The specifications and information regarding the products in this manual are subject to change without notice. All statements, information, and recommendations in this manual are believed to be accurate but are presented without warranty of any kind, express or implied. Users must take full responsibility for their application of any products.

The software license and limited warranty for the accompanying product are set forth in the information packet that shipped with the product and are incorporated herein by this reference. If you are unable to locate the software license or limited warranty, contact your Cisco representative for a copy.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided “as is” with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice. In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020–2021 Cisco Systems, Inc. All rights reserved.
CONTENTS

Full Cisco Trademarks with Software License 1

CHAPTER 1  New and Changed Information 7

CHAPTER 2  About NFVIS Documentation 7

CHAPTER 3  Host System Management 9

System Access Configuration 9
Host System Requirements 9
System Setting Hostname 10
Dual WAN Support 11
Restrictions for Dual WAN Support 11
Dual WAN Bridge and DHCP Toggle 11
Accessing NFVIS 12
Configuring VLAN for NFVIS Management Traffic 16
Configuring the IP Receive ACL 17
Port 22222 and Management Interface ACL 17
Configuring Secondary IP Address and Source Interface 18
CIMC Access Control 19
CIMC Access using NFVIS 19
BIOS-CIMC Update 19
BIOS and CIMC Password 20
UEFI Secure Boot on ENCS 5400 21
PXE Boot Mode 22
Enable or Disable Access to NFVIS Portal 22
Users, Roles, and Authentication 23
Local User Account Management 23
Local Authentication for a Specific Group of Users 26
RADIUS Support 27
   About RADIUS 27
   RADIUS Operation 27
   Configuring RADIUS 28
TACACS+ Support 29
   About TACACS+ 29
   TACACS Operation 29
   Configuring a TACACS+ Server 30
Default Authentication Order 31
Networking 32
   Bridges 32
      Creating Bridges 33
      Configuring Bridge Port 33
      Configuring Bridge IP Connectivity 34
Physical Network Interface Cards 38
   Configuring LLDP 38
   Configuring Administrative Status of a Port 39
   Tracking Changes for a Port 39
   Speed, Duplex and Autonegotiation 40
   Port Channels 42
   Promiscuous mode 45
Dynamic SR-IOV 45
System Routes 47
   Configuring System Routes 47
Troubleshooting 47
Cisco Network Plug-n-Play Support 48
   PnP Discovery Methods 49
   Configuring PnP Discovery Methods 50
   Certificate for Static PnP discovery 53
   PnP Action 55
DPDK Support on NFVIS 56
Storage Access 58
Contents

CHAPTER 4  VM Life Cycle Management  77

Overview of VM Life Cycle Management  77
Workflow of VM Life Cycle Management  77
Support for NFVIS Container Lifecycle Management  79
Uploading VM Images to an NFVIS Server  79
Performing Resource Verification  81
Configuring Management IP Subnet  82

VM Image Packaging  82
VM Image Packaging Utility  82
Contents  82
Usage  83
NFVIS Specific Enhancements  87
VM Packaging Utility Usage Examples  88
Standard VM Image Packaging  89
Generating a VM Package  89

Appendix  90
VM Image Package Files  90
Image Properties Template File  98

Image Registration  99
Register VM Packages Using REST API  99
Register VM Image with Multiple Root Disks  100
Register a VM Image through a Root Disk  101
Register a Remote VM Image  101
Single IP Address Without Secure Overlay 154

CHAPTER 6

BGP Support on NFVIS 155
Configure BGP on NFVIS 155
Route Distribution 158
BGP Route Annoucement over MPLS or IPSec 159

CHAPTER 7

Security Considerations 163
Installation 164
Image Tamper Protection 164
RPM Signing 164
RPM Signature Verification 164
Image Integrity Verification 164
ENCS Secure Boot 165
Secure Unique Device Identification 165
Device Access 166
Enforced Password Change at First Login 167
Restricting Login Vulnerabilities 167
Enforcement of Strong password 167
Configuring Minimum Length for Passwords 167
Configuring Password Lifetime 168
Limit previous password reuse 168
Restrict Frequency of login attempts 168
Disable inactive user accounts 168
Integration with external AAA servers 169
Authentication Cache for External Authentication Server 170
Role Based Access Control 170
Restrict Device Accessibility 172
Attack vector reduction 172
Enabling only essential ports by default 172
Restrict Access To Authorized Networks For Authorized Services 173
Privileged Debug Access 175
Secure Interfaces 175
Console 176
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH 176</td>
</tr>
<tr>
<td>CLI Session timeout 176</td>
</tr>
<tr>
<td>NETCONF 176</td>
</tr>
<tr>
<td>REST API 177</td>
</tr>
<tr>
<td>NFVIS Web Portal 177</td>
</tr>
<tr>
<td>HTTPS 178</td>
</tr>
<tr>
<td>SNMP Access 179</td>
</tr>
<tr>
<td>Legal Notification Banners 180</td>
</tr>
<tr>
<td>Factory Default Reset 181</td>
</tr>
<tr>
<td>Infrastructure Management Network 182</td>
</tr>
<tr>
<td>Out-of-band Management 183</td>
</tr>
<tr>
<td>Pseudo out-of-band Management 183</td>
</tr>
<tr>
<td>In-band Management 183</td>
</tr>
<tr>
<td>Locally Stored Information Protection 184</td>
</tr>
<tr>
<td>Protecting Sensitive Information 184</td>
</tr>
<tr>
<td>File Transfer 184</td>
</tr>
<tr>
<td>Logging 184</td>
</tr>
<tr>
<td>Virtual Machine security 185</td>
</tr>
<tr>
<td>VNF secure boot 185</td>
</tr>
<tr>
<td>VNC Console Access Protection 185</td>
</tr>
<tr>
<td>Encrypted VM config data variables 186</td>
</tr>
<tr>
<td>Checksum verification for Remote Image Registration 186</td>
</tr>
<tr>
<td>Certification Validation for Remote Image Registration 186</td>
</tr>
<tr>
<td>VM Isolation and Resource provisioning 186</td>
</tr>
<tr>
<td>CPU Isolation 187</td>
</tr>
<tr>
<td>Memory Allocation 188</td>
</tr>
<tr>
<td>Storage Isolation 188</td>
</tr>
<tr>
<td>Interface Isolation 189</td>
</tr>
<tr>
<td>Secure Development Lifecycle 189</td>
</tr>
</tbody>
</table>

**CHAPTER 8**

Secure Operation in FIPS Mode on NFVIS 191

Configuring FIPS Mode 191

FIPS Operational Status 194
Introduction about SNMP 252
SNMP Operations 252
   SNMP Get 252
   SNMP Notifications 254
SNMP Versions 254
SNMP MIB Support 255
Configuring SNMP Support 259
System Monitoring 264
   Collection of System Monitoring Statistics 264
Host System Monitoring 265
VNF System monitoring 268
ENCS Switch Monitoring 270

CHAPTER 12 Troubleshoot and Debug Cisco NFVIS 273
   System Shell Access 273
   Log and Show Commands 276
   SPAN Session or Port Mirroring 278
      About SPAN Sessions 278
      Configuring SPAN Sessions 278
   Configuration Examples for SPAN Session Scenarios 280
      Example: SPAN Session Traffic on a Physical Interface 280
      Example: SPAN Session Traffic on a LAN SRIOV 281
      Example: SPAN Session Traffic on a VLAN 282
   Configuring Packet Capture 282

CHAPTER 13 Appendix 285
   Event Notifications 285
      nfvisEvent 285
      vmleEvent 302
   Syslog Messages 319

CHAPTER 14 Glossary 325
New and Changed Information

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

The following tables summarize the new and changed features and tell you where they are documented.

**Table 1: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 4.6.1**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Authentication for a Specific Group of Users</td>
<td>This feature allows you to create a group with specific users, who can perform only the local authentication; and don't have to authenticate externally through TACACS.</td>
<td>Local Authentication for a Specific Group of Users</td>
</tr>
<tr>
<td>Support for External Storage for Cisco Cloud Services Platforms</td>
<td>External disks are supported for Cisco Cloud Services Platforms (CSP).</td>
<td>Support for External Storage for Cisco Cloud Services Platforms</td>
</tr>
<tr>
<td>Support for Replacing HTTP Basic Authentication</td>
<td>This feature enhances NFVIS local portal capabilities, to replace HTTP basic authentication to prevent accidental leakage of credentials.</td>
<td>–</td>
</tr>
</tbody>
</table>

**Table 2: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 4.5.1**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCS Switch Monitoring</td>
<td>This feature allows you to calculate the data rate for ENCS switch ports based on the data collected from the ENCS switch.</td>
<td>ENCS Switch Monitoring, on page 270</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Authentication Cache for External Authentication Server</td>
<td>This feature supports TACACS authentication through OTP on NFVIS portal</td>
<td>Authentication Cache for External Authentication Server, on page 170</td>
</tr>
<tr>
<td>BGP Route Announcement over MPLS or IPSec</td>
<td>This feature allows you to configure NFVIS to announce routes through BGP over MPLS or in conjunction with secure overlay to announce routes over IPSec tunnel.</td>
<td>BGP Route Announcement over MPLS or IPSec, on page 159</td>
</tr>
</tbody>
</table>

**Table 3: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 4.4.1**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Catalyst 8200 Series Edge Universal CPE Support</td>
<td>The Cisco Catalyst 8200 Series is the next generation of Cisco Enterprise Network Compute System platform that combines routing, switching and application hosting into a compact one rack unit device. These platforms are designed to allow customers to run virtualized network functions and other applications as virtual machines on the same hardware platform powered by Cisco NFVIS hypervisor software.</td>
<td>Supported Hardware Platforms On NFVIS Cisco Catalyst 8200 Series Edge uCPE Hardware Installation Guide</td>
</tr>
<tr>
<td>SNMP VM Monitoring MIB</td>
<td>SNMP VM monitoring MIBs support added.</td>
<td>SNMP MIB Support, on page 255</td>
</tr>
<tr>
<td>Live Export Support</td>
<td>Supports VM backup process without shutting down the VM.</td>
<td>VM Action APIs</td>
</tr>
<tr>
<td>VM Export with Selective Disk</td>
<td>You can exclude certain disks or volumes from a VM export.</td>
<td>VM Export with Selective Disk, on page 121</td>
</tr>
<tr>
<td>BGP Support on NFVIS</td>
<td>This feature allows NFVIS system to learn routes announced from the remote BGP neighbor and apply the learnt routes to NFVIS system. This feature also allows you to announce or withdraw NFVIS local routes to the remote BGP neighbor.</td>
<td>BGP Support on NFVIS, on page 155</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>DNS Name Server Configuration Enhancement</td>
<td>NFVIS allows configuring up to three nameservers which can be used by DNS resolvers in the order specified by the user. New command introduced to configure DNS nameservers: <code>system settings name-server</code>.</td>
<td>Configure DNS Name Servers, on page 74</td>
</tr>
<tr>
<td>CIMC Secure Overlay Support</td>
<td>This is a disaster recovery feature to recover an unresponsive NFVIS. CIMC periodically monitors NFVIS health and when NFVIS health check fails, CIMC establishes a secure tunnel to the remote. At remote site, you can login to CIMC and do the necessary troubleshooting to recover NFVIS.</td>
<td>CIMC Secure Overlay Support, on page 222</td>
</tr>
</tbody>
</table>

Table 4: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 4.2.1

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhancements to backup and restore of configurations</td>
<td>New commands are introduced to view the overall status of backup and restore process. Enhancements to backup file location and factory default options are introduced. Information on how to troubleshoot failure to restore NFVIS configurations is added.</td>
<td>Backup and Restore NFVIS and VM Configurations, on page 62</td>
</tr>
<tr>
<td>HugePage memory and CPU allocation</td>
<td>The system memory allocations are enhanced and all memory apart from the amount reserved for system is converted to HugePage memory.</td>
<td>Host System Requirements, on page 9</td>
</tr>
<tr>
<td>Command for PnP certificates</td>
<td>A certificate can be used as a PnP root certificate through Command Line Interface (CLI).</td>
<td>Certificate for Static PnP discovery, on page 53</td>
</tr>
</tbody>
</table>
### Secure Operation in FIPS Mode on NFVIS

Secure Operation in FIPS Mode on NFVIS is described in the Federal Information Processing Standards (FIPS) Publication 140-2, which are publicly announced standards developed by the United States federal government for use in computer systems by non-military government agencies and government contractors.

### BIOS and CIMC Password

New password restrictions and security measures are added for CIMC and BIOS.

### CIMC TACACS Support

CIMC TACACS configuration support enabled on NFVIS.

---

#### Table 5: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 4.1.1

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storm control support</td>
<td>Storm control is used to monitor incoming traffic levels and limit excessive flow of packets on any user facing switch port that could cause a traffic storm.</td>
<td>Configuring Storm Control, on page 209</td>
</tr>
<tr>
<td>hot-add and hot-delete support on DPDK interface</td>
<td>Support interface hot add and hot delete operations on DPDK networks.</td>
<td>Update VM Image, on page 103</td>
</tr>
<tr>
<td>VM backup and restore preserving mac and UID</td>
<td>Specifying the unique MAC ID flag during VM import will ensure that the imported VM is not deployed with the same UID and interface MAC addresses.</td>
<td>Import and Export NFVIS VM, on page 135</td>
</tr>
<tr>
<td>Support placing VM's extra storage volumes to external datastore</td>
<td>This feature supports placing VNF volumes on external datastore storage locations.</td>
<td>VNF Volumes, on page 122</td>
</tr>
<tr>
<td>PNIC promiscuous mode support</td>
<td>Enabling promiscuous mode on an interface can be used to monitor all incoming packets on the interface.</td>
<td>Promiscuous mode, on page 45</td>
</tr>
</tbody>
</table>

---

#### Table 6: New and Changed Features for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 3.12.3

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Tunnel enhancements for PKI and EAP</td>
<td>EAP and PKI authentication supported on secure overlay.</td>
<td>Secure Overlay, on page 139</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Where Documented</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>APC UPS support</td>
<td>This feature provides support for monitoring battery status for an APC UPS connected to the ENCS box through a USB cable.</td>
<td>APC UPS Support and Monitoring, on page 70</td>
</tr>
<tr>
<td>SNMP support for CPU usage</td>
<td>Supported SNMP MIBs for CPU usage</td>
<td>SNMP MIB Support, on page 255</td>
</tr>
<tr>
<td>AAA auth-order</td>
<td>In this feature the supported aaa authentication order is local authentication followed by TACACS+.</td>
<td>Default Authentication Order, on page 31</td>
</tr>
</tbody>
</table>
About NFVIS Documentation

Cisco Enterprise Network Function Virtualization Infrastructure Software (Cisco Enterprise NFVIS) is designed to help service providers and enterprises to design, deploy and manage network services. For more information about Cisco NFVIS, hardware platforms and VMs supported on it see, About Cisco Enterprise NFVIS.

This chapter describes how documentation about Cisco NFVIS is structured.

NFVIS Workflow

Typically, enterprises and service providers would go through the following stages in their NFVIS journey:

- **Install NFVIS**
  - Initial Hardware Setup
  - Install NFVIS through CMC or USB
  - Verify Installation Using System API, CLI or NFVIS Portal

- **Upgrade (Optional)**
  - Register an Image
  - Upgrade an Image
  - Firmware Upgrade
  - Upgrade to the latest version of NFVIS/NSX

- **Host System Management**
  - System Access Configuration
  - Users, Roles and Authentication
  - Networks and Bridges
  - Cisco Network Plug-n-Play
  - Operate and Manage NFVIS Host

- **VM Life Cycle Management**
  - Image Registration
  - VM Deployment
  - Access VNFs
  - Image Packaging
  - Deploy service chains

- **Monitor, Troubleshoot and Debug**
  - Event Notifications
  - Syslog Messages
  - System Shell Access
  - Debug System Errors

NFVIS documentation are structured according to the stages an enterprise or service provider would go through when they decide to setup NFVIS:

- The Getting Started Guide follows the same flow as enterprise or service providers would, at the time of deploying Virtual Network Functions (VNFs) for the first time.
- **Set Up Cisco Enterprise NFVIS**: Provides detailed information on getting started with ENCS 5400 Series platform devices.

- **Install Cisco Enterprise NFVIS**: Provides information on how to install Cisco NFVIS through Cisco IMC and USB for the supported hardware platforms.

- **Upgrade Cisco NFVIS**: Provides information on how to upgrade Cisco NFVIS to the latest version of the release.

- The Configuration Guide provides detailed information once you have completed the basic installation and set up. This guide covers managing the host system, registering and managing VNFs, security considerations for your network, troubleshooting issues with NFVIS and so on.
  - **Host System Management**: Provides detailed information about operations and management of NFVIS host.
  - **VM Life Cycle Management, on page 77**: Provides information on the entire process of registering, deploying, updating, monitoring VMs, and getting them service chained as per your requirements.
  - **Troubleshoot and Debug Cisco NFVIS, on page 273**: Information provided here helps troubleshoot and debug system errors.
CHAPTER 3

Host System Management

- System Access Configuration, on page 9
- Users, Roles, and Authentication, on page 23
- Networking, on page 32
- Cisco Network Plug-n-Play Support, on page 48
- DPDK Support on NFVIS, on page 56
- Storage Access, on page 58
- Host System Operations, on page 59
- Backup and Restore NFVIS and VM Configurations, on page 62
- APC UPS Support and Monitoring, on page 70
- Resetting to Factory Default, on page 71
- Configure Banner, Message of the day and System Time, on page 72
- Configure DNS Name Servers, on page 74
- Configuring IP Host, on page 75

System Access Configuration

Host System Requirements

Table 7: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HugePage memory and CPU allocation</td>
<td>NFVIS 4.2.1</td>
<td>The system memory allocations are enhanced and all memory apart from the amount reserved for system is converted to HugePage memory. The VMs are deployed using hugepage memory even when DPDK is enabled on the system.</td>
</tr>
</tbody>
</table>

The following resources are required for a standalone Cisco Enterprise NFVIS:
Table 8: CPU Allocation

<table>
<thead>
<tr>
<th>Total Cores</th>
<th>Before NFVIS 3.12.x Release</th>
<th>NFVIS 3.12.x Without DPDK support</th>
<th>NFVIS 3.12.x and later Releases with DPDK support</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 or less</td>
<td>1</td>
<td>1</td>
<td>1 + (1 core per socket)</td>
</tr>
<tr>
<td>Between 12 and 16</td>
<td>2</td>
<td>1</td>
<td>1 + (1 core per socket)</td>
</tr>
<tr>
<td>(including 16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More than 16</td>
<td>4</td>
<td>2</td>
<td>2 + (1 core per socket)</td>
</tr>
</tbody>
</table>

**Note**  If hyper-threading is enabled on the device, each core reflects two logical CPUs.

Table 9: Memory Allocation

<table>
<thead>
<tr>
<th>Total System Memory</th>
<th>Reserved for NFVIS 3.12.x and earlier releases</th>
<th>Reserved for NFVIS 3.12.x and 4.1.x releases</th>
<th>Reserved for NFVIS 4.2.1 release</th>
<th>Reserved for NFVIS 4.2.2 and later releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 16 GB</td>
<td>3 GB</td>
<td>3 GB</td>
<td>3 GB</td>
<td>3 GB</td>
</tr>
<tr>
<td>Up to 32 GB</td>
<td>3 GB</td>
<td>3 GB</td>
<td>3 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Up to 64 GB</td>
<td>4 GB</td>
<td>4 GB</td>
<td>4 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Up to 128 GB</td>
<td>4 GB</td>
<td>8 GB</td>
<td>8 GB</td>
<td>8 GB</td>
</tr>
<tr>
<td>Greater than 128 GB</td>
<td>4 GB</td>
<td>8 GB</td>
<td>16 GB</td>
<td>16 GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total System Memory</th>
<th>Additional memory required for DPDK support per NUMA node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upto 63 GB</td>
<td>1</td>
</tr>
<tr>
<td>64 GB - 127 GB</td>
<td>2</td>
</tr>
<tr>
<td>128 GB - 256 GB</td>
<td>4</td>
</tr>
</tbody>
</table>

Starting from NFVIS 4.2 release, all memory apart from the amount reserved for system is converted to HugePage memory. The system memory is allocated on socket 0 on system with multiple CPU sockets.

**Note**  The additional memory required for DPDK support is counted per NUMA node available on the system.

System Setting Hostname

You must adhere to the following rules for hostname on NFVIS:
• Must contain minimum length of 2 and maximum length of 255.
• Must begin with a letter or digit and can contain alphabets, numbers and hyphen.
• Must not be deleted.

Dual WAN Support

Dual WAN support provides multiple links to NFVIS connectivity. Starting from NFVIS 3.10.1 release, a second WAN bridge configured with DHCP by default is supported on ENCS 5000 series platform.

During NFVIS system initialization, NFVIS attempts to establish connectivity through DHCP on both WAN bridges. This allows connectivity to NFVIS during initial deployment even if the network is down on one of the WAN bridges. Once DHCP assigns an IP address through one WAN bridge, the other WAN bridge can be configured with static IP address for connectivity to NFVIS.

Restrictions for Dual WAN Support

• The DHCP toggle behavior is not supported during the upgrade flow. It is only triggered during fresh installation of NFVIS or after a factory default reset.

• Active/standby or redundant WAN bridges are not supported. NFVIS does not detect connectivity failure from one WAN bridge and switchover to another WAN bridge. In case connectivity fails on the WAN bridge with DHCP configurations, connectivity through the other WAN bridge is established only if static IP is applied to the second WAN bridge and static routing is configured for connectivity through that bridge.

• IPv6 is not supported for dual WAN toggle.

• If wan2-br is DHCP enabled WAN bridge, you must remove DHCP from wan2-br to apply default gateway from static IP configurations.

Dual WAN Bridge and DHCP Toggle

Note
This feature is supported only on ENCS 5000 series devices.

In zero touch deployment, NFVIS requests for IPv4 assignments through DHCP for two WAN interfaces. During system initialization a second WAN bridge is configured with GE0-1 port attached. NFVIS toggles between the two default WAN bridges sending DHCP requests on any one of the WAN bridges at a time, for 30 second intervals. The toggling stops as soon as one WAN bridge is assigned an IP address through DHCP. The bridge with the assigned IP address is configured with DHCP. The other WAN bridge has no default IP configuration and can be manually configured with a static IP address if required.

If neither of the bridges is assigned an IP address through DHCP, the WAN DHCP toggle can be terminated by logging in to NFVIS using the default credentials. In this case, wan-br is configured with DHCP and wan2-br has no default IP configuration.

After zero touch deployment, the toggle feature is terminated and it is not possible to toggle between WAN bridges. To add additional connectivity to the NFVIS host, static IP address can be configured on the other WAN bridge and system static routing can be applied. A default gateway is not supported as the system default
gateway is set through DHCP. If DHCP configuration is not required, then both WAN bridges can be configured with static IP addresses, and a default gateway can then be applied under system settings.

**Accessing NFVIS**

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 returns 401 unauthorized error if the default password is not reset.

If wan-br or wan2-br have 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 DHCP assignment on wan-br until the configurations are changed. For DHCP assignment to continue to request IP address for PnP flow on both WAN interfaces answer 'n'.

You must adhere to the following 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 48).
- Using console (After the initial login using the default password, you are prompted to change the default password).

NFVIS Version: 3.12.3

Copyright (c) 2015-2020 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

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

login: admin
NFVIS service is OK
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
admin@localhost's password:

Cisco Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS Version: 3.12.3-RC8

Copyright (c) 2015-2020 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.
admin connected from ::1 using ssh on nfvis
admin logged with default credentials
Setting admin password will disable zero touch deployment behaviors.
Do you wish to proceed? [y or n]y
Please provide a password which satisfies the following criteria:
1. At least one lowercase character
2. At least one uppercase character
3. At least one number
4. At least one special character from # _ - * ?
5. Length should be between 7 and 128 characters
Please reset the password:
Please reenter the password:

Resetting admin password

New admin password is set
nfvis#
System message at 2020-01-08 03:10:10...
Commit performed by system via system using system.

To commit the target configuration to the active (running) configuration, use the commit command in any configuration mode. 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.

Connect 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 DHCP configuration and the management interface is configured with a static IP address of 192.168.1.1. If the system has a DHCP server connected to the WAN interface, the WAN interface is assigned an 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. However, to be able to use a static IP address to remotely connect to a server, the default gateway needs to be configured first.

You can connect to the system in the following 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 the Managed (M) flag, then the network stays in DHCP mode, even if there is no DHCP server in the network. If the RA shows the 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.

**Perform 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 the `no bridges bridge wan2-br dhcp` command.

If you want to disable DHCP and use static configuration, you need to perform the initial configuration 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:

```plaintext
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
```

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:

```plaintext
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
```

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.

Secure overlay is not supported when WAN interface is configured with IPv6.
To configure DHCP on the WAN interface:

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

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. You cannot have DHCP IPv6 and default IPv6 gateway 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
```

**Verify Initial Configuration**

Use the `show system settings-native` command 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.

Here is an 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
!
!
!
```

```
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 disabled
!
!
system settings-native gateway ipv4_address 209.165.201.1
system settings-native gateway interface wan-br
```

Here is an 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 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 gw ipv4_address 209.165.201.1
system settings-native gateway interface wan-br
```
Configuring VLAN for NFVIS Management Traffic

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

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

By default, WAN bridges and LAN bridges are in trunk mode and allows all VLANs. When you configure native VLAN, you must also configure all the allowed VLANs at the same time. The native VLAN becomes the only allowed VLAN if you do not configure all the VLANs. If you want a network that allows only one VLAN, then create another network on top of wan-net and lan-net and make it access network.

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.
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:

```
configure terminal
system settings ip-receive-acl 198.0.2.0/24
action accept priority 10
commit
```

**Verifying the Trusted IP Connection**

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

```
nfvvis# show running-config system settings ip-receive-acl
system settings ip-receive-acl 198.51.100.11/24
service [ ssh https scpd]
action accept
priority 100
```

**Port 22222 and Management Interface ACL**

Port 22222 is used for SCP server and is closed by default on an NFVIS system. You cannot SCP a file into NFVIS from an external server. If you need to SCP file from an external server, you must first open the port.

To open port 22222:

```
config terminal
system settings ip-receive-acl address/mask_len service scpd priority 2 action accept
commit
```

The Access Control List (ACL) is identify by address. If this ACL is removed, all ACLs sharing the same address are also removed. Ensure that you configure the ACLs that share the same address once again.

---

**Note**

From 3.8.1 release, only users with administrator privileges can use the SCP command on port 22222 to upload or download only from restricted folders like /data/intdatastore/. For more information, see Host System Operations, on page 59.

---

**Caution**

SCP command cannot be used to copy files from one NFVIS device to another NFVIS device.

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

```
nfvvis# show running-config system settings ip-receive-acl
system settings ip-receive-acl 10.156.0.0/16
service [ ssh https scpd ]
```
Configuring Secondary IP Address and Source Interface

Secondary IP Address
The Cisco Enterprise NFVIS supports multiple IP addresses per interface. You can configure a secondary IP address on the WAN interface, as an additional IP address to reach the software. Set the external routes for secondary IP address to reach the NFVIS. Routers configured with secondary addresses can route between the different subnets attached to the same physical interface.

To access secondary IP address through ISRv, the WAN physical port is removed from wan-br similar to single IP address.

To configure Secondary IP Address:

```config
nfvis(config)# system settings wan secondary ip address 1.1.2.3 255.255.255.0
```

Source Interface
The Source Interface feature lets you assign an IP address to a source interface. The IP address configured is used for packets generated by the NFVIS. The packets generated use the default route.

Prerequisites for configuring Source Interface
• The IP address must be one of the IP addresses configured in system settings.
• The source interface IP address can be one of the following:
  • mgmt
  • WAN
  • WAN Secondary IP
  • WAN2 IP or IP configured on any bridge
• Source-interface configuration must be applied if the WAN IP is static.
• For DHCP, source interface IP address is accepted but cannot be applied. The configuration takes effect once you switch from DHCP to static.

To configure source Interface:

```config
nfvis(config)# system settings source-interface 1.1.2.3
```

The secondary IP address and source interface related errors are logged in `show log nfvis_config.log` file.
Secondary IP Address and Source Interface APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/system/settings/wan/secondary</td>
<td>• system settings wan secondary</td>
</tr>
<tr>
<td>• /api/config/system/settings/source-interface</td>
<td>• system settings source-interface</td>
</tr>
</tbody>
</table>

CIMC Access Control

NFVIS administrators have authoritative control control over ENCS 5400 devices. This includes capability to change the IP address used to reach the CIMC and modifying the CIMC and BIOS passwords.

CIMC Access using NFVIS

Note

CIMC access using NFVIS is supported only on ENCS 5400.

When CIMC access is enabled on NFVIS, ISRv can gain access to the host CIMC and internal switch management console. You must have authorization from Cisco Interactive Debug (CID) to access both consoles.

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

For releases 3.8.1 and later, if the BIOS or CIMC versions on Cisco ENCS 5400 routers are lower than the image version in the NFVIS ISO or upgrade package, the BIOS and CIMC versions on the routers are automatically upgraded to the version of the bundled image during NFVIS upgrade or installation. The CPU microcode is also upgraded as part of this upgrade or installation. Note that the upgrade process takes longer than in previous releases and the process cannot be stopped midway.
For Cisco ENCS 5100 routers, BIOS is automatically upgraded to the new version, but the server needs to be rebooted manually for the upgrade to show.

## BIOS and CIMC Password

### Table 10: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS and CIMC password</td>
<td>NFVIS 4.2.1</td>
<td>New password restrictions and security measures are added for CIMC and BIOS.</td>
</tr>
</tbody>
</table>

To change the BIOS and CIMC password for ENCS 5400, use `hostaction change-bios-password newpassword` or `hostaction change-cimc-password newpassword` commands. The change in the password will take effect immediately after the commands are executed.

**Note**

New password restrictions added for CIMC and BIOS in NFVIS 4.2.1 release.

You must adhere to the following rules to create a strong password for CIMC:

- Must contain at least one upper case and one lower case letter.
- Must contain at least one number and one special character from #, @ or _.
- Length should be between 8 and 20 characters.
- Should not contain the following string (case sensitive): admin

You must adhere to the following rules to create a strong password for BIOS:

- The first letter cannot be #.
- Must contain at least one upper case and one lower case letter.
- Must contain at least one number and one special character from #, @ or _.
- Length should be between 8 and 20 characters.
- Should not contain the following string (case sensitive): bios

Starting from BIOS version 2.11 and CIMC 3.2.10, the new BIOS password security measures are:

- BIOS password can only be set through CIMC XML API or NFVIS. It can no longer be configured in the BIOS setup menu.
- BIOS password is retained after BIOS updates and it does not have to be reconfigured after a BIOS update.
- Only an admin password can be set and user-level BIOS password can no longer be set.
BIOS and CIMC Password APIs and Commands

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

UEFI Secure Boot on ENCS 5400

The Unified Extensible Firmware Interface (UEFI) Secure Boot mode ensures that all EFI drivers and applications, ROMs or operating systems are signed and verified for authenticity and integrity before they are loaded and executed. This feature can be enabled through the GUI or CLI. When you enable UEFI secure boot mode, the boot mode is set to UEFI mode and you cannot modify the configured boot mode until the UEFI boot mode is disabled.

Note

If you enable UEFI secure boot on an unsupported OS, on the next reboot, you cannot boot the device from that particular OS when you try to reboot the next time. If you try to reboot from such unsupported OS, an error is reported and recorded under System Software Events in the GUI. You must disable the UEFI secure boot option using Cisco IMC to be able to boot from the OS that does not support UEFI secure boot.

Enabling UEFI Secure Boot Mode

To enable UEFI secure boot mode:

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

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

Disabling UEFI Secure Boot Mode

To disable UEFI secure boot mode:

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

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

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

```
encs# scope bios
encs /bios # scope advanced
encs /bios/advanced # set BootOpRom UEFI
encs /bios/advanced # set BootOrderRules Loose
encs /bios/advanced *# commit
```
Reboot the device to start the installation.

To configure the UEFI virtual-mapped image as the first boot option, enter the BIOS menu using F2 key when BIOS boots up. Use direction keys to move UEFI: Cisco CIMC-mapped image or KVM-mapped image to the top of the boot option list. For BIOS v2.10 onwards, you can also configure the UEFI boot order through CIMC GUI or CLI. For more information see, Install Cisco Enterprise NFVIS.

---

**Note**

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

---

**PXE Boot Mode**

PXE (Preboot Execution Environment) boot mode is a new configuration option in the BIOS Advanced option list in CIMC, which can be configured like any other BIOS configuration option in the list. PXE boot mode allows you to configure PXE boot for legacy mode, UEFI mode, or disable it when not in use. Starting from NFVIS 4.5, BIOS 2.13 and CIMC 3.2.12.1, PXE boot is disabled by default.

It is recommended that you disable PXE boot mode when not using PXE, in order to gain boot time savings.

---

**Enable or Disable Access to NFVIS Portal**

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
```

