CONTENTS

Preface ix

About Cisco Enterprise NFVIS 1

Installing Cisco Enterprise NFVIS Using the KVM Console 5

Installing Cisco Enterprise NFVIS on Cisco UCS E-Series Servers 8

Sample Configuration on the Cisco ISR Router to Bring Up a Cisco UCS E Server 10

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.5.1
CHAPTER 3

Setting Up System Configuration 17

Default System Configuration on the Cisco ENCS 17
Default System Configuration on the Cisco UCS C220 M4 Server and Cisco CSP 2100 19
Default System Configuration on the Cisco UCS E-Series Servers 20

Setting Up Initial Configuration 20
Configuring VLAN for NFVIS Management Traffic 26
Configuring System Routes 27

User Roles and Authentication 28
Rules for User Passwords 28
Creating Users and Assigning Roles 28
Configuring Minimum Length for Passwords 29
Configuring Password Lifetime 30
Deactivating Inactive User Accounts 30
Activating an Inactive User Account 30

Certification 31
Secure Copy Command 31

Configuring the IP Receive ACL 32
Port 22222 and Management Interface ACL 33

Configuring Your Banner and Message of the Day 33
Setting the System Time Manually or With NTP 34
Enabling or Disabling the Portal Access 35

Configuring System Logs 36
Network File System Support 37

Secure Boot of host 38
CIMC Control 39
CIMC Access using NFVIS 39

BIOS-CIMC Update 39

NFVIS Password Recovery 40

DPDK Support 40
Backup and Restore NFVIS and VM Configurations 41
CHAPTER 4  
Cisco Network Plug-n-Play Support  
PnP Discovery Methods  
Configuring PnP Discovery Methods  
PnP Action  

CHAPTER 5  
VM Life Cycle Management  
Workflow of VM Life Cycle Management  
Uploading VM Images to an NFVIS Server  
VM Bootstrap Configuration Options with a VM Deployment  
OpenStack Configuration Drive Support for Third Party VMs  
Performing Resource Verification  
Configuring Management IP Address  
VM States  

CHAPTER 6  
VM Deployment Scenarios  
Registering VM Images  
Single VM Deployment  
Steps for Deploying a VM  
Service Chaining of VMs  
Service Chaining with two VM Images  
Steps for Service Chaining with Two VM Images  
Service Chaining of Multiple VMs with Windows or Linux Servers  
Steps for Service Chaining of Multiple VMs with Windows or Linux Servers  

CHAPTER 7  
SPAN Session or Port Mirroring  
About SPAN Sessions  
Configuring SPAN Sessions  
Configuration Examples for SPAN Session Scenarios  
Example: SPAN Session Traffic on a Physical Interface  
Example: SPAN Session Traffic on a LAN SRIOV  
Example: SPAN Session Traffic on a VLAN  

CHAPTER 8  
Configuring Packet Capture  

CHAPTER 9  VM Image Packaging  71
VM Image Packaging Utility  71
Contents  71
Usage  72
NFVIS Specific Enhancements  76
VM Packaging Utility Usage Examples  77
Standard VM Image Packaging  78
Generating a VM Package  79
Appendix  79
VM Image Package Files  79
Package Manifest File  80
Bootstrap Configuration File  80
VM Image Properties File  81
Example: Package.mf  85
Example: Image Properties  86
Example: Bootstrap Configuration File  87
Image Properties Template File  87

CHAPTER 10  Upgrading Cisco Enterprise NFVIS  89

CHAPTER 11  Configuring vBranch High Availability  91
Prerequisites for vBranch HA  91
vBranch HA Design and Topology  92
Enable Virtual NIC Failure Detection with Track Feature  93
Isolating LAN and Transit Link Traffic for vBranch HA  95
Packet Flow for vBranch HA  96
Configuration Examples for vBranch HA  97
   Example: Active Cisco ENCS Configuration with ISRv1  97
   Example: Standby Cisco ENCS Configuration with ISRv2  99
Cisco ENCS Failure Points  100

CHAPTER 12  Cisco ENCS Single WAN IP Deployment Scenarios  105
Single WAN IP Deployment 105
Preconfiguring the Cisco ENCS for a Single WAN IP Deployment 106
Single WAN IP Deployment with Gigabit Ethernet Interface 0/0 107
Single WAN IP Deployment with the 4G Interface 108

CHAPTER 13

Resetting to Factory Default 111

CHAPTER 14

Event Notifications 113
nfvisEvent 114
vmlevent 123
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.
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 Release 3.7.1</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>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

**Note**

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](https://www.cisco.com) 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.

```
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
!
!
vrf definition Mgmt-intf
!
  address-family ipv4
  exit-address-family
  !
  address-family ipv6
  exit-address-family
  !
no aaa new-model
!
!
ip domain name cisco.com
!
!
subscriber templating
!
multilink bundle-name authenticated
!
!
```
license udi pid ISR4331/K9 sn FDO192207MN
!
!
ucse subslot 1/0
!imc access-port shared-lom console
!imc 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 ucsel/0/0
ip unnumbered GigabitEthernet0/0/0
negotiation auto
switchport mode trunk
no mop enabled
no mop sysid
!
interface ucsel/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.
Installing Cisco Enterprise NFVIS on a Cisco ENCS 5104

Step 1  Create bootable usb with NFVIS image.

