Password Recovery**

1. Load the NFVIS 3.9.1 image, using the CIMC KVM console.
2. Select Troubleshooting from the Boot Selection menu.
3. Select Rescue a NFVIS Password.
4. Select Continue.
5. Press Return to get a shell.
6. Run the `chroot /mnt/sysimage` command.
7. Run the `./nfvis_password_reset` command to reset the password to admin.
8. Confirm the change in password and enter Exit twice.
   Remove the CD-ROM and reboot NFVIS.
9. Login to NFVIS with the default credentials admin/Admin123#.
   After login to NFVIS, enter a new password at prompt.
10. Connect to NFVIS with the new password.

**Note**

You can update and recover NFVIS 3.8.1 and older passwords using NFVIS 3.9.1.

**Overview to ENCS 5400 for UEFI Secure Boot**

You can use Unified Extensible Firmware Interface (UEFI) secure boot to ensure that all the EFI drivers, EFI applications, option ROM or operating systems prior to loading and execution are signed and verified for authenticity and integrity, before you load and execute the operating system. You can enable this option using either web UI or CLI. When you enable UEFI secure boot mode, the boot mode is set to UEFI mode and you cannot modify the configured boot mode until the UEFI boot mode is disabled.

**Note**

If you enable UEFI secure boot on a nonsupported OS, on the next reboot, you cannot boot from that particular OS. If you try to boot from the previous OS, an error is reported and recorded the under system software event in the web UI. You must disable the UEFI secure boot option using Cisco IMC to boot from your previous OS.
Enabling UEFI Secure Boot Mode

To enable UEFI secure boot mode:

Server# scope bios
Server /bios # set secure-boot enable
Setting Value : enable
Commit Pending.
Server /bios *# commit

Reboot the server to have your configuration boot mode settings take place.

Disabling UEFI Secure Boot Mode

To disable UEFI secure boot mode:

Server# scope bios
Server /bios # set secure-boot disable
Setting Value : enable
Commit Pending.
Server /bios *# commit

Reboot the server to have your configuration boot mode settings take place.

To install NFVIS in UEFI mode, map the iso image through vmedia or kvm first, then enable secure boot and change the BIOS set-up parameters.

encs# scope bios
encs /bios # scope advanced
encs /bios/advanced # set BootOpRom UEFI
encs /bios/advanced # set BootOrderRules Loose
encs /bios/advanced *# commit

Reboot the device to start installation.

Note

All VNFs and configurations are lost at reboot. Secure boot in UEFI mode works differently from the legacy mode. Therefore, there is no compatibility in between legacy mode and UEFI mode. The previous environment is not kept.

DPDK Support

The Data Plane Development Kit (DPDK) is a set of data plane libraries and network interface controller drivers for fast packet processing. DPDK support is enabled only on ENCS 5400 from NFVIS 3.10.1 release. To enable DPDK use system settings dpdk enable command. Once DPDK is enabled it cannot be disabled. You can use factory reset to disable DPDK.

Note

Factory reset option all-except-images-connectivity does not disable DPDK support.

To enable DPDK support:
configure terminal
system settings dpdk enable
commit

DPDK can be enabled if:

• No VMs are deployed.
• There are no bridges other than wan-br, wan2-br, lan-br.

When you enable DPDK, reboot the system to avoid any system resource conflict as DPDK requires system level resource realignment such as CPU, memory for system and VMs. DPDK mode is enabled on a bridge, if the bridge is created as part of a network or bridge api without any NIC ports. NIC ports can also be added later to the bridge, if no VMs are deployed on the network associated to the bridge. If a NIC port is added to the bridge, the bridge will switch to non-dpdk mode. Once a bridge enters non-dpdk mode, it will not switch back to DPDK mode again. NFVIS supports DPDK for the interface with virtio driver only.

---

**Note**

NFVIS 3.10.x release does not support tcpdump packetcapture command on DPDK enabled bridge.

If DPDK is enabled, all VMs deployed will have DPDK and HugePage support. The default hugepage size is 2MB. After DPDK is enabled the system reserves 512 hugepages for Openvswitch operations. Hugepages for VM are allocated dynamically. If the system is not able to allocate HugePages for a newly deployed VM, the VM will boot up in error state. Memory Fragmentation is the main reason why HugePage allocation fails. In this case a reboot can help solve the issue.

---

**Note**

DPDK support is only enabled on the bridges without NIC ports.

For a system without Hyper-threading one additional core is reserved by the system and for a system with Hyper-threading two additional logical cores are reserved by the system.

NFVIS does not support changing Hyper-thread option such as disabling after DPDK is enabled with Hyper-thread. The system can be unstable if you change Hyper-thread setting after DPDK is enabled.

---

**DPDK VM Migration**

From NFVIS 3.11.1 release, you can enable DPDK on a system with deployed VMs. When you enable DPDK on a system, all deployed VMs are updated to support DPDK. During the migration process all VMs are shutdown and rebooted again later.

Any user-created bridges that does not have a physical NIC attached to it is updated to support DPDK ports. VMs that are connected to a user created bridge through a vNIC before the DPDK migration, is updated to DPDK supported vNIC.

It is recommended to perform a reboot before enabling DPDK. After the system is back online, you can proceed to enable DPDK.

To start the DPDK migration process:

configure terminal
system settings dpdk enable
commit
After enabling DPDK, check DPDK status and ensure it is enabled. To get the current status of DPDK use `show system settings-native dpdk-status` command. If DPDK is in Error state, it indicates VM migration failure due to lack of free memory. Use `no system settings dpdk` command to disable DPDK if DPDK is in error state.

For system stability with DPDK:

1. Before you enable DPDK for the first time, reboot the system.
2. Once DPDK is in enabled status, reboot the system again.

## Backup and Restore NFVIS and VM Configurations

Starting from NFVIS 3.10.1 release, you can backup and restore NFVIS configurations and VMs. You can also restore a backup from one NFVIS device to another if they are running on the same version of NFVIS and have the same platform.

To backup or restore a single VM, use `vmImportAction` and `vmBackupAction` APIs.

To backup and save NFVIS and all VM configurations use `configuration-only` option. To backup and save VM disks, NFVIS and VM configurations use `configuration-and-vms` option.

You can only create a backup to datastore or uploads directory. The backup file has `.bkup` extension.

The following examples show the backup options:

```
nfvis# hostaction backup configuration-and-vms file-path intdatastore:sample.bkup
nnfvis# hostaction backup configuration-only file-path extdatastore2:sample-dir/sample.bkup
```

The following example shows the backup stored on a USB:

```
nfvis# hostaction backup configuration-only file-path usb:usb1/sample.bkup
```

Use `hostaction backup force-stop` command to stop the running backup.

To restore a previous backup on same NFVIS setup or on a new NFVIS setup use `except-connectivity` option which preserves connectivity of the NFVIS and restores everything else from backup.

```
nfvis# hostaction restore file-path intdatastore:sample.bkup
```

The following example shows how to restore a backup on a different NFVIS device:

```
nfvis# hostaction restore except-connectivity file-path extdatastore2:sample-dir/sample.bkup
```

## Grub Edit Protection

In NFVIS 3.11.1 release, the grub menu is locked down and the user cannot modify the boot parameters. The user cannot edit the grub menu and will not be able to enter the grub command line.
Route Distribution

The route distribution feature notifies routes on the WAN IP address to the headend orchestrator or VPN Router. The headend orchestrator can ping the notified routes in its IP route table. Both static and DHCP WAN IP addresses are supported on this feature.

You can use IPv4 address routes in a network to exchange routing and reachability information with the VPN router configured with BGP. Routes are announced or withdrawn using ExaBGP depending upon the successful TCP connection established with the VPN router by ExaBGP process.

Before setting up route distribution configure the following fields:

- Neighbor IPv4 address
- Local bridge or Local address (optional)
- Local autonomous system number
- Remote autonomous system number
- Network subnets with optional next-hop IPv4 address (atleast one subnet)

ExaBGP establishes TCP connection with the neighbor IP address by sending TCP payload using the default port 179. After TCP connection is established, the network subnets mentioned in the configuration along with the optional next hop field for that network subnet are announced. If TCP connection is not established, the network subnets in the configuration are not announced.

To create or update route distribution:

```
configure terminal
route-distribute 172.16.0.0 local-bridge wan-br local-as 65000 remote-as 65000 network-subnet 192.0.2.1/24
commit
```

To display the state of route distribution use `show route-distribute` command. State of route distribution determines if TCP connection was established with neighbor machine or not. To remove the route distribution configuration use `no route-distribute` command.
Cisco Network Plug-n-Play Support

Starting from 3.10.1 release, NFVIS is integrated with PnP 1.8.

The Cisco Network Plug and Play (Cisco Network PnP) solution provides a simple, secure, unified, and integrated offering for enterprise network customers to ease new branch or campus device rollouts, or for provisioning updates to an existing network. The solution provides a unified approach to provision enterprise networks comprising Cisco routers, switches, and wireless devices with a near zero touch deployment experience. This solution uses Cisco Application Policy Infrastructure Controller Enterprise Module (APIC-EM) to centrally manage remote device deployments.

Currently, you can use the Cisco Network Plug and Play client to:

- Auto discover the server
- Provide device information to the server
- Bulk provisioning of user credentials

**Bulk Provisioning of User Credentials**

You can change the default username and password of the devices using the Cisco Network PnP client. The Cisco Network PnP server sends the configuration file to Cisco Network PnP clients residing on multiple devices in the network, and the new configuration is automatically applied to all the devices.

For bulk provisioning of user credentials, ensure that you have the necessary configuration file uploaded to the Cisco APIC-EM. The following are the supported configuration formats:

**Sample Format 1**

```xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <rbac xmlns="http://www.cisco.com/nfv/rbac">
    <authentication>
      <users>
        <user>
          <name>admin</name>
          <password>Cisco123#</password>
        </user>
      </users>
    </authentication>
  </rbac>
</config>
```
PnP Discovery Methods

When a device is powered on for the first time, the Cisco Network PnP agent discovery process, which is embedded in the device, wakes up in the absence of the startup configuration file, and discovers the IP address of the Cisco Network PnP server located in the Cisco APIC-EM. The Cisco Network PnP agent uses the following discovery methods:

- **Static IP address**—The IP address of the Cisco Network PnP server is specified using the `set pnp static ip-address` command.

- **DHCP with option 43**—The Cisco PnP agent automatically discovers the IP address of the Cisco Network PnP server specified in the DHCP option 43 string. For more details on how to configure DHCP for APIC-EM controller auto-discovery, see the Solution Guide for Cisco Network Plug and Play.

- **Domain Name System (DNS) lookup**—If DHCP discovery fails to get the IP address of the APIC-EM controller, for example, because option 43 is not configured, the Cisco Plug and Play Agent falls back on a DNS lookup method. Based on the network domain name returned by the DHCP server, it constructs a fully qualified domain name (FQDN) for the APIC-EM controller, using the preset hostname "pnpserver".

Sample Format 2

If you use format 2, the system will internally convert this format into format 1.

```xml
<aaa xmlns="http://tail-f.com/ns/aaa/1.1">
  <authentication>
    <users>
      <user>
        <name>admin</name>
        <password>User123#</password>
      </user>
    </users>
  </authentication>
</aaa>
```

For more details on the Cisco Network PnP solution and how to upload a configuration file, see the Configuration Guide for Cisco Network Plug and Play on Cisco APIC-EM.

- **PnP Discovery Methods**, on page 48
- **Configuring PnP Discovery Methods**, on page 49
- **PnP Action**, on page 52
For more details on how to configure DNS for APIC-EM controller auto-discovery, see the Solution Guide for Cisco Network Plug and Play.

Note
DNS lookup method is not supported in 3.10.1 release.

- Cloud Redirection—This method uses the Cisco Cloud Device Redirect tool available in the Cisco Software Central. The Cisco Plug and Play Agent falls back on the Cloud Redirection method if DNS lookup is not successful.

**Configuring PnP Discovery Methods**

To enable static mode for PnP discovery using IPv4:

```bash
configure terminal
pnp automatic dhcp disable
pnp automatic dns disable
pnp automatic cco disable
pnp static ip-address 192.0.2.8 port 80
commit
```

To enable static mode for PnP discovery using IPv6:

```bash
configure terminal
pnp automatic dhcp-ipv6 disable
pnp automatic dns-ipv6 disable
pnp automatic cco-ipv6 disable
pnp static ipv6-address 192.0.2.8 port 80
commit
```

Note
Either IPv4 or IPv6 can be enabled at a time.

To enable static mode for PnP discovery using FQDN:

```bash
configure terminal
pnp static ip-address apic-em-fqdn.cisco.com port 80 transport http
commit
```

Note
In FQDN support for PnP, domain names can be specified as an input. FQDN that is configured with IPv6 on a DNS server is not supported.

To enable automatic mode for PnP discovery using IPv4:
By default, the automatic discovery mode for DHCP, DNS, and CCO is enabled. You can enable or disable the options as required. For example, you can enable all options or keep one enabled, and the rest disabled.

```configure terminal
pnp automatic dhcp enable
pnp automatic dns enable
pnp automatic cco enable
pnp automatic timeout 100
commit
```

To enable automatic mode for PnP discovery using IPv6:

```configure terminal
pnp automatic dhcp-ipv6 enable
pnp automatic dns-ipv6 enable
pnp automatic cco-ipv6 enable
pnp automatic timeout 30
commit
```

You cannot disable both static and automatic PnP discovery modes at the same time. You must restart PnP action every time you make changes to the PnP discovery configuration. You can do this using the `pnp action command restart`.

**Verifying the PnP Status**

Use the `show pnp` command in privileged EXEC mode to verify the configuration of PnP discovery methods. The following sample output shows that the static discovery mode is enabled, and the automatic discovery mode is disabled.

```
nfvis# show pnp
pnp status response "PnP Agent is running\n"
pnp status ip-address 192.0.2.8
pnp status port 80
pnp status transport ""
pnp status created_by user
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status timeout 100
nfvis#
```

```FQDN
nfvis# show pnp
pnp status response "PnP Agent is running\nsnserver-connection\n  status: Success\n  time: 19:59:38 Feb 27\nbackoff\n  status: Success\n  time: 19:59:38 Feb 27\n"
pnp status ip-address apic-em-fqdn.cisco.com
pnp status ip6-address ""
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by user
pnp status dhcp_opt43 0
pnp status dns_discovery 0
```
The following sample output shows that the static discovery mode is disabled, and the automatic discovery mode is enabled for DHCP, DNS, and CCO:

**DHCP:**
```
nfvis# show pnp
pnp status response "PnP Agent is running\nlcli-exec\n status: Success\ntime: 18:30:57
Apr 21\nsnserver-connection\n status: Success\ntime: 15:40:41 Apr
22\ncertificate-install\n status: Success\ntime: 18:31:03 Apr 21\ndevice-auth\n status: Success\ntime: 18:31:08 Apr 21\nbackoff\n status: Success\ntime: 15:40:41
Apr 22"
pnp status ip-address 192.0.2.8
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by dhcp_discovery
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status timeout 60
```

**DNS:**
```
nfvis# show pnp
pnp status response "PnP Agent is running\nlcli-exec\n status: Success\ntime: 17:18:42
Apr 22\nsnserver-connection\n status: Success\ntime: 17:20:00 Apr
22\ncertificate-install\n status: Success\ntime: 17:18:47 Apr 22\ndevice-auth\n status: Success\ntime: 17:18:53 Apr 22\nbackoff\n status: Success\ntime: 17:20:00
Apr 22"
pnp status ip-address 192.0.2.8
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by dns_discovery
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status timeout 60
```

**CCO:**
```
nfvis# show pnp
pnp status response "PnP Agent is running\nlcli-exec\n status: Success\ntime: 17:18:42
Apr 22\nsnserver-connection\n status: Success\ntime: 17:20:00 Apr
22\ncertificate-install\n status: Success\ntime: 17:18:47 Apr 22\ndevice-auth\n status: Success\ntime: 17:18:53 Apr 22\nbackoff\n status: Success\ntime: 17:20:00
Apr 22"
pnp status ip-address 192.0.2.8
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by cco_discovery
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status timeout 60
```
### PnP Server APIs and Commands

<table>
<thead>
<tr>
<th>PnP Server APIs</th>
<th>PnP Server Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/pnp</td>
<td>• pnp static ip-address</td>
</tr>
<tr>
<td>• /api/config/pnp?deep</td>
<td>• pnp automatic</td>
</tr>
<tr>
<td></td>
<td>• show pnp</td>
</tr>
</tbody>
</table>

### PnP Action

You can start, stop, and restart any PnP action using the PnP action command or API.

### PnP Action API and Command

<table>
<thead>
<tr>
<th>PnP Action API</th>
<th>PnP Action Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/pnp/action</td>
<td>• pnp action command</td>
</tr>
</tbody>
</table>
VM Life Cycle Management

VM life cycle management refers to the entire process of registering, deploying, updating, monitoring VMs, and getting them service chained as per your requirements. You can perform these tasks and more using a set of REST APIs or NETCONF commands or the Cisco Enterprise NFVIS portal.

VM Packaging Format

All VM images are available in the .tar.gz/qcow2/vmdk/iso format. All Cisco supplied VMs are available in the prescribed format. Vendors are responsible for packaging all third party VMs in the prescribed format.

- Workflow of VM Life Cycle Management, on page 53
- Uploading VM Images to an NFVIS Server, on page 55
- VM Bootstrap Configuration Options with a VMDeployment, on page 56
- OpenStack Configuration Drive Support for Third Party VMs, on page 57
- Performing Resource Verification, on page 58
- Configuring Management IP Address, on page 59
- VM States, on page 59

Workflow of VM Life Cycle Management

The following diagram depicts the basic workflow of the VM life cycle management using REST APIs:
1. **Register a VM Image**—To register a VM image, you must first copy or download the relevant VM image to the NFVIS server, or host the image on a http or https server. Once you have downloaded the file, you can register the image using the registration API. The registration API allows you to specify the file path to the location (on the http/https server) where the tar.gz file is hosted. Registering the image is a one-time activity. Once an image is registered on the http or https server, and is in active state, you can perform multiple VM deployments using the registered image.

2. **Customizing the Setup**—After registering a VM image, you can optionally create a custom profile or flavor for the VM image if the profiles defined in the image file do not match your requirement. The flavor creation option lets you provide specific profiling details for a VM image, such as the virtual CPU on which the VM will run, and the amount of virtual memory the VM will consume.

   Depending on the topology requirement, you can create additional networks and bridges to attach the VM to during deployment.

3. **Deploy a VM**—A VM can be deployed using the deployment API. The deployment API allows you to provide values to the parameters that are passed to the system during deployment. Depending on the VM you are deploying, some parameters are mandatory and others optional.

4. **Manage and Monitor a VM**—You can monitor a VM using APIs and commands that enable you to get the VM status and debug logs. Using VM management APIs, you can start, stop, or reboot a VM, and view statistics for a VM such as CPU usage.

   A VM can also be managed by changing or updating its profile. You can change a VM's profile to one of the existing profiles in the image file; alternatively, you can create a new custom profile for the VM.

   The vNICs on a VM can also be added or updated.
Before performing the VM life cycle management tasks, you will have to upload the VM images to the NFVIS server or http/s server.

For details on APIs, see the VM Lifecycle Management APIs chapter in the API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software.

Uploading VM Images to an NFVIS Server

You can upload VM images to an NFVIS server in the following ways. The files are copied to the default location (/data/intdatastore/uploads) on the host server.

- Copy the images from your local system to the NFVIS server—Use the Image Upload option from the Cisco Enterprise NFVIS portal.
- Copy the images using the USB drive—Ensure that you have plugged the USB drive that contains the required images into the server before mounting the USB drive.
- Copy using the scp command (scp username@external_server:/path/image.tar.gz intdatastore:image.tar.gz).

From 3.8.1 release, NFVIS supports deleting images while the download is in progress. NFVIS also supports resuming image download after a power outage or lost connectivity.

To copy an image using the USB device:

```bash
configure terminal
system usb-mount mount ACTIVE
system file-copy usb file name usb1/package/isrv-universalk9.16.03.01.tar.gz
commit
```

Use the `show system file-list disk usb` command in privileged EXEC mode to view a list of files available with the mounted USB drive. To save space, you can delete all unwanted text and TAR files from the default location using the `system file-delete` command in global configuration mode.

Verifying the Image Copied from the USB Drive

After copying the file from the USB drive to the host server, you can verify the file using the `show system file-list disk local` command:

```bash
nfvis# show system file-list disk local
```

<table>
<thead>
<tr>
<th>SI NO</th>
<th>NAME</th>
<th>PATH</th>
<th>SIZE</th>
<th>TYPE</th>
<th>DATE MODIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lastlog-20170314.gz</td>
<td>/data/intdatastore/logs/2017-03/14/10-00</td>
<td>337K</td>
<td>Other</td>
<td>2017-03-14 21:55:42</td>
</tr>
<tr>
<td>2</td>
<td>escmanager-tagged-log.log-20170314.gz</td>
<td>/data/intdatastore/logs/2017-03/14/10-00</td>
<td>167K</td>
<td>Other</td>
<td>2017-03-14 21:55:42</td>
</tr>
</tbody>
</table>
VM Bootstrap Configuration Options with a VM Deployment

You can include the bootstrap configuration (day zero configuration) of a VM in the VM deployment payload in the following three ways:

- Bundle bootstrap configuration files into the VM package—In this method, the bootstrap configuration variables can be tokenized. Token names must be in bold text. For each tokenized variable, key-value pairs must be provided during deployment in the deployment payload.

- Bootstrap configuration as part of the deployment payload—The entire bootstrap configuration is copied to the payload without tokens.

- Bootstrap configuration file in the NFVIS server—In this method, the configuration file is copied or downloaded to the NFVIS server, and referenced from the deployment payload with the filename including full path.
For examples on how to use bootstrap configuration options in the deployment payload, see the API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software.

OpenStack Configuration Drive Support for Third Party VMs

To enable staging of bootstrap configuration files at the time of a third party VM deployment as per OpenStack standards, the following cloud init format is supported:

```
openstack/content
openstack/content/0000
openstack/content/0001
openstack/latest/meta_data.json
```

In the above sample, the "0000" and "0001" files are the actual bootstrap files from the deployment payload. A third party VM can use the init file to fetch its configuration files.

The following metadata file is used to provide the file path on the configuration drive and reference to the actual bootstrap configuration files.

```
{
  "files": [
    {
      "content_path": "content/0000",
      "path": "config/day-0.txt"
    },
    {
      "content_path": "content/0001",
      "path": "sample/path/iosxe_config.txt"
    }
  ]
}
```

With this implementation, two copies of the same bootstrap configuration file will be present on the virtual CD-ROM package. The first version at the root (iosxe_config.txt) and the second inside the "openstack/content" folder.

The admin will also have to specify the bootstrap configuration file in the image properties file before packaging the VM.

**Example for the Bootstrap Configuration File in the Image Properties File**

```
--optimize=OPTIMIZE [REQUIRED] optimized VM: --optimize=true/false;
--root_file_disk_bus=ROOT_FILE_DISK_BUS root disk file type:
--root_file_disk_bus=virtio/ide; default is virtio
--virtual_interface_model=VIRTUAL_INTERFACE_MODEL
--virtual_interface_model=rtl8139; default is none
--thick_disk_provisioning=THICK_DISK_PROVISIONING
--thick_disk_provisioning=true; default is false
--bootstrap_cloud_init_bus_type=BOOTSTRAP_CLOUD_INIT_BUS_TYPE
--bootstrap_cloud_init_bus_type=virtio; default is ide
--bootstrap_cloud_init_drive_type=BOOTSTRAP_CLOUD_INIT_DRIVE_TYPE
--bootstrap_cloud_init_drive_type=disk; default is cdrom
--bootstrap=BOOTSTRAP bootstrap file/s for VM (two parameters required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; upto 20 bootstrap files are accepted.)
examples:
--bootstrap ovf-env.xml:file1,ios-xe.txt:file2 for ISRv; both files get mounted at the...
Performing Resource Verification

Given below are the APIs and commands to perform different types of resource verification:

<table>
<thead>
<tr>
<th>Task</th>
<th>API</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>To display CPU information for each CPU</td>
<td>• api/operational/resources/cpu-info/cpus</td>
<td>show resources cpu-info cpus</td>
</tr>
<tr>
<td>or the user specified CPU, and the VMs</td>
<td>• /api/operational/resources/cpu-info/cpus/cpu</td>
<td></td>
</tr>
<tr>
<td>pinned to the CPU</td>
<td>• /api/operational/resources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/cpu-info/cpus/cpu/&lt;cpu-id&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/vnfs</td>
<td>show resources cpu-info vnfs</td>
</tr>
<tr>
<td></td>
<td>• /api/operational/resources/cpu-info/vnfs/vnf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• /api/operational/resources/cpu-info/vnfs/vnf/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;deployment_name&gt;.&lt;vm_group_name&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/allocation</td>
<td>show resources cpu-info allocation</td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/allocation</td>
<td></td>
</tr>
</tbody>
</table>

Note

To display information on all CPUs, VMs pinned to the CPUs, and VMs allocated to the CPUs, use the `show resources cpu-info` command.

CPU Over-Subscription

Cisco Enterprise NFVIS does not allow CPU over-subscription for low-latency network appliance VMs (for example, Cisco ISRv and Cisco ASAv). However, the CPU over-subscription is allowed for non low-latency VMs (for example, Linux Server VM and Windows Server VM).
Configuring Management IP Address

The following commands need to be executed in a sequence to first delete an exiting subnet and then add a new subnet in the network. For these commands to work, ensure there is no managed VNF’s in the system before you change management network address.

To delete an existing subnet use `no vm_lifecycle networks network int-mgmt-net subnet int-mgmt-net-subnet` command.

To create a new subnet:

```
configure terminal
vm_lifecycle networks network int-mgmt-net subnet int-mgmt-net-subnet address 105.20.0.0
gateway 105.20.0.1 netmask 255.255.255.0 dhcp false
commit
```

VM States

<table>
<thead>
<tr>
<th>VM States</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_UNDEF_STATE</td>
<td>The initial state of VM or VNF before deployment of this VM.</td>
</tr>
<tr>
<td>VM_DEPLOYING_STATE</td>
<td>VM or VNF is being deployed on to the NFVIS.</td>
</tr>
<tr>
<td>VM_MONITOR_UNSET_STATE</td>
<td>VM or VNF is deployed in the NFVIS but the monitoring rules are not applied.</td>
</tr>
<tr>
<td>VM_MONITOR_DISABLED_STATE</td>
<td>Due to a VM action request or recovery workflow, the monitoring or KPI rules applied on the VM or VNFs were not enabled.</td>
</tr>
<tr>
<td>VM_STOPPING_STATE</td>
<td>VM or VNF is being stopped.</td>
</tr>
<tr>
<td>VM_SHUTOFF_STATE</td>
<td>VM or VNF is in stopped or shutoff state.</td>
</tr>
<tr>
<td>VM_STARTING_STATE</td>
<td>VM or VNF is being started.</td>
</tr>
<tr>
<td>VM_REBOOTING_STATE</td>
<td>VM or VNF is being rebooted.</td>
</tr>
<tr>
<td>VM_INERT_STATE</td>
<td>VM or VNF is deployed but not alive. The KPI monitor is applied and waiting for the VM to become alive.</td>
</tr>
<tr>
<td>VM_ALIVE_STATE</td>
<td>VM or VNF is deployed and successfully booted up or alive as per the monitor or kpi metric.</td>
</tr>
<tr>
<td>VM_UNDEPLOYING_STATE</td>
<td>VM or VNF is being undeployed or terminated.</td>
</tr>
<tr>
<td>VM_ERROR_STATE</td>
<td>VM or VNF will be in error state if deployment or any other operation is failed.</td>
</tr>
</tbody>
</table>
VM Deployment Scenarios

This chapter provides details on the following deployment scenarios using REST APIs. As an example, the Cisco ENCS is used to illustrate these scenarios.

- Single VM deployment
- Service chaining with two VMs
- Service chaining of multiple VMs with Windows or Linux servers

The following VM images are used to explain the deployment scenarios:

- Cisco Integrated Services Router (ISRv)—isrv-03.16.02
- Cisco Adaptive Security Virtual Appliance (ASAv)—asav951-201
- Linux server—ubuntu-14.04.3-server-amd64-disk1

Registering VM Images, on page 61
Service Chaining of VMs, on page 65

Registering VM Images

You must register all VM images before deploying them.

Note

Register all the VM images required for the VM deployment depending on the topology. A VM image registration is done only once per VM image. You can perform multiple VM deployments using the registered VM image.

To register a Cisco ISRv image:

1. Set up the http/https server to host the VM image, or upload the image to the NFVIS server using the local portal or the `scp` command.
2. Register the Cisco ISRv image using the following API method:

```bash
curl -k -v -u admin:admin -H 'Accept: application/vnd.yang.data+xml' -H 'Content-Type: application/vnd.yang.data+xml' -X
```
POST https://<NFVIS_IP>/api/config/vm_lifecycle/images -d
'<image><name>isrv-k9.16.03.01</name><src>http://filename_with_full-path-of
the-file/isrv-universalk9.16.03.01.tar.gz</src></image>'

3. Verify the image status using the following API method:

curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET
https://<NFVIS_IP>/api/operational/vm_lifecycle/opdata/images/image/isrv-9.16.03.01?deep

4. Now, repeat Steps 1 to 3 to register the Cisco ASAv and Linux server images. Ensure that you provide
the exact image name and source file location when running the API commands.

---

Note
You can run API commands from any console/server that can reach Cisco Enterprise NFVIS.

---

Single VM Deployment

In this example, a Cisco ISRv image with three network interfaces is deployed. The following diagram
illustrates the deployment topology:

*Figure 6: Single VM Deployment*

---

Steps for Deploying a VM

To deploy a Cisco ISRv image:

1. Verify that all networks required for your deployment are configured.

curl -k -v -u admin:admin -H content-type:application/vnd.yang.data+xml -X
GET https://<NFVIS_IP>/api/config/networks?deep
2. Before deploying the VM, you can perform a resource check to ensure that you have sufficient resources for the deployment.