---

**Note**

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

---

**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"
```

---

**Portal Access APIs and Commands**

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

Local User Account Management

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 11: 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 (*), or question mark (?).
- Must contain an uppercase character and a lowercase character.
- Must not be the 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".

Note

All user groups are created by the system. You cannot create or modify a user group.

Starting from NFVIS 3.9.1, create-user, delete-user, change-role and change-password operations are configurable from exec mode.

To create a user:
rbac authentication users create-user name test1 password Test1_pass role administrators

To delete a user:

rbac authentication users delete-user name test1

To change the password, use the `rbac authentication users user test1 change-password new-password newPassword old-password oldPassword` command. To change the user role to administrators, operators or auditors, use the `rbac authentication users user test1 change-role new-role newRole old-role oldRole` command.

### User Management APIs and Commands

<table>
<thead>
<tr>
<th>User Management APIs</th>
<th>User Management Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/rbac/authentication/users</td>
<td>• rbac authentication users</td>
</tr>
<tr>
<td>• /api/operations/rbac/authentication/users</td>
<td>• rbac authentication users user &lt;user-name&gt; change-password old-password &lt;old_pwd&gt; new-password &lt;new_pwd&gt;</td>
</tr>
<tr>
<td>/user/&lt;user-name&gt;/change-password</td>
<td>• rbac authentication users user &lt;user-name&gt; change-role old-role &lt;old_role&gt; new-role &lt;new_role&gt;</td>
</tr>
<tr>
<td>• /api/operations/rbac/authentication/users/user</td>
<td>• rbac authentication users create-user name &lt;user-name&gt; password &lt;password&gt; role &lt;role&gt;</td>
</tr>
<tr>
<td>/&lt;user-name&gt;/change-role</td>
<td>• rbac authentication users delete-user name &lt;user-name&gt;</td>
</tr>
<tr>
<td>• /api/operations/rbac/authentication/users/create-user</td>
<td></td>
</tr>
<tr>
<td>• /api/operations/rbac/authentication/users/delete-user</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/min-pwd-length</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>

**NFVIS Password Recovery**

1. Load the NFVIS ISO 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.
   Disconnect the NFVIS ISO image in the CIMC KVM console 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.

---

**Local Authentication for a Specific Group of Users**

**Table 12: Feature History**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Authentication for a Specific Group of Users</td>
<td>NFVIS 4.6.1</td>
<td>This feature enables you to create a group with specific users, who can perform only the local authentication; and don't have to authenticate externally through TACACS.</td>
</tr>
</tbody>
</table>

**Restrictions for Local Authentication for a Specific Group of Users**

- One user can be assigned to a maximum of one group.
Information about Local Authentication for a Specific Group of Users

Typically, users who need authentication go through the default authentication order. The default order involves external authentication through TACACS as the first step, and local authentication as the second step. The local authentication for a specific group of users feature enables you to create a group and add specific users to it, allowing these users to bypass the default authentication order. The users in this group skip the external TACACS authentication and only go through local authentication. For this group to function as expected, assign the 'local-authentication-only' policy to the group.

Create Groups and Assign Users to the Groups

The following example shows how to create a group:

```bash
nfvis#(config) aaa groups group [ group-name ] policy local-authentication-only
```

The following example shows how to assign a user:

```bash
nfvis# rbac authentication users user <username> assign-group [ group-name ]
```

The following example shows how to remove a user:

```bash
nfvis# rbac authentication users user <username> remove-group [ group-name ]
```

The following example shows how to assign a user to a group while creating the user:

```bash
nfvis# rbac authentication users create-user name <username> password <password> role <role> group [ group-name ]
```

RADIUS Support

About RADIUS

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

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

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

Note

You can configure up to four RADIUS servers. When multiple RADIUS servers are configured, if the first server is unreachable, NFVIS tries the next server in the order it is configured.

RADIUS Operation

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

1. The user is prompted to enter the username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.

3. The user receives one of the following responses from the RADIUS server:
   a. ACCEPT—The user is authenticated.
   b. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   c. CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.
   d. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.

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

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

**Configuring RADIUS**

To configure RADIUS support:

```
configure terminal
radius-server host 103.1.4.3
shared-secret cisco123
admin-priv 15
oper-priv 11
commit
```

Starting from NFVIS 3.9.2 release, RADIUS secret encryption is supported. You can only configure either secret key or encrypted secret key at a given time. Use encrypted secret if special characters are used in secret. To configure encrypted RADIUS secret:

```
configure terminal
radius-server host 103.1.4.3
encrypted-shared-secret cisco123
admin-priv 15
oper-priv 11
commit
```

**Verifying the RADIUS configuration**

Use the `show running-config radius-server` command to verify the interface configuration for a RADIUS session:

```
nfvis# show running-config radius-server

radius-server host 103.1.4.3
key 0
shared-secret cisco123
```
RADIUS Support APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/security_servers/radius-server</td>
<td>radius-server host</td>
</tr>
<tr>
<td></td>
<td>key</td>
</tr>
<tr>
<td></td>
<td>admin-priv</td>
</tr>
<tr>
<td></td>
<td>oper-priv</td>
</tr>
<tr>
<td></td>
<td>encrypted-shared-secret or shared-secret</td>
</tr>
</tbody>
</table>

TACACS+ Support

About TACACS+

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

On the TACACS+ server, ensure you configure Cisco attribute-value (AV) pair privilege level (priv-lvl) for Cisco Enterprise NFVIS service for the minimum privilege level of administrators and operators.

Note

In NFVIS 3.11.1 or earlier release, users with no privilege level or users with a privilege level that is less than the operator's privilege level are considered as auditors with read-only permission.

After NFVIS 3.12.1 release, users with privilege level zero won't be able to login to NFVIS anymore.

Note

You can configure up to four TACACS+ servers. When multiple TACACS+ servers are configured, if the first server is unreachable, NFVIS tries the next server in the order it is configured.

TACACS Operation

When a user attempts a simple ASCII login by authenticating to NFVIS using TACACS+, this process occurs:

1. When the user tries to log in, NFVIS sends user credential to TACACS+ server.
2. NFVIS will eventually receive one of the following responses from the TACACS+ server:
   a. ACCEPT—The user is authenticated and service can begin. If NFVIS is configured to require authorization, authorization begins at this time.
   b. REJECT—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ server.
c. **ERROR**—An error occurred at some time during authentication with the server or in the network connection between the server and NFVIS. If an ERROR response is received, NFVIS typically tries to use an alternative method for authenticating the user.

d. **CONTINUE**—The user is prompted for additional authentication information.

After authentication, NFVIS will send authorization request to TACACS+ server.

3. Based on authorization result, NFVIS will assign user's role.

### Configuring a TACACS+ Server

To configure TACACS+:

```plaintext
configure terminal
tacacs-server host 209.165.201.20 shared-secret test1
key 0
    admin-priv
        14
oper-priv
    9
commit
```

In this configuration, privilege level 14 is assigned to the administrator role, and privilege level 9 is assigned to the operator role. This means a user with privilege level 14 or higher will have all admin privileges when the user logs into the system, and a user with privilege level 9 or higher will have all privileges of an operator at the time of login.

Starting from NFVIS 3.9.2 release, TACACS+ secret encryption is supported. You can only configure either secret key or encrypted secret key at a given time. Encrypted secret key can contain special characters but secret key cannot. For NFVIS 3.12.1 release, the following pattern is supported for encrypted-shared-key: `[-_a-zA-Z0-9//<>%!*$€#{}()+]`.

To configure encrypted TACACS+ key:

```plaintext
configure terminal
tacacs-server host 209.165.201.20 encrypted-shared-secret test1
key 0
    admin-priv
        14
oper-priv
    9
commit
```

### Verifying the TACAC+ configuration

Use the `show running-config tacacs-server` command to verify the configuration if encrypted TACACS+ key is configured:

```plaintext
nfvis# show running-config tacacs-server
```
tacacs-server host 209.165.201.20
encrypted-shared-secret $8$mRTnL9TKZCFi1BUP7Mwbm3JVjO4Z7QvJ
admin-priv 15
oper-priv 11

TACACS+ APIs and Commands

<table>
<thead>
<tr>
<th>TACACS+ APIs</th>
<th>TACACS+ Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/config/security_servers/tacacs-server</td>
<td>• tacacs-server host</td>
</tr>
<tr>
<td>• /api/config/security_servers/tacacs-server?deep</td>
<td>• key</td>
</tr>
<tr>
<td>• /api/config/security_servers/tacacs-server</td>
<td>• admin-priv</td>
</tr>
<tr>
<td>/host/&lt;ip-address/domain-name&gt;</td>
<td>• oper-priv</td>
</tr>
<tr>
<td></td>
<td>• encrypted-shared-secret or shared-secret</td>
</tr>
</tbody>
</table>

**Default Authentication Order**

NFVIS supports both TACACS+ and RADIUS but only one authentication method can be enabled at a time. After you have identified the TACACS+ and RADIUS server and defined an associated TACACS+ and RADIUS authentication key, you must define method lists for TACACS+ and RADIUS authentication. Because TACACS+ and RADIUS authentication is operated through AAA, you need to issue the `aaa authentication` command, specifying TACACS+ or RADIUS as the authentication method.

```
nfv<config>% aaa authentication ?
Possible completions:
radius Use RADIUS for AAA

tacacs Use TACACS+ for AAA

users List of local users
```

- Only when TACACS+ or RADIUS is enabled, it can be used for authentication.
- When TACACS+ or RADIUS is not accessible, local authentication is used. It is recommended to use `aaa authentication TACACS local` command to authenticate using local database. Local authentication is disabled if the connection between TACACS+ or RADIUS and NFVIS is restored.
- If the same username is registered for both local authentication and authentication through RADIUS or TACACS+, RADIUS or TACACS+ is chosen as the authentication method.
- It is recommended to configure `Syslog, on page 249` so that it is easier to debug if TACACS+ or RADIUS does not work as expected.

All login attempts will be logged in syslogs in the local `nfvis_syslog.log`, `nfvis-ext-auth.log` files and in remote syslog servers.
**Enchancements for NFVIS 3.12.3, User Specific Authentication Order**

Starting from NFVIS 3.12.3 release, the supported aaa authentication order is local authentication followed by TACACS+.

- If the user is part of the local database, local authentication is executed and the user is permitted or denied access.
- If the user is not part of the local database, TACACS+ is used for authentication.
- If the same user is part of both the databases (local and TACACS+), the user can login with either the local password or the TACACS+ password. However, registering the same user in both the databases is not recommended.

In NFVIS 3.12.3 release, the only supported combination for authentication order is `aaa auth-order local tacacs`. Any other combinations are not supported. `aaa auth-order` configuration is mutually exclusive to `aaa authentication` and if one is configured, the other is automatically replaced.

```
nfvis(config)# aaa ?
Possible completions:
auth-order Configure authentication order; Mutually exclusive to authentication method configuration
authentication Configure external authentication method; Mutually exclusive to auth-order configuration
ios Specific IOS settings

nfvis(config)# aaa auth-order ?
Description: Configure authentication order; Mutually exclusive to authentication method configuration
Possible completions:local radius tacacs
```

**Networking**

**Bridges**

The IP configuration on bridges and the `show bridge-settings` command were added in NFVIS 3.10.1 release. A bridge can be used for NFVIS connectivity. Each bridge can be configured with IPv4 or IPv6 configurations such as Static IP, DHCP, SLAAC, or VLAN. Each bridge can have a port or port channel associated with it.

NFVIS is installed with LAN and WAN bridges by default. A service bridge can also be created. On all NFVIS systems, lan-br and wan-br are generated by default and populated with the appropriate ports for that system. On ENCS 5000 series platforms, wan2-br is also generated by default for the dual WAN initialization. For more information, see `Dual WAN Support, on page 11`. Except on ENCS 5000 Series platforms, the default LAN bridge is configured with a static IP address 192.168.1.1 and the WAN bridges uses DHCP for initial NFVIS connectivity.
On ENCS 5400 series platforms, configuration changes are not allowed on the lan-br bridge. The LAN bridge cannot be modified in any way.

**Using IPv4**

If the system has a DHCP server connected to a bridge with DHCP configured, the bridge receives the IP address from the server. You can use this IP address to connect to the system.

You can also connect to the server locally with an Ethernet cable using a static IP address. To connect to the device remotely using a static IP address, you must configure the default gateway or setup an appropriate static route.

DHCP and a default gateway cannot be configured on NFVIS simultaneously. NFVIS only supports one system level default gateway. If DHCP is configured, the default gateway is assigned to the system through the DHCP server. Also, only one bridge can be configured with DHCP at any time.

**Using IPv6**

IPv6 can be configured in static, DHCP stateful, and Stateless Auto configuration (SLAAC) modes. 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 the Managed (M) flag, then the network stays in DHCP mode, even if there is no DHCP server in the network. If the RA shows the Other (O) flag, then the network switches from DHCP server to SLAAC mode.

SLAAC provides IPv6 address and a default gateway. Stateless DHCP is enabled in the SLAAC mode. If the server has DNS and domain configured, then SLAAC also provides those values through stateless DHCP.

Similar to IPv4, IPv6 DHCP and IPv6 default gateway cannot be configured on the system simultaneously, nor can stateful and stateless IPv6 DHCP. Also, only one bridge can be configured with either stateful or stateless IPv6 DHCP at any time.

**Creating Bridges**

To configure a new bridge:

```bash
configure terminal
bridges bridge my-br
commit
```

To verify the bridge generation, use the `show bridge-settings` command:

```bash
tfvis# show bridge-settings my-br ip-info interface
ip-info interface my-br
```

**Configuring Bridge Port**

A bridge can be tied to a physical interface by applying the port configuration. A bridge can have as many ports as are available, however a port must be unique to at most one bridge. If a port channel is applied to a bridge, it must be the only port configuration on that bridge.

To configure a port on a bridge:

```bash
configure terminal
bridges bridge my-br port eth3
commit
```
To configure a port channel on a bridge:

```
configure terminal
bridges bridge my-br port pc1
commit
```

To verify the port settings applied to a bridge, use the `support ovs vsctl` command:

```
nfvis# support ovs vsctl list-ports my-br
eth3
```

The same command can be used to verify the port channel settings applied to a bridge:

```
nfvis# support ovs vsctl list-ports my-br
bond-pc1
```

### Configuring Bridge IP Connectivity

#### Configuring DHCP on Bridge

DHCP configuration can be applied to any bridge if no other bridge on the system has DHCP configured, and default gateway is not applied under system settings. Starting from NFVIS 3.12.1 release, DHCP configuration on a bridge automatically triggers a DHCP renew request from the bridge. For an additional DHCP renew trigger, use the `hostaction bridge-dhcp-renew` command.

To configure DHCP on a bridge:

```
configure terminal
bridges bridge my-br dhcp
commit
```

To verify the DHCP settings applied to a bridge, use the `show bridge-settings <br_name> dhcp` command.

```
nfvis# show bridge-settings my-br dhcp
dhcp enabled
dhcp offer              true
dhcp interface         my-br
dhcp fixed_address     10.10.10.14
dhcp subnet_mask       255.255.255.128
dhcp gateway           10.10.10.1
dhcp lease_time        7200
dhcp message_type      5
dhcp name_servers      NA
dhcp server_identifier 10.10.10.1
dhcp renewal_time      3600
dhcp rebinding_time    6300
dhcp vendor_encapsulated_options NA
dhcp domain_name       NA
dhcp renew             2019-12-11T13:28:29-00:00
dhcp rebind            2019-12-11T14:17:12-00:00
dhcp expire            2019-12-11T14:32:12-00:00
```

#### Configuring Static IP on a Bridge

An IPv4 address and subnet can be configured on any bridge which does not have DHCP configured. To enable routing outside of the subnet, apply the default gateway under system settings or configure system routes.
To configure an IPv4 address on a bridge:

```
configure terminal
bridges bridge my-br ip address 172.25.220.124 255.255.255.0
commit
```

To verify the IPv4 settings applied to a bridge, use the `show bridge-settings <br_name> ip_info` command.

```
nfvis# show bridge-settings my-br ip_info
ip-info interface my-br
ip-info ipv4_address 172.25.220.124
ip-info netmask 255.255.255.0
ip-info link-local ipv6 address fe80::4e00:82ff:fead:e802
ip-info link-local ipv6 prefixlen 64
ip-info global ipv6 address::
ip-info global ipv6 prefix len0
ip-info mac_address 4c:00:82:ad:e8:02
ip-info mtu 9216
ip-info txqueuelen 1000
```

**Configuring IPv6 DHCP on a Bridge**

IPv6 DHCP configuration can be applied to any bridge if no other bridge on the system has IPv6 DHCP or IPv6 SLAAC configured, and IPv6 default gateway is not applied under system settings. Starting from NFVIS 3.12.1 release, an IPv6 DHCP configuration on a bridge automatically triggers an IPv6 DHCP renew request from the bridge. For an additional IPv6 DHCP renew trigger use the `hostaction bridge-dhcp-renew` command.

To configure IPv6 DHCP on a bridge:

```
configure terminal
bridges bridge my-br dhcp-ipv6
commit
```

To verify the IPv6 DHCP settings applied to a bridge, use the `show bridge-settings <br_name> dhcp-ipv6` command.

```
nfvis# show bridge-settings my-br dhcp-ipv6
dhcp-ipv6 offer true
dhcp-ipv6 interface my-br
dhcp-ipv6 ia-naec:d2:7d:b4
dhcp-ipv6 starts 1554792146
dhcp-ipv6 renew 43200
dhcp-ipv6 rebinding 69120
dhcp-ipv6 iaaddr 2001:420:30d:201::ffff:ffff:fffa:8e48
dhcp-ipv6 preferred-life 86400
dhcp-ipv6 max-life 172800
dhcp-ipv6 server-id 0:3:0:1:0:25:45:1bc2:2a
dhcp-ipv6 name_servers NA
dhcp-ipv6 domain_name NA
dhcp-ipv6 option [ ]
```

**Configuring IPv6 SLAAC on a Bridge**

IPv6 SLAAC configuration can be applied to any bridge if no other bridge on the system has IPv6 SLAAC or IPv6 DHCP configured, and IPv6 default gateway is not applied under system settings.

To configure IPv6 SLAAC on a bridge:
configure terminal
bridges bridge my-br slaac-ipv6
commit

To verify the IPv6 SLAAC settings applied to a bridge, use the show bridge-settings <br_name> slaac-ipv6 command.

nfvis# show bridge-settings my-br slaac-ipv6
slaac-ipv6 enabled

**Configuring Static IPv6 Address on a Bridge**

An IPv6 address can be configured on any bridge which does not have IPv6 DHCP or SLAAC configured. To enable routing outside of the subnet, apply the default gateway under system settings or configure system routes.

To configure an IPv6 address on a bridge:

configure terminal
bridges bridge my-br ipv6 address 2001:db8:85a3::8a2e:370:7334/64
commit

To verify the IPv6 settings applied to a bridge, use the show bridge-settings <br_name> ip_info command.

nfvis# show bridge-settings my-br ip_info
ip-info interface my-br
ip-info ipv4_address 172.25.220.124
ip-info netmask 255.255.255.0
ip-info link-local ipv6 address fe80::4e00:82ff:fead:e802
ip-info link-local ipv6 prefixlen 64
ip-info global ipv6 address 2001:db8:85a3::8a2e:370:7334
ip-info global ipv6 prefixlen 64
ip-info mac_address 4c:00:82:ad:e8:02
ip-info mtu 9216
ip-info txqueuelen 1000

**Configuring VLAN on a Bridge**

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

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

By default, Wan bridge and LAN bridge are in trunk mode and allows all VLANs. When you configure native VLAN, you must also configure all the allowed VLANs at the same time. The native VLAN becomes the only allowed VLAN if you do not configure all the VLANs. If you want a network that allows only one VLAN, then create another network on top of wan-net and lan-net and make it access network.

**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
```

To verify the VLAN settings applied to a bridge, use the `show bridge-settings my-br vlan` command.

```
nfvis# show bridge-settings my-br vlan
vlan tag 10
```

### Configuring MAC Aging Time on Bridge

MAC aging time specifies the time at which a MAC address entry ages out of the MAC address table. The max-aging-time specifies the maximum number of seconds to retain a MAC learning entry for which no packets have been seen. The default value is 300 seconds.

To configure MAC aging time on a bridge:

```
configure terminal
bridges bridge my-br mac-aging-time 600
commit
```

To verify the MAC aging time settings applied to a bridge, use the `show bridge-settings <br_name> mac-aging-time` command.

```
nfvis# show bridge-settings my-br mac-aging-time
mac-aging-time 600
```

### Bridge APIs and Commands

<table>
<thead>
<tr>
<th>Bridge APIs</th>
<th>Bridge Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/operational/bridge-settings</td>
<td>bridges bridge &lt;br_name&gt;</td>
</tr>
<tr>
<td>/api/config/bridges/bridge/</td>
<td>bridges bridge &lt;br_name&gt; port</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; ip address</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; dhcp</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; ipv6 address</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; dhcp-ipv6</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; slaac-ipv6</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; vlan</td>
</tr>
<tr>
<td></td>
<td>bridges bridge &lt;br_name&gt; mac-aging-time</td>
</tr>
<tr>
<td></td>
<td>show bridge-settings &lt;br_name&gt;</td>
</tr>
<tr>
<td></td>
<td>support ovs vsctl list-ports &lt;br_name&gt;</td>
</tr>
</tbody>
</table>

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
Physical Network Interface Cards

Configuring LLDP

Starting from NFVIS 3.7.1 release, LLDP is supported on NFVIS. The Link Layer Discovery Protocol (LLDP) is used by network devices for advertising their identity, capabilities, and neighbors. You can configure LLDP on a PNIC which is not a port channel or a DPDK port. By default, LLDP is disabled for all PNICs.

LLDP information is sent by devices from each of their interfaces at a fixed interval, in the form of an Ethernet frame. Each frame contains one LLDP Data Unit (LLDPDU). Each LLDPDU is a sequence of type-length-value (TLV) structures.

LLDP is enabled in transmit and receive mode. The LLDP agent can transmit the local system capabilities and status information and receive the remote system's capabilities and status information.

If LLDP is enabled on two connected devices, they can see each other as neighbors.

Note

LLDP packets are not propagated to VMs. LLDP cannot be enabled on port channel or DPDK ports.

To enable LLDP on a PNIC:

```bash
configure terminal
pnic eth0 lldp enabled
commit
```

To disable LLDP on a PNIC:

```bash
configure terminal
pnic eth0 lldp disabled
commit
```

Use the `show lldp neighbors` command to display the peer information:

```bash
nfvis# show lldp neighbors eth0
```

```
DEVICE
NAME ID HOLDTIME CAPS PLATFORM PORTID DESCRIPTION
-------------------------------------------------------------------
eth0 Switch1623 120 Bridge, Router Cisco IOS Software, Catalyst L3 Switch Software (CAT3K_CAA-UNIVERSALK9-M), Version 15.0(1)EX3, RELEASE SOFTWARE (fc2)Ifname: Gi1/0/4GigabitEthernet1/0/4
```

Use the `show lldp stats` command to display the tx and rx information:

```bash
nfvis# show lldp stats eth0
```

```
TX DISCARD ERROR RX DISCARDED UNREC
NAME FRAMES RX FRAMES TLVS TLVS AGEOUTS
-------------------------------------------------------------------
eth0 23 0 0 19667 0 0 0
```
LLDP Configuration APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/&lt;pnic_name&gt;/lldp</td>
<td>pnic &lt;pnic_name&gt; lldp enabled</td>
</tr>
<tr>
<td>/api/operational/lldp/neighbors</td>
<td>pnic &lt;pnic_name&gt; lldp disabled</td>
</tr>
<tr>
<td>/api/operational/lldp/stats</td>
<td>show lldp neighbors &lt;pnic_name&gt;</td>
</tr>
<tr>
<td></td>
<td>show lldp stats &lt;pnic_name&gt;</td>
</tr>
</tbody>
</table>

Configuring Administrative Status of a Port

Administrative status provides a mechanism for configuring the administrative status of a port. It can be set to up or down and the default setting is on.

Note

Administrative status cannot be enabled on port channel.

To configure the admin status on a pnic for a VM:

configure terminal
pnic GE0-1 admin status down
commit

Use the `show pnic` command to verify the admin status configuration. Use the `show pnic link_state` command to verify the admin state configuration.

nfvis# show pnic GE0-1 link_state
link_state down

Admin Status Configuration APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/&lt;pnic_name&gt;/adminstatus</td>
<td>pnic &lt;pnic_name&gt; adminstatus</td>
</tr>
</tbody>
</table>

Tracking Changes for a Port

This feature is supported only on ENCS 5400 starting from NFVIS 3.10.1 release.

In a virtual environment when the PNIC goes down there is no indication to the interfaces inside the VNFs. It is useful to track state changes of PNICs including switch ports to one or more VNF interfaces and accordingly bring down or up the vNICs. This feature brings the appropriate interfaces inside the VNF up or down based on the PNIC state changes. Most of the VNFs support this functionality.

Track state can also be configured for LAN-SRIOV. The LAN network is not physically connected to LAN-SRIOV. Switch ports are connected to an embedded switch on the LAN side. The switch has an int-LAN interface which is a 10G interface the VMs can connect to from the LAN network using VFs (virtual functions). Therefore, the VM is not directly connected to LAN-SRIOV.
Track state configuration on WAN-SRIOV is not needed, as there is a one to one connection between WAN-SRIOV and the VM.

Track state can be configured for monitored and un-monitored VMs. If a track state configuration is deleted, the PNIC or switch port state changes will not be notified to the vNICs or VFs.

The VM has to be first deployed before you can configure PNIC track state for the VM. VNFs or vNICs do not have to be attached to a bridge connected to the PNIC.

To configure track state on a pnic for a VM use the following commands:

```
pnic <pnic_name> track-state <vm_name> <vnic>
```

or

```
pnic <pnic_name> track-state <deploy_name.vm_grp_name> <vnic>
```

```
configure terminal
pnic GEO-0 track-state ROUTER 0
end
```

To verify the track state configuration on the VM use the `show interface` or `ethtool` commands or the VM specific command that displays the interface link state.

In the following example, the edgew VM deployed and vNIC 0 is being tracked by GEO-1. The `if-oper-status` command shows the state of the vNIC being tracked by pNIC. When GEO-1 is down, `if-oper-status` also shows as down.

### Track State APIs and Commands

<table>
<thead>
<tr>
<th>Track State APIs</th>
<th>Track State Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>api/config/pnics/pnic/&lt;pnic_name&gt;/track-state</td>
<td>• pnic &lt;pnic_name&gt; track-state &lt;vm_name&gt; &lt;vnic&gt;</td>
</tr>
<tr>
<td></td>
<td>• pnic &lt;pnic_name&gt; track-state &lt;vm_name&gt; &lt;vnic&gt;</td>
</tr>
<tr>
<td></td>
<td>• pnic &lt;pnic_name&gt; track-state &lt;deploy_name.vm_grp_name&gt; &lt;vnic&gt;</td>
</tr>
</tbody>
</table>

### Speed, Duplex and Autonegotiation

NFVIS supports autonegotiation by default on all PNICs. Speed and duplex are set to `auto` mode to indicate autonegotiation is enabled.

Autonegotiation allows a PNIC to communicate with the device on the other end of the link to determine the optimal duplex mode and speed for the connection. Autonegotiation can be turned off by configuring speed and duplex. Supported Ethernet speed is 10 Mbps, 100 Mbps, and 1G and 10 G.

Duplex mode displays the data flow on the interface. Duplex mode on an interface can be full or half duplex. A half-duplex interface can only transmit or receive data at any given time and a full-duplex interface can send and receive data simultaneously.

When autonegotiation is enabled on a port, it does not automatically determine the configuration of the port on the other side of the ethernet cable to match it. Autonegotiation only works if it is enabled on both sides of the link. If one side of a link has auto-negotiation enabled, and the other side of the link does not, then autonegotiation cannot determine the speed and duplex configurations of the other side. If autonegotiation is enabled on the other side of the link, the two devices decide together on the best speed and duplex mode. Each interface advertises the speed and duplex mode at which it can operate, and the best match is selected. Higher speed and full duplex is the preferred mode.
If one side of a link does not have autonegotiation enabled, then the speed and duplex on both sides must match so that the data can transmit without collisions. Autonegotiation fails on 10/100 links, if one side of the link has been set to 100/full, and the other side has been set to autonegotiation which is 100/half.

Note

Not all ports on ENCS 5000 series devices support auto-mdix feature. When autonegotiation is disabled, you need to use the correct cable to configure speed and duplex correctly. The cable type depends on the remote system, based on which you can try straight through or cross over cable.

To disable autonegotiation on a PNIC, speed and duplex must be configured:

```
configure terminal
pnic GE0-0 speed 100 duplex full
commit
```

To enable autonegotiation on a PNIC:

```
configure terminal
pnic GE0-0 speed auto duplex auto
commit
```

To configure speed and duplex with non auto values:

```
configure terminal
pnic GE0-0 speed 100 duplex full
commit
```

Use the `show pnic GE0-0 operational-speed`, `show pnic GE0-0 operational-duplex` and `show pnic GE0-0 autoneg` to verify the configurations.

```
nfvvis# show pnic GE0-0 operational-speed
operational-speed 100
```

```
nfvvis# show pnic GE0-0 operational-duplex
operational-duplex full
```

```
nfvvis# show pnic GE0-0 autoneg
autoneg off
```

To verify the PNIC speed and duplex configurations, use the `show notification stream nfvis Event` command.

```
notification
event Time 2019-12-16T22:52:49.238604+00:00
nfvisEvent
  user_id admin
  config_change true
  transaction id 0
  status FAILURE
  status_code 0
  status_message Pnic GE0-1 speed did not update successfully
  details NA
  event_type PNIC_SPEED_UPDATE
  severity INFO
  host_name nfvis
```

!
Port Channels

Information About Port Channels

Port channels combine individual links into a group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. Creating port channels helps to increase bandwidth and redundancy and to load balance traffic between the member ports. If a member port within a port channel fails, the traffic from the failed port switches to the remaining member ports.

Port channels must have at least two ports and can be configured using static mode or Link Access Control Protocol (LACP). Configuration changes that are applied to the port channel are applied to each member port of the port channel. A port channel can also be added to a bridge. When a port channel has two or more than two members and the port channel is added to a bridge, a bond is created.

A port can be a member of only one port channel and all the ports in a port channel must be compatible. Each port must use the same speed and operate in full-duplex mode.

Port Channels Bond Mode

A port channel can be configured for the following bond modes:

- **active-backup**: In this mode, one of the ports in the aggregated link is active and all others ports are in the standby mode.

- **balance-slb**: In this mode, load balancing of traffic is done based on the source MAC address and VLAN.

- **balance-tcp**: In this mode, 5-tuple (source and destination IP, source and destination port, protocol) is used to balance traffic across the ports in an aggregated link.

<table>
<thead>
<tr>
<th>Speed, Duplex and Autonegotiation APIs</th>
<th>Speed, Duplex and Autonegotiation Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics/pnic/GE0-0/speed</td>
<td>pnic GE0-0 speed auto duplex auto</td>
</tr>
<tr>
<td>/api/config/pnics/pnic/GE0-0/duplex</td>
<td>pnic GE0-0 speed 100 duplex full show</td>
</tr>
<tr>
<td>/api/operational/pnics/pnic/GE0-0/operational-speed</td>
<td>show pnic GE0-0 operational-speed</td>
</tr>
<tr>
<td>/api/operational/pnics/pnic/GE0-0/operational-duplex</td>
<td>show pnic GE0-0 operational-duplex</td>
</tr>
<tr>
<td>/api/operational/pnics/pnic/GE0-0/autoneg</td>
<td>show pnic GE0-0 autoneg</td>
</tr>
</tbody>
</table>
Port Channels LACP Mode

A port channel can be configured for the following LACP modes:

- **off**: Indicates that no mode is applicable.
- **active**: Indicates that the port initiates transmission of LACP packets.
- **passive**: Indicates that the port only responds to the LACP packets that it receives but does not initiate the LACP negotiation.

### Configuring a Port Channel

#### Creating a Port Channel

To create a port channel:

```
configure terminal
pnic egroup type port_channel lACP_type active bond_mode balance-tcp trunks 10,20
commit
```

#### Adding a Port to a Port Channel

You can add a port to a new port channel or a port channel that already contains ports. To add a port to a port channel:

Adding GE0-0 and GE0-1 to egroup:

```
configure terminal
pnic GE0-0 member_of egroup
commit
```

```
configure terminal
pnic GE0-1 member_of egroup
commit
```

#### Adding a Port Channel to a Bridge

You can add a port channel to a new bridge or an existing bridge. When a port channel is added to a bridge, a bond is added for the port channel.

To add a port channel to a bridge:

```
configure terminal
bridges bridge test-br port egroup
commit
```

#### Deleting a Port Channel

Before deleting a port channel, you must remove all members assigned to the port channel. If the port channel is configured on the bridge, you must remove the port channel from the bridge.

1. Remove ports from port channel. If GE0-0 and GE0-1 are part of port channel pc, remove them from pc first.

```
configure terminal
```
2. Remove port channel from the bridge.

   ```
   configure terminal
   no bridges bridge test-br port egroup
   commit
   ```

3. Delete port channel.

   ```
   configure terminal
   no pnic egroup
   commit
   ```

### Verifying Port Channel Configurations

To verify port channel configurations, use the `show port-channel` command.

```
nfvis# show port-channel

----bond-egroup----
bond_mode: balance-tcp
bond may use recirculation: yes, Recirc-ID : 1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
next rebalance: 6921 ms
lacp_status: negotiated >>> this should be negotiated to indicate port channel is active
lacp_fallback_ab: false
active slave mac: 38:90:a5:1b:fe:0d(GE0-1)>>should indicate active slave mac address
slave GE0-0: enabled
may_enable: true
slave GE0-1: enabled
active slave port >>> active slave port should show active
may_enable: true
```

### Port Channel APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/pnics</td>
<td>pnic &lt;port_channel_name&gt; type port_channel</td>
</tr>
<tr>
<td>/api/config/pnics/pnic/&lt;pnic_name&gt;/member_of</td>
<td>pnic &lt;pnic_name&gt; member_of</td>
</tr>
<tr>
<td>/api/config/pnics/pnic/&lt;pnic_name&gt;/bond_mode</td>
<td>&lt;portchannel_name&gt;</td>
</tr>
<tr>
<td>/api/config/pnics/pnic/&lt;pnic_name&gt;/trunks</td>
<td>show port-channel</td>
</tr>
</tbody>
</table>
Promiscuous mode

Starting from NFVIS 4.1.1 release, NFVIS allows enabling promiscuous mode on interfaces. Enabling promiscuous mode on an interface can be used to monitor all incoming packets on the interface.

To enable promiscuous mode:

```
nfvis# config terminal
nfvis(config)# pnic GE0-0 promiscuous enabled
nfvis(config-pnic-GE0-0)# commit
```

Use the `show pnic GE0-0 operational-promiscuous` command to verify if promiscuous mode has been enabled.

---

**Note**

When an interface is connected to a bridge, NFVIS enables promiscuous mode on the interface.

---

Dynamic SR-IOV

Dynamic Single-root input/output virtualization (SR-IOV) allows you to enable or disable SR-IOV on a Physical Network Interface Controller (PNIC). To disable SR-IOV on a PNIC, set the SR-IOV value to 0. To enable SR-IOV on a PNIC, set the SR-IOV value between 1 and the maximum number of virtual functions (maxvfs) supported on that PNIC. You can also create and delete SR-IOV networks based on the number of virtual functions (numvfs) set on that PNIC while enabling SR-IOV. The existing fresh installation behavior has not changed. Each PNIC has a number of VFs and SR-IOV networks created by default. You can use CLI, API, or the GUI to enable and disable SR-IOV on a PNIC and to create and delete SR-IOV networks.