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.
In this example, we used Rufus utility in Windows environment. Rufus utility can be downloaded 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.
CHAPTER 3

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
• CIMC Control, on page 39
• DPDK Support, on page 40
• Backup and Restore NFVIS and VM Configurations, on page 41

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.

Note

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

Figure 4: Default Network Configuration with a Cisco UCS E-Series Server

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 43).
- Using console - After the initial login using the default password, you are prompted to change the default password.
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:

```bash
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:

```
configure terminal
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:

```
configure terminal
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:

```
configure terminal
system settings wan slaac-ipv6
commit
```

To configure SLAAC IPv6 on the management interface:

```
configure terminal
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:

```
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
```
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 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
!
!
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:

```
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
!
!
!
system settings-native mgmt dhcp disabled
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
!
!
!
```

```
system settings-native wan dhcp enabled
```
### 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 ENCS 5400 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 destination,netmask&gt;</td>
<td>• show system routes</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:

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

To delete a user:

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

Note

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/operations/rbac/authentication/users /user/&lt;user-name&gt;/change-password</td>
<td></td>
</tr>
<tr>
<td>• /api/operations/rbac/authentication/users/user/oper/change-role</td>
<td></td>
</tr>
<tr>
<td>• /api/config/rbac/authentication/users/user?deep</td>
<td></td>
</tr>
<tr>
<td>• rbac authentication users</td>
<td></td>
</tr>
<tr>
<td>• rbac authentication users user change-password</td>
<td></td>
</tr>
<tr>
<td>• rbac authentication users user change-role</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.

```
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
t locality
organization
organization-unit-name
state

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

```
fvis# 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>• 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:

```shell
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:

```shell
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:

```shell
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:

```shell
nfvis# show system time

system time current-time 2017-01-01T17:35:39+00:00

system time current-timezone "UTC (UTC, +0000)"

REMOTE REPID ST T WHEN POLL REACH DELAY
```
**OFFSET** | **JITTER**
---|---
*calo-timeserver GPS 1 u 4 64 1 69.423 2749736 0.000
* sys.peer and synced, o pps.peer, # selected, + candidate,
- outlyer, . excess, x falseticker, 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>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/system/set-manual-time</td>
<td>• system time</td>
</tr>
<tr>
<td>• /api/config/system/time/ntp/preferred_server</td>
<td>• show system time</td>
</tr>
<tr>
<td>• /api/config/system/time/ntp/backup_server</td>
<td>• system set-manual-time</td>
</tr>
<tr>
<td>• /api/config/system/time/timezone</td>
<td></td>
</tr>
<tr>
<td>• /api/operational/system/time?deep</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:

```bash
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:

```bash
nfvis# show system portal status
system portal status "access disabled"
```
Configuring System Logs

You can view system logs for troubleshooting purpose. 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></td>
<td>[all/configuration/operational] level</td>
</tr>
<tr>
<td></td>
<td>[critical/debug/error/info/warning]</td>
</tr>
<tr>
<td>• /api/operational/system/logging-level</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:

```plaintext
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:

```plaintext
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
```

**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**

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.

**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.

---

**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.

NFVIS 3.10.x release does not support `tcpcap` packet capture 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.

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.

### 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.

**Note**

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-vm option.

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

The following examples shows the backup options:

```bash
nfvis# hostaction backup configuration-and-vms file-path intdatastore:sample.bkup
```

```bash
nfvis# hostaction backup configuration-only file-path extdatastore2:sample-dir/sample.bkup
```

The following example shows the backup stored on a USB:

```bash
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
```
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>
```
Sample Format 2

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

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 44
- Configuring PnP Discovery Methods, on page 45
- PnP Action, on page 48

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".
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.

Note 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"
**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#**
```

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
server-connection
 status: Success
 time: 19:59:38 Feb 27\nbackoff
 status: Success
 time: 19:59:38 Feb 27"n"
pnp status ip-address apic-em-fqdn.cisco.com
pnp status ipv6-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
```

**Note**
pnp status cco_discovery 0
pnp status dhcp-ipv6 0
pnp status dns-ipv6 0
pnp status cco-ipv6 0
pnp status timeout 0
nfvis#

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\nccli-exec\nstatus: Success\ntime: 18:30:57 Apr 21\nsnserver-connection\nstatus: Success\ntime: 15:40:41 Apr 22\n\ncertificate-install\nstatus: Success\ntime: 18:31:03 Apr 21\ndevice-auth\nstatus: Success\ntime: 18:31:08 Apr 21\nbackoff\nstatus: 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\nccli-exec\nstatus: Success\ntime: 17:18:42 Apr 22\n\nsnserver-connection\nstatus: Success\ntime: 17:20:00 Apr 22\ncertificate-install\nstatus: Success\ntime: 17:18:47 Apr 22\ndevice-auth\nstatus: Success\ntime: 17:18:53 Apr 22\nbackoff\nstatus: 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\nccli-exec\nstatus: Success\ntime: 17:18:42 Apr 22\nsnserver-connection\nstatus: Success\ntime: 17:20:00 Apr 22\ncertificate-install\nstatus: Success\ntime: 17:18:47 Apr 22\ndevice-auth\nstatus: Success\ntime: 17:18:53 Apr 22\nbackoff\nstatus: 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>
 CHAPTER 5

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/img/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 49
• Uploading VM Images to an NFVIS Server, on page 51
• VM Bootstrap Configuration Options with a VMDeployment, on page 52
• OpenStack Configuration Drive Support for Third Party VMs, on page 53
• Performing Resource Verification, on page 54
• Configuring Management IP Address, on page 55
• VM States, on page 55

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.