```bash
curl -k -v -u "admin:admin" -X GET https://<NFVIS_IP>/api/operational/resources/precheck/vnf/newvnf,isrv-small,true?deep
```

3. Deploy the Cisco ISRv VM.

```bash
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://<NFVIS_IP>/api/config/vm_lifecycle/tenants/tenant/admin/deployments --data <deployment>  
<name>ISR</name>  
<vm_group>  
  <name>ISR</name>  
  <image>isrv-universalk9.16.03.01/image>  
  <bootup_time>600</bootup_time>  
  <recovery_wait_time>0</recovery_wait_time>  
  <recovery_policy>  
    <action_on_recovery>REBOOT_ONLY</action_on_recovery>  
  </recovery_policy>  
  <flavor>isrv-small</flavor>  
</vm_group>  
<internet>  
  <interface>  
    <nicid>0</nicid>  
    <network>int-mgmt-net</network>  
    <port_forwarding>  
      <port>  
        <type>ssh</type>  
        <protocol>tcp</protocol>  
        <vnf_port>22</vnf_port>  
        <external_port_range>  
          <start>20022</start>  
          <end>20022</end>  
        </external_port_range>  
      </port>  
    </port_forwarding>  
  </interface>  
  <interface>  
    <nicid>1</nicid>  
    <network>lan-net</network>  
    <ip_address>209.165.201.0</ip_address>  
  </interface>  
  <interface>  
    <nicid>2</nicid>  
    <network>wan-net</network>  
    <ip_address>209.165.201.1</ip_address>  
  </interface>  
</interfaces>  
<scaling>  
  <min_active>1</min_active>  
  <max_active>1</max_active>  
</scaling>  
<kpi_data>  
  <kpi>  
    <event_name>VM_ALIVE</event_name>  
    <metric_value>1</metric_value>  
    <metric_cond>GT</metric_cond>  
    <metric_type>UINT32</metric_type>  
    <metric_collector>  
      <type>ICMPPing</type>  
    </metric_collector>  
  </kpi>  
</kpi_data>
```
If two VM's are connected to the same physical interface, one over S/RIOV and another over virtio net from a bridge connected to the packet filter, you cannot ping between the two VM's between these interfaces. Use S/RIOV or virtio net on both the VM's to connect to the packet filter over a bridge.

4. Verify the deployment status.

```
```

To enable NIM support on a Cisco ISRv running on Cisco ENCS, you must use the following variable in the ISRv deployment payload.

```
<variable>
  <name>ngio</name>
  <val>enable</val>
</variable>
```
Service Chaining of VMs

Service chaining here refers to a set of network services in the form of VMs using an intermediate network. Cisco Enterprise NFVIS supports service chaining of two or more VMs eliminating the need of dedicated hardware devices for different types of network services.

To service chain traffic between two or more VMs, you will have to create the following:

- Bridge—For example, you can create a new bridge called sc-br.
- Network—For example, you can create a new network called sc-net.
- Launch VM1 and VM2 with an interface from each VM to the service chain network (sc-net).

For more details on how to configure service chaining using APIs, see the following topics:

- Service Chaining with two VM Images, on page 65
- Service Chaining of Multiple VMs with Windows or Linux Servers, on page 66

Service Chaining with two VM Images

In this example, a Cisco ISRv VM and a Cisco ASAv VM are service chained. For that, you will have to deploy both VMs.

Figure 7: Service Chaining with two VM Images

Steps for Service Chaining with Two VM Images

1. Create a new bridge for service chaining.

   ```
   curl -skv -u admin:admin -H content-type:application/vnd.yang.data+xml -X POST https://<NFVIS_IP>/api/config/bridges --data '<bridge><name>sc-br</name></bridge>'
   ```

2. Create a new network for service chaining, and attach the bridge to the network.

   ```
   curl -skv -u admin:admin -H content-type:application/vnd.yang.data+xml -X POST https://<NFVIS_IP>/api/config/networks --data
   ```
3. Verify that all bridges and networks are configured.
4. Deploy the Cisco ISRv VM, and verify the deployment status.
5. Deploy the cisco ASAv VM, and verify the deployment status.

See Steps for Deploying a VM, on page 62 for API command details for Steps 3 to 5.

Service Chaining of Multiple VMs with Windows or Linux Servers

In this example, multiple VMs will be service chained. Cisco ISRv and Cisco ASAv VMs can be deployed as explained in Service Chaining with two VM Images, on page 65.

This section covers Linux server deployment (Windows 2012 server can also be deployed using the same steps.)

Figure 8: Service Chaining of Multiple VMs with Windows or Linux Servers

Steps for Service Chaining of Multiple VMs with Windows or Linux Servers

1. Create networks and bridges as required.

   See Steps 1 and 2 in Steps for Service Chaining with Two VM Images, on page 65 for details on creating networks and bridges.

2. Deploy Cisco ISRv and Cisco ASAv, and verify their deployment status.

3. Deploy the Linux server VM.

4. Verify the server deployment status.

   See the Steps for Deploying a VM, on page 62 for API command details for Steps 2 to 4.
SPAN Session or Port Mirroring

- About SPAN Sessions, on page 67
- Configuring SPAN Sessions, on page 67
- Configuration Examples for SPAN Session Scenarios, on page 69

About SPAN Sessions

The Switched Port Analyzer (SPAN) or Port Mirroring feature helps you analyze network traffic passing through interfaces or VLANs by using SPAN sessions. The SPAN sessions send a copy (mirror) of the traffic to another interface or VLAN on the switch that has been connected to a network analyzer or monitoring device. SPAN does not affect the switching of network traffic on the source interfaces.

Note

You must dedicate a destination port for SPAN use. Except for traffic that is required for the SPAN session, destination ports do not receive or forward traffic. When the SPAN is configured on the system, there might be some performance hit.

SPAN Session Interfaces

The interface can be:

- Physical interface
- LAN SRIOV
- VM's vNIC (virtio net)

In the case of virtio net or SRIOV VF, you have to specify the VM group name and NIC ID of the VM interface. If the VM vNIC is virtio net type, then the SPAN session is applied on the OVS bridge. If VM vNIC is SRIOV VF, then the mirror is applied to the hardware bridge. The interface name is specified for a physical interface, for example, GE0-0 or eth0.

Configuring SPAN Sessions

The SPAN session configuration has the following four parameters:
• Session number—Each SPAN session is identified with a unique number.

• Bridge name—The SPAN session is applied to a bridge. For VLAN mirroring, the bridge must be specified. The bridge name is optional if the source or destination interface is configured for the session.

• Source configuration—The source of the mirror traffic can be one of the following:
  • Packets entering (Rx), or exiting (Tx), or both. You can specify multiple interfaces of any type.
  • You can also specify all interfaces on the OVS bridge.
  • All packets entering a VLAN. You can also specify a list of VLANs.

• Destination configuration—The destination for the mirrored traffic can be one of the following:
  • The mirrored traffic can be sent to interfaces of any type.
  • The mirrored traffic can be sent to a specific VLAN. In this case, the original VLAN tag is stripped in the mirrored traffic in favor of the destination VLAN. This loss of original VLAN information might make the mirrored traffic hard to interpret.

To configure a SPAN session:

```
configure terminal
monitor session 2
bridge wan-br
source interface GE0-0
destination vm-vnic Linux2 0
commit
```

Verifying the SPAN Session Configuration

Use the `show system monitor session` command to verify the SPAN session configuration.

```
nfvis# show system monitor session
system monitor session 2
  bridge wan-br
  destination_vlan ""
  destination_interface vnic0
  source_vlans ""
  source_rx_interfaces "GE0-0"
  source_tx_interfaces "GE0-0"
  source_all false
  statistics "tx_bytes=142660, tx_packets=1380"
```

Use the `show running-config monitor session` command to verify the interface configuration for a SPAN session:

```
nfvis# show running-config monitor session
monitor session 2
  destination vm-vnic Linux2 0
  source vm-vnic Linux1 0 both
  source interface GE0-0 both
```
### SPAN Session APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/monitor</td>
<td>• monitor session</td>
</tr>
<tr>
<td>• /api/operational/monitor/?deep</td>
<td>• bridge</td>
</tr>
<tr>
<td>• /api/config/monitor/?deep</td>
<td>• source</td>
</tr>
<tr>
<td>• /api/operational/system/monitor/session/?deep</td>
<td>• destination</td>
</tr>
<tr>
<td></td>
<td>• show system monitor session</td>
</tr>
<tr>
<td></td>
<td>• show monitor session status</td>
</tr>
<tr>
<td></td>
<td>• show running-config monitor session</td>
</tr>
</tbody>
</table>

### Configuration Examples for SPAN Session Scenarios

#### Example: SPAN Session Traffic on a Physical Interface

The following example shows how to configure all traffic coming in or going out on GE0-0 (physical interface) and VM Linux1 (vnic0). And traffic is mirrored to the VM Linux2 (vnic1). With this configuration, any traffic arriving on vnet1 will be dropped.

An existing SPAN session will be in FAIL state after the system reboot. In this case, you need to recreate (delete and create) the SPAN session after the system bootup.

VM deployment interfaces:
- SPAN source: GE0-0 (traffic in both directions)
- SPAN source: Linux1/vnic0, and wan-net (traffic in both directions)
- SPAN destination: Linux2/vnic0, and wan-net

```bash
nfvis# show running-config monitor session
monitor session 20
   destination vm-vnic Linux2 0
   source vm-vnic Linux1 0 both
      source interface GE0-0 both
!

nfvis#

nfvis# show system monitor session
system monitor session 20
   bridge wan-br
   destination_vlan ""
   destination_interface vnic11
   source_vlans
   source_rx_interfaces "vnic10, GE0-0"
```
Example: SPAN Session Traffic on a LAN SRIOV

The following example shows how to configure all traffic coming in or going out on an SRIOV interface (VF0). It is also mirrored to VF1.

Note

This scenario is applicable only to the Cisco ENCS.

VM deployment for VF-VF scenario:
CentOS_SRIOV, C3, and C5 are CentOS VMs with SRIOV support.

- CentOS_SRIOV: vnic0: wan-net/vnic1: LAN-SRIOV-1 (192.168.1.36)
- C3: vnic0: LAN-SRIOV3 (192.168.1.3)
- C5: vnic0: LAN-SRIOV5 (192.168.1.5)

SPAN destination and source:

- SPAN destination: CentOS_SRIOV (vnic0: wan-net/vnic1: LAN-SRIOV-1)
- SPAN source: C3 (vnic0: LAN-SRIOV-3); traffic in both directions (rx, tx)
- Ping target: C5 (vnic0: LAN-SRIOV-5)
Example: SPAN Session Traffic on a VLAN

The following example shows how to configure the SPAN session for all traffic entering in VLAN 10 and 11. It is also mirrored to VLAN 20.

nfvis# show running-config monitor session
monitor session 11
  bridge lan-br
  destination vlan 20
  source vlan [ 10 11 ]

nfvis# show system monitor session
system monitor session 11
  bridge lan-br
  destination_vlan 20
  destination_interface ""
  source_vlans "10, 11"
  source_rx_interfaces ""
  source_tx_interfaces ""
  source_all true
  statistics "tx_bytes=0, tx_packets=0"

nfvis# show monitor session 11
NUMBER STATUS
------------------------
11 CREATE_SUCCESS
Example: SPAN Session Traffic on a VLAN
CHAPTER 8

Configuring Packet Capture

The Packet Capture feature helps you capture all packets being transmitted and received over physical and virtual network interface controllers (physical port and vNIC) for analysis. These packets are inspected to diagnose and solve network problems. Packets are stored in the /data/intdatastore/pktcaptures folder on the host server.

Benefits

- You can customize the configuration to capture specific packets such as Internet Control Message Protocol (ICMP), TCP, UDP, and Address Resolution Protocol (ARP).
- You can specify a time period over which packets are captured. The default is 60 seconds.

To configure packet capture on a physical port:

```
configure terminal
tcpdump port eth0
```

Output: pcap-location /data/intdatastore/pktcaptures/tcpdump_eth0.pcap

To configure packet capture on a vNIC:

```
configure terminal
tcpdump vnic tenant-name admin deployment-name 1489084431 vm-name ROUTER vnic-id 0 time 30
```

Output: pcap-location /data/intdatastore/pktcaptures/1489084431_ROUTER_vnic0.pcap

Types of Errors

<table>
<thead>
<tr>
<th>Error</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port/vnic not found</td>
<td>When non-existing interface is given as input.</td>
</tr>
<tr>
<td>File/directory not created</td>
<td>When the system is running out of disk space.</td>
</tr>
<tr>
<td>The tcpdump command fails</td>
<td>When the system is running out of disk space.</td>
</tr>
</tbody>
</table>

These errors are logged in the nfvis_config.log. By default, warnings and errors are logged,
### Packet Capture APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/packet-capture/tcpdump</td>
<td>• tcpdump port</td>
</tr>
<tr>
<td></td>
<td>• tcpdump vnic</td>
</tr>
</tbody>
</table>
VM Image Packaging

VM Image Packaging is a tool for converting qcow2 and img images into a tar.gz format with additional properties and profiles. VM image packaging can be done in two ways:

- **VM Image Packaging Utility**: This is an enhanced packaging process that allows the VM owner to run the `nfvpt.py` utility as a command with a combination of parameters to package the VM.

- **Standard Image Packaging**: This is a manual process in which a raw disk image (qcow2, img) is packaged along with the image properties file and bootstrap files (if needed) into a TAR archive file.

### VM Image Packaging Utility

A VM image package is a TAR archive file with the root disk image and other descriptor files. This packaging method simplifies the process of a VM image registration and deployment. The attributes specified for the image enable resource requirement specification, creation of VM profiles, and a host of other properties for the VM.

The Cisco Enterprise NFVIS VM image packaging tool, `nfvpt.py`, helps VM owners package their VMs. The tool takes one or more qcow2 images (raw disk file) as the input file along with VM specific properties, bootstrap configuration files (if any), and generates a compressed TAR file.

### Contents

The VM image packaging utility contains the following:

- **nfvpt.py**—It is a python based packaging tool that bundles the VM raw disk image/s along with VM specific properties.

- **image_properties_template.xml**—This is the template file for the VM image properties file, and has the parameters with default values. If the user provides new values to these parameters while creating the VM package, the default values get replaced with the user-defined values.

- **nfvis_vm_packaging_utility_examples.txt**—This file contains examples on how to use the image packaging utility to package a VM image.
Usage

To get the list of parameters that can be included in the command, and to get an explanation of each of the parameters, run the help command for the tool.

nfvpt.py --help
optional arguments:
  -h, --help              show this help message and exit
  --json JSON             Provide JSON input for bootstrap variables; mutually exclusive with custom and bootstrap configs
  --newjson NEWJSON       Provide JSON input for bootstrap variables; mutually exclusive with custom and bootstrap configs
  --log_dir LOG_DIR       Log Directory to for logfiles
  --multi_use             Add options for use in multiple use-cases
  --console_type_serial {true,false}   Attach the console serial to the VM; default is false; --console_type_serial=true/false;
  --root_file_disk_bus {virtio,ide}    root disk file type: --root_file_disk_bus=virtio/ide; default is virtio
  --virtual_interface_model {rtl8139}   --virtual_interface_model=rtl8139; default is none
  --thick_disk_provisioning {true,false} --thick_disk_provisioning=true/false; default is false
  --eager_zero {true,false}          --eager_zero=true/false; default is false
  --nocloud {true,false}       --nocloud=true/false; default is false
  --bootstrap_cloud_init_bus_type {ide,virtio} --bootstrap_cloud_init_bus_type=ide/virtio; default is ide
  --bootstrap_cloud_init_drive_type {cdrom,disk} --bootstrap_cloud_init_drive_type=cdrom/disk; default is cdrom
  --bootstrap BOOTSTRAP
    Every bootstrap file should be a different option Non HA format: --bootstrap
    <mountpoint>:<file1>,<mountpoint>:<file2>... See usage.txt for more details HA format for SDWAN
    NetworkHub: --bootstrap mount_point::<value>,file::<file2mount>[],<attrib>::<value> mount_point::<value> and
    file::<file2mount> are mandatory followed by one or more attributes in the format <attrib>::<value>
  --interface_hot_add {true,false}              VM supports interface add without power off. Default is set to true; --interface_hot_add=true/false
  --interface_hot_delete {true,false}            VM supports interface delete without power off. Default is set to false; --interface_hot_delete=true/false
  -v, --verbose             verbose
  -q, --quiet               quiet
  --no_compress             creates tar file without compressing the input files
  --cleanup                 deletes all the input and configuration files upon tar file created
  --tablet {true,false}     : Add input device of type tablet --tablet=true/false;
  --ha_package              enable HA packaging
  --mgmt_vnic MGMT_VNIC     VM management interface identifier
  --pack_dir <DIR> PACK     package all files in directory
Required:
- ` PACKAGE_FILENAME, --package_filename PACKAGE_FILENAME`  
  [REQUIRED] file name for the target VNF package name- 
  default is root disk image name with extension .tar.gz
- ` ROOT_DISK_IMAGE, --root_disk_image ROOT_DISK_IMAGE`  
  [REQUIRED] List of root disk images to be bundled  
  example: --root_disk_image isrv.qcow2;  
  --root_disk_image isrv1.qcow2,isrv2.qcow2
- ` --prop_template PROP_TEMPLATE`  
  image properties template file name including path  
  default path is the current dir of the tool and name  
  is image_properties_template.xml if the user doesn't  
  input this option example: --prop_template  
  /usr/bin/image_properties_template.xml
- ` VNF_TYPE, --vnf_type VNF_TYPE`  
  [REQUIRED] VNF type, e.g. ROUTER, FIREWALL, vWAAS,  
  vWLC, and OTHER
- ` NAME, --vnf_name NAME`  
  [REQUIRED] Name of the VNF image
- ` VNF_VERSION, --vnf_version VNF_VERSION`  
  [REQUIRED] VNF version, e.g. --vnf_version 1.0 or  
  --vnf_version 0.9
- ` --app_vendor APP_VENDOR`  
  Application Vendor e.g. Cisco, Juniper etc
- ` --monitored {true,false}`  
  [REQUIRED] Monitored VNF: --monitored=true/false;
- ` --optimize {true,false}`  
  [REQUIRED] optimized VM: --optimize=true/false;

HA options:
- ` --ha_capable`  
- ` --ha_vnic HA_VNIC VM HA vnic`  
- ` --ha_vnic_count HA_VNIC_COUNT`  
  Number of ha_vnics

Resources:
Resources: min and max = vCPU, memory and disk
- ` --min_vcpu VCPU_MIN`  
  min #vCPU : min number of vCPU supported by VM  
  example:--min_vcpu 2
- ` --max_vcpu VCPU_MAX`  
  max #vCPU : max number if vCPU required for VM  
  example:--max_vcpu 4
- ` --min_mem MEMORY_MB_MIN`  
  min mem : min mem in MB required for VM  
  example:--min_mem 1024
- ` --max_mem MEMORY_MB_MAX`  
  max mem : max mem in MB required for VM  
  example:--max_mem 4196
- ` --min_disk ROOT_DISK_GB_MIN`  
  min disk : min disk in GB required for VM  
  example:--min_disk 8
- ` --max_disk ROOT_DISK_GB_MAX`  
  max disk : max disk in GB required for VM  
  example:--max_disk 8
- ` --vnic_max VNIC_MAX`  
  max number of Vnics allowed for VM example:--vnic_max 8
- ` --vnic_names VNIC_NAMES`  
  list of vnic number to name mapping in format  
  number:name example --vnic_names  
  1:GigabitEthernet2,2:GigabitEthernet4

Profile Options:
- ` --profile PROFILE`  
  enter the profile name, profile description, no of  
  vCPU required, min memory required in MB, min disk
space required in MB, example: --profile
profile1,"This is profile 1",2,2048,4096 --profile
profile2,"This is profile 2",4,4096,4096
--default_profile DEFAULT_PROFILE
default profile

Driver Support Options:
   --sriov {true,false} Enable/Disable SRIOV support: --sriov=true/false;
default is false
   --sriov_list SRIOV_DRIVER_LIST
       list of SRIOV drivers example: --sriov_list
       igb,igbvf,i40evf
   --pcie {true,false} Not supported
   --pcie_list PCIE_DRIVER_LIST
       Not supported

Privilege/Priority Options:
   --privileged {true,false}
       Not supported

Custom Properties:
   --custom CUSTOM custom properties format: --custom ["propattr_"<attr>:
   <value>],key:<value>,[keyattr_<attr>:<value>],Type:<value>]
   Allows specification of custom properties: 0 or more
   propattr_<attr>:<value> pairs - 'propattr' is a
   keyword and used to specify property attributes
   key:<value> pairs 0 or more keyattr_<attr>:value pairs
   - 'keyattr' is a keyword and is used to specify key
   attributes type:<value> pair - type of value
   valN:<value> pair - val1:value,val2:value etc 0 or
   more valNattr_<attr>:<value> pairs - 'valN attrib'
is an attribute for valN See usage_examples.txt

The table lists the parameters that can be passed to the nfvpt.py command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Not applicable</td>
<td>Show program's version number and exit.</td>
</tr>
<tr>
<td>help</td>
<td>Not applicable</td>
<td>Show this help message and exit.</td>
</tr>
<tr>
<td>package_file_name</td>
<td>Mandatory</td>
<td>File name for the target VNF package. The default is the root disk image name with extension .tar.gz.</td>
</tr>
<tr>
<td>disk_img_names</td>
<td>Mandatory</td>
<td>List of root disk images to be bundled. Only the qcow2 images are supported.</td>
</tr>
<tr>
<td>img_name</td>
<td>Mandatory</td>
<td>Name of the VNF image.</td>
</tr>
<tr>
<td>vnf_type</td>
<td>Mandatory</td>
<td>VNF type Supported types are: ROUTER, FIREWALL, vWAAS, vWLC, and OTHER.</td>
</tr>
<tr>
<td>vnf_version</td>
<td>Mandatory</td>
<td>VNF version</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>monitored</td>
<td>Mandatory</td>
<td>VM health monitoring for those VMs that can be bootstrapped</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options are: true/false</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitoring timeout period for a monitored VM is 600 seconds by default</td>
</tr>
<tr>
<td>optimize</td>
<td>Mandatory</td>
<td>Optimized VM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options are: true/false</td>
</tr>
<tr>
<td>virtual_interface_model</td>
<td>Optional</td>
<td>Default is none.</td>
</tr>
<tr>
<td>thick_disk_provisioning</td>
<td>Optional</td>
<td>Default is false.</td>
</tr>
<tr>
<td>eager_zero</td>
<td>Optional</td>
<td>Default is false.</td>
</tr>
<tr>
<td>bootstrap_cloud_init_bus_type</td>
<td>Optional</td>
<td>Default is IDE.</td>
</tr>
<tr>
<td>bootstrap_cloud_init_drive_type</td>
<td>Optional</td>
<td>Mounts the day0 configuration file as disk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is CD-ROM.</td>
</tr>
<tr>
<td>bootstrap</td>
<td>Optional</td>
<td>Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRx and --bootstrap day0-config for ASAv.</td>
</tr>
<tr>
<td>min_vcpu</td>
<td>Optional</td>
<td>Minimum number of vCPUs supported by the VM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is 1.</td>
</tr>
<tr>
<td>max_vcpu</td>
<td>Optional</td>
<td>Maximum number of vCPUs required for the VM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is 8.</td>
</tr>
<tr>
<td>min_mem</td>
<td>Optional</td>
<td>Minimum memory in MB required for the VM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is 4 GB.</td>
</tr>
<tr>
<td>max_mem</td>
<td>Optional</td>
<td>Maximum memory in MB required for the VM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical memory: 2 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is 8 GB.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>min_disk</td>
<td>Optional</td>
<td>Minimum disk in GB required for the VM. The default is 8 GB.</td>
</tr>
<tr>
<td>max_disk</td>
<td>Optional</td>
<td>Maximum disk in GB required for the VM. Available disks are SSD and HDD: 15 GB. The default is 16 GB</td>
</tr>
<tr>
<td>vnic_max</td>
<td>Optional</td>
<td>Maximum number of VNICS allowed for the VM. The default is 8.</td>
</tr>
<tr>
<td>profile</td>
<td>Optional</td>
<td>The profile name, profile description, number of vCPUs required, minimum memory required in MB and minimum disk space required in MB.</td>
</tr>
<tr>
<td>default_profile</td>
<td>Optional</td>
<td>The default profile.</td>
</tr>
<tr>
<td>sriov</td>
<td>Optional</td>
<td>Enable or disable SRIOV support. The default is false.</td>
</tr>
<tr>
<td>sriov_list</td>
<td>Optional</td>
<td>List of SRIOV drivers.</td>
</tr>
<tr>
<td>pcie</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>pcie_list</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>privileged</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>custom</td>
<td>Optional</td>
<td>Custom properties to be supported and/or passed to the bootstrap configuration with tokenized variables. This is only used for the local portal to display options for the user to choose while deploying.</td>
</tr>
<tr>
<td>pack_dir</td>
<td>Optional</td>
<td>package all files in directory</td>
</tr>
</tbody>
</table>

**NFVIS Specific Enhancements**

- **Note**
  - Use `pack_dir` option if the *.tar.gz already exists and you want to modify the bootstrap configuration file or `image_properties.xml` manually.

The following parameters are added as part of the NFVIS specific enhancements:
--pack_dir <DIR> PACK

package all files in directory

Resources:

--vnic_names VNIC_NAMES

list of vnic number to name mapping in format
number:name example --vnic_names
1:GigabitEthernet2,2:GigabitEthernet4

Usage

Follow the steps to change a single line in day-0 configuration file or add a single option in image_properties.xml:

1. Get the working VM packaging image - isrv*.tar.gz.
2. Extract the contents - tar -xvf isrv*.tar.gz.
3. Modify the file contents as required.
4. nfvpt.py --pack_dir current-working-dir-with-files -i isrv.qcow2 -o isrv.tar.gz

VM Packaging Utility Usage Examples

Given below are the contents of the file nfvis_vm_packaging_utility_examples.txt:

Example 1: Usage for TinyLinux

nfvpt.py -o TinyLinux -i TinyLinux.qcow2 -n TinyLinux -t linux -r 1.0 --monitored false
--min_vcpu 1 --max_vcpu 2 --min_mem 1024 --max_mem 1024 --min_disk 1 --max_disk 2
--vnic_max 1 --optimize false

Example 2: Usage for ASAv

Note

The bootstrap filename has to be day0-config. This cannot be modified as ASAv looks for the exact filename.

nfvpt.py -o asav961-201 -i asav961-201.qcow2 -n ASAv -t firewall -r 961-201 --monitored true --bootstrap day0-config:filename1
--min_vcpu 1 --max_vcpu 4 --min_mem 1024 --max_mem 8192 --min_disk 8 --max_disk 16 --vnic_max 8 --optimize true
--profile ASAv5,"ASAv5 profile",1,1024,8192 --profile ASAv10,"ASAv10 profile",1,4096,8192
--profile ASAv30,"ASAv30 profile",4,8192,16384
--default_profile ASAv5

Example 3: Usage for ISRv
The bootstrap filename has to be **ovf-env.xml**. This cannot be modified as ISRv looks for the exact filename.

---