---

**Note**

The number of SR-IOV networks, numvfs or inusevfs, created per PNIC on fresh installation of NFVIS depends on the link speed of that particular pnic.

---

Restrictions and Limitations

- The supported platforms are CSP-2100, CSP-5228, CSP-5436, CSP-5444 (Beta), Cisco Catalyst 8200 UCPE, UCSC-C220-M5X, and UCS-E-M3.

  Dynamic SR-IOV is not supported on ENCS 5000 series.

- Dynamic SR-IOV is not supported on certain PNICs:
  - PNIC with driver i40e

  **Note**

  PNIC with driver i40e is supported on default SR-IOV.

  - PNIC that does not support SR-IOV

  - NFVIS release 3.12.1 supports Virtual Ethernet Bridge (VEB) in switch mode only.
• Resizing the number of virtual functions is not supported. SR-IOV should be disabled and then enabled with desired number of virtual functions.

**Disable SR-IOV on a PNIC**

To disable SR-IOV on a PNIC, ensure that:

• All SR-IOV networks on a PNIC are deleted.
• The PNIC is not attached to a bridge.

```bash
configure terminal
no pnic eth0-1 sriov
commit
```

**Enable SR-IOV on a PNIC**

To enable SR-IOV on a PNIC, ensure that:

• The PNIC supports SR-IOV.
• The numvfs field is populated with a value that is less than the maximum number of virtual functions (maxvfs) supported on that PNIC.
• The PNIC is not attached to a bridge.

```bash
configure terminal
pnic eth0-1 sriov numvfs 20
commit
```

To display the SR-IOV status of all PNICS, use the `show pnic sriov` command. To display the SR-IOV state of an individual PNIC use the `show pnic eth0-1 sriov` command.

**Create SR-IOV Networks**

To create SR-IOV networks, the PNIC must have SR-IOV enabled and configured with numvfs. The SR-IOV network name must have the following format: `<pnic_name>-SRIOV-<num>` where `<pnic_name>` is a valid PNIC name and `<num>` is a value that is greater than 0 and less than the number of VFs (numvfs).

To create an SR-IOV network in trunk mode:

```bash
configure terminal
networks network eth0-1-SRIOV-1 sriov true
commit
```

To create an SR-IOV network in access mode:

```bash
configure terminal
networks network eth0-1-SRIOV-1 sriov true trunk false vlan 30
commit
```

**Delete SR-IOV Networks**

To delete an SR-IOV network, ensure that no VMs are attached to the network.

```bash
configure terminal
```
no networks network eth0-1-SRIOV-1
commit

To verify the system networks, use the **show system networks** command.

### System Routes

You can configure static system routes along with the default routes in the system. Static routes are used for traffic that should not go through the default gateway. When certain destinations are not reachable through the default routes, this configuration is effective. Also the configured static routes updates the system routing table.

You can create a route by providing the destination and prefix length, but a valid route requires a specified device, a gateway or both. The gateway input represents the address of the next-hop router in the address family. The dev input is the name of the outbound interface for the static route.

### Configuring System Routes

The following example shows how to configure additional static routes:

```bash
configure terminal
system routes route 172.25.222.0/24 gateway 172.25.221.1
system routes route 172.25.223.0/24 dev wan-br
commit
```

To verify the system routes configuration, use the **show system routes** command.

```
nfvos# show system routes
```

<table>
<thead>
<tr>
<th>DESTINATION</th>
<th>PREFIXLEN</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.25.222.0</td>
<td>24</td>
<td>Success</td>
</tr>
<tr>
<td>172.25.223.0</td>
<td>24</td>
<td>Success</td>
</tr>
</tbody>
</table>

### 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>

### Troubleshooting

To troubleshoot errors in configured routes, use the **show system routes** command to identify the failed route. The following example shows common failures with system routes:

```
nfvos# show system routes
```

<table>
<thead>
<tr>
<th>DESTINATION</th>
<th>PREFIXLEN</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.25.222.0</td>
<td>24</td>
<td>Failure(1)</td>
</tr>
<tr>
<td>172.25.223.1</td>
<td>24</td>
<td>Failure(2)</td>
</tr>
</tbody>
</table>

You can find the cause for each error in the nfvos-confd log.
Failure 1) result=RTNETLINK answers: Network is unreachable

The example above indicates that the failure is caused because the network is unreachable. To resolve this issue you can either reconfigure the route with a reachable gateway or identify network connectivity issue.

Failure 2) result=RTNETLINK answers: Invalid argument

This failure is caused due to a mismatch between the subnet address and the prefix length. To resolve this issue you can reconfigure the route with the correct subnet address (in this case 172.25.223.0 for prefix length of 24).

Cisco Network Plug-n-Play Support

Note
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

Note
NFVIS SD-Branch Zero Touch Provisioning (ZTP) PnP redirect fails if the WAN DHCP IP address is in the same subnet as management IP address 192.168.1.0/24. For a successful PnP redirect, you can change the DHCP pool to a different subnet or manually change the management IP address to a different subnet.

Bulk Provisioning of User Credentials

You can change the default user name 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.

Note
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>
          <role>administrators</role>
        </user>
        <user>
          <name>test1</name>
          <password>Test1239#</password>
          <role>administrators</role>
        </user>
        <user>
          <name>test2</name>
          <password>Test2985#</password>
          <role>operators</role>
        </user>
      </users>
    </authentication>
  </rbac>
</config>
```

Sample Format 2

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

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

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, starts 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 PnP server 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 PnP server, for example, because option 43 is not configured, the Cisco Plug and Play Agent falls back on a DNS lookup.
lookup method. Based on the network domain name returned by the DHCP server, it constructs a fully qualified domain name (FQDN) for the PnP server, using the preset hostname "pnpserver". For more details on how to configure DNS for PnP server auto-discovery, see the Solution Guide for Cisco Network Plug and Play.

**Note**

DNS FQDN Only lookup method is supported since 3.10.1 release.

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

### Configuring PnP Discovery Methods

To enable static mode for PnP discovery using IPv4:

```
configure terminal
pnp automatic dhcp disable dhcp-ipv6 disable dns disable dns-ipv6 disable cco disable cco-ipv6 disable
pnp static ip-address 192.0.2.8 port 80 transport http
commit
pnp action command restart
```

To enable static mode for PnP discovery using IPv6:

```
configure terminal
pnp automatic dhcp disable dhcp-ipv6 disable dns disable dns-ipv6 disable cco disable cco-ipv6 disable
pnp static ipv6-address 0:0:0:0:ffff:c000:208 port 80 transport http
commit
pnp action command restart
```

**Note**

Either IPv4 or IPv6 can be enabled at a time.

To enable static mode for PnP discovery using FQDN:

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

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.

Note: 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 ipv6-address ""
  pnp status port 80
  pnp status transport http
  pnp status cafile ""
  pnp status created_by user
  pnp status dhcp_opt43 0
  pnp status dns_discovery 0
  pnp status cco_discovery 0
  pnp status dhcp-ipv6 0
  pnp status dns-ipv6 0
  pnp status cco-ipv6 0
  pnp status timeout 100
nfvis#

FQDN
nfvis# show pnp
pnp status response "PnP Agent is running\nserver-connection\n status: Success\ntime: 06:23:11 Jun 17\ndevice-info\n status: Success\ntime: 06:23:06 Jun 17\nbackoff\n status: Success\ntime: 06:23:11 Jun 17\ncertificate-install\n status: Success\ntime: 06:21:38 Jun 17\ncli-exec\n status: Success\ntime: 06:22:50 Jun 17\ntopology\n status: Success\ntime: 06:23:00 Jun 17"

Host System Management

Configuring PnP Discovery Methods
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\nsuccess\nserver-connection
status: Success\ntime: 05:05:59 Jun 17\ninterface-info
status: Success\ntime: 05:05:56 Jun 17\ndevice-info
status: Success\ntime: 05:05:38 Jun 17\nbackoff
status: Success\ntime: 05:05:59 Jun 17\ncertificate-install
status: Success\ntime: 05:01:19 Jun 17\ncli-exec
status: Success\ntime: 05:05:49 Jun 17\n"
pnp status ip-address 192.0.2.8
pnp status ipv6-address ""
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 1
pnp status dns_discovery 1
pnp status cco_discovery 1
pnp status dhcp-ipv6 1
pnp status dns-ipv6 1
pnp status cco-ipv6 1
pnp status timeout 60
```

**DNS:**
```
nfvis# show pnp
pnp status response "PnP Agent is running\nsuccess\nserver-connection
status: Success\ntime: 05:13:55 Jun 17\ndevice-info
status: Success\ntime: 05:13:49 Jun 17\nbackoff
status: Success\ntime: 05:13:55 Jun 17\ncertificate-install
status: Success\ntime: 05:12:26 Jun 17\ncli-exec
status: Success\ntime: 05:13:34 Jun 17\ntopology
status: Success\ntime: 05:13:45 Jun 17\n"
pnp status ip-address pnpserver.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 dns_discovery
pnp status dhcp_opt43 1
pnp status dns_discovery 1
pnp status cco_discovery 1
pnp status dhcp-ipv6 1
pnp status dns-ipv6 1
pnp status cco-ipv6 1
pnp status timeout 60
```

**CCO:**
nfvis# show pnp
pnp status response "PnP Agent is running
server-connection
 status: Success
 time: 05:24:25 Jun 17\ninterface-info
 status: Success
 time: 05:23:13 Jun 17\ndevice-info
 status: Success
 time: 05:24:25 Jun 17\ncapability\ncapability\n status: Success
 time: 05:24:25 Jun 17\nredirection\ncapability\n status: Success
 time: 05:23:06 Jun 17\ntopology\ntopology
 status: Success
 time: 05:23:10 Jun 17\n"
pnp status ip-address 192.0.2.8
pnp status ipv6-address 

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>

Certificate for Static PnP discovery

Table 13: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command for PnP certificates</td>
<td>NFVIS 4.2.1</td>
<td>A certificate can be used as a PnP root certificate through Command Line Interface (CLI).</td>
</tr>
</tbody>
</table>

Starting from NFVIS 4.2.1 release, a certificate can be used as a PnP root certificate through Command Line Interface (CLI). The following command is used to upload a certificate:

```
system certificate input filepath <filepath> pem-data <certificate contents>
```

- The file containing the certificate information is created inside /data/intdatastore/uploads directory.
- The certificate should be in PEM encoding. Any invalid content or format is rejected with an error message.
- Multiple certificates can also be added to form a certificate chain and they should be separated by a new line.
- The certificate content must have:
Maximum size of base64 content in each certificate limited to 6144 bytes,
Maximum number of certificates allowed in a certificate chain input as 10.
For both single certificate and a chain of certificates, the input should end with a new line.
If a file with the same name as the certificate file name already exists inside /data/intdatastore/uploads directory, the user gets an appropriate error message.

The following are examples to show how to upload certificates:

nfvis# system certificate input filepath intdatastore:uploads/apic_em_oneline_02.pem pem-data
"-----BEGIN CERTIFICATE-----
MIIDvzCCAqegAwIBAgIUKBZWJ4U2c1gBYJFBFWNlNSVd2MwDQQYJKoZIhvcNAQEL
BQAwgYgxLTArBgNVBAMTJDEyYjQ0ZDhmLWRhZWItNGIxNi1iYWZhLWRiYTQxZGRk
MWY3ZmElMAkGA1UEBhMCVVMxEDAOBgNVBAsTCkFQSUNFTS1TRE4wMBM4
DTE4M0cyMT1wNDMyNiODMDt1MDMyNlooXjEVMHMGALUEAxMMMTcyJjI1
B1NhkpCmC2vUEA1e1dCMBkGhAcZvLx58H+b3Mh3025zfLXxLW+0i2Q6m4w
-----END CERTIFICATE-----
" nfvis#

nfvis# system certificate input filepath
Value for 'filepath' (<string>): intdatastore:uploads/test_cert.pem
Value for 'pem-data' (<string>):
> -----BEGIN CERTIFICATE-----
> MIIDvzCCAqegAwIBAgIUKBZWJ4U2c1gBYJFBFWNlNSVd2MwDQQYJKoZIhvcNAQEL
> BQAwgYgxLTArBgNVBAMTJDEyYjQ0ZDhmLWRhZWItNGIxNi1iYWZhLWRiYTQxZGRk
> MWY3ZmElMAkGA1UEBhMCVVMxEDAOBgNVBAsTCkFQSUNFTS1TRE4wMBM4
> DTE4M0cyMT1wNDMyNiODMDt1MDMyNlooXjEVMHMGALUEAxMMMTcyJjI1
> B1NhkpCmC2vUEA1e1dCMBkGhAcZvLx58H+b3Mh3025zfLXxLW+0i2Q6m4w
> -----END CERTIFICATE-----
>
> -----BEGIN CERTIFICATE-----
> MIIDrzCCApegAwIBAgIGCMgojJNOMA0GCSqGSIb3DQEBCwUAMIGIMS0wKwYDVQQD
> EyQxMmI0NGQ4Zi1kYWViLTRiMTYtYmFmYS1kYmE0MWRkZDFmNzMxCzAJBgNVBAYT
> AlVTMRMwEQYDVQQKEwpBUElDRU0tU0ROMQ4wDAYDVR0TBAUwAwEB/wQFMBQF
> GoDIDxNh10Z0e5wT1Jcd1532t+IY4Fj1dC8Y7h98In+eW+w2WwIMOM084G6s6w
> gOv2v8S7y+bB6vYxtxvBwEwvTz8J1b0kVp7JZ9Fw9mA599Yx0Qq17H91FW
> dOTzB7yCQ4KoL3M7wOz5La+fQgQJ3S53XAk7OFlCP3q+5ePa9dOC6UbJH
> -----END CERTIFICATE-----
>
nfvis# system certificate input filepath

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
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>
DPDK Support on NFVIS

Data Plane Development Kit (DPDK) support on NFVIS increases network throughput. DPDK allows applications to pull data directly from the Network Interface Card (NIC) without involving the kernel, therefore delivering high-performance user-space network I/O. DPDK support on NFVIS allows network traffic to bypass NFVIS kernel and directly reach deployed VNFs and service chains. For DPDK adoption NFVIS reserves additional cores and memory to enhance system performance.

DPDK support on NFVIS was first introduced in NFVIS 3.10.1 release and enhancements were added in subsequent releases:

- NFVIS 3.10.1 – DPDK support only for service bridges. DPDK support can be enabled only when the device is in the factory default state. DPDK support is supported only on ENCS 5400 series devices.
- NFVIS 3.11.1 – DPDK support can be enabled at any time. All Virtual NICs connected to service bridges for all VNFs are upgraded to DPDK support. DPDK support is supported only on ENCS 5400 series devices.
- NFVIS 3.12.1 – DPDK support is extended to all supported platforms. Physical NICs can also use DPDK.

DPDK support on NFVIS includes:

- Upgrading existing bridges to enable DPDK
- Upgrading virtual NICs attached to VNFs to enable DPDK
- Upgrading physical NICs to enable DPDK

Note: NICs and WAN side are not upgraded as they are configured with SR-IOV.

Once DPDK support is successfully enabled, you can disable DPDK only by resetting NFVIS to factory settings.

Restrictions

- You must enable DPDK using the system settings dpdk enable command before you commit any other configurations.
- DPDK does not support wan-br and wan2-br on ENCS 5400 devices.
- SR-IOV interfaces and DPDK support:

  To enable DPDK, every device driver must be supported by DPDK. NFVIS does not support SR-IOV interface upgrade to enable DPDK because SR-IOV device drivers are not supported by DPDK. If any SR-IOV network has been configured on an interface, that interface will not support DPDK. Also if an SR-IOV interface is attached to a bridge, the bridge does not support DPDK and if a bridge supports DPDK, no SR-IOV interface can be attached to it.

Note: This restriction does not apply to ENCS 5000 series devices.
DPDK Support on NFVIS

- VNF downtime:

  When DPDK support is enabled on a system, NFVIS upgrades virtual NICs attached to the VNFs. The VNFs are powered down causing a downtime for the VNF service for a short duration of time. After the upgrade is complete, all VNFs are powered up again.

System Requirements

DPDK support optimizes the performance by utilizing additional resources such as CPU and memory. If NFVIS is not able to acquire additional processing or memory, DPDK support can not be enabled.

Enabling DPDK support requires an additional core from each socket available in the system. Depending upon the number of sockets present in the system, NFVIS acquires an additional core for DPDK support.

Configuring DPDK Support on NFVIS

Configuring DPDK support takes up to a minute and network changes can be observed during the process. NFVIS provides an operational status for DPDK support which indicates if DPDK support is enabled or not. The different values for operational status are listed in the table below.

<table>
<thead>
<tr>
<th>DPDK Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>The system is not using DPDK.</td>
</tr>
<tr>
<td>enabled</td>
<td>DPDK support is successfully enabled on the system. Additional CPU and memory resources are reserved for DPDK.</td>
</tr>
<tr>
<td>enabling</td>
<td>The system is in the process of enabling DPDK.</td>
</tr>
<tr>
<td>error</td>
<td>The system is unable to acquire the required resources to support DPDK. All of the resources that were acquired by DPDK are released again.</td>
</tr>
</tbody>
</table>

If DPDK status is in error state, DPDK support can be manually disabled. Before enabling DPDK again, reboot the system to defragment the system memory and increase the chance of resource allocation for a successful configuration.

After enabling DPDK, physical NICs configured with SR-IOV will not be able to interact with DPDK bridges. To add a physical NIC to a DPDK bridge, all SR-IOV networks created on the interface should be removed first. NFVIS will not allow adding an SR-IOV configured interface to a DPDK bridge. For more information, see Dynamic SR-IOV, on page 45.

To enable DPDK support:

```
config terminal
system setting dpdk enable
commit
```

To display the operational status that indicates DPDK support, use `show system native settings` command.

```
nfvis# show system settings-native dpdk-status
system settings-native dpdk-status enabled
```

If NFVIS is unable to acquire sufficient resources, it shows an error state, and DPDK configuration can be removed. After removing the configuration, DPDK can be enabled again.
Storage Access

Network File System Support

Network File System (NFS) is an application where you can view, store, and update the files on a remote device. NFS allows you 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.

Mount and Unmount NFS

The following example shows how to mount NFS:

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

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

Image Registration on NFS

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

To register tar.gz images on NFS:

```bash
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 the `placement type zone_host host nfs_storage` command.
Host System Operations

This section describes operations that can be performed on the NFVIS host.

**Power Cycle System**
To power cycle NFVIS, use the following command:

```
nfvis# hostaction powercycle
```
A notification and syslog are sent to indicate that a power cycle was performed.

**Reboot System**
To reboot NFVIS, use the following command:

```
nfvis# hostaction reboot
```
A notification and syslog are sent to indicate the system reboot.

**Shut down System**
To shut down NFVIS, use the following command:

```
nfvis# hostaction shutdown
```
A notification and syslog will be sent to indicate that the system was shutdown.

**System file-list**
To view a list of files on the system, use the `show system file-list` command.

```
nfvis# show system file-list [disk [local | nfs | usb] ]
```

<table>
<thead>
<tr>
<th>Disk Type</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>Files present in the internal datastore and external datastores</td>
</tr>
<tr>
<td>nfs</td>
<td>Files on NFS</td>
</tr>
<tr>
<td>usb</td>
<td>Files on the mounted USB drive</td>
</tr>
</tbody>
</table>

**System file-copy**
To copy a file from the USB drive to the `/data/intdatastore/uploads` directory, use the `system file-copy` command. To copy a VM image from the USB drive:

```
configure terminal
system usb-mount mount active
system file-copy usb file name usb1/package/isrv-universalk9.16.03.01.tar.gz
commit
```
The **system file-copy** command can also be used to copy a file from the given source path to the given destination path. The allowed directories for source path and destination path are:

- /data/intdatastore
- /mnt/extdatastore1
- /mnt/extdatastore2
- /data/mount

```
nfvis# system file-copy source <path-to-source-file> destination <path-to-destination-file>
```

**System file-delete**

The **system file-delete** command is used to delete a file from one of these directories: /data/intdatastore, /mnt/extdatastore1, /mnt/extdatastore2, /mnt-usb or /data/mount

```
nfvis# system file-delete file name
/data/intdatastore/uploads/isrv-universalk9.16.03.01.tar.gz
```

**Secure Copy**

The secure copy (**scp**) command allows only the admin user to securely copy files from NFVIS to an external system, or from an external system to NFVIS. For example, this command can be used to copy an upgrade package to NFVIS.

The syntax for this command is:

```
scp <source> <destination>
```

**Note**

For detailed information about how to use the **scp** command to copy to or from supported locations, see the **scp** section in **Cisco Enterprise Network Function Virtualization Infrastructure Software Command Reference**. SCP between two NFVIS devices is not supported.

**Examples**

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

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

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

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

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

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

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

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

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

```
nfvis# scp user@203.0.113.2:/user/Desktop/sample.txt user@203.0.113.3:usb:/test.txt
```
The following example copies the nfvis_scp.log file to an external system.

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

The following example shows how to secure copy from techsupport as source:

```
nfvis# scp logs:nfvis_techsupport.tar.gz user@203.0.113.2:/Users/user/Desktop/copied_techsupport.tar.gz
```

**Change BIOS Password**

This command is applicable only to the ENCS platform. It allows the user to change the BIOS password. A notification and syslog are sent regarding the password change.

To change the BIOS password:

```
nfvis# hostaction change-bios-password <new-password>
```

There is a strong password check enforced for the new BIOS password. The new password should contain:

- At least one lowercase character
- At least one uppercase character
- At least one number
- At least one special character from #, @ or _
- Password length should be between 7 and 20 characters
- The first character cannot be a #

**Change CIMC Password**

This command is applicable only to the ENCS platform. It allows the user to change the CIMC password. A notification and syslog are sent regarding the password change.

To change CIMC password:

```
nfvis# hostaction change-cimc-password <new-password>
```

There is a strong password check enforced for the new CIMC password. The new password should contain:

- At least one lowercase character
- At least one uppercase character
- At least one number
- At least one special character from #, @ or _
- Password length should be between 8 and 20 characters
Backup and Restore NFVIS and VM Configurations

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhancements to backup and restore of configurations</td>
<td>NFVIS 4.2.1</td>
<td>New commands are introduced to view the overall status of backup and restore process. Enhancements to backup file location and factory default options are introduced. Information on how to troubleshoot failure to restore NFVIS configurations is added.</td>
</tr>
</tbody>
</table>

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 and restore a single VM, use vmExportAction (for VM backup) and vmImportAction (for VM restore) APIs.

Restrictions for Backup and Restore on NFVIS

- The backup includes all deployed VMs and the registered images except uploaded files.
- VM restore using hostaction restore and vmImportAction requires original registered image to be on the system, on the same datastore. Missing registered image or image registered in a different datastore results in VM restore failure.
  
  For NFVIS 4.2 release, only VM restore using hostaction restore does not require original registered images on the system.

- For NFVIS 4.1 and earlier releases, NFVIS VM backup does not support differential disk backup and every backup is a full VM backup.

- For NFVIS 4.1 and earlier releases, in case of multiple deployments based on a single registered image, every VM backup includes the registered image disk.

- The time taken to backup a VM depends on the option you choose:
  
  - configuration-only - within 1 min.
  
  - configuration-and-vms - depends on the number of VM deployments on your system, system disk write speed, and compress the VM disks into one bundle.

- You can either backup all the VMs or none.
• The final backup is a compressed file which requires temporary disk space to create the VM backup file. If the system has only one datastore, the maximum deployment backups in a single file is around one-third to half of the datastore disk space. If the deployments occupies more disk space, use \texttt{vmExportAction} to backup an individual VM instead of relying on host backup for all VM deployments.

• NFVIS only supports backup or restore on the same release. For example, backup created in NFVIS 4.1.1 cannot be used to restore on NFVIS 4.2.1.

• Starting from NFVIS 4.5 release, secure-overlay configuration with EAP authentication is supported. However, it will be discarded if restored on a different box or on the same box after fresh-install because of encrypted password.

• Starting from NFVIS 4.5 release, single ip configuration is supported. However, it will be discarded if restored on a different box because the single ip bootstrap is tailored towards a particular box.

\textbf{Feature Comparison Table for Backup and Restore}

Backup using hostaction backup:

<table>
<thead>
<tr>
<th>Feature</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default file location for backup</td>
<td>/data/intdatastore/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore1/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore2/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/data/intdatastore/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore1/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore2/backup.bkup</td>
<td></td>
</tr>
<tr>
<td>VM backup format</td>
<td>Full backup</td>
<td>Diff disk backup</td>
</tr>
<tr>
<td>Registered Image and Flavors</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Status monitoring</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Check disk space before backup</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Restore using hostaction restore:

<table>
<thead>
<tr>
<th>Feature</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default file location for backup</td>
<td>/data/intdatastore/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore1/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore2/uploads/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/data/intdatastore/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore1/backup.bkup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/mnt/extdatastore2/backup.bkup</td>
<td></td>
</tr>
<tr>
<td>Restore images and flavors</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Unique Mac Uid for VM</td>
<td>No for NFVIS 3.12.3 and earlier release</td>
<td>Yes</td>
</tr>
<tr>
<td>Status monitoring</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Backup and Restore NFVIS and VM Configurations

<table>
<thead>
<tr>
<th>Feature</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP v3 user/passphrase restore (with uniqMacUid)</td>
<td>v3 user/passphrase restore</td>
<td>If system engine ID is the same as backup, restore all v3 users. If system engine ID is different from backup, ignore v3 users restoration.</td>
</tr>
<tr>
<td>SNMP engine ID restore on different system</td>
<td>No</td>
<td>Engine ID changed to same as backup bundle</td>
</tr>
</tbody>
</table>

VM backup using vmExportAction:

<table>
<thead>
<tr>
<th>Feature</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM backup format</td>
<td>Full backup</td>
<td>Diff disk backup</td>
</tr>
</tbody>
</table>

Backup and Restore

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

You can only create a backup and save into datastore, or mounted USB storage device. Without specifying, the backup file will have `.bkup` extension.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Backup configuration-only</th>
<th>Backup configuration-and-vms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save system configurations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Save system upgrade configurations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Save system upgrade file</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Save images and flavors configurations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Save image disks</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Save deployments configurations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Save deployments disks</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The following examples shows the backup options:

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

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

The following example shows the backup stored on a USB:
Use the `hostaction backup force-stop` command to stop the running backup.

Starting from NFVIS 4.2 release, use the `show hostaction backup status` command to view the status of the overall backup process and each component like system, image and flavors, vm and so on. The following is an example of the show command output after the backup process is complete:

```
nfvis# show hostaction backup status
hostaction backup status 2020-07-16T02:45-00:00
destination intdatastore:backup_20200704.bkup
status BACKUP-SUCCESS
size "2798.0 MB"
components FIREWALL
  status BACKUP-SUCCESS
  last update 2020-07-16T07:38-00:00
  size "20.49 MB"
  details ""
components Linux
  status BACKUP-SUCCESS
  last update 2020-07-16T07:36-00:00
  size "0.01 MB"
  details ""
components NFS
  status BACKUP-SUCCESS
  last update 2020-07-16T07:44-00:00
  size "0.01 MB"
  details ""
components NFVIS
  status BACKUP-SUCCESS
  last update 2020-07-16T07:02:48-00:00
  size "0.72 MB"
  details ""
components ROUTER
  status BACKUP-SUCCESS
  last update 2020-07-16T07:35-00:00
  size "579.89 MB"
  details ""
components VM_Images_Flavors
  status BACKUP-SUCCESS
  last update 2020-07-16T07:45-00:00
  size "2197.73 MB"
  details ""
nfvis#
```

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

The restore is based on the system condition created during backup.

<table>
<thead>
<tr>
<th></th>
<th>Restore configuration-only</th>
<th>Restore configuration-and-vms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore system configurations</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Restore upgrade configurations</td>
<td>Restore configuration-only</td>
<td>Restore configuration-and-vms</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td>Yes, requires same upgrade files in system if the host backup was taken has such upgrade files.</td>
<td>Yes, requires same upgrade files in system if the host backup was taken has such upgrade files.</td>
</tr>
<tr>
<td></td>
<td>No, if host where backup was taken did not have any upgrade files registered. Restore will fail.</td>
<td>No, if host where backup was taken did not have any upgrade files registered. Restore will fail.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Restore registered images and flavors</th>
<th>Restore configuration-only</th>
<th>Restore configuration-and-vms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes, if images sources are still available (URL link is still valid, or uploaded files are still in the same locations).</td>
<td>Yes, restore from backup file.</td>
<td>Yes, restore from backup file.</td>
</tr>
<tr>
<td>No, if images sources are not available (URL link is invalid, upload files are deleted or moved to new location). The restore process will fail.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Restore deployments</th>
<th>Restore configuration-only</th>
<th>Restore configuration-and-vms</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td></td>
<td>Yes, restore from backup file.</td>
</tr>
</tbody>
</table>

**Note**

This means if there are upgrade files registered in the NFVIS. The backup create on this host will contain those information. If using this backup on new host or same host after factory-default-reset, the restore will fail.

<table>
<thead>
<tr>
<th>dpdk-disabled while backup</th>
<th>dpdk-enable while backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>dpdk-disabled while restore</td>
<td>Yes (system is dpdk-disabled)</td>
</tr>
<tr>
<td>dpdk-enabled while restore</td>
<td>No support</td>
</tr>
</tbody>
</table>

**Note**

In hostaction restore process, the full file name (with .bkup extension) is required in the CLI.

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

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

```bash	nfvis# hostaction restore except-connectivity file-path extdatastore2:sample-dir/sample.bkup
```
Starting from NFVIS 4.2 release, use the `show hostaction restore-status` command to view the status of the overall restore process and each components like system, image and flavors, vm and so on. The following is an example of the show command output after the restore process is complete:

```
nfvis# show hostaction restore-status
hostaction restore-status 2020-07-16T07:18:54-00:00
source intdatastore:backup_20200704.bkup
status RESTORE-SUCCESS
components FIREWALL.vmbkp
  status RESTORE-SUCCESS
  last update 2020-07-16T07:26:34-00:00
  details ""
components Linux.vmbkp
  status RESTORE-SUCCESS
  last update 2020-07-16T07:26:03-00:00
  details ""
components NFS.vmbkp
  status RESTORE-SUCCESS
  last update 2020-07-16T07:25:36-00:00
  details ""
components NFVIS
  status RESTORE-SUCCESS
  last update 2020-07-16T07:22:03-00:00
  details ""
components ROUTER.vmbkp
  status RESTORE-SUCCESS
  last update 2020-07-16T07:26:55-00:00
  details ""
components VM_Images_Flavors
  status RESTORE-SUCCESS
  last update 2020-07-16T07:26:01-00:00
  details ""
components intdatastore:backup_20200704.bkup
  status VERIFICATION-SUCCESS
  last update 2020-07-16T07:18:54-00:00
  details ""
nfvis#
```

Starting from NFVIS 4.2 release, you can backup registered images and flavors into backup package and restore these images and flavors into the system. The new system does not require a pre-registered image before system restore. If the system has existing images, flavors or deployments, the system restore erases them all and restores from its own backup. VM backup is now faster and uses less disk space compared to NFVIS 4.1 release, but it also takes up additional process, time or disk space to backup registered images and flavors.

**Backup, Restore, and Factory-Default-Reset**

To perform `hostaction backup -> factory-default-reset -> hostaction restore` on the same box without any external storage (like USB or NFS mount), check the following issues:
<table>
<thead>
<tr>
<th>Backup file location</th>
<th>NFVIS 4.1.x and earlier releases</th>
<th>NFVIS 4.2.x and later releases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The system backup bundle is saved under /datastore/uploads/ by default.</td>
<td>• The system backup bundle is saved under /datastore/ by default.</td>
</tr>
<tr>
<td></td>
<td>• Factory-default-reset cleans up all files under /datastore/uploads/, but leave files under /datastore/ intact.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>hostaction restore</strong> requires backup bundle saved under /datastore/uploads/. The restore process will not start if the backup bundle is saved in another location (bundle saved on USB or NFS should be copied to datastore/uploads/ folder).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System requirements if system backup bundle contains VM backups</th>
<th>NFVIS 4.1.x and earlier releases</th>
<th>NFVIS 4.2.x and later releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>• VM restoration requires the original image or template registered in NFVIS.</td>
<td></td>
<td>• The backup package includes original registered images.</td>
</tr>
<tr>
<td>• Factory-default-reset all clean up all registered images and uploaded files. You need to configure minimum setup, like host connection and upload registered images to the same datastore.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Prevent backup bundle from deleting with factory-default-reset

<table>
<thead>
<tr>
<th>NFVIS 4.1.x and earlier releases</th>
<th>NFVIS 4.2.x and later releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Save the backup bundle in remote locations. Then restore the connectivity and upload the backup bundle after reset.</td>
<td>• Save backup bundle in local /intdatastore/ and not in /intdatastore/uploads/ or copy backup bundle from /intdatastore/uploads/to /intdatastore/:</td>
</tr>
<tr>
<td>• Save backup bundle in local /datastore/ and not in /datastore/uploads/ or copy backup bundle from /datastore/uploads/to /datastore/:</td>
<td></td>
</tr>
</tbody>
</table>

```
# Backup & Restore on the same NFVIS box without NFS & USB
# [[ BACKUP ]]
# before executing factory-default-reset

nfvis# nfvis# hostaction backup configuration-only file-path extdatastore1:configBackup-01.bkup
nfvis# system file-copy source /mnt/extdatastore1/uploads/configBackup-01.bkup destination /mnt/extdatastore2/

# after factory-default-reset all-except-images or all-except-images-connnectivity,
# file /mnt/extdatastore1/uploads/configBackup-01.bkup will be deleted
# but /mnt/extdatastore2/configBackup-01.bkup won't.