Note

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`).

Note

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:

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

Note

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:

```
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>337</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</td>
<td>• api/operational/resources/cpu-info/cpus&lt;br&gt;• /api/operational/resources/cpu-info/cpus/cpu&lt;br&gt;• /api/operational/resources/cpu-info/cpus/cpu/&lt;cpu-id&gt;</td>
<td>show resources cpu-info cpus</td>
</tr>
<tr>
<td>for each CPU or the user specified CPU, and the VMs pinned to the CPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To display information on</td>
<td>• /api/operational/resources/cpu-info/vnf&lt;br&gt;• /api/operational/resources/cpu-info/vnf/vnf&lt;br&gt;• /api/operational/resources/cpu-info/vnf/vnf/&lt;deployment_name&gt;.&lt;vm_group_name&gt;</td>
<td>show resources cpu-info vnf</td>
</tr>
<tr>
<td>the VMs running in all the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>physical CPUs or a specific physical CPU in the system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To get information on the</td>
<td>/api/operational/resources/cpu-info/ allocation</td>
<td>show resources cpu-info allocation</td>
</tr>
<tr>
<td>number of CPUs allocated to VMs and the CPUs that are already used by the VMs</td>
<td></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
vmlifecycle 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 57
- Service Chaining of VMs, on page 61

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:

   ```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X
```
3. Verify the image status using the following API method:

```bash
```

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.

```bash
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>
    <interfaces>
      <interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
        <port_forwarding>
          <port>
            <type>ssh</type>
            <protocol/tcp>protocol</protocol>
            <vnf_port>22</vnf_port>
            <external_port_range>
              <start>2022</start>
              <end>2022</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>
  </vm_group>
  <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>
</deployment>
```

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.5.1
If two VM's are connected to the same physical interface, one over SRIOV 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 SRIOV 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 61
- Service Chaining of Multiple VMs with Windows or Linux Servers, on page 62

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.

   ```bash
curl -k -v -u admin:admin -H 'Content-Type: application/vnd.yang.data+xml' --header 'Accept: application/vnd.yang.data+xml, application/json' -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.

   ```bash
curl -k -v -u admin:admin -H 'Content-Type: application/vnd.yang.data+xml' --header 'Accept: application/vnd.yang.data+xml, application/json' -X POST https://<NFVIS_IP>/api/config/networks --data '
    <network>
      <name>sc-net</name>
      <bridge ref="/api/config/bridges/sc-br"/>
    </network>'
```
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 58 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 61.

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 61 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 58 for API command details for Steps 2 to 4.
CHAPTER 7

SPAN Session or Port Mirroring

• About SPAN Sessions, on page 63
• Configuring SPAN Sessions, on page 63
• Configuration Examples for SPAN Session Scenarios, on page 65

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

```
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)

```
nfvis# show running-config monitor session
monitor session 6
  destination vm-vnic CentOS_SRIOV 1
  source vm-vnic C3 0
!
nfvis#
```

```
nfvis# show system monitor session
system monitor session 6
  bridge ""
  destination_vlan ""
  destination_interface LAN-SRIOV-1
  source_vlans ""
  source_rx_interfaces LAN-SRIOV-3
  source_tx_interfaces LAN-SRIOV-3
  source_all ""
  statistics ""
nfvis#
```

```
nfvis# show monitor session status
NUMBER  STATUS
------------------------
20  CREATE_SUCCESS
```
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
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:

```bash
configure terminal
tcpdump port eth0
```

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

To configure packet capture on a vNIC:

```bash
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 <code>tcpdump</code> 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.

**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;
  --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; default is false
  --eager_zero {true,false} --eager_zero=true; default is false
  --nocloud {true,false} --nocloud=true/false; default is false
  --bootstrap_cloud_init_bus_type {ide,virtio} --bootstrap_cloud_init_bus_type=virtio; default is ide
  --bootstrap_cloud_init_drive_type {cdrom,disk} --bootstrap_cloud_init_drive_type=disk; default is
  --mgmt_vnic MGMT_VNIC VM management interface identifier
  --pack_dir <DIR> PACK package all files in directory
```
Required:
- **o** 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
- **-1** 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
- **-t** VNF_TYPE, --vnf_type VNF_TYPE
  [REQUIRED] VNF type, e.g. ROUTER, FIREWALL, vWAAS,
  vWLC, and OTHER
- **-n** NAME, --vnf_name NAME
  [REQUIRED] Name of the VNF image
- **-r** 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>:\n</value>\], key:<value>, [keyattr_<attr>:\n</value>\], type:<value>, val<val>N\n</value>
Not supported

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&lt;br&gt;Options are: true/false&lt;br&gt;Monitoring timeout period for a monitored VM is 600 seconds by default</td>
</tr>
<tr>
<td>optimize</td>
<td>Mandatory</td>
<td>Optimized VM&lt;br&gt;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&lt;br&gt;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: &lt;br&gt;--bootstrap ovf-env.xml for ISRv 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.&lt;br&gt;The default is 1.</td>
</tr>
<tr>
<td>max_vcpu</td>
<td>Optional</td>
<td>Maximum number of vCPUs required for the VM.&lt;br&gt;The default is 8.</td>
</tr>
<tr>
<td>min_mem</td>
<td>Optional</td>
<td>Minimum memory in MB required for the VM.&lt;br&gt;The default is 4 GB.</td>
</tr>
<tr>
<td>max_mem</td>
<td>Optional</td>
<td>Maximum memory in MB required for the VM.&lt;br&gt;Physical memory: 2 GB&lt;br&gt;The default is 8 GB.</td>
</tr>
</tbody>
</table>
### Parameter | Mandatory/Optional | Description
--- | --- | ---
min_disk | Optional | Minimum disk in GB required for the VM. The default is 8 GB.
max_disk | Optional | Maximum disk in GB required for the VM. Available disks are SSD and HDD: 15 GB. The default is 16 GB.
vnic_max | Optional | Maximum number of VNICS allowed for the VM. The default is 8.
profile | Optional | The profile name, profile description, number of vCPUs required, minimum memory required in MB and minimum disk space required in MB.
default_profile | Optional | The default profile.
sriov | Optional | Enable or disable SRIOV support. The default is false.
sriov_list | Optional | List of SRIOV drivers.
pcie | Optional | Not supported.
pcie_list | Optional | Not supported.
privileged | Optional | Not supported.
custom | Optional | 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.
pack_dir | Optional | package all files in directory

### 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

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
```