```
nfvpt.py -o isrv.16.03.01 -i isrv-universalk9.16.03.01.qcow2 -n ISRv.16.03.01 -t ROUTER -r 16.03.01 --monitored true --privileged true --bootstrap ovf-env.xml:file1,ios-xe.txt:file2 --min_vcpu 2 --max_vcpu 8 --min_mem 4096 --max_mem 8192 --min_disk 8 --max_disk 8 --vnic_max 8 --optimize true --profile ISRv-small,"ISRv small profile",2,4096,8192 --profile ISRv-medium,"ISRv medium profile",4,4096,8192 --default_profile ISRv-small --sriov_list igb,igbvf,i40evf --custom tech_package,ax
```

**Example 4:** Usage for a third party VM with config drive (ISO) mounted at specific path on the VM:

```
nfvpt.py -o test.1.0 -i test-1.0.qcow2 -n TEST -t OTHER -r 1.0 --monitored true --privileged true --bootstrap /:bootstrap.xml,/license/lic.txt:license.txt --min_vcpu 2 --max_vcpu 8 --min_mem 4096 --max_mem 8192 --min_disk 8 --max_disk 8 --vnic_max 8 --optimize true --profile small,"small profile",2,4096,8192 --profile medium,"medium profile",4,4096,8192 --default_profile small
```

In this case, `test.1.0.pkg : bootstrap.xml` gets mounted as `bootstrap.xml` at the root, and the `license.txt` gets mounted as `/license/lic.txt`.

**Example 5:** Usage for Palo Alto Firewall

```
nfvpt.py -o PA_L3_HA -i PA-VM-KVM-8.0.5.qcow2 --app_vendor PA --monitor true --ha_package
```

**Example 6:** Usage for asav

```
nfvpt.py -i foo.qcow2 -o asav.tar.gz --json d.json -t firewall -n "PA FIREWALL" -r 8.0.5 --app_vendor cisco --monitor true --ha_package
```

**Example 7:** Usage for csr

```
nfvpt.py --ha_package --pack_dir /data/intdatastore -i csr1000v-universalk9.16.09.01.qcow2 -o csr1000v-universalk9.16.09.01-ha.tar.gz
```

---

### Standard VM Image Packaging

The standard VM packaging is based on the Open Virtualization Format (OVF) packaging standard, in which a single file is distributed in open virtualization appliance (OVA) format. The VM image is shared using a TAR archive file with the root disk image and descriptor files.

---

**Note**

Cisco Enterprise NFVIS supports VM packaging in `.tar.gz` (compressed form of OVA) format. Ensure that all supported third party VM images are available in the supported format.
Generating a VM Package

Package files are provided for Cisco ISRv, Cisco ASAv, and tiny Linux and Windows server 2000. Vendors are responsible for packaging all third party VMs in the supported format.

1. Create a VM qcow2 image.

2. Create an `image_properties.xml` file with the VM properties. Ensure that you add all mandatory fields. Include the profiles supported for the VM in this file, and select one default profile. If you do not want to monitor the VM bootup, make the bootup time as -1.

3. Create `bootstrap-config` or `day0-config`, if any bootstrap configuration is required for the VM. If the bootstrap configuration requires inputs from the user, use the tokens in the xml or text file. These tokens are populated during the VM deployment with the provided data.

---

**Note**

A VM deployment may fail, if there are tokens in the configuration, and the user does not provide the token values in the deployment payload.

4. Create a `package.mf` file, which lists all the files to be bundled into the `.tar.gz` file along with checksums.

5. Generate the packaging file using "tar -cvzf ova_file_name list_of_files_to_be_bundled".
   
   For example, `tar -cvzf isrv.tar.gz isrv-universalk9.03.16.02.S.155-3.S1a-ext-serial.qcow2 image_properties.xml isr_ovf_env.xml package.mf`.

Appendix

VM Image Package Files

The table lists the contents of the VM package that are generated using the packaging tool:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Manifest (package.mf)</td>
<td>Lists the files in the package and the expected checksum for the files.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>VM image properties (vmname_properties.xml)</td>
<td>XML file with resources and features supported by the VM</td>
<td>Mandatory</td>
</tr>
<tr>
<td>VM image (vmname.qcow2)</td>
<td>Image file of the VM. Multiple images are supported. One root_disk image file is mandatory.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>
Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv.

**Package Manifest File**

The package manifest XML file provides a list of the files in the package with their names and their expected checksum. SHA1 algorithm (sha1sum) is used to calculate the checksum. This is a mandatory file to be bundled in the VM package. The manifest file must be named as *package.mf*.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Property Tag</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>File information</td>
<td>XML tree with details of file name, file type, and expected checksum. The root_image and image_properties files are required.</td>
<td>&lt;file_info&gt;</td>
<td>Mandatory</td>
</tr>
<tr>
<td>File name</td>
<td>Name of the file</td>
<td>&lt;name&gt;</td>
<td>Mandatory</td>
</tr>
<tr>
<td>File type</td>
<td>Describes the file type. Supported types:</td>
<td>&lt;type&gt;</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>• root_image</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• image_properties</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• bootstrap_config_file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ephemeral_disk1_image</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ephemeral_disk2_image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expected checksum</td>
<td>The calculated SHA1 checksum to be validated.</td>
<td>&lt;sha1_checksum&gt;</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

**Bootstrap Configuration File**

The bootstrap configuration file is an XML or a text file, and contains properties specific to a VM and the environment. Properties can have tokens, which can be populated during deployment time from the deployment payload.
VM Image Properties File

This XML file provides information about the resources supported or required for the VM operation. All mandatory parameters have to be defined. It also supports custom attributes. This is a mandatory file to be bundled in the VM package. The VM package supports up to 10 disks to be bundled into the package.

Table 4: VM Image Properties File Details

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Property Tag</th>
<th>Possible Values</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNF Type</td>
<td>VM functionality provided. Router and firewall are predefined types.</td>
<td>&lt;vnf_type&gt;</td>
<td>Router, firewall, Windows, Linux, and custom_type</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Name</td>
<td>Name associated with the VM packaging. This name is referenced for VM deployment.</td>
<td>&lt;name&gt;</td>
<td>Any</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Version</td>
<td>Version of the package</td>
<td>&lt;version&gt;</td>
<td>Any</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Boot-up time</td>
<td>Boot-up time (in seconds) of the VNF before it can be reachable via ping.</td>
<td>&lt;bootup_time&gt;</td>
<td>Any in seconds, (-1) to not monitor boot-up</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Root Disk Image Bus</td>
<td>Root image disk bus</td>
<td>&lt;root_file_disk_bus&gt;</td>
<td>virtio, scsi, and ide</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Disk-1 bus type</td>
<td>Additional disk 1 image disk bus</td>
<td>&lt;disk_1_file_disk_bus&gt;</td>
<td>virtio, scsi, and ide</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-2 bus type</td>
<td>Disk2 image disk bus</td>
<td>&lt;disk_2_file_disk_bus&gt;</td>
<td>virtio, scsi, and ide</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-10 bus type</td>
<td>Disk10 image disk bus</td>
<td>&lt;disk_10_file_disk_bus&gt;</td>
<td>virtio, scsi, and ide</td>
<td>Optional</td>
</tr>
<tr>
<td>Root Disk Image format</td>
<td>Root image disk format</td>
<td>&lt;root_image_disk_format&gt;</td>
<td>qcow2 and raw</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Disk-1 Image format</td>
<td>Additional disk 1 image format</td>
<td>&lt;disk_1_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-2 Image format</td>
<td>Disk 2 image format</td>
<td>&lt;disk_2_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-10 Image format</td>
<td>Disk 10 image format</td>
<td>&lt;disk_10_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Property</td>
<td>Required</td>
<td>Optional</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Serial Console</td>
<td>Serial console supported</td>
<td><code>&lt;console_type_serial&gt;</code></td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>Minimum vCPU</td>
<td>Minimum vCPUs required for a VM operation</td>
<td><code>&lt;vcpu_min&gt;</code></td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Maximum vCPU</td>
<td>Maximum vCPUs supported by a VM</td>
<td><code>&lt;vcpu_max&gt;</code></td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Minimum memory</td>
<td>Minimum memory in MB required for VM operation</td>
<td><code>&lt;memory_mb_min&gt;</code></td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Maximum memory</td>
<td>Maximum memory in MB supported by a VM</td>
<td><code>&lt;memory_mb_max&gt;</code></td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Minimum root disk size</td>
<td>Minimum disk size in GB required for VM operation</td>
<td><code>&lt;root_disk_gb_min&gt;</code></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Maximum root disk size</td>
<td>Maximum disk size in GB supported by a VM</td>
<td><code>&lt;root_disk_gb_max&gt;</code></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Maximum vNICs</td>
<td>Maximum number of vNICs supported by a VM</td>
<td><code>&lt;vnic_max&gt;</code></td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>SRIOV support</td>
<td>SRIOV supported by VM interfaces. This should have a list of supported NIC device drivers.</td>
<td><code>&lt;sriov_supported&gt;</code></td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>SRIOV driver list</td>
<td>List of drivers to enable SRIOV support</td>
<td><code>&lt;sriov_driver_list&gt;</code></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>PCI passthrough support</td>
<td>PCI passthrough support by VM interfaces</td>
<td><code>&lt;pcie_supported&gt;</code></td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>PCIE driver list</td>
<td>List of VNICS to enable PCI passthrough support</td>
<td><code>&lt;pcie_driver_list&gt;</code></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Optional disk, cdrom</td>
<td>&lt;bootstrap_cloud_init_drive_type&gt;</td>
<td>disk, cdrom</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------</td>
<td>------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>Optional virtio, ide</td>
<td>&lt;bootstrap_cloud_init_bus_type&gt;</td>
<td>virtio, ide</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>BOOTSTRAP</td>
<td>&lt;bootstrap_file&gt;</td>
<td>File name of the bootstrap file</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Bootstrapping files</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom properties</td>
<td>&lt;custom_property&gt;</td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

- **Optional disk, cdrom**: Mounts day0 config file as disk (default is CD-ROM)
- **Optional virtio, ide**: Default is IDE
- **BOOTSTRAP**: Bootstrap files for the VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv
- **Custom properties**: List of properties can be defined within the custom_property tree. (Example: For ISRv, the technology packages are listed in this block.) If the Cisco Enterprise NFV portal is used to deploy the VM, the portal prompts you for inputs for custom properties fields, and can pass the values to the bootstrap configuration.
<table>
<thead>
<tr>
<th>Profiles for VM deployment</th>
<th>List of VM deployment profiles. Minimum one profile is required</th>
<th><code>&lt;profiles&gt;</code></th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default profile</td>
<td>The default profile is used when no profile is specified during deployment.</td>
<td><code>&lt;default_profile&gt;</code></td>
<td>Optional</td>
</tr>
<tr>
<td>Monitoring Support</td>
<td>A VM supports monitoring to detect failures.</td>
<td><code>&lt;monitoring_supported&gt;</code></td>
<td>true, false</td>
</tr>
<tr>
<td>Monitoring Method</td>
<td>A method to monitor a VM. Currently, only ICMP ping is supported.</td>
<td><code>&lt;monitoring_methods&gt;</code></td>
<td>ICMP Ping</td>
</tr>
<tr>
<td>Low latency</td>
<td>If a VM's low latency (for example, router and firewall) gets dedicated resource (CPU) allocation. Otherwise, shared resources are used.</td>
<td><code>&lt;low_latency&gt;</code></td>
<td>true, false</td>
</tr>
<tr>
<td>Privileged-VM</td>
<td>Allows special features like promiscuous mode and snooping. By default, it is false.</td>
<td><code>&lt;privileged_vm&gt;</code></td>
<td>true, false</td>
</tr>
<tr>
<td>Virtual interface model</td>
<td></td>
<td><code>&lt;virtual_interface_model&gt;</code></td>
<td>Optional</td>
</tr>
<tr>
<td>Thick disk provisioning</td>
<td>By default, it is false.</td>
<td><code>&lt;thick_disk_provisioning&gt;</code></td>
<td>true, false</td>
</tr>
<tr>
<td>Profile for VM deployment</td>
<td>A profile defines the resources required for VM deployment. This profile is referenced during VM deployment.</td>
<td><code>&lt;profile&gt;</code></td>
<td>Optional</td>
</tr>
<tr>
<td>Name</td>
<td>Profile name</td>
<td><code>&lt;name&gt;</code></td>
<td>Any</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the profile</td>
<td>&lt;description&gt;</td>
<td>Any</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------</td>
<td>--------------</td>
<td>-----</td>
</tr>
<tr>
<td>vCPU</td>
<td>vCPU number in a profile</td>
<td>&lt;vcpu&gt;</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Memory - MB in profile</td>
<td>&lt;memory_mb&gt;</td>
<td></td>
</tr>
<tr>
<td>Root Disk Size</td>
<td>Disk size - MB in profile</td>
<td>&lt;root_disk_mb&gt;</td>
<td></td>
</tr>
<tr>
<td>VNIC Offload</td>
<td>List of properties that can be set for vnic offload</td>
<td>&lt;vnic_offload&gt;</td>
<td></td>
</tr>
<tr>
<td>Generic Segmentation Offload</td>
<td>Turn generic segmentation offload on or off</td>
<td>&lt;generic_segmentation_offload&gt; (parent: &lt;vnic_offload&gt;)</td>
<td>on, off</td>
</tr>
<tr>
<td>Generic Receive Offload</td>
<td>Turn generic receive offload on or off</td>
<td>&lt;generic_receive_offload&gt; (parent: &lt;vnic_offload&gt;)</td>
<td>on, off</td>
</tr>
<tr>
<td>RX Checksumming</td>
<td>Turn RX checksumming on or off</td>
<td>&lt;rx_checksumming&gt; (parent: &lt;vnic_offload&gt;)</td>
<td>on, off</td>
</tr>
<tr>
<td>TX Checksumming</td>
<td>Turn TX checksumming on or off</td>
<td>&lt;tx_checksumming&gt; (parent: &lt;vnic_offload&gt;)</td>
<td>on, off</td>
</tr>
<tr>
<td>TCP Segmentation Offload</td>
<td>Turn TCP segmentation offload on or off</td>
<td>&lt;tcp_segmentation_offload&gt; (parent: &lt;vnic_offload&gt;)</td>
<td>on, off</td>
</tr>
</tbody>
</table>

A virtual console is supported by default. Specify the root disk size as zero for multiple disks (for example, vWaas deployment) as the system does not support populating multiple disk sizes. Actual disk sizes are calculated from the root_disk files.

**Example: Package.mf**

```xml
<PackageContents>
  <File_Info>
    <name>ISRv_serial_3.16.02.qcow2</name>
    <type>root_image</type>
    <sha1_checksum>93de73ee3531f74fddf99377972357a8a0eac7b</sha1_checksum>
  </File_Info>
  <File_Info>
    <name>image_properties.xml</name>
  </File_Info>
```

Note: Example: Package.mf

** **sha1sum - for calculating checksum**

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.11.1
Example: Image Properties

<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
  <vnf_type>ROUTER</vnf_type>
  <name>isrv-universalk9</name>
  <version>03.16.02</version>
  <bootup_time>600</bootup_time>
  <root_file_disk_bus>virtio</root_file_disk_bus>
  <root_image_disk_format>qcow2</root_image_disk_format>
  <vcpu_min>1</vcpu_min>
  <vcpu_max>8</vcpu_max>
  <memory_mb_min>4096</memory_mb_min>
  <memory_mb_max>8192</memory_mb_max>
  <vnic_max>8</vnic_max>
  <root_disk_gb_min>8</root_disk_gb_min>
  <root_disk_gb_max>8</root_disk_gb_max>
  <console_type_serial>true</console_type_serial>
  <sriov_supported>true</sriov_supported>
  <sriov_driver_list>igb</sriov_driver_list>
  <sriov_driver_list>igbvf</sriov_driver_list>
  <sriov_driver_list>i40evf</sriov_driver_list>
  <pcie_supported>true</pcie_supported>
  <pcie_driver_list>igb</pcie_driver_list>
  <pcie_driver_list>igbvf</pcie_driver_list>
  <pcie_driver_list>i40evf</pcie_driver_list>
  <bootstrap_file_1>/ovf-env.xml</bootstrap_file_1>
  <monitoring_supported>true</monitoring_supported>
  <monitoring_methods>ICMP Ping</monitoring_methods>
  <low_latency>true</low_latency>
  <privileged_vm>true</privileged_vm>
  <cdrom>true</cdrom>
  <custom_property>
    <tech_package>ax</tech_package>
    <tech_package>sec</tech_package>
    <tech_package>ipbase</tech_package>
    <tech_package>appx</tech_package>
  </custom_property>
  <profiles>
    <profile>
      <name>ISRv1kv-small</name>
      <description>ISRv upto 50MBPS performance</description>
      <vcpus>1</vcpus>
      <memory_mb>4096</memory_mb>
      <root_disk_mb>8</root_disk_mb>
    </profile>
  </profiles>
</image_properties>
Example: Bootstrap Configuration File

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.ISRv.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.isrv.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.isrv.login-password.1" oe:value="admin"/>
    <Property oe:key="com.cisco.isrv.login-username.1" oe:value="lab"/>
    <Property oe:key="com.cisco.isrv.mgmt-interface.1" oe:value="GigabitEthernet1"/>
    <Property oe:key="com.cisco.isrv.mgmt-ipv4-addr.1" oe:value="${NICID_0_IP_ADDRESS}/24"/>
    <Property oe:key="com.cisco.isrv.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.isrv.license.1" oe:value="${TECH_PACKAGE}"/>
    <Property oe:key="com.cisco.isrv.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>
    <Property oe:key="com.cisco.isrv.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.isrv.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.isrv.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.isrv.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.isrv.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.isrv.ios-config-0007" oe:value="interface GigabitEthernet1"/>
    <Property oe:key="com.cisco.isrv.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>
    <Property oe:key="com.cisco.isrv.ios-config-0009" oe:value="ip address ${NICID_0_IP_ADDRESS} ${NICID_0_NETMASK}"/>
    <Property oe:key="com.cisco.isrv.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.isrv.ios-config-0011" oe:value="exit"/>
    <Property oe:key="com.cisco.isrv.ios-config-0012" oe:value="ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 ${NICID_0_GATEWAY}"/>
  </PropertySection>
</Environment>
```

Image Properties Template File

The parameters that go into the image properties file are listed in the code extract below.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
  <vnf_type>ROUTER</vnf_type>
  <name>TEMPLATE</name>
  <version>1.0</version>
  <bootup_time>600</bootup_time>
</image_properties>
```
<root_file_disk_bus>virtio</root_file_disk_bus>
<root_image_disk_format>qcow2</root_image_disk_format>
<vcpu_min>1</vcpu_min>
<vcpu_max>8</vcpu_max>
<memory_mb_min>4096</memory_mb_min>
<memory_mb_max>8192</memory_mb_max>
<vnic_max>8</vnic_max>
<root_disk_gb_min>8</root_disk_gb_min>
<root_disk_gb_max>16</root_disk_gb_max>
<console_type_serial>false</console_type_serial>
<sriov_supported>true</sriov_supported>
<sriov_driver_list>s1</sriov_driver_list>
<sriov_driver_list>s2</sriov_driver_list>
<sriov_driver_list>s3</sriov_driver_list>
<pciie_supported>false</pciie_supported>
<monitoring_supported>true</monitoring_supported>
<monitoring_methods>ICMPPing</monitoring_methods>
<low_latency>true</low_latency>
<privileged_vm>false</privileged_vm>
<cdrom>true</cdrom>
<bootstrap_file_1>b1.xml</bootstrap_file_1>
<bootstrap_file_2>b2.txt</bootstrap_file_2>
<custom_property>
  <key>val</key>
  </custom_property>
<profiles>
  <profile>
    <name>small</name>
    <description>small</description>
    <vcpus>1</vcpus>
    <memory_mb>1024</memory_mb>
    <root_disk_mb>4096</root_disk_mb>
  </profile>
  <profile>
    <name>medium</name>
    <description>medium</description>
    <vcpus>2</vcpus>
    <memory_mb>4096</memory_mb>
    <root_disk_mb>8192</root_disk_mb>
  </profile>
</profiles>
<default_profile>small</default_profile>
Upgrading Cisco Enterprise NFVIS

The Cisco Enterprise NFVIS upgrade image is available as a .nfvispkg file. Currently, downgrade is not supported. All RPM packages in the Cisco Enterprise NFVIS upgrade image are signed to ensure cryptographic integrity and authenticity. In addition, all RPM packages are verified during Cisco Enterprise NFVIS upgrade. For more information about the Image Signing and Verification feature, see Image Signing and Verification, on page 6.

Ensure that you copy the image to the Cisco Enterprise NFVIS server before starting the upgrade process. Always specify the exact path of the image when registering the image. Use the scp command to copy the upgrade image from a remote server to your Cisco Enterprise NFVIS server. When using the scp command, you must copy the image to the "/data/intdatastore/uploads" folder on the Cisco Enterprise NFVIS server. The following is an example on how to use the scp command to copy the upgrade image:

```
scp -P 22222 nfvis-351.nfvispkg admin@192.0.2.9:/data/intdatastore/uploads/nfvis-351.nfvispkg
```

Alternatively, you can upload the image to the Cisco Enterprise NFVIS server using the System Upgrade option from the Cisco Enterprise NFVIS portal.

The upgrade process comprises two tasks:

- Registering the image using the `system upgrade image-name` command.
- Upgrading the image using the `system upgrade apply-image` command.

### Registering an Image

To register an image:

```
configure terminal
system upgrade image-name nfvis-351.nfvispkg location /data/intdatastore/uploads
commit
```

You must verify the image registration status before upgrading the image using the `system upgrade apply-image` command. The package status must be valid for the registered image.

### Verifying the Image Registration

Use the `show system upgrade reg-info` command in the privileged EXEC mode to verify the image registration.
Upgrading the Registered Image

To upgrade the registered image:

```
configure terminal
system upgrade apply-image nfvis-351.nfvispkg scheduled-time 5
commit
```

Verifying the Upgrade Status

Use the `show system upgrade apply-image` command in the privileged EXEC mode

```
nfvis# show system upgrade apply-image
UPGRADE
NAME STATUS FROM UPGRADE TO
---------------------------------------------------------------------------------------------
nfvis-351.nfvispkg SUCCESS 3.5.0 3.5.1
```

The only upgrade supported when BIOS secured boot (UEFI mode) is enabled on ENCS 5400 platform is:

NFVIS 3.8.1 + BIOS 2.5(legacy) --> NFVIS 3.9.1 + BIOS 2.6(legacy)

The following upgrade requires re-installation:

NFVIS 3.8.1 + BIOS 2.5(legacy) --> NFVIS 3.9.1 + BIOS 2.6(UEFI)
NFVIS 3.9.1 + BIOS 2.6(legacy) --> NFVIS 3.9.1 + BIOS 2.6(UEFI)

Upgrade APIs and Commands

<table>
<thead>
<tr>
<th>Upgrade APIs</th>
<th>Upgrade Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/system/upgrade</td>
<td>• system upgrade image-name</td>
</tr>
<tr>
<td>• /api/config/system/upgrade/image-name</td>
<td>• system upgrade apply-image</td>
</tr>
<tr>
<td>• /api/config/system/upgrade/reg-info</td>
<td>• show system upgrade reg-info</td>
</tr>
<tr>
<td>• /api/config/system/upgrade/apply-image</td>
<td>• show system upgrade apply-image</td>
</tr>
</tbody>
</table>

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.11.1
CHAPTER 11

Configuring vBranch High Availability

The vbranch high availability (HA) solution is a box-to-box HA. It is similar to the traditional branch, which uses physical boxes for routing and other services. This solution uses the Hot Standby Router Protocol (HSRP), a default gateway redundancy (or a first hop redundancy), which allows the network to recover from the failure of the device acting as the default gateway for the LAN side end points (devices). The routing protocols are configured to converge the traffic on the WAN side, when there are failures. So, this solution uses HSRP to provide redundancy for the branch connectivity on the LAN side. The Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF) routing protocols, and Embedded Event manager (EEM) scripts are configured to converge on the WAN side. The following section explains the redundancy solutions for the branch, with each ENCS having separate active WAN link.

Note
You can use this recommended HA design as is, or modify as per the field requirement.

- Prerequisites for vBranch HA, on page 95
- vBranch HA Design and Topology, on page 96
- Enable Virtual NIC Failure Detection with Track Feature, on page 97
- Isolating LAN and Transit Link Traffic for vBranch HA, on page 99
- Packet Flow for vBranch HA, on page 100
- Configuration Examples for vBranch HA, on page 101
- Cisco ENCS Failure Points, on page 104

Prerequisites for vBranch HA

- Cisco ISRv must run HSRP on the LAN facing interface.

- The WAN links are active on both Cisco ENCS1 and Cisco ENCS2. Each of the ENCS WAN link is connected to the WAN network (most cases with two SPs), with two ENCSs in an active-active mode.

- The LAN facing links of both Cisco ENCS devices are connected to an external switch (as an uplink), and all the devices on the LAN segment are also connected to the external switch. There should be no LAN device connecting directly to the Cisco ENCS internal switch.

- A transit link, which is L3 routed, is configured between the Cisco ENCS devices. Since the LAN HSRP makes only one device active, the transit link is used to forward traffic. This link is used to forward traffic from the standby ENCS WAN to LAN or LAN to WAN. This link can be back-to-back connected on the ENCS internal switch ports.
• VMs and VNFs on both ENCS devices must be configured identical.

vBranch HA Design and Topology

Figure 9: vBranch HA Design

Physical Devices Connection

Each Cisco ENCS has a WAN traffic connected to the Gigabit Ethernet interface, GE0-0, in this dual-WAN topology.

There are two Cisco ENCS devices namely ENCS1 and ENCS2. There is an external switch connecting one of the LAN ports from each Cisco ENCS. There is a back-to-back connection between ENCS1 and ENCS2 connecting one of the LAN ports from each Cisco ENCS. The WAN port from each Cisco ENCS is connected to the service provider's network.

ISRv1 on ENCS1 and ISRv2 on ENCS2 are responsible for handling packets from LAN to WAN and WAN to LAN. If the WAN connection goes down or if the ISRv1 becomes unavailable, fast converging routing protocols, such as EIGRP and OSPF, can respond within seconds so that ISRv2 is prepared to transfer packets.

VM and Service Chain Network Connection

The Cisco ISRv should be created with an additional vNIC mapped to the transit link between two Cisco ENCS devices, apart from the regular WAN and LAN or service net links. The Cisco ISRv on both ENCS should have identical resource configurations (vNICs, vCPU, memory, etc.) and feature configurations.

Each Cisco ENCS is running an instance of service VNFs (for example, Cisco ASAv and Cisco vWAAS), and should have the identical service chain VNFs configured on both Cisco ENCS devices. Service VNFs
should also have same features configured on both Cisco ENCS devices. The traffic goes through the service VNFs on the active Cisco ENCS only, even though both Cisco ENCS devices are actively forwarding on the WAN link. On a failover, the traffic will go over the service VNFs on the newly active ENCS (ENCS2).

This HA solution requires a transit link configured between two Cisco ENCS devices. One of the LAN ports from each of the Cisco ENCS can be connected back to back. This transit link port should be extended to the Cisco ISRv.

Enable Virtual NIC Failure Detection with Track Feature

You can enable the Track feature to detect virtual NIC failure in the following two scenarios:

- When the underlying physical link fails, the HSRP or routing protocols cannot detect the failure—This is because the line protocol does not go down when the underlying physical link fails if the Cisco ISRv is using a virtual NIC.

- With EEM scripts unconfigured, when the underlying physical link fails, the virtual NIC line protocol does not go down. In this case the routing protocol does not withdraw the routes.

Configuration Example for the Track Feature with Scenario 1 (HSRP)

In the virtual environment, you can enable higher protocols like HSRP to take action when the link failure happens. One way to achieve this is by configuring the Track feature on some object (ICMP ping) in Cisco IOS XE.

In Cisco ISRv, if the LAN interface where HSRP is running is a virtual NIC, then you can configure the track object to ping some device on the LAN segment, and monitor the connection failures. So, when the track object is down due to some connection failure, you can configure an action as to shut down the HSRP group, so that the peer will take over the active role making the default Gateway IP active. Without this track object, both Cisco ENCS devices will become active getting into a split-brain scenario.

The following example shows how to configure the track object on the active ISRv1, and monitor the connection failures by pinging the device IP in the network.