# [[RESTORE]]
# after NFVIS rebooted and login to console, copy file to uploads/ directory

nfvis# system file-copy source /mnt/extdatastore2/configBackup-01.bkup destination /mnt/extdatastore2/uploads/
nfvis# hostaction restore file-path extdatastore2:configBackup-01.bkup
```

You can copy backup file to intdatastore/ if there is sufficient storage space. If the backup is larger than free disk space in intdatastore/, you can copy to a remote server like scp or NFVIS web portal.

The following table lists the data erased and retained upon using NFVIS factory default reset options:

<table>
<thead>
<tr>
<th>Factory-default-reset all</th>
<th>Factory-default-reset all-except-images</th>
<th>Factory-default-reset all-except-images-connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>files under intdatastore</td>
<td>Retain</td>
<td>Retain</td>
</tr>
<tr>
<td>files under intdatastore/uploads/</td>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can copy backup file to intdatastore/ if there is sufficient storage space. If the backup is larger than free disk space in intdatastore/, you can copy to a remote server like scp or NFVIS web portal.

The following table lists the data erased and retained upon using NFVIS factory default reset options:
### Failure to Restore

NFVIS configurations fail to restore if:

- There is no sufficient disk space. Restore requires temporary disk space to save un-compressed files. You can Move, Copy or Upload the backup file to a larger datastore and run system restore.

nfvis# show hostaction restore-status
hostaction restore-status 2020-07-16T21:29:08-00:00
source intdatastore:encs07-configVms-dpdk-2020-07101600.bkup
status RESTORE-ERROR
components intdatastore:encs07-configVms-dpdk-2020-07101600.bkup
status VERIFICATION-ERROR
last update 2020-07-16T21:49:18-00:00
details “Backup package could not be inflated. No space left on device”
nfvis#

- The application communication fails. You can see this error after the first restore attempt has failed, and when you try to restore for the second time. You can reboot NFVIS before you attempt restore again.

nfvis# hostaction restore file-path extdatastore2:backup_20200704.bkup
Error: application communication failure

### APC UPS Support and Monitoring

This feature is supported only on ENCS 5400.

This feature provides support for monitoring battery status for an APC UPS connected to the ENCS box through a USB cable. NFVIS gracefully shuts down when the UPS battery reaches 5% and boots up again when the battery reaches 15%. This feature is available only through NFVIS CLI and is disabled by default.
In case of a prolonged power outage that drains the UPS battery completely, the box is powered off. When power is restored to the UPS, CIMC boots up which in turn boots up the NFVIS.

To enable APC UPS:

`apcups enable`

To disable APC UPS:

`apcups disable`

To check the battery status of an APC UPS:

`apcups battery-status`

---

**Resetting to Factory Default**

Factory default reset is available on all NFVIS supported hardware platforms.

You can reset the host server to factory default with the following 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**—Delete VMs and volumes, files including logs, user uploaded files and certificates. Erases all configuration except registered images. Connectivity will be lost, and the admin password will be changed to factory default password.

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

---

**Note**

Factory default reset must be used only for troubleshooting purpose. We recommend you contact Cisco Technical Support before performing factory default reset. This feature will reboot the system. Do not perform any operations until the system reboots successfully.

To reset to factory default:

`nfvis#factory-default-resetall|all-except-images|all-except-images-connectivity`

---

**Note**

Enter **Yes** when you are prompted with the factory default warning message or **no** to cancel.
Configure Banner, Message of the day and System Time

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```

Note 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:

```configure terminal```
NTP is automatically disabled when the time clock is set manually.

To set the system time using NTP IPv4:

configure terminal
system time ntp preferred_server 209.165.201.20 backup_server 1.ntp.es1.cisco.com
commit

To set the system time using NTP IPv6:

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:

nfvis# show system time

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

<table>
<thead>
<tr>
<th>REMOTE OFFSET</th>
<th>REFID</th>
<th>ST</th>
<th>T</th>
<th>WHEN</th>
<th>POLL</th>
<th>REACH</th>
<th>DELAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>*calo-timeserver .GPS.</td>
<td>2749736</td>
<td>0.000</td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>69.423</td>
<td></td>
</tr>
</tbody>
</table>

* 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 is not 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>
Configuring DNS Name Servers

Table 15: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS Name Server Configuration</td>
<td>NFVIS 4.4.1</td>
<td>You can now configure up to three nameservers, which the DNS resolvers can use in the order you specify. New command introduced to configure DNS nameservers: <code>system settings name-server</code>.</td>
</tr>
</tbody>
</table>

Restrictions

- The statically configured DNS name servers using the `system settings dns-server` command does not work with DNS name servers provided by a configured DHCP server. The statically configured DNS name servers overwrites the ones configured by DHCP, and vice versa.

- In NFVIS 4.4 release, only `system settings name-server` command or `system settings dns-server` command can be used for the configuration of DNS name servers at a given time. To configure DNS name servers using any one of the commands, the other one has to be unconfigured if it is configured previously.

This example shows how to configure name servers:

```
config terminal
commit
```

This example shows how to unconfigure name servers:

```
config terminal
no system settings name-server
commit
```

This example shows how to update name servers:

```
config terminal
no system settings name-server
system settings name-server 209.165.201.23 2001:420:30d:201:ffff:ffff:fff4:33 209.165.201.27
commit
```

DNS name servers configured using the `system settings name-server` command is prepended to DNS name servers provided by a DHCP server automatically. To view the list of configured name servers, use the `show system settings-native dns` command.

In releases prior to NFVIS 4.4.1, DNS name servers could be configured only as shown in the example below:

```
config terminal
commit
```

To unconfigure name servers:
config terminal
no system settings dns-server
commit

The above configurations are still supported on NFVIS 4.4 release.

---

**Note**

Configurations that use the `system settings dns-server` command don't necessarily guarantee that the order in the DNS resolver uses the name servers is the same as the order you specified.

---

### DNS Server APIs and Commands

<table>
<thead>
<tr>
<th>DNS nameservers APIs</th>
<th>DNS nameserver commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/system/settings/dns-server</td>
<td>system settings dns-server</td>
</tr>
<tr>
<td>/api/config/system/settings/name-server</td>
<td>system settings name-server</td>
</tr>
<tr>
<td>/api/operational/system:system/settings-native/dns</td>
<td>show system settings-native dns</td>
</tr>
</tbody>
</table>

---

### Configuring IP Host

NFVIS IP host feature allows you to specify static mapping of host name and IP addresses.

To configure IP host mapping:

```bash
configure terminal
ip host test2.com 2.2.2.3 2.2.2.1
```

### IP Host APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>CLI Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/ip/host</td>
<td>ip host</td>
</tr>
</tbody>
</table>
Overview of 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.

Workflow of VM Life Cycle Management

The following diagram depicts the basic workflow of the VM life cycle management using REST APIs:
VM Life Cycle Management

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 VM**—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 deployed VM can also be added or updated.
Before performing the VM life cycle management tasks, you will have to upload the VM images to the NFVIS server or http/s server.

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

## Support for NFVIS Container Lifecycle Management

**Table 16: Feature History Table**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for NFVIS Container Lifecycle Management</td>
<td>NFVIS 4.6</td>
<td>This feature provides support for container lifecycle management. Note: Cisco, may at its discretion, provide support (as defined by Cisco) to host container based applications for NFVIS 4.6.1. This capability is a beta feature in this release and should not be used in production because, Cisco may make changes in subsequent releases without providing any backward compatibility. The beta is provided AS-IS and without any warranty of any kind.</td>
</tr>
</tbody>
</table>

The NFVIS container lifecycle management follows the same process and data model as VM lifecycle management, with a few changes in the image registration and deployment process. For information about processes supported on container lifecycle management, refer to:

- NFVIS Container Image Registration
- NFVIS Container Deployment

### 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).

To copy an image using the USB device:

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

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

---

### Verifying the Image Copied from the USB Drive

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

```plaintext
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/2017-03/14/10-00 337</td>
<td>Other 2017-03-14 21:55:42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>escmanager-tagged-log.log-20170314.gz</td>
<td>/data/intdatastore/logs/2017-03/14/00 167K</td>
<td>Other 2017-03-18 05:58:26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>conffd_audit.log-20170317.gz</td>
<td>/data/intdatastore/logs/2017-03/17/09-30 4.6K</td>
<td>Other 2017-03-17 21:29:59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>esc_postinit.log-20170317.gz</td>
<td>/data/intdatastore/logs/2017-03/17/05-00 605K</td>
<td>Other 2017-03-17 16:40:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>error.log-20170317.gz</td>
<td>/data/intdatastore/logs/2017-03/17/05-00 1.3K</td>
<td>Other 2017-03-17 16:40:15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ovs-ctl.log-20170317.gz</td>
<td>/data/intdatastore/logs/2017-03/17/12-00 20</td>
<td>Other 2017-03-16 00:00:01 4:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>ovs-ctl1.log-20170323.gz</td>
<td>/data/intdatastore/logs/2017-03/23/12-00 20</td>
<td>Other 2017-03-22 00:00:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>CentOS-7-x86_64-Everything-1511.ova</td>
<td>/data/intdatastore/uploads 1.1G VM 2017-03-15 19:20:03 Package</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>TinyLinux.tar.gz</td>
<td>/data/intdatastore/uploads 17M VM 2017-03-15 18:25:00 Package</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>Cisco-KVM-vWAAS-1300-6.3.0-b98.tar.gz</td>
<td>/data/intdatastore/uploads 979M VM 2017-03-15 19:19:11 Package</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>isrv-universalk9.16.03.01.tar.gz</td>
<td>/data/intdatastore/uploads 1.3G VM 2017-03-15 19:19:53</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Related APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>CLI Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>• /api/operations/system/file-copy/usb/file</td>
<td>• system file-copy usb file name</td>
</tr>
<tr>
<td>• /api/config/system/usb-mount</td>
<td>• system usb-mount mount ACTIVE</td>
</tr>
<tr>
<td>• /api/operations/system/file-copy/usb/file</td>
<td>• system file-delete</td>
</tr>
<tr>
<td>• /api/config/system/usb-mount</td>
<td>• show system file-list disk usb</td>
</tr>
<tr>
<td>• /api/operations/system/file-copy/usb/file</td>
<td>• show system file-list disk local</td>
</tr>
</tbody>
</table>

Performing Resource Verification

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

<table>
<thead>
<tr>
<th>Task</th>
<th>API</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>To display CPU information for each CPU or the user specified CPU, and the VMs pinned to the CPU</td>
<td>/api/operational/resources/cpu-info/cpus</td>
<td>show resources cpu-info cpus</td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/cpus/cpu</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/cpus/cpu/&lt;cpu-id&gt;</td>
<td></td>
</tr>
<tr>
<td>To display information on the VMs running in all the physical CPUs or a specific physical CPU in the system</td>
<td>/api/operational/resources/cpu-info/vnfs</td>
<td>show resources cpu-info vnfs</td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/vnfs/vnf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/api/operational/resources/cpu-info/vnfs/vnf/&lt;deployment_name&gt;.&lt;vm_group_name&gt;</td>
<td></td>
</tr>
<tr>
<td>To get information on the number of CPUs allocated to VMs and the CPUs that are already used by the VMs</td>
<td>/api/operational/resources/cpu-info/allocation</td>
<td>show resources cpu-info allocation</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 Subnet

By default, all VMs with management interfaces will be provisioned with an IP address in the subnet of 10.20.0.1. To change the default subnet, the following commands need to be executed in a sequence to first delete an exiting subnet and then add a new subnet in the network. For these commands to work, ensure there is no managed VNF’s in the system before you change management network address.

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

To create a new subnet:

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

VM Image Packaging

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

- **VM Image Packaging Utility:** This is an enhanced packaging process that allows the VM owner to run the **nfvpt.py** utility as a command with a combination of parameters to package the VM.
- **Standard Image Packaging:** This is a manual process in which a raw disk image (qcow2, img) is packaged along with the image properties file and bootstrap files (if needed) into a TAR archive file.

VM Image Packaging Utility

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

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

Contents

The VM image packaging utility contains the following:

- **nfvpt.py**—It is a python based packaging tool that bundles the VM raw disk image/s along with VM specific properties.
- **image_properties_template.xml**—This is the template file for the VM image properties file, and has the parameters with default values. If the user provides new values to these parameters while creating the VM package, the default values get replaced with the user-defined values.
- **nfvvis_vm_packaging_utility_examples.txt**—This file contains examples on how to use the image packaging utility to package a VM image.
Usage

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

`nfvpt.py --help`

Optional arguments:

- `-h, --help`
  show this help message and exit

- `--json JSON`
  Provide JSON input for bootstrap variables; mutually exclusive with custom and bootstrap configs

- `--newjson NEWJSON`
  Provide JSON input for bootstrap variables; mutually exclusive with custom and bootstrap configs

- `--log_dir LOG_DIR`
  Log Directory to for logfiles

- `--multi_use`
  Add options for use in multiple use-cases

- `--console_type_serial {true,false}`
  Attach the console serial to the VM; default is false;
  ```
  --console_type_serial=true/false;
  ```

- `--root_file_disk_bus {virtio,ide}`
  root disk file type: --root_file_disk_bus=virtio/ide;
  default is virtio

- `--virtual_interface_model {rt18139}`
  default is none

- `--thick_disk_provisioning {true,false}`
  default is false

- `--eager_zero {true,false}`
  ```
  --eager_zero=true; default is 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 cdrom
  ```

- `--bootstrap BOOTSTRAP`
  Every bootstrap file should be a different option Non HA format: --bootstrap
  ```
  <mountpoint>:<file1>,<mountpoint>:<file2>... See usage.txt for more details HA format for SDWAN
  NetworkHub: --bootstrap mount_point:<value>,file:<file2mount>[,<attrib>:<value>] mount_point:<value> and
  file:<file2mount> are mandatory followed by one or more attributes in the format <attrib>:<value>
  ```

- `--interface_hot_add {true,false}`
  VM supports interface add without power off. Default is set to true;
  ```
  --interface_hot_add=true/false
  ```

- `--interface_hot_delete {true,false}`
  VM supports interface delete without power off. Default is set to false;
  ```
  --interface_hot_delete=true/false
  ```

- `-v, --verbose`
  verbose

- `-q, --quiet`
  quiet

- `--no_compress`
  creates tar file without compressing the input files

- `--cleanup`
  deletes all the input and configuration files upon tar file created

- `--tablet {true,false}`
  Add input device of type tablet --tablet=true/false;

- `--ha_package`
  enable HA packaging

- `--mgmt_vnic MGMT_VNIC`
  VM management interface identifier

- `--pack_dir <DIR> PACK`
  package all files in directory

Required:
Usage

-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

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

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

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

The table lists the parameters that can be passed to the nfyp.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. Options are: true/false. Monitoring timeout period for a monitored VM is 600 seconds by default</td>
</tr>
<tr>
<td>optimize</td>
<td>Mandatory</td>
<td>Optimized VM</td>
</tr>
<tr>
<td>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. Default is CD-ROM.</td>
</tr>
<tr>
<td>bootstrap</td>
<td>Optional</td>
<td>Bootstrap files for VNF. Two parameters are required in the format of dst:src;dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for 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. The default is 1.</td>
</tr>
<tr>
<td>max_vcpu</td>
<td>Optional</td>
<td>Maximum number of vCPUs required for the VM. The default is 8.</td>
</tr>
<tr>
<td>min_mem</td>
<td>Optional</td>
<td>Minimum memory in MB required for the VM. The default is 4 GB.</td>
</tr>
<tr>
<td>max_mem</td>
<td>Optional</td>
<td>Maximum memory in MB required for the VM. Physical memory: 2 GB. The default is 8 GB.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>min_disk</td>
<td>Optional</td>
<td>Minimum disk in GB required for the VM. The default is 8 GB.</td>
</tr>
<tr>
<td>max_disk</td>
<td>Optional</td>
<td>Maximum disk in GB required for the VM. Available disks are SSD and HDD: 15 GB. The default is 16 GB</td>
</tr>
<tr>
<td>vnic_max</td>
<td>Optional</td>
<td>Maximum number of VNICs allowed for the VM. The default is 8.</td>
</tr>
<tr>
<td>profile</td>
<td>Optional</td>
<td>The profile name, profile description, number of vCPUs required, minimum memory required in MB and minimum disk space required in MB.</td>
</tr>
<tr>
<td>default_profile</td>
<td>Optional</td>
<td>The default profile.</td>
</tr>
<tr>
<td>sriov</td>
<td>Optional</td>
<td>Enable or disable SRIOV support. The default is false.</td>
</tr>
<tr>
<td>sriov_list</td>
<td>Optional</td>
<td>List of SRIOV drivers.</td>
</tr>
<tr>
<td>pcie</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>pcie_list</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>privileged</td>
<td>Optional</td>
<td>Not supported.</td>
</tr>
<tr>
<td>custom</td>
<td>Optional</td>
<td>Custom properties to be supported and/or passed to the bootstrap configuration with tokenized variables. This is only used for the local portal to display options for the user to choose while deploying.</td>
</tr>
<tr>
<td>pack_dir</td>
<td>Optional</td>
<td>package all files in directory</td>
</tr>
</tbody>
</table>

**NFVIS Specific Enhancements**

*Note*

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

The following parameters are added as part of the NFVIS specific enhancements:

```
--pack_dir <DIR> PACK
```
package all files in directory

Resources:

--vnic_names VNIC_NAMES

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

Usage

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

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

VM Packaging Utility Usage Examples

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

Example 1: Usage for TinyLinux

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

Example 2: Usage for ASAv

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 isr961-201 -i isr961-201.qcow2 -n ISRv -t isr -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 ISRv,"ISRv profile",1,1024,8192 --profile ISRv10,"ISRv10 profile",1,4096,8192
--profile ISRv30,"ISRv30 profile",4,8192,16384
--default_profile ISRv5
Example 4: Usage for a third party VM with config drive (ISO) mounted at specific path on the VM:

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

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

Example 5: Usage for Palo Alto Firewall

```
nfvpt.py -o PA_L3_HA -i PA-VM-KVM-8.0.5.qcow2 --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 pa1.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.

---

**Note**

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

---

### Generating a VM Package

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

1. Create a VM qcow2 image.
2. Create an *image_properties.xml* file with the VM properties. Ensure that you add all mandatory fields. Include the profiles supported for the VM in this file, and select one default profile. If you do not want to monitor the VM bootup, make the bootup time as -1.

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

**Note**

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

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

5. Generate the packaging file using "tar -cvzf ova_file_name list_of_files_to_be_bundled".

For example, `tar -cvzf isrv.tar.gz isrv-universalk9.03.16.02.S.155-3.1a-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 17: VM Image Package Files*

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Manifest (package.mf)</td>
<td>Lists the files in the package and the expected checksum for the files.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>VM image properties (vmname_properties.xml)</td>
<td>XML file with resources and features supported by the VM</td>
<td>Mandatory</td>
</tr>
<tr>
<td>VM image (vmname.qcow2 )</td>
<td>Image file of the VM. Multiple images are supported. One root_disk image file is mandatory.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Bootstrap (bootstrap_file)</td>
<td>Optional</td>
<td>Bootstrap files for VNF. Two parameters are required in the format of dst:src; dst file name 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>
</tr>
</tbody>
</table>
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 18: 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 19: 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>Description</td>
<td>Value Field</td>
<td>Type</td>
<td>Requirement</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>-------------</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 disk 1 image format</td>
<td>&lt;disk_1_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-2 Image format</td>
<td>Disk 2 image format</td>
<td>&lt;disk_2_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-10 Image format</td>
<td>Disk 10 image format</td>
<td>&lt;disk_10_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Serial Console</td>
<td>Serial console supported</td>
<td>&lt;console_type_serial&gt;</td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>Minimum vCPU</td>
<td>Minimum vCPUs required for a VM operation</td>
<td>&lt;vcpu_min&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Maximum vCPU</td>
<td>Maximum vCPUs supported by a VM</td>
<td>&lt;vcpu_max&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Minimum memory</td>
<td>Minimum memory in MB required for VM operation</td>
<td>&lt;memory_mb_min&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Type</td>
<td>Requirement</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Maximum memory</td>
<td>Maximum memory in MB supported by a VM</td>
<td>&lt;memory_mb_max&gt;</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>Minimum root disk size</td>
<td>Minimum disk size in GB required for VM operation</td>
<td>&lt;root_disk_gb_min&gt;</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Maximum root disk size</td>
<td>Maximum disk size in GB supported by a VM</td>
<td>&lt;root_disk_gb_max&gt;</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Maximum vNICs</td>
<td>Maximum number of vNICs supported by a VM</td>
<td>&lt;vnic_max&gt;</td>
<td>Mandatory</td>
<td></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>&lt;sriov_supported&gt;</td>
<td>true, false</td>
<td></td>
</tr>
<tr>
<td>SRIOV driver list</td>
<td>List of drivers to enable SRIOV support</td>
<td>&lt;sriov_driver_list&gt;</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>PCI passthru support</td>
<td>PCI passthru support by VM interfaces</td>
<td>&lt;pcie_supported&gt;</td>
<td>true, false</td>
<td></td>
</tr>
<tr>
<td>PCIE driver list</td>
<td>List of VNICS to enable PCI passthru support</td>
<td>&lt;pcie_driver_list&gt;</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>bootstrap_cloud_init_drive_type</td>
<td>Mounts day0 config file as disk (default is CD-ROM)</td>
<td>&lt;bootstrap_cloud_init_drive_type&gt;</td>
<td>disk, cdrom</td>
<td></td>
</tr>
<tr>
<td>bootstrap_cloud_init_bus_type</td>
<td>Default is IDE</td>
<td>&lt;bootstrap_cloud_init_bus_type&gt;</td>
<td>virtio, ide</td>
<td></td>
</tr>
</tbody>
</table>
### VM Image Properties File

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

**Note**
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**

```
** sha1sum - for calculating checksum
<PackageContents>
  <File_Info>
    <name>ISRv_serial_3.16.02.qcow2</name>
    <type>root_image</type>
    <sha1_checksum>93de73ee3531f74fddf99377972357a8a0eac7b</sha1_checksum>
  </File_Info>
  <File_Info>
    <name>image_properties.xml</name>
    <type>image_properties</type>
    <sha1_checksum>c5bb6a9c5e8455b8698f49a489af3082c1d9e0a9</sha1_checksum>
  </File_Info>
  <File_Info>
    <name>ISRv_ovf_env.xml</name>
    <type>bootstrap_file_1</type>
    <sha1_checksum>c5bb6a9c5e8455b8698f49a489af3082c1d9e0a9</sha1_checksum>
  </File_Info>
```
Example: Image Properties

```xml
<image_properties>
  <vnf_type>ROUTER</vnf_type>
  <name>ISRv-universal9</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>
  <pci_driver_list>igb</pci_driver_list>
  <pci_driver_list>igbvf</pci_driver_list>
  <pci_driver_list>i40evf</pci_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>
    <profile>
      <name>ISRv1kv-medium</name>
      <description>ISRv upto 250MBPS performance</description>
      <vcpus>2</vcpus>
      <memory_mb>4096</memory_mb>
      <root_disk_mb>8</root_disk_mb>
    </profile>
  </profiles>
  <default_profile>small</default_profile>
</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>
  <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>
</image_properties>
```
Image Registration

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. All VM images are available in VM packaging and VM package content. For more information see, VM Image Packaging Utility, on page 82

Register VM Packages Using REST API

This example shows the sequence of registering a tar.gz package on Cisco Enterprise NFVIS using REST API.

Post Image Registration

curl -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/images -d 
Register VM Image with Multiple Root Disks

If any image requires multiple root disks, you can specify it in the image properties file.

This example shows how to specify multiple root disks in image properties.

```xml
<image_properties>
  ...
  <root_file_disk_bus>virtio</root_file_disk_bus>
  <root_image_disk_format>qcow2</root_image_disk_format>
  <disk_1_file_disk_bus>virtio</disk_1_file_disk_bus>
  <disk_1_image_format>qcow2</disk_1_image_format>
  <disk_2_file_disk_bus>virtio</disk_2_file_disk_bus>
  <disk_2_image_format>qcow2</disk_2_image_format>
  ...
</image_properties>
```
Register a VM Image through a Root Disk

A VM can also be registered using just a disk image (qcow2 or iso) without packaging into a tar.gz. As there will be no image properties in this scenario, the default image properties are used.

Default Properties for Root Disk Registration

The following table lists the default properties that are provisioned if you register a VM image by uploading the root disk for the image.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Tag</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>&lt;version&gt;</td>
<td>NA</td>
</tr>
<tr>
<td>VNF Type</td>
<td>&lt;vnf_typ&gt;</td>
<td>OTHER</td>
</tr>
<tr>
<td>VCPU Min</td>
<td>&lt;vcpu_min&gt;</td>
<td>1</td>
</tr>
<tr>
<td>VCPU Max</td>
<td>&lt;vcpu_max&gt;</td>
<td>64</td>
</tr>
<tr>
<td>Memory Min (MB)</td>
<td>&lt;memory_mb_min&gt;</td>
<td>256</td>
</tr>
<tr>
<td>Memory Max (MB)</td>
<td>&lt;memory_mb_max&gt;</td>
<td>1048576</td>
</tr>
<tr>
<td>Root Disk Min Size (GB)</td>
<td>&lt;root_disk_gb_min&gt;</td>
<td>1</td>
</tr>
<tr>
<td>Root Disk Max Size (GB)</td>
<td>&lt;root_disk_gb_max&gt;</td>
<td>10240</td>
</tr>
<tr>
<td>VNIC Max</td>
<td>&lt;vnic_max&gt;</td>
<td>8</td>
</tr>
<tr>
<td>Bootup Time</td>
<td>&lt;bootup_time&gt;</td>
<td>-1</td>
</tr>
<tr>
<td>Interface Hot Add</td>
<td>&lt;interface_hot_add&gt;</td>
<td>true</td>
</tr>
<tr>
<td>Interface Hot Delete</td>
<td>&lt;interface_hot_delete&gt;</td>
<td>false</td>
</tr>
</tbody>
</table>

Register a Remote VM Image

Cisco Enterprise NFVIS allows you to register a VM image that is stored at a remote location or a web server by specifying the URL to the image location in the source field.

If the web server supports HTTPS, you can choose to enable Certificate Validation to validate its authenticity. Certificate Validation requires the server’s public key to be specified, either in string or file format, in the image registration payload. This allows NFVIS to perform asymmetric encryption and download/register the image file securely over HTTPS.
Example: POST Remote Image Registration from Webserver over HTTPS

curl -k -v -u admin:Esc123# -H
Accept:application/vnd.yang.data+xml -H
Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST
https://172.29.91.28/api/config/vm_lifecycle/images/ -d ' 
  <image>
    <name>ASAV</name>
    <src>https://172.20.117.124/nfvis/asav982.tar.gz</src>
  </image>'

HTTP/1.1 201 Created

Example: POST Remote Image Registration from Webserver over HTTPS

curl -k -v -u admin:Esc123# -H
Accept:application/vnd.yang.data+xml -H
Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST
https://172.29.91.28/api/config/vm_lifecycle/images/ -d ' 
  <image>
    <name>ASAV</name>
    <src>https://172.20.117.124/nfvis/asav982.tar.gz</src>
    <certificate_validation>true</certificate_validation>
    <certificate_file>/data/intdatastore/uploads/pub_key.cert</certificate_file>
  </image>'

HTTP/1.1 201 Created

Specify Storage Location for a VM Image

Cisco Enterprise NFVIS allows users to specify the location where the register image should be stored, using the `placement` property tag.

The following table lists the placement values supported and their respective mappings.

<table>
<thead>
<tr>
<th>storage name</th>
<th>directory map</th>
</tr>
</thead>
<tbody>
<tr>
<td>datastore1</td>
<td>/data</td>
</tr>
<tr>
<td>datastore2</td>
<td>/mnt/extdatastore1</td>
</tr>
<tr>
<td>datastore3</td>
<td>/mnt/extdatastore2</td>
</tr>
<tr>
<td>nfs:nfs_mount_name</td>
<td>/data/mount/nfs/nfs_mount_name</td>
</tr>
<tr>
<td>nfs or nfs:nfs</td>
<td>/data/mount/nfs/</td>
</tr>
<tr>
<td>nfs:nfs_storage or nfs_storage</td>
<td>/data/mount/nfs_storage</td>
</tr>
</tbody>
</table>

**Note**

If your preferred storage location is `nfs`, you must have it configured to be mounted on NFVIS using appropriate CLIs before registering the image on it.

Example: VM Image Storage Placement

curl -k -v -u admin:Esc123# -H
Accept:application/vnd.yang.data+xml -H
You can only update the following image properties after a VM image has been registered.

- interface_hot_add
- interface_hot_delete

Starting from NFVIS 4.1 release, interface hot add and delete are now supported on DPDK networks.

**Note**

When using the REST API, the previously set value of the property must be deleted before updating it with the new value.

**Example: Delete Value and Add New Value**

1. Delete the previously set property value as shown below.

   ```
   HTTP/1.1 204 No Content
   ```

2. Add (PUT) the new property value to replace the one you deleted in the previous step.

   ```
curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://172.29.91.28/api/config/vm_lifecycle/images/image/ISR_IMAGE/properties/property/interface_hot_add --data '<value>true</value>'
   HTTP/1.1 201 Created
   ```

**Example: Get All Properties and Values**

```
curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Accept:application/vnd.yang.data+xml -H
```
Type: application/vnd.yang.data+xml
HTTP/1.1 200 OK

```xml
<properties xmlns="http://www.cisco.com/nfvis/vm_lifecycle"
    <property>
        <name>interface_hot_add</name>
        <value>true</value>
    </property>
    <property>
        <name>interface_hot_delete</name>
        <value>false</value>
    </property>
</properties>
```

## Image Properties

The `image_properties.xml` file in the tar.gz package contains the property configuration data for a particular image. Figure 3 depicts an example `image_properties.xml` file.

Some of the properties are mandatory and must be specified to register an image. If any of the mandatory properties are omitted, the image registration fails.

Optional properties can be specified on an image-by-image basis and are not required.

The following table lists all the image properties that are supported in Cisco Enterprise NFVIS.