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 --json d.json -t firewall -n "PA FIREWALL" -r 8.0.5 --app_vendor PA --monitor true --ha_package
```

Example 6: Usage for Asav

```
nfvpt.py -i foo.qcow2 -o asav.tar.gz --json p1.json --app_vendor cisco -t firewall -r 10 --optimize true -n asav --monitored true --ha_package --ha_capable
```

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.

```
Standard VM Image Packaging
```

```
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.

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</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</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 3: Package Manifest File Details**

<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 disk1 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 disk1 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>Disk2 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>Disk10 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>Value</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><strong>&lt;bootstrap_cloud_init_drive_type&gt;</strong></td>
<td>Mounts day0 config file as disk (default is CD-ROM)</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><strong>&lt;bootstrap_cloud_init_bus_type&gt;</strong></td>
<td>Default is IDE</td>
<td>virtio, ide</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td><strong>BOOTSTRAP</strong></td>
<td>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</td>
<td>&lt;bootstrap_file&gt;</td>
<td>File name of the bootstrap file</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>Custom properties</strong></td>
<td>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.</td>
<td>&lt;custom_property&gt;</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Profiles for VM deployment</td>
<td>List of VM deployment profiles. Minimum one profile is required</td>
<td><code>&lt;profiles&gt;</code></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------</td>
<td>--------------</td>
<td>----------</td>
<td></td>
</tr>
<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>
<td></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>
<td>Mandatory</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>
<td>Mandatory if monitoring is true</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>
<td>Mandatory</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>
<td>Optional</td>
</tr>
<tr>
<td>Virtual interface model</td>
<td></td>
<td><code>&lt;virtual_interface_model&gt;</code></td>
<td>Optional</td>
<td></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>
<td>Optional</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>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Profile name</td>
<td><code>&lt;name&gt;</code></td>
<td>Any</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the profile</td>
<td>&lt;description&gt;</td>
<td>Any</td>
<td>Mandatory</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------</td>
<td>--------------</td>
<td>-----</td>
<td>-----------</td>
</tr>
<tr>
<td>vCPU</td>
<td>vCPU number in a profile</td>
<td>&lt;vcpus&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory - MB in profile</td>
<td>&lt;memory_mb&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Root Disk Size</td>
<td>Disk size - MB in profile</td>
<td>&lt;root_disk_mb&gt;</td>
<td></td>
<td>Mandatory</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>
<td>Optional</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>
<td>Optional</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>
<td>Optional</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>
<td>Optional</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>
<td>Optional</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>
<td>Optional</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
** sha1sum - for calculating checksum
<PackageContents>
  <File_Info>
    <name>ISRv_serial_3.16.02.qcow2</name>
    <type>root_image</type>
    <sha1_checksum>93de73ee3531f74fdff99377972357a8a0eac7b</sha1_checksum>
  </File_Info>
  <File_Info>
    <name>image_properties.xml</name>
  </File_Info>
</PackageContents>
```
Example: Image Properties

```xml
<?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>ICMPPing</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>
<pcie_supported>false</pcie_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.
nfvis# show system upgrade reg-info

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>LOCATION</th>
<th>VERSION</th>
<th>STATUS</th>
<th>UPLOAD DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfvis-351.nfvispkg</td>
<td>/data/upgrade/register/nfvis-351.nfvispkg</td>
<td>3.6.1-722</td>
<td>Valid</td>
<td>2017-04-25T10:29:58.052347-00:00</td>
</tr>
</tbody>
</table>

**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

```
fvis# show system upgrade apply-image

<table>
<thead>
<tr>
<th>UPGRADE</th>
<th>NAME</th>
<th>STATUS</th>
<th>FROM</th>
<th>UPGRADE TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfvis-351.nfvispkg</td>
<td>SUCCESS</td>
<td>3.5.0</td>
<td>3.5.1</td>
<td></td>
</tr>
</tbody>
</table>
```