```
track 1 ip sla 1 reachability
ip sla 1
  icmp-echo 192.0.2.1 source-ip 198.51.100.1
  frequency 5
ip sla schedule 1 life forever start-time now 
!
track 5 ip sla 5 reachability
ip sla 5
  icmp-echo 192.0.2.2 source-ip 198.51.100.2
  frequency 5
ip sla schedule 5 life forever start-time now 
!
```

The following output shows that the Track 1 reachability is failed, and Track 5 is up.

```
device1# show track
```
Track 1
  IP SLA 1 reachability
  Reachability is Down
  11 changes, last change 00:01:22
  Latest operation return code: Timeout
  Tracked by:
  HSRP GigabitEthernet3 25
Track 5
  IP SLA 5 reachability
  Reachability is Up
  4 changes, last change 00:02:32
  Latest operation return code: OK
  Latest RTT (milliseconds) 1
  Tracked by:
  HSRP GigabitEthernet3 25
ISRv1#

The following example shows how to configure the Track object to monitor the line protocol state of the interface:

```
track 2 interface GigabitEthernet2 line-protocol
```

The following output shows that the line protocol state is down:

```
device# show track
Track 2
  Interface GigabitEthernet2 line-protocol
  Line protocol is Down ((hw down))
  8 changes, last change 00:01:25
  Tracked by:
  HSRP GigabitEthernet3 25
```

Configuration Example for the Track Feature with Scenario 2 (EEM Scripts)

With EEM scripts unconfigured, when an underlying link fails, the virtual NIC line protocol does not go down. This causes the problem as the routing protocol will not withdraw the routes. You can configure a Track object (can use the same object defined for HSRP above) to detect the failure. When the failure happens, the active Cisco ISRv has to withdraw the routes or network, so that the WAN link does not receive any traffic. One way to withdraw the routes is configure the EEM script, and delete the network from EIGRP.

The following example shows how to configure the EEM scripts, and remove the network from EIGRP:

```
track 5 ip sla 5 reachability
!
ip sla 5
  icmp-echo 192.0.2.1 source-ip 192.0.2.18
  frequency 5
ip sla schedule 5 life forever start-time now
!
  event manager applet noshut_int
  event track 5 state up
  action 1.1 cli command "enable"
  action 1.2 cli command "config t"
  action 1.3 cli command "router eigrp 10"
  action 1.4 cli command "network 192.0.2.1 0.0.0.255"
  action 1.5 cli command "end"
  event manager applet shut_int
  event track 5 state down
  action 1.1 cli command "enable"
  action 1.2 cli command "config t"
  action 1.3 cli command "router eigrp 10"
```
In the virtual environment HSRP, make sure to use standby use-bia.

The following configuration example shows how to use the Track object (Track 5) to shut down HSRP group in ISRv1, when reachability is down for Track 5:

interface GigabitEthernet4
  description Service-NET-Virtio
  ip address 192.0.2.1 255.255.255.0
  standby use-bia
  standby 25 ip 192.0.2.22
  standby 25 timers 1 5
  standby 25 priority 105
  standby 25 preempt
  standby 25 track 5 shutdown

Isolating LAN and Transit Link Traffic for vBranch HA

LAN traffic and transit link traffic shall be isolated by configuring different VLANs for each traffic since both links are connected to the same ENCS internal switch. If you do not isolate these traffic, both LAN traffic and transit link will flow through the same internal switch on the Cisco ENCS.

The following Cisco ENCS switch configuration example shows how to isolate traffic. In this example, the Cisco ISRv is configured to send HSRP traffic as an untag and transit traffic in VLAN 46. So, to isolate HSRP traffic and transit traffic on the internal switch, the Gigabit Ethernet interface 1/0 is connected to a LAN network and Gigabit Ethernet interface 1/1 is configured as the transit link. The Gigabit Ethernet interface 1/1 allows the VLAN 46 to pass the transit traffic. It should also have non-default (other than 1) native VLAN (for example, VLAN 2), because the Cisco ENCS internal switch uplink (internal) has the native VLAN 1 configured.

Enable MSTP on all switches before isolating traffic.

switch
  interface gigabitEthernet1/0
    negotiation auto
    no shutdown
    switchport access vlan 1
    switchport mode access
    switchport trunk native vlan 1
    switchport trunk allowed vlan 1-2349,2450-4093
  !
  !
switch
  interface gigabitEthernet1/1
    negotiation auto
    no shutdown
    spanning-tree mst 1 cost 200000000
    spanning-tree mst 2 cost 200000000
    switchport access vlan 46
    switchport mode trunk
    switchport trunk native vlan 2
    switchport trunk allowed vlan 1-2349,2450-4093
  !
Packet Flow for vBranch HA

This section explains high-level packet flow in failure and non-failure cases.

Non-Failure Case

In the non-failure case, both active and standby Cisco ENCS devices are up and running.

• LAN to WAN through the standby ENCS1 WAN link
  
  • The device in the LAN segment is configured with the default gateway as the HSRP virtual IP address, and since ENCS1 is an active HSRP, LAN traffic first comes to the active ENCS1.
  
  • LAN traffic goes through the service chain VM (Cisco ASAv), and then hits the Cisco ISRv. In this case, the destination IP is routable through the ENCS1 WAN interface. The Cisco ISRv sends traffic over the WAN link.

Use the `show switch vlan detailed` command to verify the configuration as shown below:

device# show switch vlan detailed

platform-detail hardware_info Manufacturer "Cisco Systems, Inc."
platform-detail hardware_info PID ENCS5412/K9
platform-detail hardware_info SN FGL212681GK
platform-detail hardware_info hardware-version M3
platform-detail hardware_info UUID 7BBEBDE0-CE3C-42E5-B564-CFEE8F18AE97
platform-detail hardware_info Version 3.8.1-FC3
platform-detail hardware_info Compile_Time "Sunday, April 15, 2018 [20:38:10 PDT]"
platform-detail hardware_info CPU_Information "Intel(R) Xeon(R) CPU D-1557 @ 1.50GHz 12 cores"
platform-detail hardware_info Memory_Information "16227148 kB"
platform-detail hardware_info Disk_Size "64.0 GB"
platform-detail hardware_info CDMC_IP NA
platform-detail hardware_info Device-Name ENCS
platform-detail hardware_info Device_desc "Enterprise Network Compute System"
platform-detail software_packages Kernel_Version 3.10.0-514.21.1.1.el7.x86_64
platform-detail software_packages QEMU_Version 1.5.3
platform-detail software_packages LibVirt_Version 3.2.0
platform-detail software_packages OVS_Version 2.5.2
platform-detail switch_detail UUID NA
platform-detail switch_detail Type NA
platform-detail switch_detail Name NA
platform-detail switch_detail Ports 8

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>MEDIA</th>
<th>LINK</th>
<th>SPEED</th>
<th>MTU</th>
<th>MAC</th>
<th>DETAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO-0</td>
<td>physical</td>
<td>Twisted Pair</td>
<td>up</td>
<td>1000</td>
<td>9216</td>
<td>70:db:98:c3:f3:64</td>
<td>02:00.0</td>
</tr>
<tr>
<td>GEO-1</td>
<td>physical</td>
<td>Twisted Pair</td>
<td>up</td>
<td>1000</td>
<td>9216</td>
<td>70:db:98:c3:f3:65</td>
<td>02:00.1</td>
</tr>
<tr>
<td>MGMT</td>
<td>physical</td>
<td>Twisted Pair</td>
<td>up</td>
<td>1000</td>
<td>1500</td>
<td>70:db:98:c3:f3:d8</td>
<td>0e:00.0</td>
</tr>
</tbody>
</table>
• LAN to WAN through the standby ENCS2 WAN link—In this case, the LAN to WAN traffic uses the transit link between the active and standby devices.
  - Devices in the LAN segment are configured with the default gateway as the HSRP virtual IP address, and since ENCS1 is an active HSRP, the LAN traffic first comes to the active ENCS1.
  - The LAN traffic goes through the service chain VMs (Cisco ASAv), and then hits the active Cisco ISRv. In this case, the destination IP is routable through the ENCS2 WAN interface. The traffic is sent to the Cisco ISRv on ENCS2 over the transit link, and then sent out over the WAN link to the destination.

• WAN to LAN through the active ENCS1
  - The WAN traffic hits the Cisco ISRv on ENCS1, then it goes through the service chain VMs, and sent to the LAN device.

• WAN to LAN through the standby ENCS2 WAN link—In this case, the WAN to LAN traffic uses the transit link between the active and standby devices.
  - The WAN traffic comes to the Cisco ISRv on ENCS2. The PBR/PFR configuration forces the traffic to use the transit link instead of the directly connected LAN port. So, the traffic is sent to the Cisco ISRv on ENCS1 over the transit link.
  - Then, the traffic on ENCS1 goes through the service chain VMs, and sent to the LAN device.

**Failure Case**

In the failure case, the active device goes down, and the standby device becomes active.

The virtual IP (default gateway) address becomes active on ENCS2. The transit link will not be used. The traffic now goes through the service chain VMs on ENCS2, and gets forwarded directly between WAN and LAN interfaces. The PBR/PFR configuration should monitor the HSRP state, and use the LAN port instead of the transit link to forward LAN traffic.

**Configuration Examples for vBranch HA**

This sample configuration is for Cisco ENCS HA with a dual-WAN scenario. The Cisco ISRv is configured with vNICs connected to the wan-net, service-net, and transit link. HSRP is configured on the service-net interface. Each Cisco ENCS is provisioned with the Cisco ASAv (service-net) and Cisco vWAAS (service-net).

**Note**

You can use this design as is, or modify as per the field requirement.

**Example: Active Cisco ENCS Configuration with ISRv1**

```bash
interface GigabitEthernet1
vrf forwarding Mgmt-intf
ip address 192.0.2.1 255.255.255.0
negotiation auto
!
```
interface GigabitEthernet2
  description WAN-GE0-0-SRIOV-1
  ip address 192.0.2.2 255.255.255.0
  negotiation auto
!
interface GigabitEthernet3
  description LAN-NET
  no ip address
  shutdown
  negotiation auto
!
interface GigabitEthernet4
  description Service-NET-Virtio
  ip address 192.0.2.3 255.255.255.0
  standby use-bia
  standby 25 ip 192.0.2.20
  standby 25 timers 1 5
  standby 25 priority 105
  standby 25 preempt
  standby 25 track 1 decrement 10
  standby 25 track 2 decrement 10
  standby 25 track 3 decrement 10
  standby 25 track 5 shutdown
  standby 25 track 6 shutdown
  standby 25 track 7 shutdown
  negotiation auto
  bfd interval 9000 min_rx 9000 multiplier 3
!
interface GigabitEthernet5
  ip address 192.0.2.4 255.255.255.0
!
router eigrp stub 10
  network 25.25.25.0 0.0.0.255
  network 38.38.38.0 0.0.0.255
  network 46.46.46.0 0.0.0.255
!
track 1 ip sla 1 reachability
!
track 2 interface GigabitEthernet2 line-protocol
!
track 3 interface GigabitEthernet4 line-protocol
!
track 5 ip sla 5 reachability
!
track 6 ip sla 6 reachability
!
track 7 ip sla 7 reachability
!
ip sla 1
  icmp-echo 9.9.9.29 source-ip 192.0.2.2
  frequency 5
  ip sla schedule 1 life forever start-time now
!
ip sla 5
  icmp-echo 25.25.25.11 source-ip 192.0.2.3
  frequency 5
  ip sla schedule 5 life forever start-time now
!
ip sla 6
  icmp-echo 25.25.25.51 source-ip 192.0.2.3
  frequency 5
  ip sla schedule 6 life forever start-time now
Example: Standby Cisco ENCS Configuration with ISRv2

```plaintext
interface GigabitEthernet1
vrf forwarding Mgmt-intf
ip address 192.0.2.1 255.255.255.0
negotiation auto
! interface GigabitEthernet2
description WAN-GE0-0-SRIOV-1
ip address 192.0.2.21 255.255.255.0
```
Cisco ENCS Failure Points

<table>
<thead>
<tr>
<th>Failure Points</th>
<th>Sequence of Events</th>
</tr>
</thead>
</table>
| ENCS chassis hardware failure: | 1. HSRP on ENCS2 detects the reachability failure to ENCS1, and triggers the failover. LAN virtual IP becomes active on ENCS2.  
  - Power down  
  - Power cycle  
  - Reboot |
| Cisco Enterprise NFVIS software failure | 2. WAN-IP1 on ENCS1 becomes unreachable, and all the routes converge towards WAN-IP2 on ENCS2. WAN-IP2 is the only IP for branch connectivity.  
  - Crash |
| Cisco ISRv software failure | 3. All the WAN to LAN, and LAN to WAN traffic will now flow through ENCS2.  
  - Stop (shutdown)  
  - Reboot  
  - Crash  
  - Error |

ISRv1 (Active) Before the Failure

ISRv1# show platform software vnic-if interface-mapping
Interface Name | Driver Name | Mac Addr
----------------|-------------|----------------------
GigabitEthernet5 | i40evf | 5254.003a.1020 (LAN-SRIOV-2)
GigabitEthernet4 | virtio | 5254.0053.e392 (service-net)
GigabitEthernet3 | i40evf | 5254.00c4.b925 (LAN-SRIOV-1)
GigabitEthernet2 | igbvf | 5254.00d2.cc9a (GE0-0-SRIOV-1)
GigabitEthernet1 | virtio | 5254.00d2.1b1c (int-mgmt-net)

ISRv1# show standby brief
| P indicates configured to preempt. |
| Interface | Grp | Pri | P | State | Active | Standby | Virtual IP |
| Gi4 | 25 | 105 | P | Active | local | 192.0.2.1 | 192.0.2.222 |

ISRv2 (Standby) Before the Failure

ISRv2# show platform software vnic-if interface-mapping
| Interface Name | Driver Name | Mac Addr |
----------------|-------------|----------------------|
GigabitEthernet5 | i40evf | 5254.00cc.ce9f (LAN-SRIOV-2)
GigabitEthernet4 | virtio | 5254.00e7.523f (Service-net)
GigabitEthernet3 | i40evf | 5254.0055.ee45 (LAN-SRIOV-1)
GigabitEthernet2 | igbvf | 5254.00a3.d443 (GE0-0-SRIOV-1)
GigabitEthernet1 | virtio | 5254.0048.e84c (int-mgmt-net)

ISRv2# show standby brief
| P indicates configured to preempt. |
| Interface | Grp | Pri | P | State | Active | Standby | Virtual IP |
| Gi4 | 25 | 100 | P | Standby | 192.0.2.20 | local | 192.0.2.222 |

ISRv2 After the Failure

ISRv1 becomes unreachable. ISRv2: The HSRP failover occurs, and the state changes from Standby to Active. The virtual IP (LAN side default gateway) becomes active on ENCS2 ISRv2.

ISRv2# show standby brief
| P indicates configured to preempt. |
| Interface | Grp | Pri | P | State | Active | Standby | Virtual IP |
| Gi4 | 25 | 100 | P | Active | local | unknown | 192.0.2.222 |

ISRv2# show logging

Sequence of Events

1. ISRv1 HSRP on ENCS1 detects the WAN connection failure. It reduces the LAN-HSRP priority. This failure is detected when the interface goes down due to VF going down or track object going down.

Failure Points

- WAN Net1 failure (WAN SRIOV VF connected to ISRv1)
  - Link down
- WAN Phy link failure
### Failure Points
- Switch failure
- End-to-end connectivity failure

### Sequence of Events
1. WAN-IP1 becomes unreachable, and all the routes converge towards WAN-IP2 on ENCS2. WAN-IP2 is the only IP for branch connectivity.
2. HSRP on ENCS2 becomes higher priority in the group, and takes over the active role. LAN-virtual IP becomes active on ENCS2.
3. The PBR/PFR configuration will now select the LAN port as the preferred path instead of the transit link for the traffic destined to LAN.
4. All the WAN to LAN, and LAN to WAN traffic will now flow through ENCS2.

### ISRv1 After the Failure
ISRv1 becomes standby.

ISRv1# show ip interface brief
<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet1</td>
<td>192.0.2.1</td>
<td>YES NVRAM up</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet2</td>
<td>192.0.2.2</td>
<td>YES NVRAM down</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet3</td>
<td>unassigned</td>
<td>YES NVRAM administratively down</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet4</td>
<td>192.0.2.3</td>
<td>YES NVRAM up</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet5</td>
<td>unassigned</td>
<td>YES NVRAM up</td>
<td></td>
</tr>
</tbody>
</table>

ISRv1# show standby brief
| Interface | Grp Pri P State Active Standby Virtual IP |
|-----------|-------|-----------------|-------------|
| GI4       | 25    | 85 P Standby    | 192.0.2.22  | 192.0.2.222 |

ISRv1# show log

ISRv1# show track

ISRv2 After the Failure

ISRv2# show standby brief

### ISRv2 After the Failure
### Sequence of Events

1. ISRv1 HSRP on ENCS1 detects the LAN connection failure, and shut down the HSRP group. This failure is detected when the interface goes down due to the track object going down.

2. EEM script on ISRv1 withdraws the routes (for example, delete EIGRP networks). All the branch traffic routes will now converge towards WAN-IP2 on ENCS2. WAN-IP2 is the only IP for branch connectivity.

3. HSRP on ENCS-2 becomes active in the group. LAN virtual IP becomes active on ENCS2.

4. On ISRv2, the PBR/PFR configuration will now select the LAN port as the preferred path, instead of the transit link for the traffic destined to LAN.

5. All the WAN to LAN and LAN to WAN traffic will now flow through ENCS2.

### ISRv1 After the Failure

```
ISRv1# show track
Track 7
   IP SLA 7 reachability
      Reachability is Down
      7 changes, last change 00:01:40
      Latest operation return code: Timeout
Tracked by:
   HSRP GigabitEthernet3 25
   EEM 2450904616
   EEM 2450905656
ISRv1#
```

```
ISRv1# show ip eigrp topo
EIGRP-IPv4 Topology Table for AS(10)/ID(53.53.53.51)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply, r - reply Status, s - sia Status

P 19.19.19.0/24, 1 successors, FD is 3328
   via 38.38.38.38 (3328/3072), GigabitEthernet2
P 9.9.9.0/24, 1 successors, FD is 3328
   via 38.38.38.38 (3328/3072), GigabitEthernet2
P 25.25.25.0/24, 0 successors, FD is Infinity
   via 38.38.38.38 (3840/3584), GigabitEthernet2
P 27.27.27.0/24, 1 successors, FD is 3328
   via 38.38.38.38 (3328/3072), GigabitEthernet2
P 38.38.38.0/24, 1 successors, FD is 2816
   via Connected, GigabitEthernet2
P 29.29.29.0/24, 1 successors, FD is 3072
   via 38.38.38.38 (3072/2816), GigabitEthernet2
P 33.33.33.0/24, 1 successors, FD is 3840
   via 38.38.38.38 (3840/3584), GigabitEthernet2
```
ISRv2 After the Failure

ISRv2# show standby brief
P indicates configured to preempt.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Grp</th>
<th>Pri</th>
<th>P</th>
<th>State</th>
<th>Active</th>
<th>Standby</th>
<th>Virtual IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi3</td>
<td>25</td>
<td>100</td>
<td>P</td>
<td>Active</td>
<td>local</td>
<td>unknown</td>
<td>192.0.2.222</td>
</tr>
</tbody>
</table>

### Failure Points

<table>
<thead>
<tr>
<th>Sequence of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ISRv2 on ENCS2 detects the WAN connection failure. This failure is detected when the interface goes down due to VF going down or the track object going down.</td>
</tr>
<tr>
<td>2. WAN-IP2 becomes unreachable, and all the routes converge towards WAN-IP1 on ENCS1. WAN-IP1 is the only IP for branch connectivity.</td>
</tr>
<tr>
<td>3. All the WAN to LAN and LAN to WAN traffic will now flow through ENCS1.</td>
</tr>
</tbody>
</table>

### Sequence of Events

<table>
<thead>
<tr>
<th>Sequence of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ISRv2 on ENCS2 detects the link going down due to VF going down or connection failure. The connection failure is detected by the track object. Then, ENCS2 WAN-IP2 link with EEM script is shut down.</td>
</tr>
<tr>
<td>2. WAN-IP2 becomes unreachable, and all the routes converge towards WAN-IP1 on ENCS1. WAN-IP1 is the only IP for branch connectivity.</td>
</tr>
<tr>
<td>3. All the WAN to LAN and LAN to WAN traffic will now flow through ENCS1.</td>
</tr>
</tbody>
</table>
Cisco ENCS Single WAN IP Deployment Scenarios

- Single WAN IP Deployment, on page 109
- Preconfiguring the Cisco ENCS for a Single WAN IP Deployment, on page 110
- Single WAN IP Deployment with Gigabit Ethernet Interface 0/0, on page 111
- Single WAN IP Deployment with the 4G Interface, on page 112

Single WAN IP Deployment

A single WAN IP deployment can be considered when the Cisco ENCS is preconfigured at the corporate main office with the service provider's WAN IP address, and shipped to the branch office for quick deployment. At the branch office, you do not have to perform any installation or configuration task. You just have to boot the system with the preconfigured setup. The single WAN IP deployment scenario could vary as per customer requirements. The following are two sample single WAN IP deployment scenarios with the Cisco ISRv:

**Note**

Ensure that you preconfigure the Cisco ENCS at the main office before shipping the device to the branch office. You cannot connect to the remote branch office from your main office in a single WAN IP deployment scenario.

- Single WAN IP Deployment with Gigabit Ethernet Interface
- Single WAN IP Deployment with the 4G Interface
Preconfiguring the Cisco ENCS for a Single WAN IP Deployment

To preconfigure the Cisco ENCS:

1. Install Cisco Enterprise NFVIS on the Cisco ENCS via CIMC. For details, see Installing Cisco Enterprise NFVIS on a Cisco ENCS 5100 and 5400, on page 12.

2. Connect your local system (laptop) to the local management interface of the host server.


4. Upload the Cisco ISRv image using the portal, and register the VM.

5. From the portal, remove the default Gigabit Ethernet 0/0 or GE0-0 WAN interface.

6. Deploy Cisco ISRv with Gigabit Ethernet 2 for SROV-1 and Gigabit Ethernet 3 for the wan-net.

7. Open the Cisco ISRv VNC.

8. From the VNC console, configure ISRv Gigabit Ethernet 2 and Gigabit Ethernet 3 interfaces with appropriate IP addresses. Then, perform a "no shut" of the interfaces.

9. Set the WAN static IP address to be on the same subnet as ISRv Gigabit Ethernet 2 IP address, and use ISRv Gigabit Ethernet 2 interface IP address as the default gateway.

10. Ping with the Cisco ISRv IP address to ensure connectivity.

11. Configure Dynamic Multipoint VPN on the Cisco ISRv, and ensure the main server can access the portal.
Single WAN IP Deployment with Gigabit Ethernet Interface 0/0

In this scenario, two Gigabit Ethernet interfaces are configured on the Cisco ISRv: GigabitEthernet2 as the outbound interface and GigabitEthernet3 as the internal interface. The outbound interface IP address is provided by the service provider. The internal interface is the WAN interface that serves as the default gateway for Cisco Enterprise NFVIS.

crypto isakmp policy 5
   authentication pre-share
   group 2
crypto isakmp key dmvpnkey address 0.0.0.0

crypto ipsec transform-set dmvpnset esp-3des esp-sha-hmac
   mode tunnel

crypto ipsec profile dmvpnprof
   set security-association lifetime seconds 1200
   set transform-set dmvpnset

! DMVPN tunnel configuration
interface Tunnel100
   ip address 192.0.2.3 255.255.255.0
   no ip redirects
   ip mtu 1440
   ip nhrp authentication dmvpnkey
   ip nhrp map 192.0.2.1 198.51.100.1
   ip nhrp network-id 90
   ip nhrp nhs 192.0.2.2
   tunnel source GigabitEthernet2
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel protection ipsec profile dmvpnprof
!
interface GigabitEthernet2
   description this is the outbound interface
   ip address 198.51.100.2 255.255.0.0

interface GigabitEthernet3
   description this is the inside interface
   ip address 192.0.2.10 255.255.255.0
!

router eigrp 90
   network 10.4.76.0 0.0.0.255
   network 192.0.2.1
   eigrp stub connected
   no auto-summary
!
ip route 20.1.0.0 255.255.0.0 198.51.100.1
!

Smart license configuration

ip name-server 198.51.100.9
ip domain lookup
service internal
do test license smart dev-cert Enable
Single WAN IP Deployment with the 4G Interface

In this scenario, a 4G interface (NIM card) is configured as the outbound interface and Gigabit Ethernet3 as the internal interface. The outbound interface IP address is provided by the service provider. The internal interface is the WAN interface that serves as the default gateway for Cisco Enterprise NFVIS.

License Level: ax
License Type: N/A (Smart License Enabled)
Next reload license Level: ax

service timestamps debug datetime msec
service timestamps log datetime msec
service internal
service call-home
no platform pmt-keepalive disable-kernel-core
platform console virtual
platform hardware throughput level MB 1000
hostname ISRv

boot-start-marker
boot system bootflash:isrv-universalk9.16.03.02.SPA.bin
boot-end-marker

clock timezone PST -7 0
call-home
contact-email-addr callhome@cisco.com
mail-server 192.0.2.8 priority 1
alert-group-config snapshot
add-command "show license tech su"
profile "CiscoTAC-1"
active
no destination transport-method email
destination transport-method http
do destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService

clock timezone PST -7
ntp server 192.0.2.9
do license smart register idtoken NIM1NJElMDAtNDV1ZX03OTg4MGEtNmRjNjg2Yjg5OTp0T002%0ANjEzNzd8e1k5SEtL2pMTGtuNSs3Q3JxdGVoSUvpTsY2l0alVqR3B5MzFj%0A%0AVWVrST0%3D%0A

ip name-server 198.51.100.2
ip domain name cisco.com

! **IPsec configuration**

crypto isakmp policy 5
   authentication pre-share
   group 2
crypto isakmp key dmvpnkey address 0.0.0.0
! crypto ipsec transform-set dmvpnset esp-3des esp-sha-hmac
   mode tunnel
! crypto ipsec profile dmvpnprof
   set security-association lifetime seconds 1200
   set transform-set dmvpnset

! **4G interface**
controller Cellular 0/2/0
   lte modem link-recovery rssi onset-threshold -110
   lte modem link-recovery monitor-timer 20
   lte modem link-recovery wait-timer 10
   lte modem link-recovery debounce-count 6
! no ip ftp passive
ip ftp username admin
ip ftp password admin

! **DMVPN tunnel configuration**
interface Tunnel100
   ip address 198.51.100.3 255.255.255.0
   no ip redirects
   ip mtu 1440
   ip nhrp authentication dmvpnkey
   ip nhrp map 198.51.100.5 192.0.2.7
   ip nhrp network-id 90
   ip nhrp nh 198.51.100.5
   tunnel source Cellular0/2/0
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel protection ipsec profile dmvpnprof
!
interface GigabitEthernet2
   ip address 198.51.100.6 255.255.255.0
   ip nat inside
   negotiation auto
!
interface GigabitEthernet3
   ip address 198.51.100.11 255.255.255.0
   negotiation auto
!
interface Cellular0/2/0
   ip address negotiated
   load-interval 30
   dialer in-band
   dialer idle-timeout 0
   dialer-group 1
   ipv6 address autoconfig
   pulse-time 1
interface Cellular0/2/1
  no ip address
!
!
router eigrp 90
  network 198.51.100.0 0.0.0.255
  network 198.52.100.0 0.0.0.255
  network 99.0.0.0
  eigrp stub
  connected
!
virtual-service csr_mgmt
  ip shared host-interface GigabitEthernet1
    activate
  ip forward-protocol nd
  ip http server
  ip http authentication local
  ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 Cellular0/2/0
ip route 192.0.2.12 255.255.255.0 198.51.100.5
ip route 192.0.2.13 255.255.255.0 198.51.100.5
ip route 192.0.2.14 255.255.255.255 198.51.100.5
ip route 192.0.2.15 255.255.255.255 198.51.100.5
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 198.51.100.20
ip ssh authentication-retries 5
ip ssh rsa keypair-name ssh-key
ip ssh version 2
ip scp server enable
!
dialer-list 1 protocol ip permit
!
line con 0
  stopbits 1
line vty 0 4
  password cisco123
  login local
  transport input telnet ssh
!
nftp server 198.51.100.17
Resetting to Factory Default

You can reset the host server to factory default with the following three options:

- **Reset all**—Deletes VMs and volumes, files including logs, images, and certificates. Erases all configuration. Connectivity will be lost, and the admin password will be changed to factory default password.

- **Reset all (except images)**—Deletes VMs and volumes, files including logs, images, and certificates. Erases all configuration except images. Connectivity will be lost, and the admin password will be changed to factory default password.

- **Reset all (except images and connectivity)**—Deletes VMs and volumes, files including logs and certificates. Erases all configuration except images, network, and connectivity.

**Note**

This option must be used only for troubleshooting purpose. We recommend you contact Cisco Technical Support before choosing this option. This option will reboot the system. Do not perform any operations for at least twenty minutes until the system is rebooted successfully.

To reset to factory default:

```
configure terminal
factory-default-reset all
```

**Note**

Click **Yes** when you are prompted with the factory default warning message.

### Factory Default APIs and Commands

<table>
<thead>
<tr>
<th>Factory Default APIs</th>
<th>Factory Default Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/api/operations/factory-default-reset/all</code></td>
<td><code>factory-default-reset</code></td>
</tr>
<tr>
<td><code>/api/operations/factory-default-reset/all-except-images</code></td>
<td></td>
</tr>
<tr>
<td><code>/api/operations/factory-default-reset/all-except-images-connectivity</code></td>
<td></td>
</tr>
</tbody>
</table>
Event Notifications

Cisco Enterprise NFVIS generates event notifications for key events. A NETCONF client can subscribe to these notifications for monitoring the progress of configuration activation and the status change of the system and VMs.

There are two types of event notifications: nfvisEvent and vmlcEvent (VM life cycle event)

To receive event notifications automatically, you can run the NETCONF client, and subscribe to these notifications using the following NETCONF operations:

- --create-subscription=nfvisEvent
- --create-subscription=vmlcEvent

You can view NFVIS and VM life cycle event notifications using the `show notification stream nfvisEvent` and `show notification stream vmlcEvent` commands respectively.

- nfvisEvent, on page 118
- vmlcEvent, on page 127
### nfvisEvent

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| WAN_DHCP_RENEW      | DHCP renew operation is performed. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T18:06:46.142089+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Wan DHCP IP address is being renewed</status_message>
<details>NA</details>
<event_type>
WAN_DHCP_RENEW
</event_type>
</nfvisEvent>
</notification> |
| BRIDGE_DHCP_RENEW   | Bridge DHCP renew operation is performed. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Bridge DHCP IP address is being renewed</status_message>
<details>NA</details>
<event_type>
BRIDGE_DHCP_RENEW
</event_type>
</nfvisEvent>
</notification> |
| INTF_STATUS_CHANGE  | Interface status is changed. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T18:12:09.963556+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<event_type>
INTF_STATUS_CHANGE
</event_type>
<intf_name>eth7</intf_name>
<intf_prv_op>up</intf_prv_op>
<intf_op>down</intf_op>
<intf_prv_link>down</intf_prv_link>
<intf_link>down</intf_link>
</nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| NETWORK_CREATE | A network is created. | ```xml
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-09-22T12:41:04.564298+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_message>Network created succesfully</status_message>
<event_type>NETWORK_CREATE</event_type>
<network_name>testn1</network_name>
<network_bridge>test-net-br</network_bridge>
<network_sriov>false</network_sriov>
<network_vlan/>
<network_trunk/>
</nfvisEvent>
</notification>``` |
| NETWORK_UPDATE | A network is updated. | ```xml
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-09-22T12:42:03.391986+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_message>Network updated succesfully</status_message>
<event_type>NETWORK_UPDATE</event_type>
<network_name>testn1</network_name>
<network_bridge/>
<network_sriov/>
<network_vlan/>
<network_trunk/>
</nfvisEvent>
</notification>``` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| NETWORK_DELETE | A network is deleted. | ```
<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-09-22T12:42:03.391986+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_message>Network deleted successfully</status_message>
<event_type>NETWORK_DELETE</event_type>
<network_name>testn1</network_name>
<network_bridge/>
<network_sriov/>
<network_vlan/>
<network_trunk/>
</nfvisEvent>
</notification>
``` |
| UPGRADE_REGISTER | System upgrade is registered. | ```
<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T15:57:50.434636+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Upgrade package registration successful: Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg</status_message>
<event_type>UPGRADE_REGISTER</event_type>
</nfvisEvent>
</notification>
``` |
| UPGRADE_APPLY | System upgrade is applied. | ```
<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T16:02:43.885516+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Upgrade Process: In Progress</status_message>
<event_type>UPGRADE_APPLY</event_type>
</nfvisEvent>
</notification>
``` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| ROTATED_LOGS_DELETE | Rotated logs older than 30 days are deleted by the system. | `<xml version="1.0" encoding="UTF-8"?>
<rpc-reply
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
<ok/>
</rpc-reply>`

`<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T17:38:10.321152+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Delete rotated logs from archive older than 30 days</status_message>
<details>NA</details>
<event_type>ROTATED_LOGS_DELETE</event_type>
</nfvisEvent>
</notification>` |

| ROTATED_LOGS_DELETE | Older logs deleted by the system when the total file size of rotated logs exceeds 2GB. | `<xml version="1.0" encoding="UTF-8"?>
<rpc-reply
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
<ok/>
</rpc-reply>`

`<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T17:42:10.321152+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Rotated logs had exceeded 2G, older logs have been deleted to make space</status_message>
<details>NA</details>
<event_type>ROTATED_LOGS_DELETE</event_type>
</nfvisEvent>
</notification>` |
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
**REBOOT** | system reboot | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:37:47.387525+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>System will be rebooted</status_message>
<details>NA</details>
<event_type>REBOOT</event_type>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>
```  

**SHUTDOWN** | system shutdown | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>System will be shutdown</status_message>
<details>NA</details>
<event_type>SHUTDOWN</event_type>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>
```  

**SECURE** | create secure overlay | ```xml
<notification
<eventTime>2018-11-02T04:23:02.641317+00:00</eventTime>
<nfvisEvent
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Secure Overlay mgmthub initial creation. Active local bridge: wan-br</status_message>
<details>NA</details>
<event_type>SECURE_OVERLAY_CREATING</event_type>
<severity>INFO</severity>
<hostname>nfvis</hostname>
</nfvisEvent>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| SECURE_OVERLAY_UP | Secure Overlay is UP | ```
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Secure Overlay mgmthub up. Active bridge: wan-br</status_message>
<details>Secure overlay initial creation</details>
<event_type>SECURE_OVERLAY_UP</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
| WAN_DHCP_SWITCHOVER | WAN bridge toggle | ```
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Switch over to bridge wan2-br for auto DHCP enablement successful</status_message>
<details>NA</details>
<event_type>WAN_DHCP_SWITCHOVER</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| WAN_DHCP_TOGGLE_END | WAN bridge toggle | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Disabling bridge toggle for auto DHCP enablement.</status_message>
<details>NA</details>
<event_type>WAN_DHCP_TOGGLE_END</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification> |
| ROUTE_DISTRIBUTION_START | To start route distribution | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Route Distribution initial creation. Neighbor Address: 172.25.221.106</status_message>
<details>NA</details>
<event_type>ROUTE_DISTRIBUTION_START</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| ROUTE_DISTRIBUTION_DOWN | Route distribution is down                                                            | `<?xml version="1.0" encoding="UTF-8"?>  
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">  
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>  
<nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">  
  <user_id>admin</user_id>  
  <config_change>true</config_change>  
  <transaction_id>0</transaction_id>  
  <status>SUCCESS</status>  
  <status_code>0</status_code>  
  <status_message>Neighbor Address: 172.25.221.106</status_message>  
  <details>NA</details>  
  <event_type>ROUTE_DISTRIBUTION_DOWN</event_type>  
  <severity>INFO</severity>  
  <hostname>NFVIS</hostname>  
</nfvisEvent>  
</notification>` |
| ROUTE_DISTRIBUTION_ERROR   | Route distribution in error                                                            | `<?xml version="1.0" encoding="UTF-8"?>  
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">  
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>  
<nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">  
  <user_id>admin</user_id>  
  <config_change>true</config_change>  
  <transaction_id>0</transaction_id>  
  <status>SUCCESS</status>  
  <status_code>0</status_code>  
  <status_message>Neighbor Address: 172.25.221.106</status_message>  
  <details>NA</details>  
  <event_type>ROUTE_DISTRIBUTION_ERROR</event_type>  
  <severity>INFO</severity>  
  <hostname>NFVIS</hostname>  
</nfvisEvent>  
</notification>` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| ROUTE_DISTRIBUTION_DELETE  | Route distribution deleted | ```<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
  <user_id>admin</user_id>
  <config_change>true</config_change>
  <transaction_id>0</transaction_id>
  <status>SUCCESS</status>
  <status_code>0</status_code>
  <status_message>All Neighbor Addresses deleted</status_message>
  <details>NA</details>
  <event_type>ROUTE_DISTRIBUTION_DELETE</event_type>
  <severity>INFO</severity>
  <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
| ROUTE_DISTRIBUTION_UP      | Route distribution up | ```<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
  <user_id>admin</user_id>
  <config_change>true</config_change>
  <transaction_id>0</transaction_id>
  <status>SUCCESS</status>
  <status_code>0</status_code>
  <status_message>Neighbor Address: 172.25.221.106</status_message>
  <details>NA</details>
  <event_type>ROUTE_DISTRIBUTION_UP</event_type>
  <severity>INFO</severity>
  <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
| OVS_DPDK_SUCCESS           | Enable DPDK           | ```<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
  <user_id>admin</user_id>
  <config_change>false</config_change>
  <transaction_id>0</transaction_id>
  <status>SUCCESS</status>
  <status_code>0</status_code>
  <status_message>OVS-DPDK enabled</status_message>
  <details>NA</details>
  <event_type>OVS-DPDK_SUCCESS</event_type>
  <severity>INFO</severity>
  <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
### Notification Output Example

**Event Type** | **Notification Trigger** | **Notification Output Example**
---|---|---
OVS_DPDK_FAILURE | DPDK failure | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Unable to allocate CPU</status_message>
<details>NA</details>
<event_type>OVS_DPDK_FAILURE</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>
```}

### vmlcEvent

**Event Type** | **Notification Trigger** | **Notification Output Example**
---|---|---
CREATE_IMAGE | The VM image is registered. | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:12:30.76+00:00</eventTime>
vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Image creation completed successfully.</status_message>
<image>isrv-universalk9.16.03.01.tar.gz</image>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
<type>CREATE_IMAGE</type>
</event>
</vmlcEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| DELETE_IMAGE| The VM image is unregistered. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:14:51.169+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Image deletion completed successfully.</status_message>
<image>isrv-universalk9.16.03.01.tar.gz</image>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
  <type>DELETE_IMAGE</type>
</event>
</vmlcEvent>
</notification> |
| CREATE_FLAVOR| A flavor is created.       | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:12:29.685+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Flavor creation completed successfully.</status_message>
<flavor>ISRv-small</flavor>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
  <type>CREATE_FLAVOR</type>
</event>
</vmlcEvent>
</notification> |
| DELETE_FLAVOR| A flavor is deleted.      | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:14:51.425+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Flavor deletion completed successfully.</status_message>
<flavor>ISRv-small</flavor>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
  <type>DELETE_FLAVOR</type>
</event>
</vmlcEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_deployed</td>
<td>The VM is deployed.</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-17T17:19:16.927+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>VIM Driver: VM successfully created, VM Name: [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223edcb036c]</status_message>
    <depname>1479341445</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <depid>c64d79de-3a29-41a8-8114-c80d4731a5b</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
      <interfaces>
        <interface>
          <nicid>0</nicid>
          <port_id>vnet0</port_id>
          <network>int-mgmt-net</network>
          <subnet>N/A</subnet>
          <ip_address>10.20.0.2</ip_address>
          <mac_address>52:54:00:31:c5:7f</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>10.20.0.1</gateway>
        </interface>
        <interface>
          <nicid>1</nicid>
          <port_id>vnet1</port_id>
          <network>wan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:59:52:41</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
      </interfaces>
    </vm_source>
    <vm_target></vm_target>
    <event>
      <type>VM_DEPLOYED</type>
    </event>
  </vmlcEvent>
</notification>
```
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
VM_ALIVE | The state of a monitored VM becomes ACTIVE. | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:22:47.306+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>VM_Alive event received, VM ID:
[SystemAdminTenant_ROUTER_0_df673c1-0768-4ae6-8dce-b223ecdb036c]</status_message>
<depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<vm_group>ROUTER</vm_group>
<vmlc_event>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
<host_id>NFVIS</host_id>
<host_name>NFVIS</host_name>
<interfaces>
</interfaces>
</vmlcEvent>
</notification>
```
The VM is **undeployed**.

**VM_UNDEPLOYED**

### Event Type
**VM_UNDEPLOYED**

### Notification
```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:31:40.6+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>204</status_code>
<status_message>VIM Driver: VM successfully deleted</status_message>
<depname>1479341445</depname>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
<interfaces>
<intface>
<nicid>0</nicid>
<port_id>vnet0</port_id>
<network>int-mgmt-net</network>
<subnet>N/A</subnet>
<ip_address>10.20.0.2</ip_address>
<mac_address>52:54:00:31:c5:7f</mac_address>
<netmask>255.255.255.0</netmask>
<gateway>10.20.0.1</gateway>
</interface>
</interfaces>
</vm_source>
<vm_target></vm_target>
<event>
<type>VM_UNDEPLOYED</type>
</event>
</vmlcEvent>
</notification>
```
### Event Type | Notification Trigger | Notification Output Example
---|---|---
SERVICE_UPDATE | The VM is updated. | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-17T17:51:45.5+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Service group update completed successfully</status_message>
    <depname>1479342258</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <depid>827e871a-30d5-4f5f-a05a-263b7ee3a734</depid>
    <vm_source></vm_source>
    <vm_target></vm_target>
    <event>
      <type>SERVICE_UPDATE</type>
    </event>
  </vmlcEvent>
</notification>
```  

VM_STOPPED | The VM is stopped per VM action request. | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-17T17:26:05.762+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Successfully stopped VM [System Admin Tenanted_ROUTER_0_df6733c1-0768-4ae6-8dce-b223edc0036c].</status_message>
    <depname>1479341445</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid>NULL</svcid>
    <depid>c6479db-3a29-41a8-8114-48042731a5b</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target></vm_target>
    <event>
      <type>VM_STOPPED</type>
    </event>
  </vmlcEvent>
</notification>
```
### Event Type: VM_STARTED

**Notification Trigger:**
The VM is started per VM action request.

**Notification Output Example:**
```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2016-11-17T17:26:40.398+00:00</eventTime>
 <vmlcEvent
 xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
 <status>SUCCESS</status>
 <status_code>200</status_code>
 <status_message>Started VM [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c].</status_message>
 <depname>1479341445</depname>
 <tenant>admin</tenant>
 <tenant_id>AdminTenantId</tenant_id>
 <svcid>NULL</svcid>
 <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
 <vm_group>ROUTER</vm_group>
 <vm_in_source>
 <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
 <hostid>NFVIS</hostid>
 <hostname>NFVIS</hostname>
 </vm_in_source>
 <vm_in_target></vm_in_target>
 <event>
 <type>VM_STARTED</type>
 </event>
 </vmlcEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_REBOOTED</td>
<td>The VM is rebooted per VM action request.</td>
<td>&lt;xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt; &lt;notification xmlns=&quot;urn:ietf:params:xml:ns:netconf:notification:1.0&quot;&gt; &lt;eventTime&gt;2016-11-17T17:36:56.5+00:00&lt;/eventTime&gt; &lt;vmlcEvent xmlns=&quot;http://www.cisco.com/nfvis/vm_lifecycle&quot;&gt; &lt;status&gt;SUCCESS&lt;/status&gt; &lt;status_code&gt;200&lt;/status_code&gt; &lt;status_message&gt;Rebooted VM [SystemAdminTenra_ROUTER_0_f17fc494-8535-4b05-b88d-f0fd2ef7f87d]&lt;/status_message&gt; &lt;depmname&gt;1479342258&lt;/depmname&gt; &lt;tenant&gt;admin&lt;/tenant&gt; &lt;tenant_id&gt;AdminTenantId&lt;/tenant_id&gt; &lt;svcid=NULL&lt;/svcid&gt; &lt;depid&gt;827e871a-30d5-4f5f-a05a-263b7ee3a734&lt;/depid&gt; &lt;vm_group&gt;ROUTER&lt;/vm_group&gt; &lt;vm_source&gt; &lt;vmid&gt;d918a3b1-f2a9-4065-9d8e-2135b0a37d87&lt;/vmid&gt; &lt;hostid&gt;NFVIS&lt;/hostid&gt; &lt;hostname&gt;NFVIS&lt;/hostname&gt; &lt;/vm_source&gt; &lt;vm_target&gt; &lt;event&gt; &lt;type&gt;VM_REBOOTED&lt;/type&gt; &lt;/event&gt; &lt;/vm_target&gt; &lt;/vmlcEvent&gt; &lt;/notification&gt;</td>
</tr>
</tbody>
</table>
VM_RECOVERY_INIT

A monitored VM is not reachable.

---

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_RECOVERY_INIT | A monitored VM is not reachable. | <?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-17T16:27:51.627+00:00</eventTime>
  <vmrcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Recovery event for VM [SystemAdminTena_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] triggered. Processing auto healing. Proceeding with Recovery.</status_message>
    <depname>1479328919</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid>NULL</svcid>
    <depid>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target/>
    <event>
      <type>VM_RECOVERY_INIT</type>
    </event>
  </vmrcEvent>
</notification> |
Recovery reboot starts for the monitored VM, which is not reachable.

VM_RECOVERY_REBOOT

<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T16:27:53.979+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_message>VM
[SystemAdminTena_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] is
being rebooted.
</status_message>
<depid>1479328919</depid>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<svcid>NULL</svcid>
<depname>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depname>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target>
<event>
<type>VM_RECOVERY_REBOOT</type>
</event>
</vm_target>
</vmlcEvent>
</notification>
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_RECOVERY_COMPLETE   | Recovery reboot completes for the monitored VM, which is not reachable.               | ?xml version="1.0" encoding="UTF-8"?><notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"><eventTime>2016-11-17T16:31:26.934+00:00</eventTime><vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"><status>SUCCESS</status><status_code>200</status_code><status_message>Successfully recovered VM [SystemAdminTen_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b].</status_message><depid>9e7fe4f8-a5f4-4a6d-aad7-21405be4ba4</depid><tenant>admin</tenant><vm_group>ROUTER</vm_group><vm_source><vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid><hostid>NFVIS</hostid><hostname>NFVIS</hostname><interfaces><interface><nicid>0</nicid><port_id>vnet0</port_id><network>int-mgmt-net</network><subnet>N/A</subnet><ip_address>10.20.0.2</ip_address><mac_address>52:54:00:7b:3f:de</mac_address><netmask>255.255.255.0</netmask><gateway>10.20.0.1</gateway></interface><interface><nicid>1</nicid><port_id>vnet1</port_id><network>wan-net</network><subnet>N/A</subnet><mac_address>52:54:00:96:8a:4d</mac_address><netmask>255.255.255.0</netmask><gateway>172.19.181.152</gateway></interface></interfaces></vm_source><vm_target><vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid><hostid>NFVIS</hostid><hostname>NFVIS</hostname><interfaces><interface><nicid>0</nicid><port_id>vnet0</port_id><network>int-mgmt-net</network><subnet>N/A</subnet><ip_address>10.20.0.2</ip_address><mac_address>52:54:00:7b:3f:de</mac_address><netmask>255.255.255.0</netmask><gateway>10.20.0.1</gateway></interface><interface><nicid>1</nicid><port_id>vnet1</port_id><network>wan-net</network><subnet>N/A</subnet><mac_address>52:54:00:96:8a:4d</mac_address><netmask>255.255.255.0</netmask><gateway>172.19.181.152</gateway></interface></interfaces></vm_target><event><type>VM_RECOVERY_COMPLETE</type></event></vmlcEvent></notification>
### Event Type

**VM_MONITOR_UNSET**

Monitoring is disabled per VM action request.

**VM_MONITOR_SET**

Monitoring is enabled per VM action request.

### Notification Output Example

```xml
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-18T13:36:43.613+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Unset monitor completed successfully</status_message>
    <depid>1479413090</depid>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svclid>NULL</svclid>
    <depidid>742dd335-330c-4bf0-a75d-a44003c645c5</depidid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target/>
    <event>
      <type>VM_MONITOR_UNSET</type>
    </event>
  </vmlcEvent>
</notification>
```

```xml
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-18T13:40:15.276+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Set monitor completed successfully</status_message>
    <depid>1479413090</depid>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svclid>NULL</svclid>
    <depidid>742dd335-330c-4bf0-a75d-a44003c645c5</depidid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target/>
    <event>
      <type>VM_MONITOR_SET</type>
    </event>
  </vmlcEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_UPDATED</td>
<td>VM's flavor is changed.</td>
<td></td>
</tr>
</tbody>
</table>
<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-12-08T00:50:39.034+00:00</eventTime>
<vmlcEvent
xmlns='http://www.cisco.com/nfvis/vm_lifecycle'>
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>VM is resized with flavor [ISRv-medium].</status_message>
<user_name>admin</user_name>
<depname>1512766000</depname>
<tenant>admin</tenant>
<tenant_id>adminUUID</tenant_id>
<depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid>
<vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname>
</vm_source>
<event>
<type>VM_UPDATED</type>
</event>
</vmlcEvent>
</notification>
</xml>
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;![CDATA[</td>
</tr>
<tr>
<td>Event Type</td>
<td>Notification Trigger</td>
<td>Notification Output Example</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>VM_UPDATED</td>
<td>VNIC is added, deleted or updated.</td>
<td></td>
</tr>
</tbody>
</table>
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-12-08T02:10:56.184+00:00</eventTime>
  <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'>
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Added 1 interface: [managed, net=my-net-1, nicid=3]
      Updated 2 interface: [managed, net=lan-net, nicid=1], [managed, net=wan-net, nicid=2]</status_message>
    <user_name>admin</user_name>
    <depname>1512766000</depname>
    <tenant>admin</tenant>
    <tenant_id>adminUUID</tenant_id>
    <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid>
      <hostname>nfvis</hostname>
      <hostid>NFVIS</hostid>
      <interfaces>
        <interface>
          <nicid>0</nicid>
          <type>virtual</type>
          <port_id>vnic0</port_id>
          <network>int-mgmt-net</network>
          <subnet>N/A</subnet>
          <ip_address>10.20.0.3</ip_address>
          <mac_address>52:54:00:3c:ee:5b</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>10.20.0.1</gateway>
        </interface>
        <interface>
          <nicid>1</nicid>
          <type>virtual</type>
          <port_id>vnic1</port_id>
          <network>lan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:70:06:4a</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>192.168.1.1</gateway>
        </interface>
        <interface>
          <nicid>2</nicid>
          <type>virtual</type>
          <port_id>vnic2</port_id>
          <network>wan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:c7:30:1c</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
      </interfaces>
    </vm_source>
  </vmlcEvent>
</notification>
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_UPDATED</td>
<td></td>
<td>&lt;network&gt;my-net-1&lt;/network&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;subnet&gt;N/A&lt;/subnet&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;mac_address&gt;52:54:00:66:b5:c1&lt;/mac_address&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/vm_source&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;event&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;type&gt;VM_UPDATED&lt;/type&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/event&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/vmlcEvent&gt;</td>
</tr>
</tbody>
</table>
Syslog Support

Cisco enterprise NFVIS can send syslog messages to syslog servers configured by the user. Syslogs are sent for Network Configuration Protocol (NETCONF) notifications from NFVIS.

Syslog Message Format

Syslog messages have the following format:

<Timestamp> hostname %SYS-<Severity>-<Event>: <Message>

Sample Syslog messages:

2017 Jun 16 11:20:22 nfvis %SYS-6-AAA_TYPE_CREATE: AAA authentication type tacacs created successfully AAA authentication set to use tacacs server
2017 Jun 16 11:20:23 nfvis %SYS-6-RBAC_USER_CREATE: Created rbac user successfully: admin
2017 Jun 16 15:36:12 nfvis %SYS-6-CREATE_FLAVOR: Profile created: ISRv-small
2017 Jun 16 15:36:12 nfvis %SYS-6-CREATE_FLAVOR: Profile created: ISRv-medium
2017 Jun 16 15:36:13 nfvis %SYS-6-CREATE_IMAGE: Image created: ISRv_IMAGE_Test
2017 Jun 19 10:57:27 nfvis %SYS-6-NETWORK_CREATE: Network testnet created successfully
2017 Jun 21 13:55:57 nfvis %SYS-6-VM_ALIVE: VM is active: ROUTER

Configuring Syslog Servers

To configure a remote Syslog server:

```
configure terminal
system settings logging host 172.24.22.186
port 3500
transport tcp
commit
```
configure terminal
system settings logging severity error

Note
The severity levels are:

• debug
• informational
• alert
• notice
• warning
• error
• critical
• emergency

By default, the logging severity of syslogs is informational which means all syslogs at informational severity and higher will be logged.

To configure syslog facility:

configure terminal
system settings logging facility local5

Note
The logging facility can be changed to a facility from local0 to local7

By default, NFVIS sends syslogs with the facility of local7

Syslog Support APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
</table>
| • /api/config/system/settings/logging  
• /api/operational/system/settings/logging | • system settings logging host  
• system settings logging severity  
• system settings logging facility |

• Syslog Messages, on page 147
## Syslog Messages

<table>
<thead>
<tr>
<th>Event</th>
<th>Trigger Condition</th>
<th>Syslog Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK_CREATE</td>
<td>create a new network</td>
<td>nfvis %SYS-6-NETWORK_CREATE: Network my-net created successfully</td>
</tr>
<tr>
<td>NETWORK_UPDATE</td>
<td>modify an existing network</td>
<td>nfvis %SYS-6-NETWORK_UPDATE: Network my-net updated successfully</td>
</tr>
<tr>
<td>NETWORK_DELETE</td>
<td>delete a network</td>
<td>nfvis %SYS-6-NETWORK_DELETE: Network my-net deleted successfully</td>
</tr>
<tr>
<td>BRIDGE_CREATE</td>
<td>create a new bridge</td>
<td>nfvis %SYS-6-BRIDGE_CREATE: Bridge created successfully: my-bridge</td>
</tr>
<tr>
<td>BRIDGE_UPDATE</td>
<td>modify an existing bridge</td>
<td>nfvis %SYS-6-BRIDGE_UPDATE: Updated bridge successfully: my-bridge</td>
</tr>
<tr>
<td>BRIDGE_DELETE</td>
<td>delete a bridge</td>
<td>nfvis %SYS-6-BRIDGE_DELETE: Bridge deleted successfully: my-bridge</td>
</tr>
<tr>
<td>WAN_DHCP_RENEW</td>
<td>dhcp renew from wan interface</td>
<td>nfvis %SYS-6-WAN_DHCP_RENEW:wan-br DHCP IP address is being renewed</td>
</tr>
<tr>
<td>BRIDGE_DHCP_RENEW</td>
<td>bridge dhcp renew</td>
<td>nfvis %SYS-6-BRIDGE_DHCP_RENEW: Bridge DHCP IP address is being renewed</td>
</tr>
<tr>
<td>MGMT_DHCP_RENEW</td>
<td>dhcp renew from MGMT interface</td>
<td>nfvis %SYS-6-MGMT_DHCP_RENEW:wan-br DHCP IP address is being renewed</td>
</tr>
<tr>
<td>INTF_STATUS_CHANGE</td>
<td>interface status change</td>
<td>nfvis %SYS-6-INTF_STATUS_CHANGE: Interface eth0, changed state to up</td>
</tr>
<tr>
<td>UPGRADE_REGISTER</td>
<td>upgrade package registration</td>
<td>nfvis %SYS-6-UPGRADE_REGISTER: Upgrade package registration successful: Cisco_NFVIS_Upsgrade-3.6.1-698-20170402_042811.nfvispkg</td>
</tr>
<tr>
<td>UPGRADE_APPLY</td>
<td>upgrade process</td>
<td>nfvis %SYS-6-UPGRADE_APPLY: Upgrade Process: In Progress</td>
</tr>
<tr>
<td>RBAC_USER_CREATE</td>
<td>create a new user</td>
<td>nfvis %SYS-6-RBAC_USER_CREATE: Created user admin as administrators successfully</td>
</tr>
<tr>
<td>RBAC_USER_PASSWORD_UPDATE</td>
<td>change user's password</td>
<td>nfvis %SYS-6-RBAC_USER_PASSWORD_UPDATE: Set admin password successfully</td>
</tr>
<tr>
<td>RBAC_USER_ROLE_UPDATE</td>
<td>change user's role</td>
<td>nfvis %SYS-6-RBAC_USER_ROLE_UPDATE: Modified user: somebody successfully</td>
</tr>
<tr>
<td>RBAC_USER_DELETE</td>
<td>delete a user</td>
<td>nfvis %SYS-6-RBAC_USER_DELETE: Deleted rbac user successfully: somebody</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RBAC_USERS_INACTIVATED</td>
<td>disable the user</td>
<td>nfvis %SYS-6-RBAC_USERS_INACTIVATED: Following users have been marked as INACTIVE, [user1, user2]. Please take necessary action.</td>
</tr>
<tr>
<td>RBAC_USER_ACTIVATED</td>
<td>activate the user</td>
<td>nfvis %SYS-6-RBAC_USER_ACTIVATED: Modified user user1 successfully.</td>
</tr>
<tr>
<td>RBAC_PWD_EXPIRED</td>
<td>password expired</td>
<td>nfvis %SYS-6-RBAC_PWD_EXPIRED: User user1's password is older than 60 days. Please reset password.</td>
</tr>
<tr>
<td>RBAC_LOGIN_FAILURE</td>
<td>invalid user login</td>
<td>nfvis %SYS-3-RBAC_LOGIN_FAILURE: Login with invalid username from maapi failed.</td>
</tr>
<tr>
<td>SECURITY_SERVER_CREATE</td>
<td>create server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_CREATE: TACACS+ server config created successfully.</td>
</tr>
<tr>
<td>SECURITY_SERVER_UPDATE</td>
<td>update server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_UPDATE: TACACS+ server configuration updated successfully.</td>
</tr>
<tr>
<td>SECURITY_SERVER_DELETE</td>
<td>delete server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_DELETE: TACACS+ server deleted successfully.</td>
</tr>
<tr>
<td>AAA_TYPE_CREATE</td>
<td>create AAA authentication type</td>
<td>nfvis %SYS-6-AAA_TYPE_CREATE: AAA authentication type TACACS created successfully.</td>
</tr>
<tr>
<td>AAA_TYPE_UPDATE</td>
<td>update AAA authentication type</td>
<td>nfvis %SYS-6-AAA_TYPE_UPDATE: AAA authentication type TACACS+ updated successfully. AAA authentication updated to use TACACS+ server</td>
</tr>
<tr>
<td>RECREATE_CERTIFICATE</td>
<td>recreate self-sign certificate</td>
<td>nfvis%SYS-6-RECREATE_CERTIFICATE: SelfSigned Certificate re-created. Application connection may become temporarily unavailable.</td>
</tr>
<tr>
<td>CERT_CSR_CREATE</td>
<td>create a CSR file</td>
<td>nfvis %SYS-6-CERT_CSR_CREATE: signing-request created /data/intdatastore/download/nfvis.csr</td>
</tr>
<tr>
<td>CERT_SWITCH_CERT</td>
<td>switch to use different certificate</td>
<td>nfvis%SYS-6-CERT_SWITCH_CERT: switch certificate from ca-signed to self-signed.</td>
</tr>
<tr>
<td>CERT_CA_CERT_INSTALL</td>
<td>install CA signed certificate</td>
<td>nfvis%SYS-6-CERT_CA_CERT_INSTALL: ca-signed certificate file:// installed</td>
</tr>
<tr>
<td>REBOOT</td>
<td>system reboot</td>
<td>nfvis %SYS-6-REBOOT: System will be rebooted</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>system shutdown</td>
<td>nfvis %SYS-6-SHUTDOWN: System will be shutdown</td>
</tr>
<tr>
<td>LOGGING_FAILURE</td>
<td>logging failure</td>
<td>nfvis %SYS-6-LOGGING_FAILURE: Unable to write to log file nfvis_config.log. Log message: log_config.CONFIG_LOGGER: File not found.</td>
</tr>
<tr>
<td>DISK_SPACE_ALMOST_FULL</td>
<td>disk space almost full</td>
<td>nfvis %SYS-6-DISK_SPACE_ALMOST_FULL: 'lv_data' currently occupies 95% of available disk space, which is more than or equal to the threshold of 90%.</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ROTATED_LOGS_DELETE</td>
<td>delete rotated log files when accumulated rotated log files reach 2GB</td>
<td>nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated log files from archive older than 30 days</td>
</tr>
<tr>
<td>TIME_UPDATE</td>
<td>Change system time manually</td>
<td>nfvis %SYS-6-TIME_UPDATE: Manual time updated successfully Manual time is now set to 2018-04-26 11:43:00</td>
</tr>
<tr>
<td>TIMEZONE_UPDATE</td>
<td>Change system timezone</td>
<td>nfvis %SYS-6-TIMEZONE_UPDATE: Timezone updated successfully. Timezone is now set to US/Eastern</td>
</tr>
<tr>
<td>FILE_COPY_STATUS</td>
<td>copy status of file</td>
<td>nfvis %SYS-6-FIELD_COPY_STATUS: hostaction.py Copied Successfully.</td>
</tr>
<tr>
<td>CREATE_IMAGE</td>
<td>create image</td>
<td>nfvis %SYS-6-CREATE_IMAGE: Image creation successful: TinyLinux.tar.gz</td>
</tr>
<tr>
<td>DELETE_IMAGE</td>
<td>delete image</td>
<td>nfvis %SYS-6-DELETE_IMAGE: Image deletion successful: TinyLinux.tar.gz</td>
</tr>
<tr>
<td>CREATE_FLAVOR</td>
<td>create flavor</td>
<td>nfvis %SYS-6-CREATE_FLAVOR: Profile creation successful: small</td>
</tr>
<tr>
<td>DELETE_FLAVOR</td>
<td>delete flavor</td>
<td>nfvis %SYS-6-DELETE_FLAVOR: Profile deletion successful: small</td>
</tr>
<tr>
<td>VM_DEPLOYED</td>
<td>vm deployment</td>
<td>nfvis %SYS-6-VM_DEPLOYED: VM deployment successful: SystemAdmin_Tena_ROUTER_0_d1673c1-0768-4abc-8dce-b23c9d03c</td>
</tr>
<tr>
<td>VM_ALIVE</td>
<td>vm alive</td>
<td>nfvis %SYS-6-VM_ALIVE: VM active successful: SystemAdmin_Tena_ROUTER_0_d1673c1-0768-4abc-8dce-b23c9d03c</td>
</tr>
<tr>
<td>SERVICE_ALIVE</td>
<td>service alive</td>
<td>nfvis %SYS-6-SERVICE_ALIVE: Service group deployment completed successfully!</td>
</tr>
<tr>
<td>VM_UNDEPLOYED</td>
<td>vm undeployed</td>
<td>nfvis %SYS-6-VM_UNDEPLOYED: VM undeployment successful: SystemAdmin_Tena_ROUTER_0_d1673c1-0768-4abc-8dce-b23c9d03c SERVICE_UNDEPLOYED service undeployed nfvis %SYS-6-SERVICE_UNDEPLOYED: Service group undeployment completed successfully</td>
</tr>
<tr>
<td>VM_UPDATED</td>
<td>vm updated</td>
<td>nfvis %SYS-6-VM_UPDATED: VM update successful: VM is resized with flavor [ISRv-medium].</td>
</tr>
<tr>
<td>VM_UPDATED</td>
<td>vm updated</td>
<td>nfvis %SYS-6-VM_UPDATED: VM update successful: Added 1 interface: [managed, net=my-net-1, nicid=3] Updated 2 interface: [managed, net=lan-net, nicid=1],[managed, net=wan-net, nicid=2]</td>
</tr>
</tbody>
</table>

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.11.1
<table>
<thead>
<tr>
<th>Event</th>
<th>Trigger Condition</th>
<th>Syslog Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVICE_UPDATED</td>
<td>service updated</td>
<td>nfvis %SYS-6-SERVICE_UPDATED: Service group update completed successfully</td>
</tr>
<tr>
<td>VM_STOPPED</td>
<td>vm stopped</td>
<td>nfvis %SYS-6-VM_STOPPED: VM stop successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_STARTED</td>
<td>vm started</td>
<td>nfvis %SYS-6-VM_STARTED: VM start successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_REBOOTED</td>
<td>vm rebooted</td>
<td>nfvis %SYS-6-VM_REBOOTED: VM reboot successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_RECOVERY_INIT</td>
<td>vm recovery initiation</td>
<td>nfvis %SYS-6-VM_RECOVERY_INIT: VM recovery initiation successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_RECOVERY_REBOOT</td>
<td>vm recovery reboot</td>
<td>nfvis %SYS-6-VM_RECOVERY_REBOOT: VM recovery reboot successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_RECOVERY_COMPLETE</td>
<td>vm recovery complete</td>
<td>nfvis %SYS-6-VM_RECOVERY_COMPLETE: VM recovery successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_MONITOR_UNSET</td>
<td>vm monitoring unset</td>
<td>nfvis %SYS-6-VM_MONITOR_UNSET: Unsetting VM monitoring successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>VM_MONITOR_SET</td>
<td>vm monitoring set</td>
<td>nfvis %SYS-6-VM_MONITOR_SET: Setting VM monitoring successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ed606c</td>
</tr>
<tr>
<td>ROTATED LOGS DELETE</td>
<td>delete rotated logs</td>
<td>nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days</td>
</tr>
<tr>
<td>(When logs older than 30 days are present)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROTATED LOGS DELETE</td>
<td>delete rotated logs</td>
<td>nfvis %SYS-6-ROTATED_LOGS_DELETE: Rotated logs had exceeded 2G, older logs have been deleted to make space</td>
</tr>
<tr>
<td>(When Log file size exceed 2GB, older logs are deleted)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMC_PASSWORD_UPDATE</td>
<td>cimc password update operation</td>
<td>nfvis %SYS-6-CIMC_PASSWORD_UPDATE: CIMC password change is successful</td>
</tr>
<tr>
<td>BIOS_PASSWORD_UPDATE</td>
<td>bios password update operation</td>
<td>nfvis %SYS-6-BIOS_PASSWORD_UPDATE: BIOS password change is successful</td>
</tr>
<tr>
<td>SECURE_OVERLAY_CREATING</td>
<td>create secure overlay</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_CREATING: Secure Overlay mgmthub initial creation. Active local bridge: wan-br</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SECURE_OVERLAY_UP</td>
<td>secure overlay is up</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_UP: Secure Overlay mgmt hub up. Active bridge: wan-br Secure Overlay up after network interruption</td>
</tr>
<tr>
<td>SECURE_OVERLAY_DELETE</td>
<td>secure overlay is deleted</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_DELETE: Secure Overlay deleted</td>
</tr>
<tr>
<td>SECURE_OVERLAY_ERROR</td>
<td>error in secure overlay</td>
<td>nfvis %SYS-3-SECURE_OVERLAY_ERROR: Secure Overlay mgmt hub creation in error. Active bridge: wan-br Secure overlay initial creation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nfvis %SYS-3-SECURE_OVERLAY_ERROR: Secure Overlay mgmt hub creation in error. Active bridge: wan-br Cannot ping remote system ip address 10.0.0.1</td>
</tr>
<tr>
<td>WAN_DHCP_SWITCHOVER</td>
<td>WAN bridge toggle</td>
<td>nfvis %SYS-6-WAN_DHCP_SWITCHOVER: Switch over to bridge wan-br for auto DHCP enablement successful</td>
</tr>
<tr>
<td>WAN_DHCP_TOGGLE_END</td>
<td>WAN bridge toggle</td>
<td>nfvis %SYS-6-WAN_DHCP_TOGGLE_END: Disabling bridge toggle for auto DHCP enablement.</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_DOWN</td>
<td>Route distribution down</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_DOWN: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_START</td>
<td>Route distribution start</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_START: Route Distribution initial creation. Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_ERROR</td>
<td>Route distribution in error state</td>
<td>nfvis %SYS-3-ROUTE_DISTRIBUTION_ERROR: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_DELETE</td>
<td>Route distribution deleted</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_DELETE: All Neighbor Addresses deleted</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_UP</td>
<td>Route distribution up</td>
<td>nfvis %SYS-3-ROUTE_DISTRIBUTION_UP: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>OVS_DPDK_SUCCESS</td>
<td>Enable DPDK</td>
<td>nfvis %SYS-3-OVS_DPDK_SUCCESS: OVS-DPDK enabled</td>
</tr>
<tr>
<td>OVS_DPDK_FAILURE</td>
<td>DPDK failure</td>
<td>nfvis %SYS-3-OVS_DPDK_FAILURE: Unable to allocate CP</td>
</tr>
</tbody>
</table>
CHAPTER 16

SNMP Support on NFVIS

- Introduction about SNMP, on page 153
- SNMP Operations, on page 153
- SNMP Versions, on page 155
- SNMP MIB Support, on page 156
- Configuring SNMP Support, on page 158

Introduction about SNMP

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

The SNMP framework has three parts:

- SNMP manager - The SNMP manager is used to control and monitor the activities of network hosts using SNMP.
- SNMP agent - The SNMP agent is the software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems.
- MIB - The Management Information Base (MIB) is a virtual information storage area for network management information, which consists of collections of managed objects.

A manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited notifications (traps or informs) to the manager to notify the manager of network conditions.

SNMP Operations

SNMP applications perform the following operations to retrieve data, modify SNMP object variables, and send notifications:

- SNMP Get - The SNMP GET operation is performed by a Network Management Server (NMS) to retrieve SNMP object variables.
- SNMP Set - The SNMP SET operation is performed by a Network Management Server (NMS) to modify the value of an object variable.
SNMP Get

The SNMP GET operation is performed by a Network Management Server (NMS) to retrieve SNMP object variables. There are three types of GET operations:

- **GET**: Retrieves the exact object instance from the SNMP agent.
- **GETNEXT**: Retrieves the next object variable, which is a lexicographical successor to the specified variable.
- **GETBULK**: Retrieves a large amount of object variable data, without the need for repeated GETNEXT operations.

The command for SNMP GET is:

```
snmpget -v2c -c [community-name] [NFVIS-box-ip] [tag-name, example ifSpeed].[index value]
```

SNMP Walk

SNMP walk is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information.

An object identifier (OID) may be given on the command line. This OID specifies which portion of the object identifier space will be searched using GETNEXT requests. All variables in the subtree below the given OID are queried and their values presented to the user.

The command for SNMP walk with SNMP v2 is:

```
snmpwalk -v2c -c [community-name] [nfvis-box-ip]
```

```
snmpwalk -v2c -c myUser 172.19.147.115 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Cisco NFVIS
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.12.3.1.3.1291
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (43545580) 5 days, 0:57:35.80
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 70
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
IF-MIB::ifIndex.11 = INTEGER: 11
IF-MIB::ifDescr.1 = STRING: GE0-0
IF-MIB::ifDescr.2 = STRING: GE0-1
IF-MIB::ifDescr.3 = STRING: MGMT
IF-MIB::ifDescr.4 = STRING: gigabitEthernet1/0
IF-MIB::ifDescr.5 = STRING: gigabitEthernet1/1
IF-MIB::ifDescr.6 = STRING: gigabitEthernet1/2
```
The following is a sample configuration of SNMP walk with SNMP v3:

```
snmpwalk -v 3 -u user3 -a sha -A changePassphrase -x aes -X changePassphrase -l authPriv
-n snmp 172.16.1.101 system
```

**SNMP Notifications**

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Unsolicited (asynchronous) notifications can be generated as traps or inform requests. Traps are messages alerting the SNMP manager to a condition on the network. Inform requests (informs) are traps that include a request for confirmation of receipt from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

**Note**

Starting from Release 3.8.1 NFVIS has SNMP Trap support for switch interfaces. If a trap server is setup in the NFVIS snmp configuration, it will send trap messages for both NFVIS and switch interfaces. Both the interfaces are triggered by the link state up or down by unplugging a cable or setting admin_state up or down when a cable is connected.

**SNMP Versions**

Cisco enterprise NFVIS supports the following versions of SNMP:
• SNMP v1—The Simple Network Management Protocol: A Full Internet Standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.

• SNMP v2c—The community-string based Administrative Framework for SNMPv2. SNMPv2c (the "c" stands for "community") is an Experimental Internet Protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic), and uses the community-based security model of SNMPv1.

• SNMPv3—Version 3 of SNMP. SNMPv3 is an interoperable standards-based protocol defined in RFCs 3413 to 3415. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network.

The security features provided in SNMPv3 are as follows:

• Message integrity—Ensuring that a packet has not been tampered with in transit.

• Authentication—Determining that the message is from a valid source.

• Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.

Both SNMP v1 and SNMP v2c use a community-based form of security. The community of managers able to access the agent MIB is defined by an IP address Access Control List and password.

SNMPv3 is a security model in which an authentication strategy is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

Authentication of the community with the user configuration is implemented even though SNMP v1 and v2 traditionally do not require a user configuration to be set. For both SNMP v1 and v2 on NFVIS, the user must be set with the same name and version as the corresponding community name. The user group must also match an existing group with the same SNMP version for snmpwalk commands to work.

SNMP MIB Support

The following MIB’s are supported for SNMP on NFVIS:

IF-MIB:
- ifDescr
- ifType
- ifPhysAddress
- ifSpeed
- ifOperStatus
- ifAdminStatus
- ifMtu
- ifName
• ifHighSpeed
• ifPromiscuousMode
• ifConnectorPresent
• ifInErrors
• ifInDiscards
• ifInOctets
• ifOutErrors
• ifOutDiscards
• ifOutOctets
• ifOutUcastPkts
• ifHCInOctets
• ifHCInUcastPkts
• ifHCOutOctets
• ifHCOutUcastPkts
• ifInBroadcastPkts
• ifOutBroadcastPkts
• ifInMulticastPkts
• ifOutMulticastPkts
• ifHCInBroadcastPkts
• ifHCOutBroadcastPkts
• ifHCInMulticastPkts
• ifHCOutMulticastPkts

Entity MIB:
• entPhysicalIndex
• entPhysicalDescr
• entPhysicalVendorType
• entPhysicalContainedIn
• entPhysicalClass
• entPhysicalParentRelPos
• entPhysicalName
• entPhysicalHardwareRev
• entPhysicalFirmwareRev
Configuring SNMP Support

Though SNMP v1 and v2c is using community-based string, the following is still required:

- Same community and user name.
- Same SNMP version for user and group.

To configure SNMP v2 support:
configure terminal
snmp community public community-access readOnly

snmp group testgroup snmp 2 noAuthNoPriv read read-access write write-access notify notify-access

snmp user public user-group testgroup user-version 2

snmp host host2 host-ip-address 2.2.2.2 host-port 162 host-user-name public host-version 2 host-security-level noAuthNoPriv

snmp enable traps linkUp

To configure SNMP v3 support:

configure terminal
snmp group testgroup3 snmp 3 authPriv notify test write test read test
snmp user user3 user-version 3 user-group testgroup3 auth-protocol sha priv-protocol aes
passphrase changePassphrase
! configure snmp host to enable snmp v3 trap
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
host-security-level authPriv host-port 162
!
! Change to different security level
!
snmp group testgroup4 snmp 3 authNoPriv notify test write test read test
snmp user user4 user-version 3 user-group testgroup4 auth-protocol md5 passphrase
changePassphrase
! configure snmp host to enable snmp v3 trap
snmp host host4 host-ip-address 4.4.4.4 host-version 3 host-user-name user4
host-security-level authNoPriv host-port 162
!
!

snmp enable traps linkUp
snmp enable traps linkDown

Note
SNMP host configuration is supported for NFVIS 3.6.1 release. Host trap server configuration will be officially supported for NFVIS 3.7.1 release.

Note
SNMP v3 context snmp is added automatically when configured from the web portal. To use a different context value or empty context string, use NFVIS CLI or API for configuration.

NFVIS SNMP v3 only supports single passphrase for both auth-protocol and priv-protocol.

Note
NFVIS 3.11.1 release enhances the special character support for passphrase. Now the following characters are supported: @#$-!&*

Verify the configuration for SNMP support
Use the show snmp agent command to verify the snmp agent description and ID.
nfvis# show snmp agent

snmp agent sysDescr "Cisco NFVIS"
snmp agent sysOID 1.3.6.1.4.1.9.12.3.1.3.1291

Use the `show snmp traps` command to verify the state of snmp traps.

nfvis# show snmp traps

<table>
<thead>
<tr>
<th>TRAP</th>
<th>TRAP NAME</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>linkDown</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td>linkUp</td>
<td>enabled</td>
<td></td>
</tr>
</tbody>
</table>

Use the `show snmp stats` command to verify the snmp stats.

nfvis# show snmp stats

snmp stats sysUpTime 57351917
snmp stats sysServices 70
snmp stats sysORLastChange 0
snmp stats snmpInPkts 104
snmp stats snmpInBadVersions 0
snmp stats snmpInBadCommunityNames 0
snmp stats snmpInBadCommunityUses 0
snmp stats snmpInASNParseErrs 0
snmp stats snmpSilentDrops 0
snmp stats snmpProxyDrops 0

Use the `show running-config snmp` command to verify the interface configuration for snmp.

nfvis# show running-config snmp

snmp agent enabled true
snmp agent engineID 00:00:00:09:11:22:33:44:55:66:77:88
snmp enable traps linkUp
snmp community pub_comm
community-access readOnly
snmp community tachen
community-access readOnly
snmp group tachen snmp 2 noAuthNoPriv
read test
write test
notify test
snmp group testgroup snmp 2 noAuthNoPriv
read read-access
write write-access
notify notify-access
snmp user public
user-version 2
user-group 2
auth-protocol md5
priv-protocol des
snmp user tachen
user-version 2
user-group tachen
!

snmp host host2
host-port 162
host-ip-address 2.2.2.2
host-version 2
host-security-level noAuthNoPriv
host-user-name public
!

Upper limit for SNMP configurations:

- Communities: 10
- Groups: 10
- Users: 10
- Hosts: 4

SNMP Support APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/snmp/agent</td>
<td>agent</td>
</tr>
<tr>
<td>/api/config/snmp/communities</td>
<td>community</td>
</tr>
<tr>
<td>/api/config/snmp/enable/traps</td>
<td>trap-type</td>
</tr>
<tr>
<td>/api/config/snmp(hosts</td>
<td>host</td>
</tr>
<tr>
<td>/api/config/snmp/user</td>
<td>user</td>
</tr>
<tr>
<td>/api/config/snmp/groups</td>
<td>group</td>
</tr>
</tbody>
</table>
TACACS and RADIUS Support on NFVIS

• About RADIUS, on page 163
• RADIUS Operation, on page 163
• Configuring a TACACS+ Server, on page 164
• Configuring RADIUS, on page 165
• Specifying TACACS and RADIUS Authentication, on page 166

About RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client-server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its AAA security paradigm. RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

1. The user is prompted to enter the username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   1. ACCEPT—The user is authenticated.
   2. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   3. CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.
4. **REJECT**—The user is not authenticated and is prompted to reenter the username and password, or access is denied.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including connections such as Telnet, rlogin, or local-area transport (LAT), and services such as PPP, Serial Line Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

## Configuring a TACACS+ Server

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must configure a TACACS+ server before the configured TACACS+ features on your network access server are available.

On the TACACS+ server, ensure you configure Cisco attribute-value (AV) pair privilege level (priv-lvl) for Cisco Enterprise NFVIS service for the minimum privilege level of administrators and operators.

For more details on TACACS+ configuration, see the Configuring TACACS module in TACACS+ Configuration Guide, Cisco IOS XE Release 3S.

### Note

Users with no privilege level or users with a privilege level that is less than the operator's privilege level are considered as auditors with read-only permission.

To configure TACACS+:

```
configure terminal
tacacs-server host 209.165.201.20 shared-secret test1
key 0
admin-priv 14
oper-priv 9
commit
```

In this configuration, privilege level 14 is assigned to the administrator role, and privilege level 9 is assigned to the operator role. This means a user with privilege level 14 or higher will have all admin privileges when the user logs into the system, and a user with privilege level 9 or higher will have all privileges of an operator at the time of login.

Starting from NFVIS 3.9.2 release, TACACS+ secret encryption is supported. You can only configure either secret key or encrypted secret key at a given time. To configure encrypted TACACS+ key:

```
configure terminal
tacacs-server host 209.165.201.20 encrypted-shared-secret test1
key 0
admin-priv 14
oper-priv 9
commit
```
Verifying the TACAC+ configuration

Use the `show running-config tacacs-server` command to verify the configuration if encrypted TACACS+ key is configured:

```bash
nfvis# show running-config tacacs-server
```

```
tacacs-server host 209.165.201.20
    encrypted-shared-secret $8$mRTnL9TKZCFi1BUP7Mwbm3JVIo4Z7QvJ
    admin-priv 15
    oper-priv 11

Verifying the RADIUS configuration

Use the `show running-config radius-server` command to verify the interface configuration for a RADIUS session:

```bash
nfvis# show running-config radius-server
```

```
radius-server host 103.1.4.3
    key 0
    shared-secret cisco123
    admin-priv 2
    oper-priv 1
    commit
```

### TACACS+ APIs and Commands

<table>
<thead>
<tr>
<th>TACACS+ APIs</th>
<th>TACACS+ Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/security_servers/tacacs-server</td>
<td>tacacs-server host</td>
</tr>
<tr>
<td>/api/config/security_servers/tacacs-server?deep</td>
<td>key</td>
</tr>
<tr>
<td>/api/config/security_servers/tacacs-server/host/&lt;ip-address/domain-name&gt;</td>
<td>admin-priv</td>
</tr>
<tr>
<td></td>
<td>oper-priv</td>
</tr>
</tbody>
</table>

### Configuring RADIUS

To configure RADIUS support:

```bash
radius-server host 103.1.4.3
    key 0
    shared-secret cisco123
    admin-priv 2
    oper-priv 1
    commit
```

Starting from NFVIS 3.9.2 release, TACACS+ secret encryption is supported. You can only configure either secret key or encrypted secret key at a given time. To configure encrypted RADIUS key:

```bash
radius-server host 103.1.4.3
    key 0
    encrypted-shared-secret cisco123
    admin-priv 2
    oper-priv 1
    commit
```

Verifying the RADIUS configuration

Use the `show running-config radius-server` command to verify the interface configuration for a RADIUS session:

```bash
nfvis# show running-config radius-server
```

```
radius-server host 103.1.4.3
    key 0
```
shared-secret cisco123
admin-priv 2
oper-priv 1

RADIUS Support APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/security_servers/radius-server</td>
<td>host</td>
</tr>
</tbody>
</table>

Specifying TACACS and RADIUS Authentication

NFVIS supports both TACACS+ and RADIUS but only one authentication method can be enabled at a time. After you have identified the TACACS+ and RADIUS server and defined an associated TACACS+ and RADIUS authentication key, you must define method lists for TACACS+ and RADIUS authentication. Because TACACS+ and RADIUS authentication is operated through AAA, you need to issue the aaa authentication command, specifying TACACS+ or RADIUS as the authentication method.

```
nfvis(config)# aaa authentication ?
Possible completions:
radius Use RADIUS for AAA
tacacs Use TACACS+ for AAA
users List of local users
```

- Only when TACACS+ or RADIUS is enabled, it can be used for authentication.
- When TACACS+ or RADIUS is not accessible, local authentication is used. Local authentication is disabled if the connection between TACACS+ or RADIUS and NFVIS is restored.
- If the same username exists on both local and TACACS+ or RADIUS, then TACACS+ or RADIUS user is chosen for authentication.
- It is recommended to configure Syslog Support, on page 145 so that it is easier to debug if TACACS+ or RADIUS does not work as expected.