<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><code>&lt;vnf_type&gt;</code></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><code>&lt;name&gt;</code></td>
<td>Any</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Version</td>
<td>Version of the package</td>
<td><code>&lt;version&gt;</code></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><code>&lt;bootup_time&gt;</code></td>
<td>Any in seconds, (-1) to not monitor boot-up</td>
<td>Optional, default -1</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Property Tag</td>
<td>Possible Values</td>
<td>Mandatory/Optional</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>----------------------------------</td>
<td>--------------------</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>Boot Mode</td>
<td>Specifies the mode in which the VNF will boot. Used for Secure Boot feature</td>
<td>&lt;boot_mode&gt;</td>
<td>efi-secure-boot, bios</td>
<td>Optional, default bios</td>
</tr>
<tr>
<td>Shim Signature</td>
<td>If using efi-secure-boot boot mode, a shim signature must be provided</td>
<td>&lt;shim_signature&gt;</td>
<td>microsoft, N/A</td>
<td>Required if Boot Mode is specified</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 disk 1 image format</td>
<td>&lt;disk_1_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-2 Image format</td>
<td>Disk 2 image format</td>
<td>&lt;disk_2_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Disk-10 Image format</td>
<td>Disk 10 image format</td>
<td>&lt;disk_10_image_format&gt;</td>
<td>qcow2 and raw</td>
<td>Optional</td>
</tr>
<tr>
<td>Serial Console</td>
<td>Serial console supported</td>
<td>&lt;console_type_serial&gt;</td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>Minimum vCPU</td>
<td>Minimum vCPUs required for a VM operation</td>
<td>&lt;vcpu_min&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Maximum vCPU</td>
<td>Maximum vCPUs supported by a VM</td>
<td>&lt;vcpu_max&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Property Tag</td>
<td>Possible Values</td>
<td>Mandatory/ Optional</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Minimum memory</td>
<td>Minimum memory in MB required for VM operation</td>
<td>&lt;memory_mb_min&gt;</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Maximum memory</td>
<td>Maximum memory in MB supported by a VM</td>
<td>&lt;memory_mb_max&gt;</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>&lt;root_disk_gb_min&gt;</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>&lt;root_disk_gb_max&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Maximum vNICs</td>
<td>Maximum number of vNICs supported by a VM</td>
<td>&lt;vnic_max&gt;</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>&lt;sriov_supported&gt;</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>&lt;sriov_driver_list&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>PCI passthru support</td>
<td>PCI passthru support by VM interfaces</td>
<td>&lt;pcie_supported&gt;</td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>PCIE driver list</td>
<td>List of VNICS to enable PCI passthru support</td>
<td>&lt;pcie_driver_list&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>bootstrap_cloud_init_drive_type</td>
<td>Mounts day0 config file as disk (default is CD-ROM)</td>
<td>&lt;bootstrap_cloud_init_drive_type&gt;</td>
<td>disk, cdrom</td>
<td>Optional</td>
</tr>
<tr>
<td>bootstrap_cloud_init_bus_type</td>
<td>Default is IDE</td>
<td>&lt;bootstrap_cloud_init_bus_type&gt;</td>
<td>virtio, ide</td>
<td>Optional</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Property Tag</td>
<td>Possible Values</td>
<td>Mandatory/Optional</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>BOOTSTRAP</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 ASA v</td>
<td>&lt;bootstrap_file&gt;</td>
<td>File name of the bootstrap file</td>
<td>Optional</td>
</tr>
<tr>
<td>Custom properties</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></td>
<td>Optional</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Property Tag</td>
<td>Possible Values</td>
<td>Mandatory/Optional</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Profiles for VM deployment</td>
<td>List of VM deployment profiles. Minimum one profile is required</td>
<td>&lt;profiles&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Default profile</td>
<td>The default profile is used when no profile is specified during deployment.</td>
<td>&lt;default_profile&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Monitoring Support</td>
<td>A VM supports monitoring to detect failures.</td>
<td>&lt;monitoring_supported&gt;</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>&lt;monitoring_methods&gt;</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>&lt;low_latency&gt;</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>&lt;privileged_vm&gt;</td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>Disable Spoof Check</td>
<td>Used to disable spoof check for Privileged VMs</td>
<td>&lt;disable_spoof_check&gt;</td>
<td>true, false</td>
<td>Optional</td>
</tr>
<tr>
<td>Virtual interface model</td>
<td></td>
<td>&lt;virtual_interface_model&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Property Tag</td>
<td>Possible Values</td>
<td>Mandatory/Optional</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Interface Hot Add</td>
<td>If true, an active VNF’s virtual interface can be added/updated without shutting down the VNF.</td>
<td>&lt;interface_hot_add&gt;</td>
<td>true, false</td>
<td>Optional, default true</td>
</tr>
<tr>
<td>Interface Hot Delete</td>
<td>If true, an active VNF’s virtual interface can be deleted without shutting down the VNF.</td>
<td>&lt;interface_hot_delete&gt;</td>
<td>true, false</td>
<td>Optional, default false</td>
</tr>
<tr>
<td>Thick disk provisioning</td>
<td>During deployment, VM will be a fully allocated root disk with size specified by flavor.</td>
<td>&lt;thick_disk_provisioning&gt;</td>
<td>true, false</td>
<td>Optional, default false</td>
</tr>
<tr>
<td>Eager Zero</td>
<td>Used in conjunction with Thick disk provisioning. During deployment, root disk is zeroed out to improve I/O operations.</td>
<td>&lt;eager_zero&gt;</td>
<td>true, false</td>
<td>Optional, only valid if Thick disk provisioning is enabled. Default false</td>
</tr>
<tr>
<td>Profile for VM deployment</td>
<td>A profile defines the resources required for VM deployment. This profile is referenced during VM deployment.</td>
<td>&lt;profile&gt;</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Name</td>
<td>Profile name</td>
<td>&lt;name&gt;</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>
</tbody>
</table>
### Property Name | Description | Property Tag | Possible Values | Mandatory/Optional
--- | --- | --- | --- | ---
vCPU | vCPU number in a profile | `<vcpus>` |  | Mandatory
Memory | Memory - MB in profile | `<memory_mb>` |  | Mandatory
Root Disk Size | Disk size - MB in profile | `<root_disk_mb>` |  | Mandatory
VNIC Offload | List of properties that can be set for vnic offload | `<vnic_offload>` |  | Optional
Generic Segmentation Offload | Turn generic segmentation offload on or off | `<generic_segmentation_offload>(parent: `<vnic_offload>`)` | on, off | Optional
Generic Receive Offload | Turn generic receive offload on or off | `<generic_receive_offload>(parent: `<vnic_offload>`)` | on, off | Optional
RX Checksumming | Turn RX checksumming on or off | `<rx_checksumming>(parent: `<vnic_offload>`)` | on, off | Optional
TX Checksumming | Turn TX checksumming on or off | `<tx_checksumming>(parent: `<vnic_offload>`)` | on, off | Optional
TCP Segmentation Offload | Turn TCP segmentation offload on or off | `<tcp_segmentation_offload>(parent: `<vnic_offload>`)` | on, off | Optional

---

#### Example: Contents of an image_properties.xml File

```xml
<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
  <vnf_type>ROUTER</vnf_type>
  <name>ISRV</name>
  <version>16.06.05</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>
  <vnic_names>vnics:1:GigabitEthernet2</vnic_names>
  <vnic_names>vnics:2:GigabitEthernet3</vnic_names>
  <vnic_names>vnics:3:GigabitEthernet4</vnic_names>
  <vnic_names>vnics:4:GigabitEthernet5</vnic_names>
  <vnic_names>vnics:5:GigabitEthernet6</vnic_names>
</image_properties>
```
<vnics>vnics:6:GigabitEthernet7</vnics>
<vnics>vnics:7:GigabitEthernet8</vnics>
<source_disk_gb_min>8</source_disk_gb_min>
<source_disk_gb_max>8</source_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>
<monitoring_supported>true</monitoring_supported>
<monitoring_methods>ICMPPing</monitoring_methods>
<low_latency>true</low_latency>
<privileged_vm>true</privileged_vm>
<cdrom>true</cdrom>
<bootstrap_file_1>ovf-env.xml</bootstrap_file_1>
<bootstrap_file_2>iosxe_config.txt</bootstrap_file_2>
<custom_property>
  <tech_package>ax</tech_package>
  <tech_package>security</tech_package>
  <tech_package>ipbase</tech_package>
  <tech_package>appx</tech_package>
</custom_property>
<custom_property>
  <ngio>enable</ngio>
</custom_property>
<custom_property>
  <SSH_USERNAME></SSH_USERNAME>
</custom_property>
<custom_property>
  <SSH_PASSWORD></SSH_PASSWORD>
</custom_property>
<profiles>
  <profile>
    <name>ISRv-mini</name>
    <description>ISRv-mini</description>
    <vcpus>1</vcpus>
    <memory_mb>4096</memory_mb>
    <root_disk_mb>8192</root_disk_mb>
  </profile>
  <profile>
    <name>ISRv-small</name>
    <description>ISRv-small</description>
    <vcpus>2</vcpus>
    <memory_mb>4096</memory_mb>
    <root_disk_mb>8192</root_disk_mb>
  </profile>
  <profile>
    <name>ISRv-medium</name>
    <description>ISRv-medium</description>
    <vcpus>4</vcpus>
    <memory_mb>4096</memory_mb>
    <root_disk_mb>8192</root_disk_mb>
  </profile>
</profiles>
<default_profile>ISRv-small</default_profile>
NFVIS Container Image Registration

The NFVIS container image registration follows the same workflow as VM image registration. However, there is a difference in the container image registration payload, where a `<locator>` field is included. This field must be populated with the word "container", and it signifies that it is a container image and not a VM image. Only files with the extension .docker are supported and can be uploaded for container image registration.

Here's a sample payload of container image registration:

```
<vm_lifecycle xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
               xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
               xmlns:ns3="http://www.cisco.com/esc/esc_notifications"
               xmlns:ns0="http://www.cisco.com/nfvis/vm_lifecycle"
               xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <images>
    <image>
      <name>ubuntu-1</name>
      <src>file:///data/intdatastore/uploads/ubuntu.docker</src>
      <remove_src_on_completion>true</remove_src_on_completion>
      <locator>
        <vim_id>container</vim_id>
      </locator>
    </image>
  </images>
</vm_lifecycle>
```

The container image does not include any profiles. Before deploying a container, you have to either create a customized profile or use a profile from another image that is suitable.

VM Profiles or Flavors

Flavors or profiles define VMs in terms of number of parameters for how to run the VM. Some of the parameters that you can define in a VM profile or flavor are: number of vCPUs, RAM, disk size and so on.

Flavors are created as part of image registration if you use the tar.gz image packages for registering a VM. However, for other image packages such as .qcow2, iso, and raw, you must define custom flavors based on your requirements.

```
Note
The container image does not include any profiles. Before deploying a container, you have to either create a customized profile or use a profile from another image that is suitable.
```

Example: Create VM Profile Using Rest API

The following example shows how to create a VM profile or flavor using REST API.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/flavors -d '=<flavor>
  <name>windows</name>
  <ephemeral_disk_mb>0</ephemeral_disk_mb>
```

```
Note
Unless specified otherwise in the deployment payload, the value assigned to the custom image property default_profile is used at the time of deploying the VM. Only applicable to tar.gz image packages.
```

```
Example: Create VM Profile Using Rest API

The following example shows how to create a VM profile or flavor using REST API.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/flavors -d '=<flavor>
  <name>windows</name>
  <ephemeral_disk_mb>0</ephemeral_disk_mb>
```

```
Note
Unless specified otherwise in the deployment payload, the value assigned to the custom image property default_profile is used at the time of deploying the VM. Only applicable to tar.gz image packages.
```
Configure Internal Management Network

Monitored VNFs must have one internal management network specified to the NIC ID “0” in the deployment payload. If the IP address is configured, it needs to belong to the same network as the internal subnet. By default, the internal management network has the range 10.20.0.0. If the IP address is not specified in the VM deployment payload, the system assigns an IP address for internal management network using IP address pool.

Here an example of configuring internal management network.

```xml
<interface>
  <nicid>0</nicid>
  <network>int-mgmt-net</network>
  <ip_address>10.20.0.21</ip_address>
</interface>
```

VM Deployment and Management

VM Deployment

A VM can be deployed through API or CLI. Ensure that you have registered a VM image before attempting to deploy it. For more details, see Image Registration, on page 99.

Note

The VM name must meet the following requirements:

- Must contain an uppercase character and a lowercase character.
- Must contain a digit.
- Must contain one of the following special characters: dot (.), underscore (_) and hyphen (-).
- Must not have more than 256 characters.

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.

Example: Deploy VMs Using REST API

Use the following API to deploy a VM.
Here's a sample payload of deploying a VM.

curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/tenants /tenant/admin/deployments --data '
<deployment>
  <name>ISRdep</name>
  <vm_group>
    <name>ISRvmgrp</name>
    <image>ISR_IMAGE</image>
  </vm_group>
  <bootup_time>500</bootup_time>
  <recovery_wait_time>0</recovery_wait_time>
  <interfaces>
    <interface>
      <nicid>0</nicid>
      <network>int-mgmt-net</network>
      <ip_address>10.20.0.21</ip_address>
      <port_forwarding>
        <port>
          <type>ssh</type>
          <protocol>tcp</protocol>
          <vnf_port>22</vnf_port>
          <external_port_range>
            <start>20022</start>
            <end>20022</end>
          </external_port_range>
        </port>
      </port_forwarding>
    </interface>
  </interfaces>
  <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>
        <nicid>0</nicid>
        <poll_frequency>3</poll_frequency>
        <polling_unit>seconds</polling_unit>
        <continuous_alarm>false</continuous_alarm>
      </metric_collector>
    </kpi>
  </kpi_data>
  <rules>
    <admin_rules>
      <rule>
        <event_name>VM_ALIVE</event_name>
        <action>ALWAYS log</action>
        <action>TRUE servicebooted.sh</action>
        <action>FALSE recover autohealing</action>
      </rule>
    </admin_rules>
  </rules>
  <config_data>
    <configuration>
      <dst>bootstrap_config</dst>
    </configuration>
  </config_data>
</deployment>
Verify VM Deployment

The following example shows how to get the operational data for a VM deployment using the command

```
show vm_lifecycle opdata tenants tenant admin deployments
<deployment_name>/<deployment_id>/<vmgroup_name>
```

```
nfvis# show vm_lifecycle opdata tenants tenant admin deployments ROUTER
deployments ROUTER
deployment_id SystemAdminTenantIdROUTER
vm_group ROUTER
bootup_time 600
vm_instance d1c462e9-2706-4868-befd-d8f7806b9444
name ROUTER
host_id NFVIS
hostname nfvis
interfaces interface 0
model virtio
type virtual
port_id vnic0
network int-mgmt-net
subnet N/A
ip_address 10.20.0.2
mac_address 52:54:00:fe:34:53
netmask 255.255.255.0
gateway 10.20.0.1
interfaces interface 1
type virtual
port_id vnic1
network GE0-1-SRIOV-1
subnet N/A
mac_address 52:54:00:e4:13:67
state_machine state SERVICE_ACTIVE_STATE
VM
NAME STATE
------------------------
ROUTER VM_ALIVE_STATE
```

If the deployment was accepted first but failed later, NFVIS sends a notification with error status and message. Also on VM deployment failure, the operational data may or may not show the complete data about the failed VM.

---

**VM Deployment Parameters**

VNFs can be deployed using multiple mandatory and option parameters. The following table lists some of the parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notes</th>
<th>Example</th>
</tr>
</thead>
</table>

---

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
### VM Deployment Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image</td>
<td>Mandatory. The image needs to be registered and active when it is being referred in deployment.</td>
</tr>
<tr>
<td>bootup_time</td>
<td>This parameter is no longer mandatory from 3.12 and later, provided that it is specified in image properties.   Accepted Values:</td>
</tr>
<tr>
<td></td>
<td>- For unmonitored VMs: -1</td>
</tr>
<tr>
<td></td>
<td>- For monitored VMs: Number of seconds</td>
</tr>
<tr>
<td>vim_vm_name</td>
<td>Optional. If a custom VM name is provided at the time of deployment, it may be used for all commands that accept VM name as an input.</td>
</tr>
<tr>
<td><strong>kpi_data</strong></td>
<td>Mandatory for monitored VMs.</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>rules</strong></td>
<td>Mandatory for monitored VMs.</td>
</tr>
<tr>
<td><strong>config_data</strong></td>
<td>Mandatory if the day-0 configuration has variables that have tokens assigned to them.</td>
</tr>
<tr>
<td><strong>Encrypted config variable</strong></td>
<td>Optional. Only one value for a variable is allowed to be encrypted.</td>
</tr>
</tbody>
</table>

```xml
<variable>
  <name>TEST_VARIABLE</name>
  <encrypted_val>test_value</encrypted_val>
</variable>
```
| **placement** | Optional.  
The *placement* tag under vm group points to the location where the VNF would be deployed. This parameter supports deploying a VNF in a local data store (default-if not specified), external data store (datastore2), or NFS. |
|----------------|--------------------------------------------------|
| **volumes**    | Optional.  
Up to 2 volumes could be added to a deployment. Location of the volumes can be local or NFS (needs NFS mount name to be specified in case of NFS). |
<table>
<thead>
<tr>
<th><strong>port_forwarding</strong></th>
<th><strong>Optional. If port forwarding is included, all elements under it are mandatory.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port_forwarding&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;type&gt;ssh&lt;/type&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;protocol&gt;tcp&lt;/protocol&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;vnf_port&gt;22&lt;/vnf_port&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;external_port_range&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;start&gt;20022&lt;/start&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;end&gt;20022&lt;/end&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/external_port_range&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/port&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/port_forwarding&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ngio interface</strong></th>
<th><strong>Optional. Used in config_data. To enable NIM support on a Cisco ISRv running on Cisco ENCS, you must use the variables in the ISRv deployment payload.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;variable&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;name&gt;ngio&lt;/name&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;val&gt;enable&lt;/val&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/variable&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>interface model</strong></th>
<th><strong>Optional. If the model is not specified for an interface, the default model is used. For Windows, the default model is rtl8139.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;nicid&gt;3&lt;/nicid&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;network&gt;wan-net&lt;/network&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;model&gt;virtio&lt;/model&gt;</td>
<td></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 ways:

- Bundle bootstrap configuration files into the VM package: In this method, the bootstrap configuration variables can be assigned tokens. Token names must be in bold text. For each variable with a token assigned, 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, which includes the 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.

VM Monitoring

After VMs are deployed, they are monitored periodically based on the metrics defined in the KPI section of deployment data model. Monitoring can be enabled or disabled by modifying the <actionType> tag. See the VM Actions section for details on the allowed values for the action Type tag and what they mean.

The following example shows how to disable monitoring for a VM.

```
curl -k -v -u "admin:password" -H "Accept:application/vnd.yang.data+xml" -H "Content-Type:application/vnd.yang.data+xml" -X POST https://<NFVIS_IP>/api/operations/vmAAction --data '<vmAAction><actionType>DISABLE_MONITOR</actionType><vmName><vm-instance name></vmName></vmAction>'
```

- If the bootup_time is set to -1, it signifies that VM monitoring is disabled.
- You are not required to set a boot up time during image registration. However, you must set it during VM deployment.
- If a qcow2 image is used during registration, the bootup_time defaults to -1.

VNF Deployment Placement

Cisco NFVIS allows you to specify where a VNF should be deployed using the placement tag for parameter deployment. See VM Deployment Parameters for more information on supported placement parameters and their accepted values.
If you are placing the VNF deployment on nfs, ensure that you have configured this storage option to be mounted on NFVIS using appropriate CLIs before deploying the VNF.

Example: VM Deployment Using Placement

curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/vm_lifecycle/tenants /tenant/admin/deployments --data 
  '<deployment>
    <name>WINIsodep</name>
    <vm_group>
      <name>WINIsovmgrp</name>
      <image>WinServer2012R2.iso</image>
      <bootup_time>-1</bootup_time>
      <recovery_wait_time>0</recovery_wait_time>
      <kpi_data>
        <enabled>true</enabled>
      </kpi_data>
      <scaling>
        <min_active>1</min_active>
        <max_active>1</max_active>
        <elastic>true</elastic>
      </scaling>
      <placement>
        <type>zone_host</type>
        <enforcement>strict</enforcement>
        <host>datastore1</host>
      </placement>
      <recovery_policy>
        <recovery_type>AUTO</recovery_type>
        <action_on_recovery>REBOOT_ONLY</action_on_recovery>
      </recovery_policy>
    </vm_group>
  </deployment>'

VM Export with Selective Disk

Table 20: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM Export with Selective Disk</td>
<td>NFVIS 4.4.1</td>
<td>You can exclude certain disks or volumes from a VM export.</td>
</tr>
</tbody>
</table>

Starting from NFVIS 4.4, you can exclude certain disks or volumes from a VM export.

nfvis# show running-config vm_lifecycle tenants tenant admin deployments
vm_lifecycle tenants tenant admin
deployments deployment linuxdep
vm_group gp_01
  image  Linux_IMAGE
  flavor  centos-disk-large
  vim_vm_name  linux_vm
bootup_time =-1
recovery_wait_time 0
recovery_policy action_on_recovery REBOOT_ONLY
interfaces interface 0
    model virtio
    network wan-net

! scaling min_active 1
! scaling max_active 1
! placement zone_host
! host datastore1
!
vmexport_policy disk_exclusion Linux_IMAGE:512G-file.qcow2 =====> Disk to be excluded on Export
!
!

VNF Volumes

A VNF can be created and deployed with maximum two volumes. Currently, NFVIS supports a maximum of two volumes per VNF. The empty volume can be used as an extra storage by the VNF. Starting from NFVIS 4.1 release, storage volumes can now be deployed on external datastores and NFS.

Restrictions for VNF Volumes

- Volumes cannot be updated for a VNF after a VNF has already been deployed.
- NFVIS currently supports only two volumes per VNF.
- Starting from NFVIS 4.1 release, volumes can be stored in local storage or NFS, and other data stores like datastore1, datastore2 and datastore3.

Example: Payload for Creating Volumes

...<volumes>
    <volume>
        <name>Volume1</name>
        <volid>1</volid>
        <bus>ide</bus>
        <size>1</size>
        <sizeunit>GiB</sizeunit>
        <format>qcow2</format>
        <device_type>disk</device_type>
        <storage_location>local</storage_location>
    </volume>
...<volumes>

Volume Storage Locations

The following are the accepted values for storage_location tags:

- <storage_location>local</storage_location>
- <storage_location>nfs:NFS_MOUNT_NAME</storage_location>
- <storage_location>datastore1</storage_location>
The storage locations datastore1, datastore2 and datastore3 are not supported for NFVIS 3.12.3 release.

In NFVIS 4.1.1 release, local and datastore1 are identical in storage_location.

### Volume Deployment Parameters

Cisco NFVIS supports the following fields for volume deployment.

<table>
<thead>
<tr>
<th>Property</th>
<th>Allowed Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string, { length 1..255 }</td>
<td>Name of the volume</td>
</tr>
<tr>
<td>valid</td>
<td>uint16</td>
<td>Volumes will be presented to the VM sorted by volume ID.</td>
</tr>
<tr>
<td>bus</td>
<td>virtio, ide, scsi</td>
<td>The bus type</td>
</tr>
<tr>
<td>size</td>
<td>unit 16</td>
<td>Size of the Volume</td>
</tr>
<tr>
<td>sizeunit</td>
<td>MiB, GiB, TiB, PiB</td>
<td>Size units. MiB/GiB/TiB/PiB</td>
</tr>
<tr>
<td>format</td>
<td>qcow2, raw</td>
<td>Format of the disk to be created.</td>
</tr>
<tr>
<td>device_type</td>
<td>disk</td>
<td>Type of the device being attached to the VM.</td>
</tr>
<tr>
<td>storage_location</td>
<td>local, datastore1, datastore2, datastore3, nfs:NFS_MOUNT_NAME</td>
<td>Storage location name</td>
</tr>
</tbody>
</table>

**Note** Starting from NFVIS 4.1 release, external datastore storage location is supported.

### Port Forwarding

By default, the wan bridge interface (wan-br) is used to redirect incoming traffic from WAN to access the internal management network (int-mgmt-net) of the VM.

The bridge interface that is used to redirect traffic coming from the WAN side can be modified using the source_bridge tag in the deployment payload as shown below.

```xml
<port_forwarding>
  <port>
    <type>ssh</type>
    <protocol>tcp</protocol>
    <vnf_port>22</vnf_port>
    <source_bridge>MGMT</source_bridge>
    <external_port_range>
      <start>20122</start>
      <end>20122</end>
    </external_port_range>
  </port>
</port_forwarding>
```
With the payload above, the traffic coming from the WAN side is redirected through the management interface (MGMT) instead of the default WAN bridge (wan-br) interface.

**NGIO**

Next Generation Input/Output (NGIO) is supported on Cisco ENCS platforms from NFVIS release 3.11 and later. Using the NGIO flag, the VM image informs its capability. Between ISRv and NFVIS image, NGIO is used to decide the NIM enablement capability available for the VM. This is only for Cisco ISRv with Cisco IOS XE image or Cisco IOS XE with SD-WAN image.

To enable NGIO on a VNF, a config data variable is added to the deployment payload as shown below.

```xml
<config_data>
  <configuration>
    <dst>bootstrap_config</dst>
    <variable>
      <name>TECH_PACKAGE</name>
      <val>ax</val>
    </variable>
    <variable>
      <name>ngio</name>
      <val>enable</val>
    </variable>
  </configuration>
</config_data>
```

The following is an example of the interfaces added to a VNF, when the NGIO variable is detected and set to enable.

```xml
<interface type='bridge'>
  <source bridge='csxbr'/>
  <model type='e1000'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' multifunction='on'/>
</interface>

<interface type='hostdev' managed='yes'>
  <driver name='vfio'/>
  <source>
    <address type='pci' domain='0x0000' bus='0x0e' slot='0x10' function='0x1'/>
  </source>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x2' multifunction='on'/>
</interface>

<interface type='hostdev' managed='yes'>
  <driver name='vfio'/>
  <source>
    <address type='pci' domain='0x0000' bus='0x0e' slot='0x10' function='0x3'/>
  </source>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x2' multifunction='on'/>
</interface>
```

**Requirements for NGIO Validation**

When NGO deployment request is initiated, it is validated against the specified image and the platform itself. To be accepted, the platform must be supported and the registered image must have the following properties.

```xml
<image_properties>
  ...
  <custom_property>
    <ngio>enable</ngio>
  </custom_property>
</image_properties>
```
VM States

The following table describes various 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 a VM or VNF before deployment of this VM.</td>
</tr>
<tr>
<td>VM_DEPLOYING_STATE</td>
<td>The VM or VNF is being deployed on to the NFVIS.</td>
</tr>
<tr>
<td>VM_MONITOR_UNSET_STATE</td>
<td>The VM or VNF is deployed in 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 to the VM 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 shown in the KPI metric.</td>
</tr>
<tr>
<td>VM_UNDEPLOYING_STATE</td>
<td>The deployment of a VM or VNF is being terminated.</td>
</tr>
<tr>
<td>VM_ERROR_STATE</td>
<td>The VM or VNF is in an error state because the deployment or some other operation has failed.</td>
</tr>
</tbody>
</table>

NFVIS Container Deployment

NFVIS container deployment follows the same workflow as VM deployment.

Restrictions for NFVIS Container Deployment

- Import and export of containers is not supported.

The container deployment process in container lifecycle management supports the following:

- SRIOV/Host Interface
- Day 0 Configuration
- Configuration Options
- vCPU Pinning
- Volumes
SRIOV/Host Interface

To attach an SRIOV or host interface to the container's interface, specify the SRIOV or host interface in the configuration. Either physical interfaces or SRIOV VFs can be specified in the below configuration.

To attach an SRIOV or host interface to the container's interface, use the following configuration:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<vm_lifecycle xmlns:ns0="http://www.cisco.com/nfvis/vm_lifecycle"
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<tenants>
	<tenant>
		<name>admin</name>
		<deployments>
		<deployment>
			<name>ubuntu-2</name>
			<vm_group>
				...
				<interfaces>
					<interface>
						<nicid>0</nicid>
						<network>int-LAN-vf-1</network>
					</interface>
				</interfaces>
			</vm_group>
			</deployment>
		</deployments>
	</tenant>
</tenants>
</vm_lifecycle>
```

Day 0 Configuration

Day 0 support for container deployments follows the same process as that of VM deployments. Refer to VM Bootstrap Configuration Options with a VM Deployment.

Configuration Options

There are multiple boot options available, that allow you to customize the container behavior.

To customize the container behavior using different configuration options, use the following configuration:

```xml
<config_data>
<config_type>CONFIG_DATA_OPTIONS</config_type> <!-- type is required or options below are ignored -->
<config_options>
<options>
<option><name>ENV_VARIABLE</name><value>env=test</value></option>
<option><name>ENV_VARIABLE</name><value>env2=test2</value></option>
</options>
</config_options>
<configuration>
<dst>/etc/ovf-env.xml</dst>
...
```
Table 21: Supported Configuration Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENV_VARIABLE</td>
<td>String</td>
<td>Environment variable to set within container: name/value pair (example: varName=varValue)</td>
</tr>
<tr>
<td>LOW_LATENCY</td>
<td>Boolean</td>
<td>vCPU pinning option: When the value is True, the container is allocated a dedicated core.</td>
</tr>
</tbody>
</table>

**vCPU Pinning**

Similar to the NFVIS VM model, containers can also be pinned to a specific vCPU. If a container requires a dedicated core, a low latency flag can be set using the configuration options. By default, no low latency flag is set, this means that the core may be shared with other containers. Because the workflow for container vCPU pinning is the same as VM deployments, VimManager reuses the Node.js CPU allocation API to interface with the NFVIS CPU allocation scripts.

**Volumes**

Volumes support for container deployments uses the same datamodel as VM deployments and provides external storage options to the containers. The `storage_location` tag in the payload refers to the file path within the container where the volume is mounted.

- Volumes are equivalent to docker binds.
- All container volumes are stored on datastore1.

To mount a container using a volume or a specific path, use the following configuration:

```
<vm_group>
  <name>docker-nginx</name>
  <image>nginx</image>
  . . .
  <volumes>
    <volume>
      <name>test1</name>
      <volid>1</volid>
      <storage_location>/etc/datastore</storage_location> <!-- the path to mount inside the container -->
      <size>120</size>
      <sizeunit>MiB</sizeunit>
    </volume>
    <!-- using default size allocations - 10 gibibytes (GiB) -->
    <volume>
      <name>test2</name>
      <volid>2</volid>
      <storage_location>/var/logs</storage_location>
    </volume>
  </volumes>
</vm_group>
```

For more information, see VNF Volumes.
VNF Deployment Update

After you have deployed a VNF, you can update it in terms of its flavor, CPU topology, or interfaces.

Update VNF Flavor

You can update a VNF deployment to have a different flavor from the one you deployed it with. The flavor can also be custom-defined.

**Note**

Before updating a VNF with another flavor, we recommend that you check whether CPUs are available for the required update.

Updating a VNF flavor only supports CPU and Memory changes and does not support disk size change.

The following is a typical workflow for updating a VNF flavor.

**Get Available Flavors**

Use the following command to get a list of available flavors.

```
curl -k -v -u admin:admin -X GET https://209.165.201.1/api/operational/vm_lifecycle/flavors?deep
```

**Check System CPU Usage**

Use the following command to check the CPU usage of the system.

```
curl --tlsv1.2 -k -i -u admin:Esc123# -H Accept:application/vnd.yang.data+json -H content-type:application/vnd.yang.data+json -X GET https://<nfvis_ip>/api/operational/resources/cpu-info/allocation
```

**Update VNF Flavor**

Use the following command to update the flavor of a VNF.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/<deploymentID>/vm_group/<VMGroupName>/flavor --data '<flavor><FlavorName></flavor>
```

**Example: Changing the Flavor of a VNF from Flavor ASAv5 to ASAv10**

```
curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://172.29.91.32/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/flavor --data '<flavor>ASAv10</flavor>
```

**Update CPU Topology**

Updating the CPU topology of a VNF involves updating a VM to a custom-defined topology that you would have created at the time of creating a VM flavor. For more details, see VM Profiles or Flavors, on page 112.
The process to update the CPU topology is similar to updating a VNF flavor—by replacing the name of the CPU topology. The following command is used to update a CPU topology.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/<deploymentID>/vm_group/<VMGroupName>/flavor --data "<flavor><FlavorName_withCPUTopology></flavor>"
```

**Example: Updating ISRv to Include a CPU Topology**

```
curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://172.29.91.32/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ROUTER/vm_group/ROUTER/flavor --data '<flavor>Isrv_CPUToplogy</flavor>'
```

**Verify Changed Configuration**

Use the `support virsh dumpxml <uid>` command to verify whether the changed configuration was applied to the VNF.

The following is an example of the output you would see.

```
<vcpu placement='static'>3</vcpu>
<putune>
  <vcpupin vcpu='0' cpuset='11'/>
  <vcpupin vcpu='1' cpuset='10'/>
  <vcpupin vcpu='2' cpuset='9'/>
  <emulatorpin cpuset='21-23'/>
</cputune>
.
<cpu mode='host-passthrough' check='none'>
  <topology sockets='1' cores='3' threads='1'/>
</cpu>
```

**About Updating VNF Interfaces**

NFVIS supports updating VNF interfaces at the time of updating a VM deployment. VM interface updates include adding an interface, deleting an interface, and moving a VNIC of a VM from one interface to another.

**Hot and Cold Updates**

All the options related to updating interfaces mentioned above can be done in two ways: hot or cold.

- **Hot Update**: Hot update refers to an update operation that runs when a VM is in ACTIVE state. In such cases, the VM does not reboot during the update.

- **Cold Update**: Cold update refers to an update operation that runs after a VM is put in a VM_SHUTOFF_STATE. In such cases, the VM reboots during the update.
The need to do a hot or cold update depends on the custom image properties set during image registration. Refer to the table below for various custom image properties related to hot and cold updates.

<table>
<thead>
<tr>
<th>Custom Image Property</th>
<th>Value</th>
<th>Interface Update Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface_hot_add</td>
<td>true</td>
<td>Hot add an interface to a VM</td>
<td>true</td>
</tr>
<tr>
<td>interface_hot_add</td>
<td>false</td>
<td>Cold add an interface to VM</td>
<td>False</td>
</tr>
<tr>
<td>interface_hot_delete</td>
<td>true</td>
<td>Hot delete a VM interface</td>
<td>False</td>
</tr>
<tr>
<td>interface_hot_delete</td>
<td>false</td>
<td>Cold delete a VM interface</td>
<td></td>
</tr>
</tbody>
</table>

Starting from ISRv 17.1, interface_hot_add and interface_hot_delete are set to true by default.

**Prerequisites for Updating VNF Interfaces**

- The VNF should support hot add or hot delete operations for the interface.
- The custom image properties for interface update should be set during image registration to allow values other than the default.
- The VM to be updated needs to be in one of the following states: VM_ALIVE_STATE, VM_ERROR_STATE or VM_SHUTOFF_STATE.

**Supported Interface Update Operations by Interface Types**

The following table shows which hot interface update operations are supported for various interface types.

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Hot Add</th>
<th>Hot Delete</th>
<th>Hot Update for Moving VNIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIRTIO</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SRIOV</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DPDK</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note**

NFVIS also supports moving VNICS from one interface to another. For example, you can move a VNIC from a VIRTIO interface to SRIOV, or from SRIOV to DPDK, and so on.

If the VNIC is updated to a different interface type like SRIOV or DPDK, the configuration of the vnic will not be preserved.

Syslog is not generated in ISRv when an interface is updated from a DPDK enabled network to another DPDK enabled network.

Interface update of any SRIOV interface to remove from one interface and add the same to another or swap, in a single transaction fails. Use two separate requests for a successful update.
Update Interfaces

This topic walks you through how to perform various tasks related to updating interfaces such as adding an interface, deleting an interface, and so on.

Add Interfaces

Single Interface: The following example shows how to add a single interface to a VM deployment.

```
curl -k -v -u admin:Esc123# -H "Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml" -X PUT https://<NfvisIpAddress>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data "
<interfaces>
  <interface>
    <nicid>0</nicid>
    <network>int-mgmt-net</network>
  </interface>
  <interface>
    <nicid>newNIC</nicid>
    <network>networkName</network>
  </interface>
</interfaces>
```

Multiple Interfaces: The following example shows how to add multiple interfaces (in this case, two) to a VM deployment.

```
curl -k -v -u admin:Esc123# -H "Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml" -X PUT https://172.29.91.32/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data "
<interfaces>
  <interface>
    <nicid>0</nicid>
    <network>int-mgmt-net</network>
  </interface>
  <interface>
    <nicid>1</nicid>
    <network>wan-net</network>
  </interface>
  <interface>
    <nicid>2</nicid>
    <network>lan-net</network>
  </interface>
</interfaces>
```

HTTP/1.1 204 No Content

Delete Interfaces

The following REST API is used to delete an interface from a VM deployment.

```
curl -k -v -u admin:<password> -H "Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml" -X PUT https://<NfvisIpAddress>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data "
<interfaces>
  <interface>
    <nicid>0</nicid>
    <network>int-mgmt-net</network>
  </interface>
</interfaces>
```

HTTP/1.1 204 No Content
**** Note: Remove the required nicID along with content between <interface> and </interface> ' 
******

Example: The following example shows how to delete an interface called NIC ID2, which was added in the example of adding multiple interfaces in the Add Interfaces section above.

curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://172.29.91.32/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data ' 
  <interfaces> 
  <interface> 
  <nicid>0</nicid> 
  </interface> 
  <interface> 
  <nicid>1</nicid> 
  </interface> 
  </interfaces>'

HTTP/1.1 204 No Content

Notice that in the example above, NIC ID 2 has been excluded from the REST API for it to be deleted from the deployment.

Move VNICS from One Network to Another

curl -k -v -u admin:<password> -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://<NfvisIpAddress>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data ' 
  <interfaces> 
  <interface> 
  <nicid>0</nicid> 
  <network>int-mgmt-net</network> 
  </interface> 
  <interface> 
  <nicid>1</nicid> 
  <network>wan-net</network> 
  </interface> 
  <interface> 
  <nicid>selectedNicId</nicid> 
  <network>NewNetworkSelected</network> 
  </interface> 
  </interfaces>'

Example: The following example shows how to move nicid 1 from wan-net to wan2-net.

curl -k -v -u admin:Esc123# -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X PUT https://172.29.91.32/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces --data ' 
  <interfaces> 
  <interface> 
  <nicid>0</nicid> 
  <network>int-mgmt-net</network> 
  </interface> 
  <interface> 
  <nicid>1</nicid> 
  <network>wan2-net</network> 
  </interface> 
  </interfaces>'

HTTP/1.1 204 No Content
Access VNFs

In Cisco NFVIS, you can access VNFs in three ways after they have been deployed:

- Through VNC Console
- Through Serial Console.
- Through Port Forwarding if VM is deployed with port_forwarding configuration in deployment payload.