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>
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 91
- vBranch HA Design and Topology, on page 92
- Enable Virtual NIC Failure Detection with Track Feature, on page 93
- Isolating LAN and Transit Link Traffic for vBranch HA, on page 95
- Packet Flow for vBranch HA, on page 96
- Configuration Examples for vBranch HA, on page 97
- Cisco ENCS Failure Points, on page 100

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

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

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

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

```plaintext
device# show track
Track 2
    Interface GigabitEthernet2 line-protocol
        Line protocol is Down ((hw down))
```

### 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:

```plaintext
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"
```
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 ISRs 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.
• 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

!```

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.5.1
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

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 Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.5.1
negotiation auto
!
interface GigabitEthernet3
  no ip address
  shutdown
  negotiation auto
!
interface GigabitEthernet4
  description Service-NET-virtio
  ip address 192.0.2.22 255.255.255.0
  standby use-bia
  standby 25 ip 192.0.2.20
  standby 25 timers 1 5
  standby 25 preempt
  negotiation auto
  bfd interval 9000 min_rx 9000 multiplier 3
!
interface GigabitEthernet5
  ip address 192.0.2.23 255.255.255.0
!
!
router eigrp 10
  network 8.8.8.0 0.0.0.255
  network 25.25.25.0 0.0.0.255
  network 46.46.46.0 0.0.0.255
!

Cisco ENCS Failure Points

<table>
<thead>
<tr>
<th>Failure Points</th>
<th>Sequence of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCS chassis hardware failure:</td>
<td>1. HSRP on ENCS2 detects the reachability failure to ENCS1, and triggers the failover. LAN virtual IP becomes active on ENCS2.</td>
</tr>
<tr>
<td>• Power down</td>
<td>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.</td>
</tr>
<tr>
<td>• Power cycle</td>
<td>3. All the WAN to LAN, and LAN to WAN traffic will now flow through ENCS2.</td>
</tr>
<tr>
<td>• Reboot</td>
<td>4. The PBR/PFR configuration will now select the LAN port as the preferred path instead of the transit link for the traffic heading to LAN.</td>
</tr>
<tr>
<td>Cisco Enterprise NFVIS software failure</td>
<td></td>
</tr>
<tr>
<td>• Crash</td>
<td></td>
</tr>
<tr>
<td>Cisco ISRv software failure</td>
<td></td>
</tr>
<tr>
<td>• Stop (shutdown)</td>
<td></td>
</tr>
<tr>
<td>• Reboot</td>
<td></td>
</tr>
<tr>
<td>• Crash</td>
<td></td>
</tr>
<tr>
<td>• Error</td>
<td></td>
</tr>
</tbody>
</table>

ISRv1 (Active) Before the Failure

ISRv1# show platform software vnic-if interface-mapping

---

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 3.5.1

100
### Interface Table

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Driver Name</th>
<th>Mac Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet5</td>
<td>i40evf</td>
<td>5254.003a.1020 (LAN-SRIOV-2)</td>
</tr>
<tr>
<td>GigabitEthernet4</td>
<td>virtio</td>
<td>5254.0053.e392 (service-net)</td>
</tr>
<tr>
<td>GigabitEthernet3</td>
<td>i40evf</td>
<td>5254.00c4.b925 (LAN-SRIOV-1)</td>
</tr>
<tr>
<td>GigabitEthernet2</td>
<td>igbvf</td>
<td>5254.00d2.cc9a (GE0-0-SRIOV-1)</td>
</tr>
<tr>
<td>GigabitEthernet1</td>
<td>virtio</td>
<td>5254.00d2.1b1c (int-mgmt-net)</td>
</tr>
</tbody>
</table>

### ISRv1# show standby brief

<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>Gi4</td>
<td>25</td>
<td>105</td>
<td>P</td>
<td>Active</td>
<td>local</td>
<td>192.0.2.1</td>
<td>192.0.2.222</td>
</tr>
</tbody>
</table>

**ISRv2 (Standby) Before the Failure**

### ISRv2# show platform software vnic-if interface-mapping

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Driver Name</th>
<th>Mac Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet5</td>
<td>i40evf</td>
<td>5254.00cc.ce9f (LAN-SRIOV-2)</td>
</tr>
<tr>
<td>GigabitEthernet4</td>
<td>virtio</td>
<td>5254.00e7.523f (Service-net)</td>
</tr>
<tr>
<td>GigabitEthernet3</td>
<td>i40evf</td>
<td>5254.0055.ee45 (LAN-SRIOV-1)</td>
</tr>
<tr>
<td>GigabitEthernet2</td>
<td>igbvf</td>
<td>5254.00a3.d443 (GE0-0-SRIOV-1)</td>
</tr>
<tr>
<td>GigabitEthernet1</td>
<td>virtio</td>
<td>5254.0048.e84c (int-mgmt-net)</td>
</tr>
</tbody>
</table>

### ISRv2 After the Failure

**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

<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>Gi4</td>
<td>25</td>
<td>100</td>
<td>P</td>
<td>Standby</td>
<td>192.0.2.20</td>
<td>local</td>
<td>192.0.2.222</td>
</tr>
</tbody>
</table>

### ISRv2# show logging

*Dec 13 21:22:32.385: %HSRP-5-STATECHANGE: GigabitEthernet4 Grp 25 state Standby -> Active*