All login attempts will be logged in syslogs in the local /var/log/nfvis_syslog.log file and in remote syslog servers. It is important to configure a remote syslog server when configuring TACACS+/RADIUS in order to be able to view logs regarding login attempts when TACACS+/RADIUS is configured.
ENCS Switch Portal Configuration

- Switch Settings, on page 167
- Configuring Spanning Tree, on page 169
- Configuring Dot1x, on page 171
- Configuring LACP, on page 172
- Configuring VLAN, on page 173
- Configuring General Settings, on page 174
- Configuring Advanced Settings, on page 175
- Configuring Spanning Tree per Interface, on page 176

Switch Settings

The **Switch** option from the Cisco Enterprise NFVIS portal allows you to configure STP/RSTP, VLAN on specified ranges, RADIUS based authentication, and port channel load balancing for various switch ports. This section describes how to configure settings on the ENCS switch portal.
You can view the Switch Interface operational data and the statistics parameters in the following table:

**Table 5: Switch Settings Interface**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwitchPort</td>
<td>Specifies the switch interface name.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Specifies the description of the interface.</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Specifies the status of the interface.</td>
<td>up or down</td>
</tr>
<tr>
<td>MAC Address</td>
<td>Specifies the MAC address of the interface.</td>
<td></td>
</tr>
<tr>
<td>PortType</td>
<td>Specifies the mode of the port interface.</td>
<td>Supported types are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dot1q-tunnel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• private-vlan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• trunk</td>
</tr>
<tr>
<td>VLAN</td>
<td>Specifies the VLAN ID.</td>
<td>Range: 1-2349 and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2450-4093</td>
</tr>
</tbody>
</table>
Configuring Spanning Tree

Spanning Tree Protocol (STP) is a Layer 2 protocol that runs on bridges and switches. The main purpose of STP is to ensure that you do not create loops when you have redundant paths in your network.

The Spanning Tree option is enabled by default. You can click on edit and make the necessary settings or disable Spanning Tree if required.
The configuration of spanning tree has the following parameters when it is enabled:

Table 6: Spanning Tree Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanning Tree</td>
<td>Specifies the state of the Spanning Tree.</td>
<td>Enable or Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is Enable.</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies the mode of the Spanning Tree.</td>
<td>stp or rstp</td>
</tr>
<tr>
<td>Forward Time</td>
<td>Specifies the Spanning Tree forward time in seconds.</td>
<td>Range: 4-30 seconds</td>
</tr>
<tr>
<td>Hello Time</td>
<td>Specifies the Hello time in seconds.</td>
<td>Range: 1 to 10 seconds</td>
</tr>
<tr>
<td>Max Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loopback Guard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Path Cost Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dot1x

This chapter describes how to configure dot1x port-based authentication on the Cisco Enterprise NFVIS portal. dot1x prevents unauthorized devices (clients) from gaining access to the network. It is a standard for media-level (Layer 2) access control, offering the capability to permit or deny network connectivity based on the identity of the end user or device. The dot1x is disabled by default. You can click on edit to enable dot1x.

### Table: Configuring Dot1x Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Age</td>
<td>Specifies the spanning-tree bridge maximum age in seconds.</td>
<td>Range: 6 to 40 seconds</td>
</tr>
<tr>
<td>Loopback Guard</td>
<td>Specifies the loopback guard status.</td>
<td>Enable or Disable</td>
</tr>
<tr>
<td>Path Cost Method</td>
<td>Specifies the speed of the interface.</td>
<td>Method:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• long - for 32 bit based values for default port path costs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• short - 16 bit based values for default port path costs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default method is long.</td>
</tr>
<tr>
<td>Priority</td>
<td>Specifies the port priority.</td>
<td>Range: 0 to 61440 in steps of 4096</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 32768.</td>
</tr>
<tr>
<td>BPDU Filtering</td>
<td>Specifies that BPDU packets are filtered when the spanning tree is disabled on an interface.</td>
<td></td>
</tr>
<tr>
<td>BPDU Flooding</td>
<td>Specifies that BPDU packets are flooded unconditionally when the spanning tree is disabled on an interface.</td>
<td></td>
</tr>
</tbody>
</table>
The configuration of dot1x has the following parameters:

**Table 7: Dot1x Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Specifies the authentication type for the port.</td>
<td>radius or none</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is radius.</td>
</tr>
<tr>
<td>Guest VLAN Timeout(s)</td>
<td>Specifies the time delay in seconds between enabling Dot1X (or port up) and adding the port to the guest VLAN.</td>
<td>Range: 30 to 180 seconds</td>
</tr>
<tr>
<td>System Auth control</td>
<td>Specifies the authentication control.</td>
<td>Enable or Disable</td>
</tr>
</tbody>
</table>

**Configuring LACP**

The Link Aggregation Control Protocol (LACP) enables you to bundle several physical ports together to form a single logical channel. LACP enables you to form a single Layer 2 link automatically from two or more Ethernet links. This protocol ensures that both ends of the Ethernet link are functional and are part of the aggregation group.
LACP uses the following parameters to control aggregation:

*Table 8: LACP Parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Priority</td>
<td>Specifies the port priority.</td>
<td>Range: 1 to 65535</td>
</tr>
<tr>
<td>Port-channel load balance</td>
<td>Specifies the load balance of the port channel.</td>
<td>Mac Based or IP Based</td>
</tr>
</tbody>
</table>

**Configuring VLAN**

You can use virtual LANs (VLANs) to divide the network into separate logical areas. VLANs can also be considered as broadcast domains. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

You can configure VLANs in the range `<1-2349>|<2450-4093>` for a specified switch port.
You can configure general settings using the following parameters for each switch interface:

- **Interface**—Name of the interface
- **Description**—Set the description per interface
- **Speed**—10/100/1000 MBPS
- **Dot1x Auth**—802.1x, mac or both
- **PoE Method**—auto, never or four-pair
- **PoE Limit**—0-60000mW
- **Admin Status**—enable or disable
Configuring Advanced Settings

You can make the advanced settings using the following parameters for each switch interface:

- **Mode**—access, dot1q-tunnel, private-vlan, or trunk
- **Access Vlan**—Specifies the number of VLANs.
- **Allowed Vlan**—All or VLAN IDs
- **Native Vlan**—Specifies the VLAN ID. You can enter a value from one of the following ranges:
  - 1 to 2349
  - 2450 to 4093
- **Protected Port**—Yes or No

![Advanced Settings configuration interface](image)
- Dot1q Tunnel Vlan—Specifies the Layer 2 tunnel port.
- Community—Specifies the community number. Range: 1 to 29
- Protected Port—Yes or No

**Note**

The VLAN configuration takes effect only if the global VLANs are also configured with the same values in Configuring VLAN, on page 173.

## Configuring Spanning Tree per Interface

<table>
<thead>
<tr>
<th>General Settings</th>
<th>Advanced Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spanning Tree</strong></td>
<td>Enable, Disable</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Choose from 1-200000000</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>128</td>
</tr>
<tr>
<td><strong>Link Type</strong></td>
<td></td>
</tr>
<tr>
<td><strong>BPDU Guard</strong></td>
<td>Enable, Disable</td>
</tr>
<tr>
<td><strong>Root Guard</strong></td>
<td>Enable, Disable</td>
</tr>
<tr>
<td><strong>Port Fast</strong></td>
<td>auto</td>
</tr>
</tbody>
</table>

[Apply] [Cancel]
You can configure spanning tree for each switch interface using the following parameters:

- Spanning Tree—Enable or Disable
- Cost—Specifies the cost. Range: 1 to 200000000
- Priority—Specifies the port priority. Range: 0 to 240, default value is 128
- Link Type—point-to-point or shared
- BPDU Guard—Enable or Disable
- Root Guard—Enable or Disable
- Port Fast—auto or enable
- BPDU Filtering—Specifies that BPDU packets are filtered when the spanning tree is disabled
- BPDU Flooding—Specifies that BPDU packets are flooded when the spanning tree is disabled
**CHAPTER 19**

**Configuring Secondary IP and Source Interface**

**Secondary IP**

The Cisco Enterprise NFVIS supports multiple IP addresses per interface. A Secondary IP feature can be configured on the WAN interface, as an additional IP to reach the software. Set the external routes for Secondary IP to reach the NFVIS. Routers configured with secondary addresses can route between the different subnets attached to the same physical interface.

To access secondary IP through ISRv, the WAN physical port is removed from wan-br similar to single IP.

**To configure Secondary IP:**

```
Configure Secondary IP
nfvis(config)# system settings wan secondary ip address 1.1.2.3 255.255.255.0
```

**Source Interface**

This feature is used to set the source interface for packets with source IP address, generated by NFVIS using the default route.

**Prerequisites for configuring Source Interface**

- IP must be one of the configured IP addresses in system settings.
- The source-interface IP address can be one of the following:
  - mgmt
  - WAN
  - WAN Secondary IP
  - WAN2 IP or IP configured on any bridge
- Source-interface configuration must be applied if the WAN IP is static.
- For DHCP, Source-interface IP is accepted but cannot be applied. The configuration takes effect once you switch from DHCP to static.
To configure Source Interface:

```plaintext
Configure source-interface ip
nfvis(config)# system settings source-interface 1.1.2.3
```

The Secondary IP and Source Interface related errors are logged in `show log /var/log/nfvis_config.log` file.

### Secondary IP and Source Interface APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/system/settings/wan/secondary</td>
<td>• system settings wan secondary</td>
</tr>
<tr>
<td>• /api/config/system/settings/source-interface</td>
<td>• system settings source-interface</td>
</tr>
</tbody>
</table>
Ports and Port Channels

This chapter contains the following sections.

- Configuring Port Channels, on page 181
- Configuring LLDP, on page 184
- Configuring Admin Status of a Port, on page 184
- Tracking Changes for a Port, on page 185
- Speed, Duplex and Autonegotiation, on page 185

Configuring Port Channels

Information About Port Channels

Port channels provide a mechanism for combining individual links into a group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. Port channels provide increased bandwidth and redundancy and balance the traffic load between the members port. If a member port within a port channel fails, traffic previously carried over the failed port switches to the remaining member ports.

Port channels can be configured using static mode (no protocol) or the Link Access Control Protocol (LACP). Any configuration changes that you apply to the port channel are applied to each member port of the port channel. A port channel must have at least two ports. A port channel can be added to a bridge. A bond is created when a port channel has more than two members and the port channel is added to a bridge.

A port can be a member of only one port channel. All the ports in a port channel must be compatible; they must use the same speed and operate in full-duplex mode.

Port Channels Bond Mode

A port channel can be configured for the following bond modes:

- **active-backup**: In this mode, one of the ports in the aggregated link is active and all others ports are in the standby mode.

- **balance-slb**: In this mode, load balancing of traffic is done based on the source MAC address and VLAN.

- **balance-tcp**: In this mode, 5-tuple (source and destination IP, source and destination port, protocol) is used to balance traffic across the ports in an aggregated link.
Port Channels LACP Mode

A port channel can be configured for the following LACP modes:

- **off**: Indicates that no mode is applicable.
- **active**: Indicates that the port initiates transmission of LACP packets.
- **passive**: Indicates that the port only responds to the LACP packets that it receives but does not initiate the LACP negotiation.

Creating a Port Channel

To create a port channel:

```
configure terminal
pnic pc type port_channel lACP_type active bond_mode balance-tcp trunks 10, 20
commit
```

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics</td>
<td>pnic name type port_channel</td>
</tr>
<tr>
<td>/api/operational/pnics</td>
<td>show pnic</td>
</tr>
</tbody>
</table>

Adding a Port to a Port Channel

A port channel must have at least two ports. A bond is created when a port channel has more than two members and the port channel is added to a bridge. You can add a port to a new port channel or a port channel that already contains ports.

To add a port to a port channel:

```
configure terminal
pnic eth1 member_of pc
commit
```

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/name/member_of</td>
<td>pnic name member_of portchannel_name</td>
</tr>
</tbody>
</table>

Adding a Port Channel to a Bridge

You can add a port channel to a new bridge or an existing bridge. When a port channel is added to a bridge, a bond is added for the port channel.

To add a port channel to a bridge:
configure terminal
bridges bridge test-br port pc
commit

Adding a Port Channel to a Bridge APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/bridges/bridge/bridgename</td>
<td>bridges bridge name port portchannel_name</td>
</tr>
</tbody>
</table>

Deleting a Port Channel

Before deleting a port channel, you must remove all members assigned to the port channel. If the port channel is configured on the bridge, you must remove the port channel from the bridge.

To delete a port channel:

configure terminal
no pnic pc
commit

Port Channel Deletion APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/portchannel_name</td>
<td>no pnic portchannel_name</td>
</tr>
<tr>
<td>/api/operational/pnics</td>
<td>show pnic</td>
</tr>
</tbody>
</table>

Removing a Port from a Port Channel

To remove a port from a port channel:

configure terminal
no pnic eth1 member_of pc
commit

Removing a Port from a Port Channel APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/name/member_of</td>
<td>no pnic name member_of portchannel_name</td>
</tr>
</tbody>
</table>

Removing a Port Channel from a Bridge

To remove a port channel from a bridge:

configure terminal
no bridges bridge test-br port pc
commit
Removing a Port Channel from a Bridge APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/bridges/bridge/bridgename</td>
<td>no bridges bridge bridgename port portname</td>
</tr>
</tbody>
</table>

Configuring LLDP

To enable LLDP on a port:

```bash
configure terminal
pnic eth0 lldp enabled
commit
```

To disable LLDP on a port:

```bash
configure terminal
pnic eth0 lldp disabled
commit
```

LLDP Configuration APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/portname/lldp</td>
<td>pnic name lldp</td>
</tr>
<tr>
<td>/api/operational/lldp</td>
<td>show lldp</td>
</tr>
<tr>
<td>/api/operational/lldp?deep</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Admin Status of a Port

To bring a port up administratively:

```bash
configure terminal
pnic eth5 adminstatus up
commit
```

To bring a port down administratively:

```bash
configure terminal
pnic eth5 adminstatus down
commit
```

Admin Status Configuration APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/portname/adminstatus</td>
<td>pnic name adminstatus</td>
</tr>
</tbody>
</table>
Tracking Changes for a Port

This feature is supported only on ENCS 5400.

The configured VNICS tracks the state of the ports based on the PNICS notifications. To verify the state of the port, use show interface or ethtool commands. You can also use commands specific to the VM, that displays the interface link state.

To configure track state on GE0-0 & GE0-1:

    configure terminal
    pnic GE0-0 track-state ROUTER 1
    end

To configure track state on switch port:

    configure terminal
    switch interface gigabitEthernet 1/0 track-state ROUTER 2
    end

Speed, Duplex and Autonegotiation

To enable autonegotiation on a port:

    configure terminal
    pnic GE0-0 speed auto duplex auto
    commit

To configure speed and duplex with non auto values:

    configure terminal
    pnic GE0-0 speed 100 duplex full
    commit

Speed can be set to 10, 100, 1G, 10G, 10G_PF, 10G_SR, 10G_VF and auto. Duplex values can be set at full, half and auto.

Use show pnic GE0-0 operational-speed, show pnic GE0-0 operational-duplex and show pnic GE0-0 autoneg to verify the configurations.

The speed and duplex configurations are dependent on the peer configuration. If the peer is set at a certain speed and duplex, NFVIS port is set to match that speed. Not all ports on ENCS 5000 series hardware devices support Automatic medium-dependent interface crossover (auto-MDIX) feature. Based on the port connected to the ENCS device, the cable type used to connect to the peer and the speed or duplex settings on the peer, you can try straight through and cross over cable.
MSTP for ENCS 5400 8-Port Switch

Multiple Spanning Trees Protocol (MSTP) is introduced to the 8-port switch on ENCS 5400. MSTP enables multiple VLANs to be mapped to the same spanning tree instance, which reduces the number of spanning-tree instances needed to support a large number of VLANs.

ENCS 5400 switch supports 15 instances. Each spanning tree instance is identified by an instance ID from 1 to 15.

To enable MST:

```plaintext
configure terminal
switch
  spanning-tree mode mst
commit
```

To configure VLAN-to-instance mapping:

```plaintext
configure terminal
switch
  spanning-tree mst configuration
    instance 1 vlan 15
    name mst_test
    revision 2
commit
```

To configure the switch priority:

```plaintext
configure terminal
switch
  spanning-tree mst 1 priority 0
commit
```

To configure the path cost per interface's mst instance:

```plaintext
configure terminal
switch
  interface gigabitEthernet 1/1
    spanning-tree mst 1 port-priority 96
    spanning-tree mst 1 cost 2
commit
```

To configure the maximum hop count for all MST instances:

```plaintext
configure terminal
```
switch
    spanning-tree mst max-hops 10
commit

Use the `show switch spanning-tree mstp configuration global` command to verify the MSTP configuration.

nfvis# show switch spanning-tree mstp configuration global
spanning-tree mstp configuration global name transit-net
spanning-tree mstp configuration global revision 2
spanning-tree mstp configuration global max-hops 20

Use the `show switch spanning-tree mstp configuration instance-list` command to verify the MSTP instance configuration.

nfvis# show switch spanning-tree mstp configuration instance-list
INSTANCE   VLANS MAPPED       STATE
-------------------------------------
0          1,2350-2353,2363   enabled
1           15               enabled

To display the global information for MSTP instance 2 use `show switch spanning-tree mstp summary instance-global-info` command.

To display interface information for MSTP instance 2 use `show switch spanning-tree mstp summary instance-interface-info 2` command.
ENCS 5400 Switch LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) on ENCS 5400.

- Understanding LLDP, on page 189
- Enabling and Disabling LLDP, on page 189
- Configuring LLDP Characteristics, on page 190

Understanding LLDP

Link Layer Discovery Protocol (LLDP), is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP is unidirectional, operating only in an advertising mode. LLDP does not solicit information or monitor state changes between LLDP nodes. LLDP periodically sends advertisements to a constrained multicast address. Devices supporting LLDP can send information about themselves while they receive and record information about their neighbors. Additionally, devices can choose to turn off the send or receive functions independently. Advertisements are sent out and received on every active and enabled interface, allowing any device in a network to learn about all devices to which it is connected.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

Enabling and Disabling LLDP

To globally enable LLDP:

```bash
configure terminal
switch
lldp run
commit
```

To globally disable LLDP:

```bash
configure terminal
switch
```
no lldp run
commit

LLDP is enabled by default on all supported interfaces. You must enable LLDP globally to allow a device to send LLDP packets. However, no changes are required at the interface level.

You can configure the interface selectively not to send and receive LLDP packets with the `no lldp transmit` and `no lldp receive` commands.

To enable LLDP on an interface:

```
configure terminal
switch
    interface gigabitEthernet1/0
    lldp transmit
    lldp receive
    commit
```

### Configuring LLDP Characteristics

To specify an interval at which LLDP packets are sent:

```
configure terminal
switch
    lldp timer 135
    commit
```

To display LLDP statistics:

```
nfvvis# show switch lldp statistics

<table>
<thead>
<tr>
<th>PORT</th>
<th>TX FRAMES</th>
<th>RX FRAMES</th>
<th>RX DISCARDED</th>
<th>TX DISCARDED</th>
<th>TLVS UNRECOGNIZED</th>
<th>RX AGEOUTS</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/2</td>
<td>792</td>
<td>756</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/3</td>
<td>791</td>
<td>756</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/6</td>
<td>792</td>
<td>756</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/7</td>
<td>791</td>
<td>756</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

To display information about LLDP neighbors:

```
nfvvis# show switch lldp neighbors

SYSTEM
INDEX PORT DEVICE ID PORT ID NAME CAPABILITIES TTL
-----------------------------------------------------------------------------------
1 gi1/1 00:1a:6c:81:f0:80 Gi1/0/31 SW-026 Bridge 93
2 gi1/6 2c:0b:e9:3c:89:00 Gi1/0/5 Switch Bridge 119
```
Secure Overlay and Single IP Configuration

- Restrictions, on page 191
- Supported Event Notifications, on page 192
- Secure Overlay over WAN, on page 192
- Single IP Address with Secure Overlay, on page 193
- Single IP Address Without Secure Overlay, on page 193

Restrictions

- Secure overlay is supported on:
  - IPSec IKEv2
  - IPv4
  - Pre-shared-key authentication
  - IKE cipher: aes128-sha1-mopd1536
  - ESP cipher: aes128-sha1
  - Local-system-ip unique to each NFVIS device
  - Local-bridge: Wan-br and Wan2-br

- When the guest VM is deployed and single-ip-mode is configured:
  - The configuration requests can be sent in one NETCONF commit.
  - If the configuration requests are sent separately, then commit single-ip-mode creation request first. NFVIS releases the WAN IP address only after the guest device is deployed.
  - If you commit the guest device deployment configuration first, commit the single-ip-mode configuration request before the guest device is active. The guest VM will have conflicting IP address if the commit is delayed.

- When the guest device and single-ip-mode configurations need to be deleted:
  - The two deletion requests can be sent in one NETCONF commit.
  - If the two deletion requests are sent separately, commit the guest device deletion first.
Supported Event Notifications

The following event types are supported

- SECURE_OVERLAY_CREATING
- SECURE_OVERLAY_UP
- SECURE_OVERLAY_DOWN
- SECURE_OVERLAY_DELETE
- SECURE_OVERLAY_ERROR
- SINGLE_IP_START
- SINGLE_IP_ACTIVE
- SINGLE_IP_FAILOVER_START
- SINGLE_IP_FAILOVER_COMPLETE
- SINGLE_IP_DELETE
- SINGLE_IP_ERROR

Secure Overlay over WAN

An overlay is a virtualized network layer on top of the physical network with the support of its infrastructure to provide additional security to the network. IPSec is a framework with protocols and algorithms to provide secured data transmission over unprotected or untrusted networks. IPSec secure tunnel is created between two networks to ensure virtual private network communication.

Secure overlay in NFVIS allows IPSec tunnel establishment between NFVIS supporting the vBranch platform and a VPN device in the headend orchestrator. This feature manages traffic only between the headend orchestrator and the vBranch platform. The orchestrator connects to NFVIS through the system IP address and manages NFVIS over the secure tunnel.

NFVIS can be configured with WAN IP, static IP or DHCP IP. NFVIS calls home PnP server, which pushes NFVIS Day-0 configurations including secure overlay configurations. NFVIS establishes IPSec connection between NFVIS and the headend management hub which has IPSec VPN configured. On NFVIS side, the tunnel end point has NFVIS local system IP address. When IPSec tunnel is up Network Services Orchestrator (NSO) solution, can connect to the NFVIS system throught the system IP address and manage NFVIS throught the IPSec tunnel.

To create secure overlay with the management IP address as local system IP address:

```
configure terminal
secure-overlay myconn local-system-ip-addr 10.0.0.1 local-system-ip-bridge int-mgmt-net
remote-interface-ip-addr 172.16.10.1 remote-system-ip-addr 10.0.0.2 local-psk Admin remote-psk Admin
commit
```
Secure Overlay APIs and Commands

<table>
<thead>
<tr>
<th>Secure Overlay APIs</th>
<th>Secure Overlay Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/secure-overlays</td>
<td>secure-overlay</td>
</tr>
<tr>
<td>/api/operational/secure-overlays</td>
<td></td>
</tr>
</tbody>
</table>

**Single IP Address with Secure Overlay**

After secure overlay over WAN is established the orchestrator sends requests to configure single-ip-mode and deploy the guest vm that takes the public IP address.

NFVIS deploys a VM with specific bootstrap and Day-0 configurations. NFVIS notes the IPSec tunnel and releases the public IP address. The VM takes the public IP address when it is active. NFVIS sets up IPSec tunnel again with the remote management hub. After the IPSec tunnel is established, the orchestrator solution can connect with NFVIS through its system IP address and manage NFVIS over the IPSec tunnel.

NFVIS reclaims the WAN IP address if the guest device has:
- Failed to deploy.
- Error state.
- Stopped.

NFVIS releases the WAN IP address if the guest device has:
- Deployed.
- Started.

**Single IP and Secure Overlay APIs**

<table>
<thead>
<tr>
<th>Secure Overlay APIs</th>
<th>Secure Overlay Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/single-ip-mode</td>
<td>single-ip-mode</td>
</tr>
<tr>
<td>/api/operational/single-ip-mode</td>
<td></td>
</tr>
</tbody>
</table>

**Single IP Address Without Secure Overlay**

---

**Note**

This feature is only supported for wan-br in this NFVIS 3.10.1 release.

To reach NFVIS when secure overlay is not configured, you must first configure the guest device and manage IP addressing. The rest of the functionality, switching IP address between NFVIS and the guest device is the same.
Single IP Address Without Secure Overlay
Dual WAN Support

Dual WAN support is introduced to provide a backup link to NFVIS connectivity in case the primary link is down. NFVIS connectivity can be maintained through multiple ports in case connectivity is lost over the primary WAN or management port. For all supported platforms on NFVIS, IP configuration is moved under bridges and user generated bridges can specify IP or DHCP connectivity.

Starting from NFVIS 3.10.1 release, a second WAN bridge configured with DHCP by default is supported on ENCS 5000 series platform.

- Bridge IP Configurations, on page 195
- Dual WAN Bridge and DHCP Toggle, on page 196

Bridge IP Configurations

Both default bridges and user generated bridges contain IP/DHCP configuration to allow IP configurations on any port. NFVIS provides two ports by default for zero touch deployment, with dhclient actively requesting DHCP IP configurations.

Each bridge can be configured with:

- IPv4 DHCP
- Static IPv4
- IPv6 DHCP
- IPv6 SLAAC
- Static IPv6
- VLAN tag

Except for ENCS 5400 management port configuration which continues to remain under system settings mgmt, use the Bridge APIs and Commands to enable IP configurations and move away from the previous System Settings APIs and commands.

On ENCS5100 use system settings mgmt for management IP configuration and bridges bridge lan-br for LAN IP configuration.
Restrictions for Bridge IP Configurations

IPv4:

• IPv4 DHCP can only be configured on one bridge at a time
• Cannot configure IPv4 DHCP on any bridge if system wide default gateway is configured
• Cannot configure system settings default-gw if IPv4 DHCP is configured on any bridge.

IPv6:

• IPv6 DHCP can only be configured on one bridge at any time, and cannot be applied on any bridge if IPv6 SLAAC is applied on any bridge or if system wide IPv6 default gateway is configured.
• IPv6 SLAAC can only be configured on one bridge at any time, and cannot be applied on any bridge if IPv6 DHCP is applied on any bridge or if system wide IPv6 default gateway is configured.
• Cannot configure system settings default-gw-ipv6 if IPv6 DHCP or IPv6 SLAAC is configured on any bridge.

Dual WAN Bridge and DHCP Toggle

This feature is supported only on ENCS 5000 series devices.

In zero touch deployment, NFVIS requests for IPv4 assignments through DHCP for two WAN ports. A second WAN bridge and network are default configurations and GE0-1 is attached to the WAN2 bridge. NFVIS toggles between the two default WAN bridges activating dhclient on any one of the WAN bridges at a time, for 30 seconds interval. The toggling stops as soon as a WAN bridge is assigned with an IP address through DHCP. The bridge with the assigned IP address is considered an active WAN bridge and DHCP configurations are applied to that bridge. dhclient is deactivated for the remaining bridge.

If neither bridge is assigned with an IP address through DHCP, you can terminate DHCP toggle by terminating zero touch deployment from NFVIS. DHCP is then applied to the WAN bridge and dhclient is activated.

After the zero touch deployment, the toggle feature is terminated. To backup NFVIS connectivity, you can add static IP address to the other WAN bridge, and setup static routing. You cannot configure default gateway, as the system default gateway is set through DHCP. You can also configure static IP address on both WAN bridges and setup static routing.

Restrictions

• The DHCP toggle behaviour is not supported in upgrade from NFVIS 3.9.x releases.
• Does not support active or standby WAN bridges. NFVIS does not detect connectivity failure from active WAN bridge to switchover to another WAN bridge. In case connectivity fails on the primary WAN bridge, connectivity through other WAN bridge is established only if static IP is enabled and static routing is configured.
• Does not support IPv6.
• If wan2-br is the primary WAN bridge, you must remove DHCP from wan2-br to apply default gateway from static IP configurations.
Configuring System Shell-Access

To debug issues in the NFVIS system, you must get into linux shell through a procedure of challenge-response between the device and the Cisco server (cid.cisco.com). Only an admin user can start this procedure.

To configure system shell-access:

1. Open cid.cisco.com in your web browser.
2. Login with Cisco username and password assigned to you.
3. Copy and paste the challenge string from the NFVIS system generated from `system shell-access` command.
4. Enter the OTP.
   A response string is generated by the system.
5. Enter the response string onto the NFVIS command page.
6. Enter the admin password. You get shell-access if password is validated
7. 8. To exit shell-access enter Exit.
Switch Port Security

- Switch Port Security, on page 201

### Switch Port Security

**Note**

Always shutdown interface before port security configurations.

Port security is not supported on port channel interfaces.

**Restrictions**

- Dynamic secure mac address are not retained over reboot and only delete-on-reset secure mode is supported.

- Static mac address must be set after port is in shutdown, and other port security commands are configured and enabled.

- Only ethernet ports are supported for port security configuration.

- Does not support trap and `show switch interface port-security` command does not show information about trap.

- If port-security violation shutdown mode is configured on a secure port and violation traffic is received, the port changes to error state. A manual interface shutdown and no shutdown is required to recover the port.

**Configuring Port Security**

To configure port security:

1. Shutdown the interface:

    ```
    configure terminal
    switch
    interface gigabitEthernet 1/1
    shutdown
    commit
    ```
2. Disable port security:

```
configure terminal
switch
  interface gigabitEthernet 1/1
  no port-security enable
commit
```

3. Configure max mac address:

```
configure terminal
switch
  interface gigabitEthernet 1/1
  port-security max 5
commit
```

4. Configure violation handling:

```
configure terminal
switch
  interface gigabitEthernet 1/1
  port-security violation discard
commit
```

5. Enable port security:

```
configure terminal
switch
  interface gigabitEthernet 1/1
  port-security enable
commit
```

6. Add static secure mac address:

```
configure terminal
switch
  mac address-table static 18:65:90:cb:e6:08 vlan 1 interface gigabitEthernet 1/1 secure
commit
```

7. Restart the disabled interface:

```
configure terminal
switch
  interface gigabitEthernet 1/1
  no shutdown
commit
```

8. Use `show switch interface port-security` to verify the configuration:

```
nfvis# show switch interface port-security
MAC
VIOLATION ADDRESS MAX MAC
PORT STATUS LEARNING HANDLING COUNT ADDRESS
-----------------------------------------------
1/0 Disabled Delete-On-Reset Discard 0 0
1/1 Enabled Delete-On-Reset Discard 1 5
1/2 Disabled Delete-On-Reset Discard 0 0
1/3 Disabled Delete-On-Reset Discard 0 0
```
1/4 Disabled Delete-On-Reset Discard 0 0
1/5 Disabled Delete-On-Reset Discard 0 0
1/6 Disabled Delete-On-Reset Discard 0 0
1/7 Disabled Delete-On-Reset Discard 0 0

9. Use `show switch mac addr-table` command to check static configured or dynamic learnt secure mac addresses:

```
nfvis# show switch mac addr-table
VLAN MAC ADDRESS PORT TYPE
----------------------------------------
1 18:65:90:cb:e6:08 gi1/1 secure
```

**Note**

If traffic with a secure MAC address that is configured on one secure port attempts to access another secure port in the same VLAN, ENCS switch port security responds to the violation by discarding the traffic always.
PART II

NFVIS Functionality Changes for Cisco SD-WAN Cloud OnRamp for Colocation

• Default System Configurations, on page 207
• NFVIS Integration with Docker Container Lifecycle, on page 211
• NFVIS Integration with vManage, on page 217
• Enhancements to VM Image Packaging, on page 221
Default System Configurations

LACP

In Cisco SD-WAN Cloud OnRamp for Colocation solution, the Link Aggregation Control Protocol (LACP) is enabled for the management port channel. The management port channel is created by default using the Ethernet links (eth0-1 and eth0-2). To ensure the port channel configuration on the management switch side is reachable, run the `support ovs appctl bond-show mgmt-bond` command and ensure OOB switch ports that are connected to the switch has the following port-channel configuration.

```
! interface Port-channel1
   switchport mode access
! interface GigabitEthernet1/0/6
   switchport mode access
   channel-group 1 mode passive
! interface GigabitEthernet1/0/7
   switchport mode access
   channel-group 1 mode passive
!
```

DHCP

In Cisco SD-WAN Cloud OnRamp for Colocation solution, DHCP is enabled by default on the management port channel. Once the DHCP server is up, the host gets the DHCP IP address on management internal port.

Sticky DHCP

Note

Sticky DHCP configurations are optional.

Configure the DHCP servers to get sticky DHCP IP address. The DHCP client identifier is the serial number on the CSP device.

The DHCP server configuration on a Linux server is:

```
host
{
    option dhcp-client-identifier "WZP22060AUR";
    fixed-address 10.20.0.2;
}```
option routers 10.20.0.1;
option domain-name-servers 198.51.100.9;
option domain-name "cisco.com";
option subnet-mask 255.255.0.0;
}

The DHCP server configuration on IOS is:

ip dhcp pool P_112
host 10.0.0.2 255.255.0.0
client-identifier 4643.4832.3133.3256.3131.48
default-router 10.0.0.1
dns-server 10.0.0.1

Here 10.0.0.2 is the sticky DHCP IP address. Use `debug ip dhcp server packet` command to find out the client identifier.

Static IPv4

To troubleshoot issues with DHCP configurations, configure static IPv4 on the management port channel:

```plaintext
configure shared
vm_lifecycle networks network int-mgmt-net subnet int-mgmt-net-subnet address 105.20.0.0
gateway 105.20.0.1 netmask 255.255.255.0 dhcp false
system settings domain cisco.com
system:system settings dns-server 209.165.201.20
system:system settings ip-receive-acl 0.0.0.0/0
action accept
priority 100
service scpd
commit
```

Since vManage is the controller in this solution, **Configure shared** writes to candidate database (CDB) which will keep the device config in sync with vManage.

---

**Note**

`Configure shared` is only applicable to static ip configurations. Any other configurations done manually using either confd cli or netconf or rest api, will be removed by vManage as NFVIS is a vManaged device in this solution.

In NFVIS NetworkHub image, networks are automated and user should not create, delete or modify networks. You can reset the host server to default configurations using the `factory reset all` command.

- SRIOV Support, on page 208

---

**SRIOV Support**

SR-IOV is statically enabled on NFVIS Cisco SD-WAN Cloud OnRamp for Colocation image with a CSP 5444 Product Identifier (PID).

- SRIOV is enabled by default on ethernet ports eth1-1, eth1-2, eth4-1, eth4-2 as Niantec NIM cards are placed in slots 1 and 4.
- SR-IOV is enabled only on Niantic NICs and onboard Niantics does not support SR-IOV.
• Thirty two virtual functions are created on each PNIC. If the NIC is connected to 1G, two virtual functions are created.

• Virtual Ethernet Port Aggregator (VEPA) mode is enabled.

• The naming convention is: <interface name>-SRIOV-1,<interface name>-SRIOV-2,<interface name>-SRIOV-3,<interface name>-SRIOV-4.

• Fortville NICs are used to create port channels for OVS data traffic and HA sync between the VM’s.
SRIOV Support
NFVIS Integration with Docker Container Lifecycle

Docker container lifecycle infrastructure is developed in NFVIS for Cisco SD-WAN Cloud OnRamp for Colocation solution. Container lifecycle APIs are developed to bring up docker containers.

- Cisco Colo Manager, on page 211

Cisco Colo Manager

Cisco colo manager (CCM) is a software stack managing switches in colo. In the Cisco SD-WAN Cloud OnRamp for Colocation solution, CCM is hosted on NFVIS software in a Docker container. CCM is hosted on the CSP devices along with VNFs and there are no dedicated CSP devices for hosting CCM. CCM is used to configure and provision PNFs (switches) in this solution.

Cisco colo manager (CCM) is bundled along with the Cisco NFVIS software which is used as the base virtualization infrastructure software running on the compute platform. The NFVIS software provides programmable Rest and netconf APIs and an orchestrator can use these APIs to configure and monitor the system, instantiate virtual network functions and configure the VNF networks and service chains. As part of colo provisioning for the orchestrator, vManage selects one device in the colo and sends netconf action command to bring up the CCM container. The CCM container is connected to the colo management network. This management network is used to transfer files and images into and out of the systems. This network will not be used for the normal customer data traffic.

CCM State Transitions from the Host Side

vManage brings up CCM on one of the CSP devices in the Network Hub solution. CCM state transitions are seen on the host side, using the container life-cycle model's state operation.

The CCM state on the host side has the following states:

- Starting: The container is booting up and starting
- Healthy: The container health monitoring has started and container is healthy. The container operational state is set to healthy
- Unhealthy: If the CCM does not boot properly, the CCM container is not usable and needs to be recovered. CCM in unhealthy state can be due to docker daemon not running, CCM is not configured with correct management IP address, gateway or CCM cannot respond to ping.
The starting state can only be seen when the container is brought up or re-spun. Healthy and unhealthy states can transition to each other during the lifetime of the container. A notification is also sent whenever the CCM state changes.

The CCM container state is tracked through container life-cycle model as one of the containers. This is not CCM-state or CCM-status oper. The state for container named ColoMgr is used for CCM state transitions.

<table>
<thead>
<tr>
<th>State</th>
<th>Action/config can be pushed</th>
<th>config status queried</th>
<th>oper model on host</th>
<th>notification for CCM state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Unhealthy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Healthy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

PNF device list is sent from vManage to the NFVIS hosting CCM when CCM is in healthy state.

To verify CCM state, when Colo Manager crashes on a CSP device use `support show container` command:

```
CSP# support show container
Possible completions:
  docker-container-ls Lists all containers
  docker-info Lists docker daemon info
  docker-inspect Inspect container or volume
  docker-volume-ls Lists all volumes
  dump Dumps all container related info
```

**CCM Notifications**

CCM health check sends CCM state transitions to vManage notification stream.

You can view the CCM event notifications using the `show notification stream vmanageEvent` command.
### CCM Recovery

When CCM is up, the Catalyst 9000 series switches are onboard successfully and CCM is restarted on the same or different CSP, the CCM recovery is initiated.

vManage brings down CCM and then brings it up again. vManage sends the device list with passwords for the switches along with all the service configurations. CCM then uses these configurations to sync with the device.

Recovery flag for device action list - false for day0, true for recovery (mandatory).

Static IP change for device action list - IP addresses for devices is sent all the time - day0 and recovery.

### Support Commands

To verify the CCM version use `support show ccm-version`:

```
CSP# support show ccm-version
Cisco Colo Manager (CCM)
Version 0.0.1-150
Build date Tue 06 Nov 2018 09:09:28 AM UTC
```

To verify the firewall state use `support show firewall`:

```
CSP# support show firewall
```
Possible completions:
- list-forward-ports  Lists all port forwarding rules
- state             Lists firewalld daemon status

To display information about OVS switch use **support ovs vsctl show**:

CSP# support ovs vsctl show
Possible completions:
| <cr>
CSP2# support ovs vsctl show
107a6588-62f1-411f-b5da-fa0fd39f2500
Bridge ovs-data-br
  Port bond-bond_data
    tag: 1
    Interface "eth2-3"
    Interface "eth2-4"
  Port ovs-data-br
    Interface ovs-data-br
      type: internal
Bridge ovs-ha-br
  Port bond-bond_ha
    tag: 1
    Interface "eth2-2"
    Interface "eth2-1"
  Port ovs-ha-br
    Interface ovs-ha-br
      type: internal
Bridge int-mgmt-net-br
  Port colo-mgmt
    Interface colo-mgmt
      type: internal
Port mgmt-bond
  Interface "eth0-2"
  Interface "eth0-1"
Port int-mgmt-net-br
  Interface int-mgmt-net-br
    type: internal

ovs_version: "2.5.2"

To display the list of NFVIS system settings use **show system:system settings-native**:

```plaintext
system:system settings-native mgmt ip-info interface colo-mgmt
system:system settings-native mgmt ip-info ipv4_address 192.168.30.163
system:system settings-native mgmt ip-info netmask 255.255.255.0
system:system settings-native mgmt ip-info link-local ipv6 address ::
system:system settings-native mgmt ip-info link-local ipv6 prefixlen 0
system:system settings-native mgmt ip-info global ipv6 address ::
system:system settings-native mgmt ip-info global ipv6 prefixlen 0
system:system settings-native mgmt ip-info mac_address b2:5d:28:aa:f1:96
system:system settings-native mgmt ip-info mtu 1500
system:system settings-native mgmt ip-info txqueuelen 1000
system:system settings-native mgmt stats rx_packets 7140693
system:system settings-native mgmt stats rx_bytes 767558248
system:system settings-native mgmt stats rx_errors 0
system:system settings-native mgmt stats rx_dropped 2
system:system settings-native mgmt stats rx_overruns 0
system:system settings-native mgmt stats rx_frame 0
system:system settings-native mgmt stats tx_packets 5259073
system:system settings-native mgmt stats tx_bytes 1008512311
system:system settings-native mgmt stats tx_errors 0
system:system settings-native mgmt stats tx_dropped 0
system:system settings-native mgmt stats tx_overruns 0
system:system settings-native mgmt stats tx_carrier 0
```
system:system settings-native mgmt stats tx_collisions 0
system:system settings-native domain NA
system:system settings-native dns nameserver1 0.0.0.0
system:system settings-native dns nameserver2 0.0.0.0
system:system settings-native dns nameserver3 0.0.0.0
system:system settings-native hostname CSP2
system:system settings-native gateway ipv4_address 192.168.30.1
system:system settings-native gateway interface colo-mgmt
system:system settings-native gateway-ipv6 ipv6_address ::
system:system settings-native gateway-ipv6 interface NA
system:system settings-native trusted-source [ "not set" ]
system:system settings-native source-interface 0.0.0.0

To display information about a bond use **support ovs appctl bond-show mgmt-bond**

CSP2# **support ovs appctl bond-show mgmt-bond**

<table>
<thead>
<tr>
<th>bond_mode: balance-slb</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond may use recirculation: no, Recirc-ID : -1</td>
</tr>
<tr>
<td>bond-hash-basis: 0</td>
</tr>
<tr>
<td>updelay: 0 ms</td>
</tr>
<tr>
<td>downdelay: 0 ms</td>
</tr>
<tr>
<td>next rebalance: 252 ms</td>
</tr>
<tr>
<td>lacp_status: negotiated</td>
</tr>
</tbody>
</table>

active slave mac: 00:fc:ba:d7:39:86(eth0-1)

slave eth0-1: enabled
  active slave
  may_enable: true
  hash 242: 8 kB load

slave eth0-2: enabled
  may_enable: true

To display the IP routing statistics use **support show route**:

CSP# **support show route**

<table>
<thead>
<tr>
<th>Kernel IP routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
</tr>
<tr>
<td>default</td>
</tr>
<tr>
<td>172.16.255.22</td>
</tr>
<tr>
<td>192.168.30.0</td>
</tr>
</tbody>
</table>
NFVIS Integration with vManage

In the Cisco SD-WAN Cloud OnRamp for Colocation solution a colo is a stack of computing and networking fabric which brings up multiple networking functions and service chains them to connect branch users or endpoints to hybrid cloud or data center. vManage is used as the orchestrator to provision the devices in a colo. This solution can be deployed in multiple locations where each colo is independent and unaware of other colos in the same site or across sites.

- Establishing DTLS Tunnel with vManage, on page 217
- NFVIS Notifications, on page 219
- Stats for Host and VM, on page 219
- System CLI, on page 219
- NFVIS Local Portal, on page 219
- Core Allocation for Host and CCM, on page 219

Establishing DTLS Tunnel with vManage

Before you begin

To establish a DTLS channel with vManage, vDaemon is integrated on NFVIS

Note
The device is vManaged and hence any configurations done out-of-band is overwritten by vManage. The show commands continue to work in the same way.

Note
If CSP devices are already added into PnP Connect, skip the instructions that are mentioned from steps 1 to 5 in topic, and perform instructions from step 6.

Note
If CSP devices are already added into vManage, perform instructions from step 13.

Step 1
Get access to PnP devices and log into Plug and Play Connect.
Step 2  Create a virtual account. See the Plug and Play Connect Configuration Guide for more information about creating a virtual account.

Step 3  In the virtual account, create a vbond controller.

**Note** Only one vbond controller profile is allowed in a virtual account.

Step 4  In the Add Controller Profile window, provide information about Organization Name, vbond IP address, root CA, and other information. Click **Next**.

Step 5  Go to the **Devices** tab, add your device by using PID and serial number. Assign the vbond profile that is created in step 3 to the device.

**Note** You can only choose and add CSP 5444, X1 and X2 devices.

If the switch and CSP devices are already added into PnP Connect, skip steps 1 to 5. Go to the next step.

Step 6  Your device should have DNS servers with connections to Plug and Play Connect.

Step 7  Verify PnP status to determine if redirection is successful. Use the `nfvis# show pnp status` command to determine PnP status.

Step 8  Go to Plug and Play Connect screen and verify if status is displayed as "Redirect Successful".

Step 9  To ensure that VPN configuration are present on NFVIS, use the `nfvis# show running-config vpn` command.

Step 10  To ensure that Organization name and vbond IP address have been configured, use the `nfvis# show running-config viptela-system:system` command.

Step 11  To ensure that root ca have been installed, use the `nfvis# show control local-properties root-ca-chain-status` command.

If the switch and CSP devices are already added into vManage, skip the next step and perform instructions from step 13.

Step 12  Upload WAN edge list into vManage. For more information, see Add Network Hub Devices into vManage in Cisco SD-WAN Cloud OnRamp for Colocation Solution Guide.

Step 13  In vManage, go to **Configuration > Network Hub** screen. Create a new cluster by clicking the **Configure & Provision Cluster** button. For more information, see Create and Activate Network Hub Cluster in Cisco SD-WAN Cloud OnRamp for Colocation Solution Guide.

Step 14  After activating the cluster, get the token that you had noted while adding devices into vManage, and then request an activate command on NFVIS. Use the following NFVIS command:

**Example:**

```
nfvis# request activate chassis-number CSP-5444-X2-FCH2118V0CY token f3117c35c3206f4adfab5ced0d57db44```

Step 15  Verify that your system IP address has been configured, VNFs to be run on CSPs such as CSR 1000v, vEdge are already installed, and connections are working. For verification, use the following NFVIS commands:

**Example:**

```
nfvis# show control local-properties certificate-status
nfvis# show control local-properties system-ip
nfvis# show control connections```

Step 16  If connections are not running, use the following NFVIS history command to debug:

**Example:**

```
nfvis# show control connections-history```
NFVIS Notifications

You can view the NFVIS notifications using the `show notification stream viptela` command. The NFVIS notifications are available at Syslog Messages, on page 147 and the same notifications are sent to viptela stream.

Stats for Host and VM

The stats for cpu/mem/disk/interface are collected periodically and the files are compressed and stored in the device in the required format for vManage. vManage collects these log files periodically and deletes the older set of log files.

System CLI

In NFVIS integration with vManage, Viptela system model is loaded into NFVIS software, due to which the existing NFVIS `system` commands become `system:system` commands.

Example:

```
show system:system status
```

NFVIS Local Portal

All the configurations from the local portal are blocked as the admin role is changed to view-only for Cisco SD-WAN Cloud OnRamp for Colocation solution. The admin can continue to use the NFVIS portal for troubleshooting and serviceability, but not for provisioning or configuring any functionalities.

Core Allocation for Host and CCM

The host CPU reserve based on the hardware core is:

- less than 12 cores: 1 pCPU for NFVIS 1pCPU=1core
- 16 cores: 2 pCPUs for NFVIS
- greater than 16 cores: 4 pCPUs for NFVIS
CHAPTER 30

Enhancements to VM Image Packaging

The Cisco Enterprise NFVIS VM image packaging tool, nfvpt.py, is enhanced to support functionality required for Cisco SD-WAN Cloud OnRamp for Colocation solution.

- NFVIS Specific Enhancements, on page 221
- Cisco SD-WAN Cloud OnRamp for Colocation Packaging Enhancements, on page 222
- VM Packaging Parameters, on page 223
- VM Packaging Utility Usage Examples, on page 224
- Packaging a VM, on page 224

NFVIS Specific Enhancements

Use pack_dir option if the *.tar.gz already exists and you want to modify the bootstrap configuration file or image_properties.xml manually.

The following parameters are added as part of the NFVIS specific enhancements:

- --pack_dir <DIR> PACK
  package all files in directory

Resources:

- --vnic_names VNIC_NAMES
  list of vnic number to name mapping in format
  number:name example --vnic_names
  1:GigabitEthernet2,2:GigabitEthernet4

Usage

Follow the steps to change a single line in day-0 configuration file or add a single option in image_properties.xml:

1. Get the working VM packaging image - isrv*.tar.gz.
2. Extract the contents - tar -xvf isrv*.tar.gz.
3. Modify the file contents as required.
4. `nfvpt.py --pack_dir current-working-dir-with-files -i isrv.qcow2 -o isrv.tar.gz`

## Cisco SD-WAN Cloud OnRamp for Colocation Packaging Enhancements

The following parameters are the enhancements specific to SD-WAN:

```
--json JSON      Provide JSON input for bootstrap variables; mutually
                 exclusive with custom and bootstrap configs
--multi_use      Add options for use in multiple use-cases
--app_vendor APP_VENDOR
                 Application Vendor e.g. Cisco, Juniper etc
--bootstrap BOOTSTRAP
                 Every bootstrap file should be a different option HA
                 packaging format: --bootstrap mount_point:<value>,file
                 :<file2mount>[,<attrib>:<value>] mount_point:<value>
                 and file:<file2mount> are mandatory followed by one or
                 more attributes in the format <attrib>:<value> Legacy
                 format: --bootstrap file1,file2... See usage.txt for
                 more details
--ha_package     enable HA packaging
--mgmt_vnic MGMT_VNIC
                 VM management interface identifier
HA options:
--ha_capable
--ha_vnic HA_VNIC VM HA vnic CSV list
Custom Properties:
--custom CUSTOM  custom properties format: --custom ["propattr_"<attr>:
                 <value>],key:<value>,[keyattr_<attr>:<value>],type:<va
```
VM Packaging Parameters

The table lists the new parameters that can be passed to the nfvpt.py command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>json</td>
<td>Optional</td>
<td>Provide JSON input for bootstrap variables. It's mutually exclusive with</td>
</tr>
<tr>
<td></td>
<td></td>
<td>custom and bootstrap configs</td>
</tr>
<tr>
<td>multi_use</td>
<td>Optional</td>
<td>option for use in multiple use-cases</td>
</tr>
<tr>
<td>ha_package</td>
<td>Optional</td>
<td>enable HA packaging</td>
</tr>
<tr>
<td>mgmt_vnic</td>
<td>Optional</td>
<td>VM management interface identifier</td>
</tr>
<tr>
<td>pack_dir</td>
<td>Optional</td>
<td>package all files in directory</td>
</tr>
<tr>
<td>app_vendor</td>
<td>Required</td>
<td>Application Vendor e.g. Cisco, Juniper</td>
</tr>
<tr>
<td>ha_capable</td>
<td>Optional</td>
<td>For HA capability</td>
</tr>
<tr>
<td>vnic_names</td>
<td>Optional</td>
<td>list of vnic number to name mapping in format number:name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--vnic_names --vnic_names 1:GigabitEthernet2,2:GigabitEthernet4</td>
</tr>
</tbody>
</table>
VM Packaging Utility Usage Examples

Given below are the contents of the file `nfvis_vm_packaging_utility_examples.txt`:

Example 1: Usage for Palo Alto Firewall

```
nfvpt.py -o PA_L3_HA -i PA-VM-KVM-8.0.5.qcow2 --json d.json -t firewall -n "PA FIREWALL" -r 8.0.5 --app_vendor PA --monitor true --ha_package
```

Example 1: Usage for Asav

```
nfvpt.py -i foo.qcow2 -o asav.tar.gz --json pa1.json --app_vendor cisco -t firewall -r 10 --optimize true -n asav --monitored true --ha_package --ha_capable
```

Example 1: Usage for csr

```
nfvpt.py --ha_package --pack_dir /data/intdatastore -i csr1000v-universalk9.16.09.01.qcow2 -o csr1000v-universalk9.16.09.01-ha.tar.gz
```

Packaging a VM

The following steps shows how to package a bundled VM image, bootstrap files and metadata into an archive:

1. Create a json file using `gen_json.py` tool. The `gen_json.py` needs a pattern that matches bootstrap files as an option. `gen_json.py --help` shows all the details about the options. Redirect the output of `gen_json.py` into a json file.

```
gen_json.py --g "boot*,ios*" --ha > temp.json
```

Include `--ha` option if the packaging is for HA. Include `--multi_use` option if the VM is a part of a service chain.

2. The `temp.json` file has two arrays - Userinput and SysGen. Userinput and SysGen are variables from bootstrap files which were tokenized. By default all the variables are included in Userinput array. The system generated variables should be moved to SysGen array. vManage generates some of these variables like MGMT and DATA IP addresses from the pool provided in the cluster creation on vManage. All other variables like DNS_SERVER, VM password etc. are user inputs at the VM/servicechain provisioning.

Example:

```
interface G0/1
ip address ${MGMT_PRIM} <-- variable
```

3. After making changes to the json file you can package the VM with the script - `nfvpt.py`.

```
nfvpt.py -i <qcow file> -o <tar file name> --json <json file> --app_vendor cisco -t firewall -r 10 --optimize true -n asav --monitored true --ha_package --ha_capable
```

The tool creates a .tar.gz file with the name you have provided.