Access VNFs Using VNC Console

A VNC console allows you to access VNFs through a VNC client. This method enables you to manipulate the VNF through the NFVIS portal.

If you are using the NFVIS portal, follow these steps to access the VNF using the VNC console.

1. On the NFVIS portal, click the VM Life Cycle tab.
2. Next, click the black console icon next to the VNF you want to access (see screenshot below).

Secure Access to the VNC Console

For added security, access to the VNC console can be restricted through a passphrase. To enable VNC passphrase, include the contents as shown in the example below.

```
<vm_lifecycle>
  <tenants>
    <name>admin</name>
    <deployments>
      <deployment>
        <vnc>
          <password>PASSWORD</password>
        </vnc>
      </deployment>
    </deployments>
  </tenant>
</vm_lifecycle>
```

Access VMs Using Serial Console

The serial console allows you to access the VM using the serial interface provided by the VM itself. This method is applicable only if the VM supports serial interfaces in both, its image and its image properties.
To access the VNF through the serial console, use the command `vmConsole` followed by the name of the VNF as shown in the CLI example below.

```
nfvis# vmConsole OTHER
Connected to domain OTHER
Escape character is ^]...
```

When you access the VM console using the ENCS console port on the device, you cannot use ctrl+] to exit the VM console. You must use ctrl+] and then enter `send escape` to exit the VM console.

```
nfvis# show system deployments
NAME ID STATE
--------------------------------------
sj-02-sj02-isrv.sj-02-sj02-isrv 1 running
sj-02-sj02-linux.sj-02-sj02-linux 2 running
sj-02-sj02-vwaas.sj-02-sj02-vwaas 3 running
sj-02-sj02-firewall.sj-02-sj02-firewall 4 running

nfvis#
nfvis# vmConsole sj-02-sj02-vwaas.sj-02-sj02-vwaas
Connected to domain sj-02-sj02-vwaas.sj-02-sj02-vwaas
Escape character is ^]...

sj-02-sj02-vwaas#
sj-02-sj02-vwaas#
sj-02-sj02-vwaas#
telnet> send escape
```

**Access a VM Using Port Forwarding**

To access a VM using port forwarding:

1. Deploy a VNF using the deployment payload with `port_forwarding` configuration:

```
<port_forwarding>
  <port>
    <type>ssh</type>
    <protocol>tcp</protocol>
    <vnf_port>22</vnf_port>
    <external_port_range>
      <start>20122</start>
      <end>20122</end>
    </external_port_range>
  </port>
  <port>
    <type>telnet</type>
    <protocol>tcp</protocol>
    <vnf_port>23</vnf_port>
    <external_port_range>
      <start>20123</start>
      <end>20123</end>
    </external_port_range>
  </port>
```

**Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x**
2. Log into VNF using SSH and port number given in the example payload (20122):

```
USER-M-G2PT:~ user$ ssh cisco@172.29.91.28 -p 20122
Password: isrv-encs#
```

## Import and Export NFVIS VM

Starting from NFVIS 3.10.1 release, you can backup or export (vmExportAction) and restore or import (vmImportAction) VMs. To backup or restore the whole NFVIS system, refer Backup and Restore NFVIS and VM Configurations.

### VM Export and Import Limitations

- The imported VM cannot change datastore.
- The original registered image must exist.
- The OVS network name must be identical to the one used by original deployment.
- VM export is dependant on the amount of free space available in the deployed datastore, regardless of the free space available in the destination datastore. For example, when the VM is deployed in the intdatastore (default), you should ensure that the available free space is at least twice that of the deployed VM.

To export a VM ensure that:

- Backup file must be saved to NFVIS datastore or USB.
- Provide a backup name for NFVIS to append .vmbkp extension to the backup name.

You can only create and save a VM backup to datastores. The backup file has .vmbkp extension. To verify the backup:

```
nfvis# show system file-list disk local | display xpath | include backup
/system/file-list/disk/local[si-no='84']/name tiny_backup.vmbkp
```

To import a VM ensure that:

- The Backup file is placed under NFVIS datastores or USB.
- The registered image used by the original deployed VM is in the same datastore, with same properties.
- The exported VM does not exist on the system.
• OVS network used by the original deployment should exist.
• Restored VM is created with the same datastore with same deployment properties.
• The full path name to backup file is used (for example, /mnt/extdatastore1/backup.vmbkp, not extdatastore1:backup)

```
nfvis# vmImportAction importPath /mnt/extdatastore1/tiny_backup.vmbkp
System message at 2019-01-31 19:53:32...
Commit performed by admin via ssh using maapi.
```

Starting from NFVIS 4.1 release, an optional unique MAC UID support is added to VM import.

```
vmImportAction importPath <vm backup file with location> uniqueMacUid
```

Specifying the uniqueMACUid flag ensures that the imported VM is not deployed with the same UID and interface MAC addresses.

The following examples show export failures:

• Original deployment is not deleted

```
nfvis# vmImportAction importPath /mnt/extdatastore1/tiny_backup.vmbkp
Error: Exception from action callback: Deployment Configuration :
'SystemAdminTenantIdtiny' already exists , can not be imported/restored due to conflict!
```

• 2. OVS network used by original deployment is deleted.

```
nfvis# vmImportAction importPath /mnt/extdatastore1/tiny_backup.vmbkp
Error: Exception from action callback: Restoration Request rejected, see logs for root cause
```

**Feature Comparison Table for Import and Export**

VM backup using vmExportAction:

<table>
<thead>
<tr>
<th>Features</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
<th>NFVIS 4.4.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default file location for backup</td>
<td>/mnt/extdatastore1/vmbackup.vmbkp</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>vmExportAction vmName</td>
<td>sampleexportName</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>sample exportName</td>
<td>vmbackup exportPath &lt;datastore&gt;</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Import and Export NFVIS VM

<table>
<thead>
<tr>
<th>Features</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
<th>NFVIS 4.4.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default file location for backup on USB</td>
<td>/mnt-usb/usb1/vmbackup.bkp</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>vmExportAction vmName sample exportName backup02 exportPath usb:usb1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check disk space before backup</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>VM backup format</td>
<td>Full backup</td>
<td>Diff disk backup</td>
<td>Diff disk backup</td>
</tr>
<tr>
<td>Backup image and flavor</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>VM live export snapshot</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>VM Export with Selective Disk</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

VM restore using vmImportAction:

<table>
<thead>
<tr>
<th>Features</th>
<th>NFVIS 4.1.1 and Earlier Releases</th>
<th>NFVIS 4.2.1 Release</th>
<th>NFVIS 4.4.1 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default file location for restore on USB</td>
<td>/mnt-extdatastore1/vmbackup.vmbkp</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>vmImportAction importPath &lt;datastore&gt;:vmbackup.vmbkp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default file location for restore on USB</td>
<td>/mnt-extdatastore2/vmbackup.vmbkp</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>vmImportAction importPath &lt;datastore&gt;:vmbackup.vmbkp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check disk space before backup</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Restore backing images and flavors</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>VM Export with Selective Disk</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>
Secure Boot of VNFs

A key aspect of a secure compute system is to ensure that the system is running the intended software without malware or tampered software. This protection must begin as soon as the system is powered-on. The UEFI (Unified Extensible Firmware Interface) specification defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature.

NFVIS already supports UEFI Secure Boot of Host. Secure boot ensures that the booted NFVIS software is genuine. Starting from NFVIS 3.11.1 release, the UEFI secure boot feature has been extended to VNFs being used in the Cisco NFV solution.

Booting a VNF securely requires the environment to support UEFI secure boot and requires modification to VNFs to support secure boot. NFVIS 3.11.1 and later releases have the infrastructure to support UEFI secure boot for VNFs that are capable of secure boot.

Note

ISRv versions 16.12 and later support secure boot.

The firmware modes in which VNFs can boot are:

- BIOS - for VNFs which are not capable of secure boot
- UEFI secure - for VNFs capable of secure boot

VNFs can indicate secure boot capability using properties in the image_properties.xml file in the tar.gz package for the VNF.

You must set the following properties to enable secure boot of VNFs:

<table>
<thead>
<tr>
<th>Name of the Image Property</th>
<th>Value for non-secure boot VNF</th>
<th>Value for secure boot VNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_mode</td>
<td>bios</td>
<td>efi-secure-boot</td>
</tr>
<tr>
<td>shim_signature</td>
<td>N/A</td>
<td>microsoft</td>
</tr>
</tbody>
</table>

Any combinations not matching the above default to the following:

- boot_mode: bios
- shim_signature: N/A

Note

On the NFVIS portal, the Image Repository page shows if the image is capable of secure boot.
Secure Overlay and Single IP Configuration

Secure Overlay

An overlay is a virtualized network layer on top of the physical network with the support of its infrastructure to provide additional security to the network. IPSec is a framework with protocols and algorithms to provide secured data transmission over unprotected or untrusted networks. IPSec secure tunnel is created between two networks to ensure virtual private network communication.

Secure overlay in NFVIS allows IPSec tunnel establishment between NFVIS supporting the vBranch platform and a VPN server and allows the orchestrator to manage NFVIS over the IPSec tunnel.

Supported Features on Secure Overlay

The following features are supported on NFVIS 3.10.x and later releases:

- IPSec IKEv2
- IPv4
- Authentication:
  - Pre-shared-key authentication
  - Introduced in NFVIS 3.12.3 release - EAP authentication
- IKE cipher:
  - aes128-sha1-mopd1536
  - Introduced in NFVIS 3.12.3 release - aes256-sha512-modp2048
  - Introduced in NFVIS 3.12.3 release - aes256-sha512-modp4096
- ESP cipher:
  - aes128-sha1
  - Introduced in NFVIS 3.12.3 release - aes256-sha512
• Introduced in NFVIS 4.2.1 release - aes256-sha512-modp2048
• Introduced in NFVIS 4.2.1 release - aes256-sha512-modp4096
• Introduced in NFVIS 4.4.1 release - aes128-sha1-modp1536, aes256-sha1-modp2048, and aes256-sha256-modp2048

• Local system IP address:
  • Unique tunnel IP address for each NFVIS system.
  • Introduced in NFVIS 3.11.1 release - Internal management network bridge (int-mgmt-net-br) gateway IP address is allowed to be used as local system IP address. In this case, the local system IP bridge must be set to internal management network (int-mgmt-net).
  • Introduced in NFVIS 4.4.1 release - local system IP address can be learnt through IPSec negotiation when local system IP address is not configured.

• Local system IP subnet:
  • local system IP subnet prefix.
  • Introduced in NFVIS 4.4.1 release - local system IP address and subnet can be learnt through IPSec negotiation. In this case local system IP subnet should not be configured.

• Local bridge for NFVIS reaching out to remote VPN server:
  • wan-br by default
  • wan2-br

• Local bridge and Dual local bridge:
  Introduced in NFVIS 3.12.1 release - Secure overlay is support on NFVIS Dual WAN feature. DHCP client toggles between wan and wan2 to request for an IP address. When IP address and default gateway are obtained from an interface with DHCP configuration, the toggling stops. If dual-local-bridge is configured, to start overlay, NFVIS selects the interface between local-bridge and dual-local-bridge, in the following order:
    • Interface with DHCP configuration.
    • Interface having static IP address.
    • If both interfaces have static IP address, local-bridge interface.

• Local identity:
  • IP address or FQDN
  • Introduced in NFVIS 3.12.3 release - email domain

• Remote identity:
  • IP address or FQDN
  • Introduced in NFVIS 3.12.3 release - Distinguish Name
  • Introduced in NFVIS 3.12.3 release - email domain
- Introduced in NFVIS 4.4.1 release - Active-Standby headend VPN responder scenario is supported. You can configure a list of Remote IDs, each of which is corresponding to one remote VPN responder.

- Introduced in NFVIS 4.5.1 release - Remote ID configuration using EAP authentication is simplified. If the added security using a distinguished name is not required when using EAP authentication, then an FQDN can be configured on NFVIS to simplify the remote ID configuration and reduce authentication complexity.

- Remote Interface IP address:
  - IP address
  - Introduced in NFVIS 4.4.1 release - FQDN
  - For the FQDN, DNS server has to be configured in system settings or learned through DHCP.

- Remote system IP address:
  - IP address
  - Introduced in NFVIS 4.4.1 release - Active-Standby headend VPN responder scenarios is supported. You can configure a list of Remote system IP addresses, each of which is corresponding to a tunnel IP address on a remote VPN responder.
  - Introduced in NFVIS 4.5.1 release - Remote system IP address can be learned from the remote VPN responder. To do so, leave the `remote-system-ip-addr` field out of the secure overlay configuration entirely.

- Remote system IP Subnet:
  - User can specify IP subnet prefix.
  - Introduced in NFVIS 4.1.1 release - User can specify 0.0.0.0/0.
    - Default route is set to Secure Overlay tunnel
    - Following routes bypass the tunnel:
      - If PNP server IP address is discovered before Secure Overlay is configured
      - If DNS server IP address is set before Secure Overlay is configured.
      - Any NFVIS local static route
      - Any route added by user using `system routes route` command

- BGP Neighbor Name:
  - Introduced in NFVIS 4.4.1 release - Active-Standby headend VPN responder scenario is supported. You can configure a list of Remote system IP subnets, each of which corresponds to a tunnel IP address on a remote VPN responder.

- BGP Neighbor Name:
  - Introduced in NFVIS 4.4.1 release, Secure Overlay feature can work together with BGP feature. BGP session can be established over IPSec tunnel. NFVIS can learn private remote subnets from BGP neighbor over the tunnel. These learnt private subnets are added to the routing table for IPSec tunnel. BGP Neighbor Name allows BGP neighbor session to be established with the active secure overlay remote system IP address if the neighbor name is also configured under `router bgp`
configuration. NFVIS will automatically determine which remote system IP address belongs to the active remote IPSec VPN responder.

- Introduced in NFVIS 4.5.1 release - Secure overlay feature works together with BGP feature to announce NFVIS subnet routes over an IPSec tunnel to a BGP neighbor. For more information, see BGP Route Annoucement over MPLS or IPSec, on page 159.

Example for Secure Overlay with Zero Touch Deployment

1. NFVIS has WAN IP address, static IP address or DHCP IP address. NFVIS calls home PnP server.
2. The PnP server pushes NFVIS Day-0 configurations including the secure overlay configuration.
3. NFVIS establishes IPSec connection between NFVIS and the headend management hub which has IPSec VPN configurations. On NFVIS side, the tunnel end point has NFVIS local system IP address.
4. After the IPSec tunnel is up, the headend can connect to NFVIS through the system IP address and manage NFVIS over the IPSec tunnel.

To configure secure overlay:

```
configure terminal
secure-overlay myconn
  local-system-ip-addr 27.27.27.1
  local-system-ip-bridge int-mgmt-net
  local-id mail@gmail.com
```
remote-interface-ip-addr nfvisoverlay.cisco.com
remote-system-ip-addr [ 166.34.121.111 166.34.121.112 ]
psk local-psk Cisco1234Admin
psk remote-psk Cisco1234Admin
!

configure terminal
secure-overlay myconn
local-system-ip-addr 28.28.28.1
local-system-ip-subnet 28.28.28.0/24
local-system-ip-bridge int-mgmt-net
local-id AxxxY@cisco.com
remote-interface-ip-addr C***d.cisco.com
remote-system-ip-addr [ 166.35.121.112 166.34.121.112 ]
remote-system-ip-subnet [ 166.35.121.112/32 166.34.121.112/32 ]
remote-id [ CN=vbranch,unstructuredAddress=10.30.1.153,unstructuredName=csr-vpn-srvr-03.cisco.com ]
ike-cipher [ aes256-sha512-modp2048 ]
esp-cipher [ aes256-sha512-modp2048 ]
eap username admin
eap password Cisco123#
eap cacert intdatastore:uploads/ca.pem
!

To get the secure overlay state:

nfvis# show secure-overlay

ACTIVE ACTIVE
ACTIVE SELECTED LOCAL REMOTE ACTIVE REMOTE
LOCAL STATE LOCAL SYSTEM IP INTERFACE SYSTEM IP ACTIVE
REMOTE
NAME STATE BRIDGE DETAILS BRIDGE ADDR IP ADDR ADDR SYSTEM IP SUBNET ACTIVE REMOTE ID
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
myconn up wan-br - wan-br 28.28.28.1 10.30.1.153 166.34.121.112 166.34.121.112/32 "CN=vbranch,unstructuredAddress=10.30.1.153,unstructuredName=csr-vpn-srvr-03.cisco.com"

nfvis#

Examples for Configuring Secure Overlay

Secure Overlay over WAN with pre-shared-key and FQDN remote-id

<config xmlns="http://tail-f.com/ns/config/1.0">  
<secure-overlays xmlns="http://www.cisco.com/nfvis/secure-overlay">  
<secure-overlay>
  <name>myconn</name>
</secure-overlays>
</config>

Note

Secure overlay configuration on NFVIS must match with VPN configuration on the VPN server. The secure overlay tunnel will not be established successfully if the configurations do not match.
Secure Overlay and Single IP Configuration

VPN configuration on VPN server:

```plaintext
aaa new-model
!
aaa group server radius radius-group
server-private 10.30.1.234 auth-port 1812 acct-port 1813 key Cisco123#
ip vrf forwarding public-vrf
ip radius source-interface GigabitEthernet2
!
aaa authentication login default group radius-group local
aaa authentication login ucpe-authen group radius-group
aaa authorization console
aaa authorization exec default local
aaa authorization network default local
!
aaa session-id common
!
crypto pki trustpoint TP-self-signed-2641177237
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-2641177237
  revocation-check none
  rsakeypair TP-self-signed-2641177237
```
crypto pki trustpoint SLA-TrustPoint
  revocation-check crl
crypto pki trustpoint router
  ip-address 10.30.1.114
  subject-name CN=vbranch
  vrf public-vrf
  revocation-check crl
  rsa key-pair router
  auto-enroll regenerate
  hash sha512

crypto ikev2 authorization policy uCPE-author-pol
  pool uCPE-pool1
  dns 166.37.162.103 166.37.218.90
  pfs
  route set interface Loopback1001
  no crypto ikev2 authorization policy default

crypto ikev2 proposal uCPE-proposal
  encryption aes-cbc-256
  integrity sha512
  group 16 14
  no crypto ikev2 policy default

crypto ikev2 policy uCPE-policy
  match fvrf public-vrf
  proposal uCPE-proposal

crypto ikev2 profile uCPE-profile
  description uCPE profile
  match fvrf public-vrf
  match identity remote email domain vpntest.com
  identity local dn
  authentication local rsa-sig
  authentication remote eap query-identity
  pki trustpoint router
  dpd 60 2 on-demand
  aaa authentication eap ucpe-authen
  aaa authorization group eap list default uCPE-author-pol
  virtual-template 1 mode auto

crypto ipsec transform-set tset_aes_256_sha512 esp-aes 256 esp-sha512-hmac
  mode tunnel
  crypto ipsec df-bit clear
  no crypto ipsec profile default

crypto ipsec profile uCPE-ips-prof
  set security-association lifetime seconds 28800
  set security-association idle-time 1800
  set transform-set tset_aes_256_sha512
  set pfs group14
  set ikev2-profile uCPE-profile

crypto call admission limit ike in-negotiation-sa 30

ip radius source-interface GigabitEthernet2
  route set interface Loopback1001
interface Loopback1
ip vrf forwarding private-vrf
ip address 90.90.90.1 255.255.255.255

interface Loopback2
ip vrf forwarding private-vrf
ip address 91.91.91.1 255.255.255.0

interface Loopback3
ip vrf forwarding private-vrf
ip address 90.90.90.201 255.255.255.128

interface Loopback1001
description MNSO Interface and Tunnel Loopback
ip vrf forwarding private-vrf
ip address 166.35.121.112 255.255.255.255

interface GigabitEthernet1
ip vrf forwarding private-vrf
ip address 92.92.92.1 255.255.255.0
negotiation auto
no mop enabled
no mop sysid

interface GigabitEthernet2
ip vrf forwarding public-vrf
ip address 10.30.1.114 255.255.255.0
negotiation auto
no mop enabled
no mop sysid

interface GigabitEthernet3
ip vrf forwarding private-vrf
ip address 91.91.91.1 255.255.255.0
shutdown
negotiation auto
no mop enabled
no mop sysid
interface Virtual-Template1 type tunnel
ip vrf forwarding private-vrf
ip unnumbered Loopback1001
ip mtu 1400
ip tcp adjust-mss 1380
tunnel vrf public-vrf
tunnel protection ipsec profile uCPE-ips-prof
ip radius source-interface GigabitEthernet2

Internal management network bridge IP address as local system IP address

Note
NFVIS internal management network has gateway IP address 12.12.12.1.

EAP authentication
The following is an example of the VPN configuration on VPN server:

aaa group server radius radius-group
server-private 172.19.160.190 auth-port 1812 acct-port 1813 key Cisco123#
ip radius source-interface GigabitEthernet

aaa authentication login default group radius-group
aaa authentication login ucpe-authen group radius-group

ip domain name headendvpn
crypto pki server ca-server
database level names
no database archive
hash sha512
lifetime certificate 3650
lifetime ca-certificate 7305 23 59
auto-rollover 365
eku server-auth client-auth
database url flash:ca
crypto pki trustpoint ca-server
revocation-check crl
rsakeypair ca-server
crypto pki trustpoint router
enrollment url http://172.19.160.75:80
ip-address 172.19.160.75
subject-name CN=vbranch
revocation-check crl
rsakeypair router
auto-enroll regenerate
hash sha512
crypto ikev2 authorization policy uCPE-athor-pol
pfs
route set interface

no crypto ikev2 authorization policy default
crypto ikev2 proposal uCPE-proposal
encryption aes-cbc-256
integrity_sha512
group 16 14

no crypto ikev2 policy default
crypto ikev2 policy uCPE-policy
match address local 172.19.160.75
Single Public IP Address and Secure Overlay

Single Public IP Address

In a virtual branch deployment, two public IP addresses are needed for each branch site, one for the NFVIS hypervisor and the other one for the WAN router. In Single Public IP Address feature on NFVIS, one public IP address assigned to a branch site, is seamlessly shared between the NFVIS hypervisor and the guest VM deployed on NFVIS. This feature ensures that the branch site is reachable even if the guest router is in failure state.

NFVIS reclaims the WAN IP address if the guest router has:

- Deployment failure.
NFVIS releases the WAN IP address if the guest router has:
  • Deployed.
  • Started.

To create a single-ip-mode:

```
configure terminal
single-ip-mode vm-name <vm_deploy_name>.<vm_group_name>
commit
```

Use the `show vm_lifecycle deployments all | include Name` command to find out the Deployment Name and VM Group Name.

To get the state of single-ip-mode use the `show single-ip-mode` command.

### Single Public IP Address with Secure Overlay

Secure overlay tunnel is established automatically when IP address is moves back and forth between NFVIS and the guest VM. The orchestrator can always reach NFVIS through the system IP address which does not change during the transitioning of the single public IP address.

*Figure 2: Example of Setting IPSec Tunnel in Single IP mode*

After secure overlay over WAN is established, the orchestrator sends requests to configure single IP mode and deploy the guest router that takes the public IP address.

1. NFVIS deploys the VM with specified bootstrap and Day-0 configuration. NFVIS takes down the current IPSec tunnel and releases the public IP address.

2. The VM takes the public IP address when it is in active state. NFVIS sets up the IPSec tunnel again with the remote management hub.

3. After the IPSec tunnel is up, the orchestrator can connect to NFVIS through its system IP address and manage NFVIS over the IPSec tunnel.
In single IP mode, NFVIS monitors the guest VM taking the public IP address. NFVIS takes WAN IP address back when the guest VM is:

- In error state.
- Stopped through vmAction.
- Undeployed.

**Figure 3: Example of NFVIS Handling Failure**

1. NFVIS takes WAN IP address.
2. NFVIS sets up IPSec tunnel to the management hub.
3. When IPSec tunnel is up, the VPN server can connect to NFVIS through its system IP address and manage NFVIS over the IPSec tunnel.

Each NFVIS has its int-mgmt-net and subnet associated to the IP address pool. The following is an example to setup the IP address pool before deploying VM:

```bash
no vm_lifecycle networks network int-mgmt-net commit
```

**Guest VM taking Public IP Address**

Guest VM must be deployed as a monitored VM which has two interfaces:

- Interface facing public with the public IP address.
- Interface on int-mgmt-net-br for traffic flow with NFVIS.

The guest VM has routing function to route traffic between the two interfaces and Network address translation (NAT) enabled. NFVIS reaches remote through int-mgmt-net-br to the guest VM.
The int-mgmt-net-br address pool and gateway IP address must be unique on each NFVIS. If secure overlay is configured, single IP mode is setup when VM is active and int-mgmt-net-br is used as a local-bridge.

**Single IP address and DHCP**

NFVIS single-ip-mode supports the public IP address acquired through DHCP by leveraging on the lease timer configuration on DHCP server. The guest VM with Day-0 configuration gets the IP address through DHCP when NFVIS client sends release message to DHCP server.

To handle failure, NFVIS:

- stops the VM, to ensure the VM dhclient does not send DHCP renew to DHCP server
- switches back to WAN and its dhclient sends DHCP renew message to DHCP server
- gets the same IP address from DHCP server when VM’s lease time expires.

**ISRv bootstrap and Day-0 Configuration**

In single-ip-mode, NFVIS reaches to the guest router and takes its IP address. Traffic must be allowed between ISRv gigabit ethernet interface 1 connected to NFVIS int-mgmt-net-br and gigabit ethernet interface 2 connected to public side having the public IP address.

The following is an example based on ISRv 16.09.01a that shows how to configure the guest router to route the traffic:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.csr1000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.csr1000v.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.csr1000v.login-username.1" oe:value="${SSH_USERNAME}"/>
    <Property oe:key="com.cisco.csr1000v.login-password.1" oe:value="${SSH_PASSWORD}"/>
    <Property oe:key="com.cisco.csr1000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
    <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-addr.1" oe:value="${NICID_0_IP_ADDRESS}/24"/>
    <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.csr1000v.license.1" oe:value="${TECH_PACKAGE}"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0007" oe:value="interface GigabitEthernet1"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0008" oe:value="ip address ${NICID_0_IP_ADDRESS} ${NICID_0_NETMASK}"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0009" oe:value="no shut"/>
    <Property oe:key="com.cisco.csr1000v.ios-config-0010" oe:value="exit"/>
  </PropertySection>
</Environment>
```

The following is an example of ISRv day-0 configuration with a static public IP address:

```bash
interface GigabitEthernet1
ip nat inside
```
negotiation auto
!
interface GigabitEthernet2
ip address 172.25.221.17 255.255.255.0
ip nat outside
negotiation auto
!
ip nat inside source list NAT interface GigabitEthernet2 overload
ip route 0.0.0.0.0.0.0.0 172.25.221.1
ip route 10.19.18.0 255.255.255.0 12.12.12.1
!
ip access-list standard NAT
permit 12.12.12.0.0.0.0.25

The following is an example of ISRv day-0 configuration with DHCP IP address:

interface GigabitEthernet1
ip nat inside
negotiation auto
!
interface GigabitEthernet2
ip address dhcp
ip nat outside
negotiation auto
!
ip nat inside source list NAT interface GigabitEthernet2 overload
ip route 10.19.18.0 255.255.255.0 12.12.12.1
!
ip access-list standard NAT
permit 12.12.12.0.0.0.0.25

To verify single IP address mode status use the show single-ip-mode and show secure-overlay command:

nfvis#
single-ip-mode state start
single-ip-mode state-details "waiting for VM ROUTER.ROUTER deployment"
nfvis#
single-ip-mode state start
single-ip-mode state-details "VM deployed"
nfvis#
nfvis# show secure-overlay

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>mgmthub</td>
<td>up</td>
<td>int-mgmt-net-br</td>
<td>VM alive</td>
</tr>
</tbody>
</table>

**Single IP and Secure Overlay APIs**

<table>
<thead>
<tr>
<th>Secure Overlay APIs</th>
<th>Secure Overlay Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/single-ip-mode</td>
<td>single-ip-mode</td>
</tr>
<tr>
<td>/api/operational/single-ip-mode</td>
<td></td>
</tr>
</tbody>
</table>
Single IP Address Without Secure Overlay

Note
This feature is only supported for WAN bridge in NFVIS 3.10.1 release.

To reach NFVIS when secure overlay is not configured, you must first configure the guest device and manage IP addressing. The rest of the functionality, switching IP address between NFVIS and the guest device is the same as IP address with secure overlay.

Typically you need two IP addresses in each site, one for NFVIS and one for the VM. You can enable the single IP feature to reduce one public IP address. The single public IP address is used by NFVIS after deploying the VM with the single IP feature. After the VM comes up, NFVIS releases the public IP address for the VM to use. NFVIS and the VM have an internal network to communicate with each other. The traffic between NFVIS and an external network will need to go through the new VM and NAT by the new VM.

For single IP without secure overlay feature to work:

- From the Deploy page on NFVIS portal select single IP or configure the single IP mode by using the `single-ip-mode router.router` command.
- Provide a bootstrap file for the VM.
- Enable Monitor for the VM and the internal network int-mgmt-net between NFVIS and VM is created automatically.

The following example is a sample bootstrapping configuration:

172.25.221.7/24 is the single public IP address that is originally used by NFVIS and later by the VM. 172.25.221.1 is the gateway to the external network and 10.20.0.x is the internal network between NFVIS and the VM. IP address in 10.20.0.x network is used to NAT by the VM:

```
interface GigabitEthernet1
ip nat inside
negotiation auto
!
interface GigabitEthernet2
ip address 172.25.221.17 255.255.255.0
ip nat outside
negotiation auto
!
ip nat inside source list NAT interface GigabitEthernet2 overload
ip route 0.0.0.0 0.0.0.0 172.25.221.1
!
ip access-list standard NAT
permit 10.20.0.0 0.0.0.25
```

When the VM is down, NFVIS takes back the single IP address and the external server can communicate with NFVIS directly.
CHAPTER 6

BGP Support on NFVIS

Table 22: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Support on Remote Subnets Over IPSec.</td>
<td>NFVIS 4.4.1</td>
<td>This feature allows the NFVIS system to learn routes that are announced from the remote BGP neighbor and apply the learnt routes to the NFVIS system.</td>
</tr>
<tr>
<td>BGP Support Announcing Local Subnets (Route Distribution)</td>
<td>NFVIS 3.10.1</td>
<td>This feature allows you to announce or withdraw NFVIS local routes to the remote BGP neighbor using route distribution.</td>
</tr>
</tbody>
</table>

Border Gateway Protocol (BGP) is the dynamic routing protocol to exchange route information between BGP autonomous systems.

The NFVIS BGP feature works together with remote BGP router. This feature allows NFVIS system to learn routes announced from the remote BGP neighbor and apply the learnt routes to the NFVIS system. This feature also allows you to announce or withdraw NFVIS local routes from the remote BGP neighbor.

Starting from NFVIS 4.4.1 release, NFVIS BGP feature works with the secure overlay feature to learn routes from the BGP neighbor over a secure overlay tunnel. These learnt routes or subnets are added into the NFVIS routing table for the secure tunnel, which makes the routes accessible over the tunnel.

- Configure BGP on NFVIS, on page 155
- Route Distribution, on page 158
- BGP Route Announcement over MPLS or IPSec, on page 159

Configure BGP on NFVIS

A BGP neighbor can be configured using a neighbor IP address or a name string.

If a BGP neighbor is specified using a name string, it must be used in conjunction with the secure overlay bgp-neighbor-name field. A BGP session is established over the secure overlay tunnel. If the neighbor name matches the bgp-neighbor-name field configured in secure-overlay configuration, then NFVIS will determine the active remote system IP address used for IPSec connection and replace the neighbor name with that IP.
This will establish a BGP neighbor session with that IP address. For more information on how to configure secure overlay with BGP name, see Secure Overlay and Single IP Configuration, on page 139.

If a BGP neighbor is specified using an IP address which is headend VPN responder’s tunnel IP address, which is the same as the IP address of a a headend VPN responder tunnel, a BGP session is established over the secure overlay tunnel.

This example shows how to create or update BGP configuration for a neighbor with a specified name string:

```bash
config terminal
router bgp 200
   neighbor csr-bgp remote-as 65000
   commit
```

This example shows how to create or update BGP configuration with a specified neighbor IP address:

```bash
config terminal
router bgp 200
   neighbor 166.34.121.112 remote-as 65000
   exit
   neighbor 166.35.121.112 remote-as 65000
   commit
```

This example shows how to delete BGP configurations:

```bash
no router bgp 200
commit
```

The following table provides the syntax description for each parameter in the commands mentioned in the examples above:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>as</td>
<td>Uint32</td>
<td>Local BGP AS number</td>
<td>Yes</td>
</tr>
<tr>
<td>router-id</td>
<td>IPv4</td>
<td>H.H.H.H: IPv4 address for local system</td>
<td>No</td>
</tr>
<tr>
<td>neighbor</td>
<td>list</td>
<td>Neighbor list</td>
<td>Yes</td>
</tr>
<tr>
<td>remote-ip</td>
<td>String</td>
<td>IPv4 address or Secure Overlay BGP neighbor name for BGP neighbor system</td>
<td>Yes</td>
</tr>
<tr>
<td>remote-as</td>
<td>Uint32</td>
<td>Remote BGP AS number</td>
<td>Yes</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>Description of neighbor</td>
<td>No</td>
</tr>
</tbody>
</table>

The following example displays the BGP session details:

```bash
nfvis# support show bgp
BIRD 1.6.8 ready.
name proto table state since info
bgp1 BGP bgptable UP 23:53:18 Established
Preference: 100
Input filter: ACCEPT
```
Output filter: Accept
Import limit: 15
Action: restart
Routes: 1 import, 0 exported, 1 preferred
Route change stats: received rejected filtered ignored accepted
Import updates: 1 0 0 0 1
Import withdraws: 0 0 --- 0 0
Export updates: 1 1 0 --- 0
Export withdraws: 0 --- --- --- 0
BGP state: Established
Neighbour address: 166.34.121.112
Neighbour AS: 65000
Neighbour ID: 166.34.121.112
Neighbour caps: refresh enhanced-refresh AS4
Session: external multihop AS4
Source Address: 112.112.112.1
Route limit: 1/15
Hold timer: 204/240
Keepalive timer: 65/80

The following example displays the BGP routes learnt through BGP:

nfvis# support show bgp route
BIRD 1.6.8 ready.
91.91.91.0/24 dev ipsec0 [bgp1 23:53:18 from 166.34.121.112] (100) [AS65000?]

Note
NFVIS can learn up to 15 prefixes.

BGP Neighbor Configuration Example

router bgp 65000
bgp router-id 166.34.121.112
bgp always-compare-med
bgp log-neighbor-changes
bgp deterministic-med
bgp listen range 112.112.0.0/16 peer-group uCPEs
bgp listen range 90.90.90.0/24 peer-group uCPEs
bgp listen range 10.20.0.0/24 peer-group uCPEs
bgp listen limit 255
no bgp default ipv4-unicast
! address-family ipv4 vrf private-vrf
  redistribute connected
  redistribute static
  neighbor uCPEs peer-group
  neighbor uCPEs remote-as 200
  neighbor uCPEs ebgp-multihop 10
  neighbor uCPEs timers 610 1835
  neighbor uCPEs prefix-list allow-list out
  exit-address-family

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
Route Distribution

The Route Distribution feature works together with a remote BGP router. It allows you to announce or withdraw specified routes to the remote BGP router.

You can use this feature to announce the route of int-mgmt-net subnet to a remote BGP router. A remote user, can access the VMs attached to int-mgmt-net through the VMs’ IP address on int-mgmt-net-br through a BGP router, when the routes are successfully inserted on the remote BGP router.

To configure or update route distribution:

```
configure terminal
route-distribute 172.25.221.17 local-bridge wan-br local-as 45.45 remote-as 65000 network-subnet 12.12.12.0/24
commit
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor-address</td>
<td>IPv4</td>
<td>BGP neighbor IPv4 address. It is the key of the route distribution list.</td>
<td>Yes</td>
</tr>
<tr>
<td>local-address</td>
<td>IPv4</td>
<td>Local IPv4 address. This address must be configured as neighbor IP address on the remote BGP router. If not configured, local-address is set to local-bridge’s IP address.</td>
<td>No</td>
</tr>
<tr>
<td>local-as</td>
<td></td>
<td>Local autonomous system number. It can be in following two formats:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;decimal number, 1.0 .. 65535.65535&gt;&lt;unsignedInt, 1 .. 4294967295&gt;</td>
<td></td>
</tr>
<tr>
<td>local-bridge</td>
<td></td>
<td>Local bridge name for advertising routes (default wan-br).</td>
<td>No</td>
</tr>
<tr>
<td>remote-as</td>
<td></td>
<td>Remote autonomous system number. It can be in following two formats:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;decimal number, 1.0 .. 65535.65535&gt;&lt;unsignedInt, 1 .. 4294967295&gt;</td>
<td></td>
</tr>
<tr>
<td>router-id</td>
<td>IPv4</td>
<td>Local router ID</td>
<td>No</td>
</tr>
</tbody>
</table>
### Property | Type | Description | Mandatory
--- | --- | --- | ---
network-subnet |  | List of network subnet to be announced. | Yes
subnet | IPv4 prefix | Network subnet to be announced H.H.H.H/N | Yes
next-hop | IPv4 | IPv4 address of next hop. Default local-address or IP address of local-bridge. | No

Use the **no route-distribute** command to delete route distribution. To verify the route-distribution status use the **show route-distribution** command.