---

**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.
Cisco ENCS Failure Points

### Sequence of Events

<table>
<thead>
<tr>
<th>Failure Points</th>
<th>Sequence of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Switch failure</td>
<td>2. WAN-IP1 becomes unreachable, and all the routes converge towards WAN-IP2 on ENCS2. WAN-IP2 is the only IP for branch connectivity.</td>
</tr>
<tr>
<td>• End-to-end connectivity failure</td>
<td>3. HSRP on ENCS2 becomes higher priority in the group, and takes over the active role. LAN-virtual IP becomes active on ENCS2.</td>
</tr>
<tr>
<td></td>
<td>4. 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.</td>
</tr>
<tr>
<td></td>
<td>5. All the WAN to LAN, and LAN to WAN traffic will now flow through ENCS2.</td>
</tr>
</tbody>
</table>

---

**ISRv1 After the Failure**

ISRv1 becomes standby.

**ISRv1# show ip interface brief**

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

**ISRv1# 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>Gi4</td>
<td>25</td>
<td>85</td>
<td>P</td>
<td>Standby</td>
<td>192.0.2.22</td>
<td>local</td>
<td>192.0.2.222</td>
</tr>
</tbody>
</table>

**ISRv1# show log**

*Dec 14 03:41:52.307: %TRACK-6-STATE: 2 interface Gi2 line-protocol Up -> Down
*Dec 14 03:42:37.744: %HSRP-5-STATECHANGE: GigabitEthernet4 Grp 25 state Active -> Speak
*Dec 14 03:42:43.663: %HSRP-5-STATECHANGE: GigabitEthernet4 Grp 25 state Speak -> Standby

**ISRv1# show track**

Track 1

  **IP SLA 1 reachability**
  
  **Reachability is Down**

  1405 changes, last change 00:03:08
  Latest operation return code: Timeout
  Tracked by:
  HSRP GigabitEthernet4 25

**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>Gi4</td>
<td>25</td>
<td>100</td>
<td>P</td>
<td><strong>Active</strong></td>
<td>local</td>
<td>192.0.2.3</td>
<td>192.0.2.222</td>
</tr>
</tbody>
</table>
# Cisco ENCS Failure Points

## 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# 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

**WAN Net2 failure (WAN SROV VF connected to ISRv2 is down)**

- Link down

### Sequence of Events

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.
2. WAN-IP2 becomes unreachable, and all the routes converge towards WAN-IP1 on ENCS1. WAN-IP1 is the only IP for branch connectivity.
3. All the WAN to LAN and LAN to WAN traffic will now flow through ENCS1.

**Transit link between ENCS1 and ENCS2 fails**

- Link down

### Sequence of Events

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.
2. WAN-IP2 becomes unreachable, and all the routes converge towards WAN-IP1 on ENCS1. WAN-IP1 is the only IP for branch connectivity.
3. All the WAN to LAN and LAN to WAN traffic will now flow through ENCS1.
Cisco ENCS Single WAN IP Deployment Scenarios

- Single WAN IP Deployment, on page 105
- Preconfiguring the Cisco ENCS for a Single WAN IP Deployment, on page 106
- Single WAN IP Deployment with Gigabit Ethernet Interface 0/0, on page 107
- Single WAN IP Deployment with the 4G Interface, on page 108

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 SRIOV-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 datagtime msec
service timestamps log datagtime msec
service internal
service call-home
no platform punt-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
do destination transport-method email
destination transport-method http
do destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService

destination address http http://10.22.183.117:8080/ddce/services/DDCEService

clock timezone PST -7
ntp server 192.0.2.9
do license smart register idtoken NXM1NJElMDaTNDV1ZC00ZLTq4MGEtNmlRjNjg2Ng5ZDV1LTE0OTg5Ndk2%0ANJIpNz8a5k5SeuoL2pMTGtuNSs3Q3JxdoGVoSUVPqFmY2IlalVqR3B5MzFj%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**

template 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 nhs 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 192.0.2.13 255.255.255.255 198.51.100.5
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
!
ntp 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>• /api/operations/factory-default-reset/all</td>
<td>• factory-default-reset</td>
</tr>
<tr>
<td>• /api/operations/factory-default-reset/all-except-images</td>
<td></td>
</tr>
<tr>
<td>• /api/operations/factory-default-reset</td>
<td></td>
</tr>
<tr>
<td>/all-except-images-connectivity</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 114
- `vmlcEvent`, on page 123
### 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 version="1.0" encoding="UTF-8"?>
<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 successfully</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 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 updated successfully</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.| &lt;?xml version="1.0" encoding="UTF-8"?>
&lt;notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
&lt;eventTime&gt;2016-09-22T12:42:03.391986+00:00&lt;/eventTime&gt;
&lt;nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier"
&lt;user_id&gt;admin&lt;/user_id&gt;
&lt;config_change&gt;true&lt;/config_change&gt;
&lt;transaction_id&gt;0&lt;/transaction_id&gt;
&lt;status&gt;SUCCESS&lt;/status&gt;
&lt;status_message&gt;Network deleted succesfully&lt;/status_message&gt;
&lt;event_type&gt;NETWORK_DELETE&lt;/event_type&gt;
&lt;network_name&gt;testn1&lt;/network_name&gt;
&lt;/nfvisEvent&gt;
&lt;/notification&gt; |
| UPGRADE_REGISTER| System upgrade is registered. | &lt;?xml version="1.0" encoding="UTF-8"?>
&lt;notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
&lt;eventTime&gt;2017-04-26T15:57:50.434636+00:00&lt;/eventTime&gt;
&lt;nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier"
&lt;user_id&gt;NA&lt;/user_id&gt;
&lt;config_change&gt;true&lt;/config_change&gt;
&lt;transaction_id&gt;0&lt;/transaction_id&gt;
&lt;status&gt;SUCCESS&lt;/status&gt;
&lt;status_code&gt;0&lt;/status_code&gt;
&lt;status_message&gt;Upgrade package registration successful:
Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg&lt;/status_message&gt;
&lt;event_type&gt;UPGRADE_REGISTER&lt;/event_type&gt;
&lt;/nfvisEvent&gt;
&lt;/notification&gt; |
| UPGRADE_APPLY | System upgrade is applied. | &lt;?xml version="1.0" encoding="UTF-8"?>
&lt;notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
&lt;eventTime&gt;2017-04-26T16:02:43.885516+00:00&lt;/eventTime&gt;
&lt;nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier"
&lt;user_id&gt;NA&lt;/user_id&gt;
&lt;config_change&gt;true&lt;/config_change&gt;
&lt;transaction_id&gt;0&lt;/transaction_id&gt;
&lt;status&gt;SUCCESS&lt;/status&gt;
&lt;status_code&gt;0&lt;/status_code&gt;
&lt;status_message&gt;Upgrade Process: In Progress&lt;/status_message&gt;
&lt;event_type&gt;UPGRADE_APPLY&lt;/event_type&gt;
&lt;/nfvisEvent&gt;
&lt;/notification&gt; |
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
**ROTATED_LOGS_DELETE** | Rotated logs older than 30 days are deleted by the system. | ```xml
<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>
</xml>
```  

**ROTATED_LOGS_DELETE** | Older logs deleted by the system when the total file size of rotated logs exceeds 2GB. | ```xml
<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>
</xml>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| SECURE_OVERLAY_CREATING | create secure overlay | `<notification
<eventTime> 2018-11-02T04:23:02.641317+00:00
<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>
</notification>` |
| 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>` |
### Event Notifications

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAN DHCP SWITCHOVER</td>
<td>WAN bridge toggle</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>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>

| 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>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>
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| ROUTE_DISTRIBUTION | 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> |
| 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> |
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
ROUTE_DISTRIBUTION_ERROR | Route distribution in error | ```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>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>
```  

ROUTE_DISTRIBUTION_DELETE | Route distribution deleted | ```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>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>
```
## Event Notifications

<table>
<thead>
<tr>
<th>EventType</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE_DISTRIBUTION_UP</td>
<td>Route distribution up</td>
<td>&lt;!-- xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVS_DPDK_SUCCESS</td>
<td>Enable DPDK</td>
<td>&lt;!-- xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVS_DPDK_FAILURE</td>
<td>DPDK failure</td>
<td>&lt;!-- xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Event Notifications

#### vmlcEvent

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| CREATE_IMAGE | The VM image is registered.                               | `<xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:30:30.76+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>
  <vm_source></vm_source>
  <vm_target></vm_target>
  <event>
    <type>CREATE_IMAGE</type>
  </event>
</vmlcEvent>
</notification>` |
| 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>
  <vm_source></vm_source>
  <vm_target></vm_target>
  <event>
    <type>DELETE_IMAGE</type>
  </event>
</vmlcEvent>
</notification>` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| 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>` |
### Event Notifications

<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><code>&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;2016-11-17T17:19:16.927+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;VIM Driver: VM successfully created, VM Name: [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c]&lt;/status_message&gt;&lt;depname&gt;1479341445&lt;/depname&gt;&lt;tenant&gt;admin&lt;/tenant&gt;&lt;tenant_id&gt;AdminTenantId&lt;/tenant_id&gt;&lt;depid&gt;c64d79db-3a29-41a8-8114-c80d42731a5b&lt;/depid&gt;&lt;vm_group&gt;ROUTER&lt;/vm_group&gt;&lt;vm_source&gt;&lt;vmid&gt;d18dd252-80c8-44f2-ab66-d4481790bb79&lt;/vmid&gt;&lt;hostid&gt;NFVIS&lt;/hostid&gt;&lt;hostname&gt;NFVIS&lt;/hostname&gt;&lt;interfaces&gt;&lt;nicid&gt;0&lt;/nicid&gt;&lt;port_id&gt;vnet0&lt;/port_id&gt;&lt;network&gt;int-mgmt-net&lt;/network&gt;&lt;subnet&gt;N/A&lt;/subnet&gt;&lt;ip_address&gt;10.20.0.2&lt;/ip_address&gt;&lt;mac_address&gt;52:54:00:31:c5:7f&lt;/mac_address&gt;&lt;netmask&gt;255.255.255.0&lt;/netmask&gt;&lt;gateway&gt;10.20.0.1&lt;/gateway&gt;&lt;/interfaces&gt;&lt;interfaces&gt;&lt;nicid&gt;1&lt;/nicid&gt;&lt;port_id&gt;vnet1&lt;/port_id&gt;&lt;network&gt;wan-net&lt;/network&gt;&lt;subnet&gt;N/A&lt;/subnet&gt;&lt;mac_address&gt;52:54:00:59:52:41&lt;/mac_address&gt;&lt;netmask&gt;255.255.255.0&lt;/netmask&gt;&lt;gateway&gt;172.19.181.152&lt;/gateway&gt;&lt;/interfaces&gt;&lt;/vm_source&gt;&lt;vm_target&gt;&lt;/vm_target&gt;&lt;event&gt;&lt;type&gt;VM_DEPLOYED&lt;/type&gt;&lt;/event&gt;&lt;/vmlcEvent&gt;&lt;/notification&gt;</code></td>
</tr>
</tbody>
</table>
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
VM_ALIVE | The state of a monitored VM becomes ACTIVE. | `<?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: [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-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>
      <vm_id>d18dd252-80c8-44f2-ab66-d4481790bb79</vm_id>
      <host_id>NFVIS</host_id>
      <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>
    </vmlc_event>
  </vmlcEvent>
</notification>`
### Event Type: VM_UNDEPLOYED

The VM **is undeployed**.