**Remote BGP Router Configuration Example**

The NFVIS route distribution feature works together with the remote BGP router. The configuration on NFVIS and on remote BGP router must match.

This example shows the configuration on a remote BGP router.

```
router bgp 65000
  bgp log-neighbor-changes
  neighbor 172.25.221.106 remote-as 45.45
  neighbor 172.25.221.106 update-source GigabitEthernet2
```

---

**BGP Route Announcement over MPLS or IPSec**

**Table 24: Feature History**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Route Announcement over MPLS or IPSec</td>
<td>NFVIS 4.5.1</td>
<td>This feature allows you to configure NFVIS to announce routes through BGP over MPLS. NFVIS allows the routes learnt through BGP available over IPSec tunnel over MPLS connection.</td>
</tr>
</tbody>
</table>

With this feature enhancement, the existing routes learnt through BGP over IPSec tunnel are now allowed over MPLS connection. Additionally, NFVIS can now announce routes through BGP, using the same `router bgp` command that is used for learning routes over BGP. For more information on this command, see the Cisco IOS XE `router bgp` command.

You can pair the secure overlay configurations to announce NFVIS routes over BGP through IPSec tunnel. The existing `router bgp` configurations can be updated to add the route announcement feature. Make sure that you remove the existing route distribution configurations before you configure the `router bgp` command.

The following example shows how to configure the announcement of 10.20.0.0/24 subnet over BGP.
BGP Route Announcement over MPLS or IPSec

router bgp 65000
neighbor 172.25.221.17 remote-as 65001
address-family ipv4 unicast
    network 10.20.0.0 mask 255.255.255.0
neighbor 172.25.221.17 activate

The following example shows how to remove the announcement of 10.20.0.0/24 subnet from BGP.

router bgp 65000
    address-family ipv4 unicast
    no network 10.20.0.0 mask 255.255.255.0

The following example shows how to remove a neighbor from the IPv4 address family, and disable route announcements for the same neighbor.

router bgp 65000
    address-family ipv4 unicast
    no neighbor 172.25.221.17 activate

To view the local BGP status for BGP over MPLS use the `show bgp ipv4 unicast` command.

```
nfvis# show bgp ipv4 unicast
```

<table>
<thead>
<tr>
<th>Family</th>
<th>Transmission</th>
<th>Router ID</th>
<th>Local AS Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4</td>
<td>unicast</td>
<td>10.20.0.1</td>
<td>65000</td>
</tr>
</tbody>
</table>

To view the BGP neighbor status for BGP over MPLS use the `show bgp ipv4 unicast summary` command.

```
nfvis# show bgp ipv4 unicast summary
```

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>IP Version</th>
<th>AS Number</th>
<th>Up/Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.25.221.17</td>
<td>4</td>
<td>65001</td>
<td>up</td>
</tr>
</tbody>
</table>

To view the BGP learned or announced routes for BGP over MPLS use the `show bgp ipv4 unicast route` command.

```
nfvis# show bgp ipv4 unicast route
```

<table>
<thead>
<tr>
<th>Network</th>
<th>Next-Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.30.30.0/24</td>
<td>172.25.221.17</td>
<td>0</td>
<td>100</td>
<td>65001 ?</td>
</tr>
<tr>
<td>10.40.40.0/24</td>
<td>172.25.221.17</td>
<td>0</td>
<td>100</td>
<td>65001 ?</td>
</tr>
<tr>
<td>10.20.0.0/24</td>
<td>0.0.0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To view the local BGP status for BGP over IPSec tunnel use the `show bgp vpnv4 unicast` command.

```
nfvis# show bgp vpnv4 unicast
```

<table>
<thead>
<tr>
<th>Family</th>
<th>Transmission</th>
<th>Router ID</th>
<th>Local AS Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpnv4</td>
<td>unicast</td>
<td>10.20.0.1</td>
<td>200</td>
</tr>
</tbody>
</table>

To show BGP neighbor status for BGP over IPSec tunnel:

```
nfvis# show bgp vpnv4 unicast summary
```

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>IP Version</th>
<th>AS Number</th>
<th>Up/Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.90.90.1</td>
<td>4</td>
<td>65000</td>
<td>up</td>
</tr>
</tbody>
</table>

To show BGP learned/announced routes for BGP over IPSec tunnel:
BGP Support on NFVIS

BGP Route Announcement over MPLS or IPSec

nfvis# show bgp vpnv4 unicast route

<table>
<thead>
<tr>
<th>Network</th>
<th>Next-Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.91.91.0/24</td>
<td>10.90.90.1</td>
<td>0</td>
<td>100</td>
<td>65000</td>
</tr>
<tr>
<td>10.92.92.0/24</td>
<td>10.90.90.1</td>
<td>0</td>
<td>100</td>
<td>65000</td>
</tr>
<tr>
<td>10.20.0.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>100</td>
<td>65000</td>
</tr>
</tbody>
</table>

**Note**
When you configure BGP route announcement over IPSec tunnel, ensure that you configure secure overlay to use the virtual IP address for the local tunnel IP address (*no local-system-ip-addr configured*).

**Note**
When you configure BGP route announcement, the only configurable address-family or transmission combination is **ipv4 unicast** for both IPSec and MPLS. To view the BGP status, the configurable address-family or transmission for IPSec is **vpnv4 unicast** and for MPLS is **ipv4 unicast**.
BGP Route Announcement over MPLS or IPSec
Security Considerations

This chapter describes the security features and considerations in NFVIS. It gives a high-level overview of security related components in NFVIS to plan a security strategy for deployments specific to you. It also has recommendations on security best practices for enforcing the core elements of network security.

The NFVIS software has security embedded right from installation through all software layers. The subsequent chapters focus on these out-of-the-box security aspects such as credential management, integrity and tamper protection, session management, secure device access and more.

- Installation, on page 164
- Secure Unique Device Identification, on page 165
- Device Access, on page 166
**Installation**

To ensure that the NFVIS software has not been tampered with, the software image is verified before installation using the following mechanisms:

**Image Tamper Protection**

NFVIS supports RPM signing and signature verification for all RPM packages in the 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 NFVIS. The private key used for signing the RPM packages is created and securely maintained by Cisco.

**RPM Signature Verification**

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

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Enterprise NFVIS 3.7.1 and later installations</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**

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 the 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:

```
% /usr/bin/sha512sum <ImageFile>
c2122783efc18b039246ae1bcd4eec4e5e027526967b5b809da5632d462dafa9b20ec318c74543b2c2706b26217bce94f6ce4832e7d4b3b3ec0e012e3925a5f7
```

### ENCS Secure Boot

Secure boot is part of the Unified Extensible Firmware Interface (UEFI) standard which ensures that a device boots only using a software that is trusted by the Original Equipment Manufacturer (OEM). When NFVIS starts, the firmware checks the signature of the boot software and the operating system. If the signatures are valid, the device boots, and the firmware gives the control to the operating system.

Secure boot is available on the ENCS but is disabled by default. Cisco recommends you to enable secure boot. For more information, see [UEFI Secure Boot on ENCS 5400](#).

### Secure Unique Device Identification

NFVIS uses a mechanism known as Secure Unique Device Identification (SUDI), which provides it with an immutable identity. This identity is used to verify that the device is a genuine Cisco product, and to ensure that the device is well-known to the customer’s inventory system.

The SUDI is an X.509v3 certificate and an associated key-pair which are protected in hardware. The SUDI certificate contains the product identifier and serial number and is rooted in Cisco Public Key Infrastructure. The key pair and the SUDI certificate are inserted into the hardware module during manufacturing, and the private key can never be exported.

The SUDI-based identity can be used to perform authenticated and automated configuration using Zero Touch Provisioning (ZTP). This enables secure, remote on-boarding of devices, and ensures that the orchestration server is talking to a genuine NFVIS device. A backend system can issue a challenge to the NFVIS device to validate its identity and the device will respond to the challenge using its SUDI based identity. This allows the backend system to not only verify against its inventory that the right device is in the right location but also provide encrypted configuration that can only be opened by the specific device, thereby ensuring confidentiality in transit.

The following workflow diagrams illustrate how NFVIS uses SUDI:
NFVIS provides different access mechanisms including console as well as remote access based on protocols such as HTTPS and SSH. Each access mechanism should be carefully reviewed and configured. Ensure that only the required access mechanisms are enabled and that they are properly secured. The key steps to securing both interactive and management access to NFVIS are to restrict the device accessibility, restrict the capabilities of the permitted users to what is required, and restrict the permitted methods of access. NFVIS ensures that the access is only granted to authenticated users and they can perform just the authorized actions. Device access is logged for auditing and NFVIS ensures the confidentiality of locally stored sensitive data.
It is critical to establish the appropriate controls in order to prevent unauthorized access to NFVIS. The following sections describe the best practices and configurations to achieve this:

**Enforced Password Change at First Login**

Default credentials are a frequent source of product security incidents. Customers often forget to change the default login credentials leaving their systems open to attack. To prevent this, the NFVIS user is forced to change the password after the first login using the default credentials (username: admin and password Admin123#).

For more information, see Accessing NFVIS, on page 12.

**Restricting Login Vulnerabilities**

You can prevent the vulnerability to dictionary and Denial of Service (DoS) attacks by using the following features.

**Enforcement of Strong password**

An authentication mechanism is only as strong as its credentials. For this reason, it is important to ensure users have strong passwords. NFVIS checks that a strong password is configured as per the following rules:

Password must contain:
- At least one uppercase character
- At least one lowercase character
- At least one number
- At least one of these special characters: hash (#), underscore (_), hyphen (-), asterisk (*), or question mark (?)
- Seven characters or more
- The password length should be between 7 and 128 characters.

**Configuring Minimum Length for Passwords**

Lack of password complexity, particularly password length, significantly reduces the search space when attackers try to guess user passwords, making brute-force attacks much easier.

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

**CLI:**

```
nfvis(config)# rbac authentication min-pwd-length 9
```

**API:**

```
/api/config/rbac/authentication/min-pwd-length
```
Configuring Password Lifetime

The password lifetime determines how long a password can be used before the user is required to change it. The admin user can configure minimum and maximum lifetime values for passwords for 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.

CLI:

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

API:

/api/config/rbac/authentication/password-lifetime/

Limit previous password reuse

Without preventing the use of previous passphrases, password expiry is largely useless since users can simply change the passphrase and then change it back to the original.

NFVIS checks that the new password is not the same as one of the 5 previously used passwords. One exception to this rule is that the admin user can change the password to the default password even if it was one of the 5 previously used passwords.

Restrict Frequency of login attempts

If a remote peer is allowed to login an unlimited number of times, it may eventually be able to guess the login credentials by brute force. Since passphrases are often easy to guess, this is a common attack. By limiting the rate at which the peer can attempt logins, we prevent this attack. We also avoid spending the system resources on unnecessarily authenticating these brute-force login attempts which could create a Denial of Service attack.

NFVIS enforces a 5 minute user lockdown after 10 failed login attempts.

Disable inactive user accounts

Monitoring user activity and disabling unused or stale user accounts helps to secure the system from insider attacks. The unused accounts should eventually be removed.

The admin user can enforce a rule to mark unused user accounts as inactive and configure the number of days after which an unused user account is marked as inactive. Once marked as inactive, that user cannot login to the system. To allow the user to login to the system, the admin user can activate the user account.
The inactivity period and the rule to check the inactivity period are not applied to the admin user.

The following CLI and API can be used to configure the enforcement of account inactivity.

**CLI:**

```
configure terminal
rbac authentication account-inactivity enforce true inactivity-days 30
commit
```

**API:**

```
/api/config/rbac/authentication/account-inactivity/
```

The default value for inactivity-days is 35.

### Activating an Inactive User Account

The admin user can activate the account of an inactive user using the following CLI and API:

**CLI:**

```
configure terminal
rbac authentication users user guest_user activate
commit
```

**API:**

```
/api/operations/rbac/authentication/users/user/username/activate
```

### Integration with external AAA servers

Users login to NFVIS through ssh or the Web UI. In either case, users need to be authenticated. That is, a user needs to present password credentials in order to gain access.

Once a user is authenticated, all operations performed by that user need to be authorized. That is, certain users may be allowed to perform certain tasks, whereas others are not. This is called authorization.

It is recommended that a centralized AAA server be deployed to enforce per-user, AAA-based login authentication for NFVIS access. NFVIS supports RADIUS and TACACS protocols to mediate network access. On the AAA server, only minimum access privileges should be granted to authenticated users according to their specific access requirements. This reduces the exposure to both malicious and unintentional security incidents.

For more information on external authentication, see [Configuring RADIUS](#), on page 28 and [Configuring a TACACS+ Server](#), on page 30.
**Authentication Cache for External Authentication Server**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Cache for External Authentication Server</td>
<td>NFVIS 4.5.1</td>
<td>This feature supports TACACS authentication through OTP on NFVIS portal.</td>
</tr>
</tbody>
</table>

The NFVIS portal uses the same One-Time Password (OTP) for all API calls after the initial authentication. The API calls fail as soon as the OTP expires. This feature supports TACACS OTP authentication with the NFVIS portal.

After you have successfully authenticated through the TACACS server using an OTP, NFVIS creates a hash entry using the username and the OTP and stores this hash value locally. This locally stored hash value has an expiration time stamp associated with it. The time stamp has the same value as the SSH session idle timeout value which is 15 minutes. All the subsequent authentication requests with the same username are authenticated against this local hash value first. If the authentication fails with the local hash, NFVIS authenticates this request with TACACS server and creates a new hash entry when the authentication is successful. If a hash entry already exists, its time stamp is reset to 15 minutes.

If you are removed from the TACACS server after successfully logging into the portal, you can continue to use the portal until the hash entry in NFVIS expires.

When you explicitly log out from the NFVIS portal or are logged out due to idle time, the portal calls a new API to notify NFVIS backend to flush the hash entry. The authentication cache and all of its entries are cleared out after NFVIS reboot, factory reset, or upgrade.

**Role Based Access Control**

Limiting network access is important to organizations that have many employees, employ contractors or permit access to third parties, such as customers and vendors. In such a scenario, it is difficult to monitor network access effectively. Instead, it is better to control what is accessible, in order to secure the sensitive data and critical applications.

Role-based access control (RBAC) is a method of restricting network access based on the roles of individual users within an enterprise. RBAC lets users access just the information they need, and prevents them from accessing information that doesn't pertain to them.

An employee's role in the enterprise should be used to determine the permissions granted, in order to ensure that employees with lower privileges can't access sensitive information or perform critical tasks.

The following user roles and privileges are defined in NFVIS

<table>
<thead>
<tr>
<th>User Role</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Can configure all available features and perform all tasks including changing of user roles. The administrator cannot delete basic infrastructure that is fundamental to NFVIS. The Admin user’s role cannot be changed; it is always &quot;administrators&quot;.</td>
</tr>
<tr>
<td>Operators</td>
<td>Can Start and stop a VM, and view all information.</td>
</tr>
</tbody>
</table>
Privilege

<table>
<thead>
<tr>
<th>User Role</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auditors</td>
<td>They are the least privileged users. They have Read-only permission and therefore, can’t modify any configuration.</td>
</tr>
</tbody>
</table>

**Benefits of RBAC**

There are a number of benefits to using RBAC to restrict unnecessary network access based on people's roles within an organization, including:

- Improving operational efficiency.

  Having predefined roles in RBAC makes it easy to include new users with the right privileges or switch roles of existing users. It also cuts down on the potential for error when user permissions are being assigned.

- Enhancing compliance.

  Every organization must comply with local, state and federal regulations. Companies generally prefer to implement RBAC systems to meet the regulatory and statutory requirements for confidentiality and privacy because executives and IT departments can more effectively manage how the data is accessed and used. This is particularly important for financial institutions and healthcare companies that manage sensitive data.

- Reducing costs.

  By not allowing user access to certain processes and applications, companies may conserve or use resources such as network bandwidth, memory and storage in a cost-effective manner.

- Decreasing risk of breaches and data leakage.

  Implementing RBAC means restricting access to sensitive information, thus reducing the potential for data breaches or data leakage.

**Best practices for role-based access control implementations**

- As an administrator, determine the list of users and assign the users to the predefined roles. For example, the user "networkadmin" can be created and added to the user group "administrators".

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

  **Note**

  The user groups or roles are created by the system. You cannot create or modify a user group.

  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.
• Terminate accounts for users who no longer require access.

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

• Periodically conduct audits to evaluate the roles, the employees who are assigned to them and the access that's permitted for each role. If a user is found to have unnecessary access to a certain system, change the user’s role.

For more details see, Users, Roles, and Authentication, on page 23

**Restrict Device Accessibility**

Users have repeatedly been caught unawares by attacks against features they had not protected because they did not know that those features were enabled. Unused services tend to be left with default configurations which are not always secure. These services may also be using default passwords. Some services can give an attacker easy access to information on what the server is running or how the network is setup. The following sections describe how NFVIS avoids such security risks:

**Attack vector reduction**

Any piece of software can potentially contain security vulnerabilities. More software means more avenues for attack. Even if there are no publicly known vulnerabilities at the time of inclusion, vulnerabilities will probably be discovered or disclosed in the future. To avoid such scenarios, only those software packages which are essential for the NFVIS functionality are installed. This helps to limit software vulnerabilities, reduce resource consumption, and reduce extra work when problems are found with those packages. All third-party software included in NFVIS is registered at a central database in Cisco so that Cisco is able to perform a company level organized response (Legal, Security, etc). Software packages are periodically patched in every release for known Common Vulnerabilities and Exposures (CVEs).

**Enabling only essential ports by default**

Only those services which are absolutely necessary to setup and manage NFVIS are available by default. This removes the user effort needed to configure firewalls and deny access to unnecessary services. The only services that are enabled by default are listed below along with the ports they open.

<table>
<thead>
<tr>
<th>Open Port</th>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22/TCP</td>
<td>SSH</td>
<td>Secure Socket Shell for remote command-line access to NFVIS</td>
</tr>
<tr>
<td>80/TCP</td>
<td>HTTP</td>
<td>Hypertext Transfer Protocol for the NFVIS portal access. All HTTP traffic received by NFVIS is redirected to port 443 for HTTPS</td>
</tr>
<tr>
<td>443/TCP</td>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure for secure NFVIS portal access</td>
</tr>
</tbody>
</table>
Open Port | Service | Description
--- | --- | ---
830/TCP | NETCONF-ssh | Port opened for the Network Configuration Protocol (NETCONF) over SSH. NETCONF is a protocol used for automated configuration of NFVIS and for receiving asynchronous event notifications from NFVIS.
161/UDP | SNMP | Simple Network Management Protocol (SNMP). Used by NFVIS to communicate with remote network-monitoring applications. For more information see, Introduction about SNMP, on page 252

**Restrict Access To Authorized Networks For Authorized Services**

Only authorized originators should be permitted to even attempt device management access, and access should be only to the services they are authorized to use. NFVIS can be configured such that access is restricted to known, trusted sources and expected management traffic profiles. This reduces the risk of unauthorized access and the exposure to other attacks, such as brute force, dictionary, or DoS attacks.

To protect the NFVIS management interfaces from unnecessary and potentially harmful traffic, an admin user can create Access Control Lists (ACLs) for the network traffic that is received. These ACLs specify the source IP addresses/networks from which the traffic originates, and the type of traffic that is permitted or rejected from these sources. These IP traffic filters are applied to each management interface on NFVIS. The following parameters are configured in an IP receive Access Control List (ip-receive-acl)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source network/Netmask</td>
<td>Network/netmask. For example: 0.0.0.0/0 172.39.162.0/24</td>
<td>This field specifies the IP address/network from which the traffic originates</td>
</tr>
<tr>
<td>Service</td>
<td>https icmp netconf scpd smtp ssh</td>
<td>Type of traffic from the specified source.</td>
</tr>
</tbody>
</table>
### Action

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>accept</td>
<td>Action to be taken on the traffic from the source network.</td>
</tr>
<tr>
<td></td>
<td>drop</td>
<td>With <strong>accept</strong>, new connection attempts will be granted.</td>
</tr>
<tr>
<td></td>
<td>reject</td>
<td>With <strong>reject</strong>, connection attempts will not be accepted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the rule is for a TCP based service such as HTTPS, NETCONF, SCP, SSH,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the source will get a TCP reset (RST) packet.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-TCP rules such as SNMP and ICMP, the packet will be dropped.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With <strong>drop</strong>, all packets will be dropped immediately, there is no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information sent to the source.</td>
</tr>
</tbody>
</table>

### Priority

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>A numeric value</td>
<td>The <strong>priority</strong> is used to enforce an order on the rules. Rules with a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>higher numeric value for priority will be added further down in the chain.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you want to make sure that a rule will be added after another one, use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a low priority number for the first and a higher priority number for the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>following.</td>
</tr>
</tbody>
</table>

The following sample configurations illustrate some scenarios that can be adapted for specific use-cases.

#### Configuring the IP Receive ACL

The more restrictive an ACL, the more limited the exposure to unauthorized access attempts. However, a more restrictive ACL can create a management overhead, and can impact accessibility to perform troubleshooting. Consequently, there is a balance to be considered. One compromise is to restrict access to internal corporate IP addresses only. Each customer must evaluate the implementation of ACLs in relation to their own security policy, risks, exposure, and acceptance thereof.

Reject ssh traffic from a subnet:

```bash
nfvis(config)# system settings ip-receive-acl 171.70.63.0/24 service ssh action reject priority 1
```

Removing ACLs:

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 just one service, configure other services again.

```bash
nfvis(config)# no system settings ip-receive-acl 171.70.63.0/24
```

For more details see, Configuring the IP Receive ACL, on page 17
Privileged Debug Access

The super-user account on NFVIS is disabled by default, to prevent all unrestricted, potentially adverse, system-wide changes and NFVIS does not expose the system shell to the user.

However, for some hard to debug issues on the NFVIS system, the Cisco Technical Assistance Center team (TAC) or development team might require shell access to the customer’s NFVIS. NFVIS has a secure unlock infrastructure to ensure that privileged debug access to a device in the field is restricted to authorized Cisco employees. To securely access the Linux shell for this kind of interactive debugging, a challenge-response authentication mechanism is used between NFVIS and the Interactive debugging server maintained by Cisco. The admin user’s password is also required in addition to the challenge-response entry to ensure that the device is accessed with the customer’s consent.

Steps to access the shell for Interactive Debugging:

1. An admin user initiates this procedure using this hidden command.

   `nfvis# system shell-access`

2. The screen will show a challenge string, for example:

   ```
   Challenge String (Please copy everything between the asterisk lines exclusively):
   ********************************************************************************
   SPH/wkAAABOcR1JU0VOQ01MINDA4L0s5AQAAABt+dcx+hBOV6r9RkdV9MeEzNTgw
   R1Hq7BxeAAA=
   DONE.
   ********************************************************************************
   ```

3. The Cisco member enters the Challenge string on an Interactive Debug server maintained by Cisco. This server verifies that the Cisco user is authorized to debug NFVIS using the shell, and then returns a response string.

4. Enter the response string on the screen below this prompt:

   Input your response when ready:

5. When prompted, the customer should enter the admin password.

6. You get shell-access if the password is valid.

7. Development or TAC team uses the shell to proceed with the debugging.

8. To exit shell-access type `Exit`.

For more information see, System Shell Access, on page 273.

Secure Interfaces

NFVIS management access is allowed using the interfaces shown in the diagram. The following sections describe security best practices for these interfaces to NFVIS.
Console

The console port is an asynchronous serial port that allows you to connect to the NFVIS CLI for initial configuration. A user can access the console with either physical access to the NFVIS or remote access through the use of a terminal server. If console port access is required via a terminal server, configure access lists on the terminal server to allow access only from the required source addresses.

SSH

Users can access the NFVIS CLI by using SSH as a secure means of remote login. The integrity and confidentiality of NFVIS management traffic is essential to the security of the administered network since administration protocols frequently carry information which could be used to penetrate or disrupt the network.

NFVIS uses SSH version 2, which is Cisco's and the Internet's de facto standard protocol for interactive logins and supports strong encryption, hash, and key exchange algorithms recommended by the Security and Trust Organization within Cisco.

CLI Session timeout

By logging in via SSH, a user establishes a session with NFVIS. While the user is logged in, if the user leaves the logged-in session unattended, this can expose the network to a security risk. Session security limits the risk of internal attacks, such as one user trying to use another user’s session.

To mitigate this risk, NFVIS times out CLI sessions after 15 minutes of inactivity. When the session timeout is reached, the user is automatically logged out.

NETCONF

The Network Configuration Protocol (NETCONF) is a Network Management protocol developed and standardized by the IETF for the automated configuration of network devices.

The NETCONF protocol uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages. The protocol messages are exchanged on top of a secure transport protocol.

NETCONF allows NFVIS to expose an XML-based API that the network operator can use to set and get configuration data and event notifications securely over SSH.

For more information see, NETCONF Event Notifications, on page 251.
REST API

NFVIS can be configured using RESTful API over HTTPS. The REST API allow the requesting systems to access and manipulate the NFVIS configuration by using a uniform and predefined set of stateless operations. Details on all the REST APIs can be found in the NFVIS API Reference guide.

When the user issues a REST API, a session is established with NFVIS. In order to limit risks related to denial of service attacks, NFVIS limits the total number of concurrent REST sessions to 100.

NFVIS Web Portal

The NFVIS portal is a web-based Graphical User Interface that displays information about NFVIS. The portal presents the user with an easy means to configure and monitor NFVIS over HTTPS without having to know the NFVIS CLI and API.

Session Management

The stateless nature of HTTP and HTTPS requires a method of uniquely tracking users through the use of unique session IDs and cookies.

NFVIS encrypts the user’s session. The AES-256-CBC cipher is used to encrypt the session contents with an HMAC-SHA-256 authentication tag. A random 128-bit Initialization Vector is generated for each encryption operation.

An Audit record is started when a portal session is created. Session information is deleted when the user logs out or when the session times out.

The default idle timeout for portal sessions is 15 minutes. However, this can be configured for the current session to a value between 5 and 60 minutes on the Settings page. Auto-logout will be initiated after this period. Multiple sessions are not permitted in a single browser. The Maximum number of concurrent sessions are set to 30.

The NFVIS portal utilizes cookies to associate data with the user. It uses the following cookie properties for enhanced security:

- **ephemeral** to ensure the cookie expires when the browser is closed
- **httpOnly** to make the cookie inaccessible from JavaScript
- **secureProxy** to ensure the cookie can only be sent over SSL.

Even after authentication, attacks such as Cross-Site Request Forgery (CSRF) are possible. In this scenario, an end user might inadvertently execute unwanted actions on a web application in which they're currently authenticated. To prevent this, NFVIS uses **CSRF** tokens to validate every REST API that is invoked during each session.

URL Redirection

In typical web servers, when a page is not found on the web server, the user gets a 404 message; for pages that exist, they get a login page. The security impact of this is that an attacker can perform a brute force scan and easily detect which pages and folders exist.

To prevent this on NFVIS, all non-existent URLs prefixed with the device IP are redirected to the portal login page with a 301 status response code. This means that irrespective of the URL requested by an attacker, they will always get the login page to authenticate themselves.

All HTTP server requests are redirected to HTTPS and have the following headers configured:
Disabling the Portal

The NFVIS portal access is enabled by default. If you are not planning to use the portal, it is recommended to disable portal access using this command:

```plaintext
Configure terminal
System portal access disabled
commit
```

HTTPS

All the HTTPS data to and from NFVIS uses Transport Layer Security (TLS) to communicate across the network. TLS is the successor to Secure Socket Layer (SSL).

The TLS handshake involves authentication during which the client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the owner of the domain.

By default, NFVIS uses a self-signed certificate to prove its identity to its clients. This certificate has a 2048-bit public key to increase the security of the TLS encryption, since the encryption strength is directly related to the key size.

Certificate Management

NFVIS generates a self-signed SSL certificate when first installed. It is a security best practice to replace this certificate with a valid certificate signed by a compliant Certificate Authority (CA).

Use the following steps to replace the default self-signed certificate:

1. Generate a Certificate Signing Request (CSR) on NFVIS.

   A Certificate Signing request (CSR) is a file with a block of encoded text that is given to a Certificate Authority when applying for an SSL Certificate. This file contains information that should be included in the certificate such as the organization name, common name (domain name), locality, and country. The file also contains the public key that should be included in the certificate. NFVIS uses a 2048-bit public key since encryption strength is higher with a higher key size.

   To generate a CSR on NFVIS, run the following command:

   ```plaintext
   nfvis# system certificate signing-request [common-name country-code locality organization organization-unit-name state]
   ```

   The CSR file is saved as /data/intdatastore/download/nfvis.csr.

2. Get an SSL certificate from a CA using the CSR.
From an external host, use the scp command to download the Certificate Signing Request.

```
[myhost:/tmp] > scp -P 22222 admin@<NFVIS-IP>:/data/intdatastore/download/nfvis.csr <destination-file-name>
```

Contact a Certificate authority to issue a new SSL server certificate using this CSR.

3. **Install the CA Signed Certificate.**

From an external server, use the scp command to upload the certificate file into NFVIS to the `data/intdatastore/uploads/` directory.

```
[myhost:/tmp] > scp -P 22222 <certificate file> admin@<NFVIS-IP>:/data/intdatastore/uploads
```

Install the certificate in NFVIS using the following command.

```
nfvis# system certificate install-cert path file:///data/intdatastore/uploads/<certificate file>
```

4. **Switch to using the CA Signed Certificate.**

Use the following command to start using the CA signed certificate instead of the default self-signed certificate.

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

---

**SNMP Access**

Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks, and for modifying that information to change device behavior.

Three significant versions of SNMP have been developed. NFVIS supports SNMP version 1, version 2c and version 3. SNMP versions 1 and 2 use community strings for authentication, and these are sent in plain-text. So, it is a security best practice to use SNMP v3 instead.

SNMPv3 provides secure access to devices by using three aspects: users, authentication, and encryption. SNMPv3 uses the USM (User-based Security Module) for controlling access to information available via SNMP. The SNMP v3 user is configured with an authentication type, a privacy type as well as a passphrase. All users sharing a group utilize the same SNMP version, however, the specific security level settings (password, encryption type, etc.) are specified per-user.