```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>c64d978b-3a29-41a8-8114-c80d42731a5b</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>d182d252-80c8-44f2-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>172.19.191.152</gateway>
        </interface>
      </interfaces>
    </vm_source>
    <vm_target>
    </vm_target>
  </vmlcEvent>
</notification>
```
### Event Notifications

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| SERVICE_UPDATED | The VM is updated. | ```<?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_UPDATED</type>
</event>
</vmlcEvent>
</notification>``` |
| VM_STOPPED | The VM is stopped per VM action request. | ```<?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:AdminTen_TENANT_ROUTER_0_df6733c1-0768-4ae6-8dce-b223edcb036c].</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_source>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
</vm_source>
<vm_target>
<vmid>NULL</vmid>
</vm_target>
<event>
<type>VM_STOPPED</type>
</event>
</vmlcEvent>
</notification>``` |
### Event Types

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_STARTED  | The VM is started per VM action request. | ```<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-c80d427315a5b</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>
      <event>
        <type>VM_STARTED</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_REBOOTED | The VM is rebooted per VM action request. |<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-11-17T17:36:56.5+00:00</eventTime>
  <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Rebooted VM [SystemAdminTena_ROUTER_0_f17fc494-8535-4b05-b88d-f0fd2effdc7d]</status_message>
    <depname>1479342258</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid=NULL</svcid>
    <depid>827e871a-30d5-4f5f-a05a-263b7ee3a734</depid>
    <vm_group>ROUTER</vm_group>
    <vmid>d918a3b1-f2a9-4065-9d8e-2135b0a37d87</vmid>
    <hostid>NFVIS</hostid>
    <hostname>NFVIS</hostname>
  </vmlcEvent>
</notification>
### Event Notifications

**Event Type** | **Notification Trigger** | **Notification Output Example**
--- | --- | ---
VMRECOVERY | A monitored VM is not reachable. | ```xml
<?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>
<vmlcEvent
 xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Recovery event for VM [SystemAdminTen;a_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>
</vmlcEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_RECOVERY_REBOOT | Recovery reboot starts for the monitored VM, which is not reachable. | ```xml
<?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_code>200</status_code>
    <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>
    <depid>9e7f4f8-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>
      <type>VM_RECOVERY_REBOOT</type>
    </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
<?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 [SystemAdminTenar_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b].</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-aaf6-0f31d888af67</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target>
      <vmid>000883fc-77f3-4b9e-aaf6-0f31d888af67</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>
  </vmlcEvent>
</notification>`
```
### Event Notifications

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_MONITOR_UNSET | Monitoring is disabled per VM action request. | `<xml version="1.0" encoding="UTF-8"?>
<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>
    <depname>1479413090</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid>NULL</svcid>
    <depid>742dd335-330c-4bf0-a75d-a44003c645c5</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target></vm_target>
    <event>
      <type>VM_MONITOR_UNSET</type>
    </event>
  </vmlcEvent>
</notification>` |

| VM_MONITOR_SET | Monitoring is enabled per VM action request. | `<xml version="1.0" encoding="UTF-8"?>
<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>
    <depname>1479413090</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid>NULL</svcid>
    <depid>742dd335-330c-4bf0-a75d-a44003c645c5</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target></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-b84c-80587f626004</vmid>
<vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname>
</vm_source>
<event>
<type>VM_UPDATED</type>
</event>
</vmlcEvent>
</notification>
```
Notification Output Example

<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-12-08T00:50:39.06+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>
    <user_name>admin</user_name>
    <depname>1512766000</depname>
    <tenant>admin</tenant>
    <tenant_id>adminUUID</tenant_id>
    <depid>92c11aa1-f6dd-47d1-948f-c865b9ef70f</depid>
    <event>
      <type>SERVICE_UPDATED</type>
    </event>
  </vmlcEvent>
</notification>
### Event Notifications

<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>VNIC is added, deleted or updated.</td>
<td></td>
</tr>
</tbody>
</table>
Event Type | Notification Trigger | Notification Output Example
---|---|---

```xml
<?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]
  <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-c0a28a481b1</vmid>
    <hostid>NFVIS</hostid>
    <hostname>nfvis</hostname>
    <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>
</table>
|            |                      | `<network>my-net-1</network>`<subnet>N/A</subnet><mac_address>52:54:00:66:55:c1</mac_address>`</interface>`</interfaces>`</vm_source>`<event>`<type>VM_UPDATED</type>`</event>`</vmcEvent>