The following table summarizes the security options within SNMP:

<table>
<thead>
<tr>
<th>Model</th>
<th>Level</th>
<th>Authentication</th>
<th>Encryption</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>noAuthNoPriv</td>
<td>Community String</td>
<td>No</td>
<td>Uses a community string match for authentication.</td>
</tr>
<tr>
<td>v2c</td>
<td>noAuthNoPriv</td>
<td>Community String</td>
<td>No</td>
<td>Uses a community string match for authentication.</td>
</tr>
<tr>
<td>Model</td>
<td>Level</td>
<td>Authentication</td>
<td>Encryption</td>
<td>Outcome</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-----------------------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>v3</td>
<td>noAuthNoPriv</td>
<td>Username</td>
<td>No</td>
<td>Uses a username match for authentication.</td>
</tr>
<tr>
<td>v3</td>
<td>authNoPriv</td>
<td>Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)</td>
<td>No</td>
<td>Provides authentication based on the HMAC-MD5-96 or HMAC-SHA-96 algorithms.</td>
</tr>
<tr>
<td>v3</td>
<td>authPriv</td>
<td>MD5 or SHA</td>
<td>Data Encryption Standard (DES) or Advanced Encryption Standard (AES)</td>
<td>Provides authentication based on the HMAC-MD5-96 or HMAC-SHA-96 algorithms. Provides DES Cipher algorithm in Cipher Block Chaining Mode (CBC-DES) or AES encryption algorithm used in Cipher FeedBack Mode (CFB), with a 128-bit key size (CFB128-AES-128)</td>
</tr>
</tbody>
</table>

Since its adoption by NIST, AES has become the dominant encryption algorithm throughout the industry. To follow the industry's migration away from MD5 and toward SHA, it is a security best practice to configure the SNMP v3 authentication protocol as SHA and privacy protocol as AES. For more details on SNMP see, Introduction about SNMP, on page 252

**Legal Notification Banners**

It is recommended that a legal notification banner is present on all interactive sessions to ensure that users are notified of the security policy being enforced and to which they are subject. In some jurisdictions, civil and/or criminal prosecution of an attacker who breaks into a system is easier, or even required, if a legal notification banner is presented, informing unauthorized users that their use is in fact unauthorized. In some jurisdictions, it may also be forbidden to monitor the activity of an unauthorized user unless they have been notified of the intent to do so.

Legal notification requirements are complex and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary. Discuss this issue with your own legal counsel to ensure that the notification banner meets company, local, and international legal requirements. This is often critical to securing appropriate
action in the event of a security breach. In cooperation with the company legal counsel, statements which may be included in a legal notification banner include:

- Notification that the system access and use is permitted only by specifically authorized personnel, and perhaps information about who may authorize use.
- Notification that unauthorized access and use of the system is unlawful, and may be subject to civil and/or criminal penalties.
- Notification that access and use of the system may be logged or monitored without further notice, and the resulting logs may be used as evidence in court.
- Additional specific notices required by specific local laws.

From a security rather than a legal point of view, a legal notification banner should not contain any specific information about the device, such as its name, model, software, location, operator or owner because this kind of information may be useful to an attacker.

The following is a sample legal notification banner which can be displayed before login:

```
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED You must have explicit, authorized permission to access or configure this device. Unauthorized attempts and actions to access or use this system may result in civil and/or criminal penalties. All activities performed on this device are logged and monitored.
```

Present a legal notification banner approved by company legal counsel.

NFVIS allows the configuration of a banner and Message of the Day (MOTD). The banner is displayed before the user logs in. Once the user logs in to NFVIS, a system-defined banner provides Copyright information about NFVIS, and the message-of-the-day (MOTD), if configured, will appear, followed by the command line prompt or portal view, depending on the login method.

It is recommended that a login banner is implemented to ensure that a legal notification banner is presented on all the device management access sessions prior to a login prompt being presented. Use this command to configure the banner and MOTD.

```
nfvis(config)# banner-motd banner <banner-text> motd <message-of-the-day-text>
```

For more information about the banner command, see Configure Banner, Message of the day and System Time, on page 72.

**Factory Default Reset**

Factory Reset removes all the customer specific data that has been added to the device since the time of its shipping. The data erased includes configurations, log files, VM images, connectivity information, and user login credentials.

It provides one command to reset the device to factory-original settings, and is useful in the following scenarios:

- Return Material Authorization (RMA) for a device—If you have to return a device to Cisco for RMA, use Factory Default reset to remove all the customer-specific data.
- Recovering a compromised device—If the key material or credentials stored on a device is compromised, reset the device to factory configuration and then reconfigure the device.
• If the same device needs to be re-used at a different site with a new configuration, perform a Factory Default reset to remove the existing configuration and bring it to a clean state.

NFVIS provides the following options within Factory default reset:

<table>
<thead>
<tr>
<th>Factory Reset Option</th>
<th>Data Erased</th>
<th>Data Retained</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>All configuration, uploaded image files, VMs and logs. Connectivity to the device will be lost.</td>
<td>The admin account is retained and the password will be changed to the factory default password.</td>
</tr>
<tr>
<td>all-except-images</td>
<td>All configuration except image configuration, VMs, and uploaded image files.</td>
<td>Image configuration, registered images and logs. The admin account is retained and the password will be changed to the factory default password.</td>
</tr>
<tr>
<td>all-except-images-connectivity</td>
<td>All configuration except image, network and connectivity configuration, VMs, and uploaded image files. Connectivity to the device is available.</td>
<td>Images, network and connectivity related configuration, registered images, and logs. The admin account is retained and the previously configured admin password will be preserved.</td>
</tr>
<tr>
<td>manufacturing</td>
<td>All configuration except image configuration, VMs, uploaded image files, and logs. Connectivity to the device will be lost.</td>
<td>Image related configuration and registered images. The admin account is retained and the password will be changed to the factory default password.</td>
</tr>
</tbody>
</table>

The user must choose the appropriate option carefully based on the purpose of the Factory Default reset.

For more information, see Reseting to Factory Default, on page 71.

### Infrastructure Management Network

An infrastructure management network refers to the network carrying the control and management plane traffic (such as NTP, SSH, SNMP, syslog, etc.) for the infrastructure devices. Device access can be through the console, as well as through the Ethernet interfaces. This control and management plane traffic is critical to network operations, providing visibility into and control over the network. Consequently, a well-designed and secure infrastructure management network is critical to the overall security and operations of a network. One of the key recommendations for a secure infrastructure management network is the separation of management and data traffic in order to ensure remote manageability even under high load and high traffic conditions. This can be achieved using a dedicated management interface.

The following are the Infrastructure management network implementation approaches:
Out-of-band Management

An Out-of-band Management (OOB) management network consists of a network which is completely independent and physically disparate from the data network that it helps to manage. This is also sometimes referred to as a Data Communications Network (DCN). Network devices can connect to the OOB network in different ways: – NFVIS supports a built-in management interface that can be used to connect to the OOB network. NFVIS allows the configuration of a predefined physical interface, the MGMT port on the ENCS, as a dedicated management interface. Restricting management packets to designated interfaces provides greater control over the management of a device, thereby providing more security for that device. Other benefits include improved performance for data packets on non-management interfaces, support for network scalability, need for fewer access control lists (ACLs) to restrict access to a device, and prevention of management packet floods from reaching the CPU.

Network devices can also connect to the OOB network via dedicated data interfaces. In this case, ACLs should be deployed to ensure that management traffic is only handled by the dedicated interfaces.

For further information, see Configuring the IP Receive ACL, on page 17 and Port 22222 and Management Interface ACL, on page 17.

Pseudo out-of-band Management

A pseudo out-of-band management network uses the same physical infrastructure as the data network but provides logical separation through the virtual separation of traffic, by using VLANs. NFVIS supports creating VLANs and virtual bridges to help identify different sources of traffic and separate traffic between VMs. Having separate bridges and VLANs isolates the virtual machine network’s data traffic and the management network, thus providing traffic segmentation between the VMs and the host. For further information see Configuring VLAN for NFVIS Management Traffic, on page 16.

In-band Management

An in-band management network uses the same physical and logical paths as the data traffic.

Ultimately, this network design requires a per-customer analysis of risk versus benefits and costs. Some general considerations include:

• An isolated OOB management network maximizes visibility and control over the network even during disruptive events.

• Transmitting network telemetry over an OOB network minimizes the chance for disruption of the very information which provides critical network visibility.

• In-band management access to network infrastructure, hosts, etc. is vulnerable to complete loss in the event of a network incident, removing all the network visibility and control. Appropriate QoS controls should be put in place to mitigate this occurrence.

• NFVIS features interfaces which are dedicated to device management, including serial console ports and Ethernet management interfaces.

• An OOB management network can typically be deployed at a reasonable cost, since management network traffic does not typically demand high bandwidth nor high performance devices, and only requires sufficient port density to support the connectivity to each infrastructure device.
Locally Stored Information Protection

Protecting Sensitive Information

NFVIS stores some sensitive information locally, including passwords and secrets. Passwords should generally be maintained and controlled by a centralized AAA server. However, even if a centralized AAA server is deployed, some locally-stored passwords are required for certain cases such as local fallback in the case of AAA servers not being available, special-use usernames, etc. These local passwords and other sensitive information are stored on NFVIS as hashes so that it is not possible to recover the original credentials from the system. Hashing is a widely accepted industry norm.

File Transfer

Files which may need to be transferred to NFVIS devices include VM image and NFVIS upgrade files. The secure transfer of files is critical for network infrastructure security. NFVIS supports Secure Copy (SCP) to ensure the security of file transfer. SCP relies on SSH for secure authentication and transport, enabling the secure and authenticated copying of files.

A secure copy from NFVIS is initiated through the scp command. The secure copy (scp) command allows only the admin user to securely copy files from NFVIS to an external system, or from an external system to NFVIS.

The syntax for the scp command is:

```
scp <source> <destination>
```

We use port 22222 for the NFVIS SCP server. By default, this port is closed and users cannot secure copy files into NFVIS from an external client. If there is a need to SCP a file from an external client, the user can open the port using:

```
system settings ip-receive-acl (address)/(mask lenth) service scpd priority (number) action accept commit
```

To prevent users from accessing system directories, secure copy can be performed only to or from intdatastore:, extdatastore1:, extdatastore2:, usb: and nfs:, if available. Secure copy can also be performed from logs: and techsupport:

Logging

NFVIS access and configuration changes are logged as audit logs to record the following information:

- Who accessed the device
- When did a user log in
- What did a user do in terms of the host configuration and the VM lifecycle
- When did a user log off
• Failed access attempts
• Failed authentication requests
• Failed authorization requests

This information is invaluable for forensic analysis in case of unauthorized attempts or access, as well as for configuration change issues and to help plan group administration changes. It may also be used real time to identify anomalous activities which may indicate that an attack is taking place. This analysis can be correlated with information from additional external sources, such as IDS and firewall logs.

All the key events on the NFVIS are sent as event notifications to NETCONF subscribers and as syslogs to the configured central logging servers. For more information on syslog messages and event notifications, see Appendix, on page 285.

Virtual Machine security

This section describes security features related to the registration, deployment and operation of Virtual Machines on NFVIS.

VNF secure boot

NFVIS supports Open Virtual Machine Firmware (OVMF) to enable UEFI secure boot for Virtual Machines which support secure boot. VNF Secure boot verifies that each layer of the VM boot software is signed, including the bootloader, the operating system kernel, and operating system drivers.

For more information see, Secure Boot of VNFs, on page 138.

VNC Console Access Protection

NFVIS allows the user to create a Virtual Network Computing (VNC) session to access a deployed VM’s remote desktop. To enable this, NFVIS dynamically opens a port to which the user can connect using their web browser. This port is only left open for 60 seconds for an external server to start a session to the VM. If no activity is seen within this time, the port is closed. The port number is assigned dynamically and thereby allows only a one-time access to the VNC console.
Encrypted VM config data variables

During VM deployment, the user provides a day-0 configuration file for the VM. This file can contain sensitive information such as passwords and keys. If this information is passed as clear text, it appears in log files and internal database records in clear text. This feature allows the user to flag a config data variable as sensitive so that its value is encrypted using AES-CFB-128 encryption before it is stored or passed to internal subsystems.

For more information see, VM Deployment Parameters, on page 115.

Checksum verification for Remote Image Registration

To register a remotely located VNF image, the user specifies its location. The image will need to be downloaded from an external source, such as an NFS server or a remote HTTPS server.

To know if a downloaded file is safe to install, it is essential to compare the file’s checksum before using it. Verifying the checksum helps ensure that the file was not corrupted during network transmission, or modified by a malicious third party before you downloaded it.

NFVIS supports the checksum and checksum_algorithm options for the user to provide the expected checksum and checksum algorithm (SHA256 or SHA512) to be used to verify the checksum of the downloaded image. Image creation fails if the checksum does not match.

Certification Validation for Remote Image Registration

To register a VNF image located on a HTTPS server, the image will need to be downloaded from the remote HTTPS server. To securely download this image, NFVIS verifies the SSL certificate of the server. The user needs to specify either the path to the certificate file or the PEM format certificate contents to enable this secure download.

More details can be found at Register a Remote VM Image

VM Isolation and Resource provisioning

The Network Function Virtualization (NFV) architecture consists of:

• Virtualized network functions (VNFs), which are Virtual Machines running software applications that deliver network functionality such as a router, firewall, load balancer, and so on.

• Network functions virtualization infrastructure, which consists of the infrastructure components—compute, memory, storage, and networking, on a platform that supports the required software and hypervisor.

With NFV, network functions are virtualized so that multiple functions can be run on a single server. As a result, less physical hardware is needed, allowing for resource consolidation. In this environment, it is essential to simulate dedicated resources for multiple VNFs from a single, physical hardware system. Using NFVIS, VMs can be deployed in a controlled manner such that each VM receives the resources it needs.
are partitioned as needed from the physical environment to the many virtual environments. The individual VM domains are isolated so they are separate, distinct, and secure environments, which are not contending with each other for shared resources.

VMs cannot use more resources than provisioned. This avoids a Denial of Service condition from one VM consuming the resources. As a result, CPU, memory, network and storage are protected.

**CPU Isolation**

The NFVIS system reserves cores for the infrastructure software running on the host. The rest of the cores are available for VM deployment. This guarantees that the VM’s performance does not affect the NFVIS host performance.

**Low-latency VMs**

NFVIS explicitly assigns dedicated cores to low latency VMs that are deployed on it. If the VM requires 2 vCPUs, it is assigned 2 dedicated cores. This prevents sharing and oversubscription of cores and guarantees the performance of the low-latency VMs. If the number of available cores is less than the number of vCPUs requested by another low-latency VM, the deployment is prevented since we do not have sufficient resources.

**Non low-latency VMs**

NFVIS assigns sharable CPUs to non low latency VMs. If the VM requires 2 vCPUs, it is assigned 2 CPUs. These 2 CPUs are shareable among other non low latency VMs. If the number of available CPUs is less than the number of vCPUs requested by another non low-latency VM, the deployment is still allowed because this VM will share the CPU with existing non low latency VMs.
**Memory Allocation**

The NFVIS Infrastructure requires a certain amount of memory. When a VM is deployed, there is a check to ensure that the memory available after reserving the memory required for the infrastructure and previously deployed VMs, is sufficient for the new VM. We do not allow memory oversubscription for the VMs.

VMs are not allowed to directly access the host file system and storage.

**Storage Isolation**

The ENCS platform supports an internal datastore (M2 SSD) and external disks. NFVIS is installed on the internal datastore. VNFs can also be deployed on this internal datastore. It is a security best practice to store customer data and deploy customer application Virtual Machines on the external disks. Having physically separate disks for the system files vs the application files helps to protect system data from corruption and security issues.
Interface Isolation

Single Root I/O Virtualization or SR-IOV is a specification that allows the isolation of PCI Express (PCIe) resources such as an Ethernet port. Using SR-IOV a single Ethernet port can be made to appear as multiple, separate, physical devices known as Virtual Functions. All of the VF devices on that adapter share the same physical network port. A guest can use one or more of these Virtual Functions. A Virtual Function appears to the guest as a network card, in the same way as a normal network card would appear to an operating system.

Virtual Functions have near-native performance and provide better performance than para-virtualized drivers and emulated access. Virtual Functions provide data protection between guests on the same physical server as the data is managed and controlled by the hardware.

NFVIS VNFs can use SR-IOV networks to connect to WAN and LAN Backplane ports.

Each such VM owns a virtual interface and its related resources achieving data protection among VMs.

Secure Development Lifecycle

NFVIS follows a Secure Development Lifecycle (SDL) for software. This is a repeatable, measurable process designed to reduce vulnerabilities and enhance the security and resilience of Cisco solutions. Cisco SDL applies industry-leading practices and technology to build trustworthy solutions that have fewer field-discovered product security incidents. Every NFVIS release goes through the following processes.

• Following Cisco-internal and market-based Product Security Requirements
• Registering 3rd party software with a central repository at Cisco for vulnerability tracking
• Periodically patching software with known fixes for CVEs.
• Designing software with Security in mind
• Following secure coding practices such as using vetted common security modules like CiscoSSL, running Static Analysis and implementing input validation for Preventing command injection, etc.
• Using Application Security tools such as IBM AppScan, Nessus, and other Cisco internal tools.
Secure Operation in FIPS Mode on NFVIS

Table 25: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Operation in FIPS Mode on NFVIS</td>
<td>NFVIS 4.2.1</td>
<td>The Federal Information Processing Standards (FIPS) Publication 140-2 are publicly announced standards developed by the United States federal government for use in computer systems by non-military government agencies and government contractors. In FIPS mode, the device tries to prevent the use of non-FIPS compatible algorithms, but you must ensure that you configure the device to use only FIPS approved algorithms. Some functionality may silently fail in FIPS mode if it attempts to use non-FIPS compliant algorithms.</td>
</tr>
</tbody>
</table>

The Federal Information Processing Standards (FIPS) Publication 140-2 are publicly announced standards developed by the United States federal government for use in computer systems by non-military government agencies and government contractors. In FIPS mode, the device tries to prevent the use of non-FIPS compatible algorithms, but you must ensure that you configure the device to use only FIPS approved algorithms. Some functionality may silently fail in FIPS mode if it attempts to use non-FIPS compliant algorithms.

- Configuring FIPS Mode, on page 191
- FIPS Operational Status, on page 194

Configuring FIPS Mode

NFVIS supports both FIPS and non-FIPS mode of operation. NFVIS is FIPS 140-2 compliant for SSH, TLS and SNMP protocols.
To enable FIPS mode:

```
config terminal
security fips
commit
```

Only after NFVIS reboot after the configuration FIPS mode is enabled.

FIPS mode can be disabled only with factory reset. If you try to disable FIPS mode with the no form of the command:

```
config terminal
no security fips
commit
```

Aborted: This command can only be removed while factory reset

The following is an example, where FIPS mode is successfully configured, but not enabled:

```
fvis# show security
security fips-status CONFIGURED_REBOOT_TO_ENABLE
```

To verify the status of the FIPS mode after reboot:

```
fvis# show security
security fips-status ENABLED
```

FIPS mode configuration is terminated when:

- SNMP v1 or v2 is configured.

The following is an example of FIPS mode configuration failure when SNMP v1 or v2 is configured:

```
config terminal
security fips
commit
```

Aborted: SNMP version 1 and/or SNMP version 2 is configure. Please unconfigure SNMPv1 and SNMPv2 and then try again
• SNMP v3 is configured with auth protocol MD5.

The following is an example of FIPS mode configuration failure when SNMP v3 is configured with auth protocol md5:

```plaintext
config terminal
security fips
commit
Aborted: SNMP version 3 MD5 auth-protocol configured other secure protocol and try again
```

After FIPS mode is enabled, SNMP v1 or v2 and SNMP v3 with auth protocol MD5 cannot be configured.

```plaintext
snmp group test_v1 snmp 1 noAuthNoPriv read test write test
commit
Aborted: Cannot configure SNMP group-version 1 because fips-status is ENABLED
```

```plaintext
config terminal
snmp user test_md5_v3
  user-version 3
  user-group test_v3
  auth-protocol md5
commit
Aborted: 'snmp user test_md5_v3_passhd auth-protocol': Cannot configure SNMP user-version 3 with auth-protocol MD5 because fips-status is CONFIGURED_REBOOT_TO_ENABLE
```

**Backup and Restore for FIPS Mode**

If you backup NFVIS configurations when FIPS mode is enabled, then upon restore, FIPS mode is configured but needs a manual reboot to enable FIPS mode.

**Backup configuration**

```plaintext
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

**After restore**

```plaintext
nfvis#
nfvis# show running-config security
security fips
nfvis# show security fips-status
security fips-status CONFIGURED_REBOOT_TO_ENABLE
nfvis#
```

**After reboot**

```plaintext
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```
When you backup NFVIS configurations with FIPS mode disabled, but the system where you restore the configurations has FIPS mode enabled, upon restore, the NFVIS configurations disable FIPS mode but the system has to reboot for FIPS mode to be in DISABLED state.

**Backup configurations**

nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#

**Restore system configurations**

nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#

**After restore**

nfvis# show running-config security
% No entries found.
nfvis# show security
security fips-status UNCONFIGURED_REBOOT_TO_DISABLE
nfvis#

**After reboot**

nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#

---

**FIPS Operational Status**

The following is a list of operational states when you try to enable FIPS mode:

- **DISABLED**
- **CONFIGURED-REBOOT-TO-ENABLE**
- **ENABLED**
- **UNCONFIGURED-REBOOT-TO-DISABLE**
- **FAILED**

The following table lists the possible operational state transitions for FIPS mode:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISABLED</td>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>If the Oper data of fips-state leafs was previously set to DISABLED and if the <code>security fips</code> configuration is pushed</td>
</tr>
<tr>
<td>DISABLED</td>
<td>FAILED</td>
<td>If there is an error while pushing the <code>security fips</code> configuration</td>
</tr>
<tr>
<td>From</td>
<td>To</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>ENABLED</td>
<td>If the fips-mode configuration is successful before and the Oper data was set to CONFIGURED-REBOOT-TO-ENABLE, then after reboot set the Oper data to ENABLED</td>
</tr>
<tr>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>FAILED</td>
<td>If the Oper data of fips-state leafs was previously set to CONFIGURED-REBOOT-TO-ENABLE and there was an error while removing fips-mode configuration</td>
</tr>
<tr>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>DISABLED</td>
<td>If the fips-mode is unconfigured while restoring from a backup package or factory-reset and the current fips-status is CONFIGURED-REBOOT-TO-ENABLE</td>
</tr>
<tr>
<td>ENABLED</td>
<td>DISABLED</td>
<td>After factory-reset (of any type)</td>
</tr>
<tr>
<td>ENABLED</td>
<td>FAILED</td>
<td>If there is an error while disabling the FIPS mode</td>
</tr>
<tr>
<td>ENABLED</td>
<td>UNCONFIGURED-REBOOT-TO-DISABLE</td>
<td>If the fips-mode is unconfigured while restoring from a backup package and the current fips-status is ENABLED</td>
</tr>
<tr>
<td>FAILED</td>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>If the Oper date of fips-state leafs was previously set to FAILED and now configuring fips-mode</td>
</tr>
<tr>
<td>FAILED</td>
<td>DISABLED</td>
<td>If the Oper date of fips-state leafs was previously set to FAILED and now issued factory-reset</td>
</tr>
<tr>
<td>UNCONFIGURED-REBOOT-TO-DISABLE</td>
<td>DISABLED</td>
<td>If the Oper data of fips-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and then NFVIS is rebooted</td>
</tr>
<tr>
<td>UNCONFIGURED-REBOOT-TO-DISABLE</td>
<td>CONFIGURED-REBOOT-TO-ENABLE</td>
<td>If the Oper data of fips-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and fips-mode config is successful</td>
</tr>
<tr>
<td>UNCONFIGURED-REBOOT-TO-DISABLE</td>
<td>FAILED</td>
<td>If the Oper data of fips-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and if the fips-mode config was unsuccessful</td>
</tr>
</tbody>
</table>
Platform Specific Configurations

- ENCS Switch Configuration, on page 197
- Configuring vBranch High Availability, on page 210
- CIMC Secure Overlay Support, on page 222
- CIMC TACACS Support, on page 223
- LTE PIM Module on Cisco Catalyst 8200 UCPE, on page 224
- External Storage for Cisco ENCS 5400 and Cisco Cloud Services Platforms, on page 243

**ENCS Switch Configuration**

Access to the ENCS switch is restricted through Consent Token. Consent Token is a security feature that is used to authenticate the network administrator of an organization to access system shell with mutual consent from the network administrator and Cisco Technical Assistance Centre (Cisco TAC).

---

**Note**

From the switch console, there is access to debug mode and an advanced debug mode. Credentials of the local user are synchronized to access debug mode. Advanced debug uses unique credentials for each device that allows for additional debugging options for Cisco engineering. To enter either debug mode permission must be granted through Consent Token.

---

**ENCS Switch Commands**

See, Cisco Enterprise Network Compute System Switch Command Reference for switch commands.

---

**ENCS Switch APIs**

See, API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software for switch related APIs.
ENCS Switch Portal Configuration

Switch Settings

The Switch option from the Cisco Enterprise NFVIS portal allows you to configure STP/RSTP, VLAN on specified ranges, RADIUS based authentication, and port channel load balancing for various switch ports. This section describes how to configure settings on the ENCS switch portal.
You can view the Switch Interface operational data and the statistics parameters in the following table:

**Table 26: Switch Settings Interface**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwitchPort</td>
<td>Specifies the switch interface name.</td>
<td></td>
</tr>
</tbody>
</table>
### Platform Specific Configurations

#### Switch Settings

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies the description of the interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Specifies the status of the interface.</td>
</tr>
<tr>
<td>MAC Address</td>
<td>Specifies the MAC address of the interface.</td>
</tr>
</tbody>
</table>
| Port Type            | Specifies the mode of the port interface.  | Supported types are:  
  - access  
  - dot1q-tunnel  
  - private-vlan  
  - trunk |
| VLAN                 | Specifies the VLAN ID.                     | Range: 1-2349 and 2450-4093 |
| Speed                | Specifies the speed of the interface.      | Speed:  
  - 10 MBPS  
  - 100 MBPS  
  - 1000 MBPS |
| RxBytes              | Specifies the received data on interface in bytes. |
| PktDrop              | Specifies the number of packet drops.       |
| PORT                 | Specifies the port number.                 |
| IN-UCAST             | Specifies the number of incoming unicast packets at the interface. |
| OUT-UCAST            | Specifies the number of outgoing unicast packets at the interface. |
| IN-MCAST             | Specifies the number of incoming multicast packets at the interface. |
| OUT-MCAST            | Specifies the number of outgoing multicast packets at the interface. |
| IN-BCAST             | Specifies the number of incoming broadcast packets at the interface. |
| OUT-BCAST            | Specifies the number of outgoing broadcast packets at the interface. |
Configuring Spanning Tree

Spanning Tree Protocol (STP) is a Layer 2 protocol that runs on bridges and switches. The main purpose of STP is to ensure that you do not create loops when you have redundant paths in your network.

The Spanning Tree option is enabled by default. You can click on edit and make the necessary settings or disable Spanning Tree if required.

The configuration of spanning tree has the following parameters when it is enabled:

**Table 27: Spanning Tree Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>dot1x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LACP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vlan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanning Tree</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>rstp</td>
<td></td>
</tr>
<tr>
<td>Forward Time</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Hello Time</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Max Age</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Loopback Guard</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable</td>
<td></td>
</tr>
<tr>
<td>Path Cost Method</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>32768</td>
<td></td>
</tr>
</tbody>
</table>

The configuration of spanning tree has the following parameters when it is enabled:

**Table 27: Spanning Tree Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>dot1x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LACP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vlan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanning Tree</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>rstp</td>
<td></td>
</tr>
<tr>
<td>Forward Time</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Hello Time</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Max Age</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Loopback Guard</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable</td>
<td></td>
</tr>
<tr>
<td>Path Cost Method</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>32768</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dot1x

This chapter describes how to configure dot1x port-based authentication on the Cisco Enterprise NFVIS portal. dot1x prevents unauthorized devices (clients) from gaining access to the network. It is a standard for media-level (Layer 2) access control, offering the capability to permit or deny network connectivity based on the identity of the end user or device. The dot1x is disabled by default. You can click on edit to enable dot1x.
The configuration of dot1x has the following parameters:

**Table 28: Dot1x Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Specifies the authentication type for the port.</td>
<td>radius or none</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is radius.</td>
</tr>
<tr>
<td>Guest VLAN Timeout(s)</td>
<td>Specifies the time delay in seconds between enabling Dot1X (or port up) and adding the port to the guest VLAN.</td>
<td>Range: 30 to 180 seconds</td>
</tr>
<tr>
<td>System Auth control</td>
<td>Specifies the authentication control.</td>
<td>Enable or Disable</td>
</tr>
</tbody>
</table>

**Configuring LACP**

The Link Aggregation Control Protocol (LACP) enables you to bundle several physical ports together to form a single logical channel. LACP enables you to form a single Layer 2 link automatically from two or more Ethernet links. This protocol ensures that both ends of the Ethernet link are functional and are part of the aggregation group.
LACP uses the following parameters to control aggregation:

### Table 29: LACP Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Priority</td>
<td>Specifies the port priority.</td>
<td>Range: 1 to 65535</td>
</tr>
<tr>
<td>Port-channel load balance</td>
<td>Specifies the load balance of the port channel.</td>
<td>Mac Based or IP Based</td>
</tr>
</tbody>
</table>

## Configuring VLAN

You can use virtual LANs (VLANs) to divide the network into separate logical areas. VLANs can also be considered as broadcast domains. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

You can configure VLANs in the range `<1-2349>|<2450-4093>` for a specified switch port.
Configuring General Settings

You can configure general settings using the following parameters for each switch interface:

- Interface—Name of the interface
- Description—Set the description per interface
- Speed—10/100/1000 MBPS
- Dot1x Auth—802.1x, mac or both
- PoE Method—auto, never or four-pair
- PoE Limit—0-60000mW
- Admin Status—enable or disable
Configuring Advanced Settings

You can make the advanced settings using the following parameters for each switch interface:

- **Mode**—access, dot1q-tunnel, private-vlan, or trunk
- **Access Vlan**—Specifies the number of VLANs.
- **Allowed Vlan**—All or VLAN IDs
  - 1 to 2349
  - 2450 to 4093
- **Native Vlan**—Specifies the VLAN ID. You can enter a value from one of the following ranges:
  - 1 to 2349
  - 2450 to 4093
- **Dot1q Tunnel Vlan**—Specifies the Layer 2 tunnel port.

You can make the advanced settings using the following parameters for each switch interface:

- **Mode**—access, dot1q-tunnel, private-vlan, or trunk
- **Access Vlan**—Specifies the number of VLANs.
- **Allowed Vlan**—All or VLAN IDs
- **Native Vlan**—Specifies the VLAN ID. You can enter a value from one of the following ranges:
  - 1 to 2349
  - 2450 to 4093
- **Dot1q Tunnel Vlan**—Specifies the Layer 2 tunnel port.
• Community—Specifies the community number. Range: 1 to 29
• Protected Port—Yes or No

---

**Note**

The VLAN configuration takes effect only if the global VLANs are also configured with the same values in Configuring VLAN, on page 204.
## Configuring Spanning Tree per Interface

<table>
<thead>
<tr>
<th>General Settings</th>
<th>Advanced Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spanning Tree</strong></td>
<td>Enable</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Choose from 1-200000000</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>128</td>
</tr>
<tr>
<td><strong>Link Type</strong></td>
<td></td>
</tr>
<tr>
<td><strong>BPDU Guard</strong></td>
<td>Enable</td>
</tr>
<tr>
<td><strong>Root Guard</strong></td>
<td>Enable</td>
</tr>
<tr>
<td><strong>Port Fast</strong></td>
<td>auto</td>
</tr>
</tbody>
</table>

[Apply] [Cancel]
You can configure spanning tree for each switch interface using the following parameters:

- **Spanning Tree**—Enable or Disable
- **Cost**—Specifies the cost. Range: 1 to 200000000
- **Priority**—Specifies the port priority. Range: 0 to 240, default value is 128
- **Link Type**—point-to-point or shared
- **BPDU Guard**—Enable or Disable
- **Root Guard**—Enable or Disable
- **Port Fast**—auto or enable
- **BPDU Filtering**—Specifies that BPDU packets are filtered when the spanning tree is disabled
- **BPDU Flooding**—Specifies that BPDU packets are flooded when the spanning tree is disabled

### Configuring Storm Control

Storm control is used to monitor incoming traffic levels and limit excessive flow of packets on any user facing switch port that could cause a traffic storm. Traffic storms can lead to device instability and unintended behavior.

You can configure storm control from NFVIS Portal, from Storm Control tab.
Storm control can be configured for specific type of traffic - unicast or multicast or broadcast. The suppression range can be in terms of a percentage level (1-100) or Kbps value (1-1000000).

**Configuring vBranch High Availability**

High availability design provides redundancy for WAN, LAN, ENCS device, vRouter, vFirewall VNF level redundancy.

A branch site can have two routers for redundancy. If vEdge-cloud router is chosen, Each of the vEdge-cloud router maintains:

- A secure control plane connection, via a DTLS connection, with each vSmart controller in its domain
- A secure data plane connection with the other vEdge routers at the site

Because both vEdge routers receive the same routing information from the vSmart controllers, each one is able to continue to route traffic if one should fail, even if they are connected to different transport providers.

Two firewalls are placed in a group and their configuration is synchronized to prevent a single point of failure on your network. A heartbeat connection between the firewall peers ensures seamless failover in the event that a peer goes down. Setting up two firewalls in an HA pair provides redundancy and allows you to ensure business continuity.

**Prerequisites for vBranch HA**

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.

Two vRouters and the Two vFirewalls have full mesh L3 connectivity.
VMs and VNFs on both ENCS devices must be configured identical.

SD-Branch HA Design and Topology

In HA design, there are two sets of VLANs. Traffic path is between the VNFs and traffic from or towards LAN.

To protect against cable connection issue and box failure, there is back-to-back cable between ENCS and connection from each ENCS to the external switch.

When using Cisco ENCS and Cisco switches, common expectation is to use PVST+, detect loops and switch specific ports to BLOCKING mode. ENCS switch does not support PVST (Per VLAN spanning tree). By Default, RSTP could end up blocking ENCS port back-to-back connection, this will result in blocking traffic path between the VNFs.

The recommended solution is to use MSTP in ENCS and the external switches. The following topology and configuration provides a step-by-step procedure with reasoning for specific configuration use. There are two instances of MSTP created. One for handling traffic path between VNFs and the second for handling traffic from or towards LAN.

Note

In cases where external switch cannot be configured for MSTP, RSTP is used and the two links back-2-back between ENCS is not in port-channel.

• One of the links carries traffic between VNFs by configuring disable spanning tree. The second back-to-back link between ENCS processes RSTP and forward or block for the traffic from or towards LAN.

• From each of the ENCS, a third physical link connects to the external switch. This also forwards or blocks the traffic from or towards LAN depending on the RSTP decisions.
In the absence of firewall in the design, the router is directly connected to the LAN side. Pt-to-Pt network extends the TLOC connection across the ENCS devices and VRRP is enabled in the router LAN facing connection.

**Isolating LAN and Transit Link Traffic for vBranch HA**

Traffic from or towards LAN and traffic between the VNFs are isolated by configuring different VLANs for each traffic since both links are connected to the same ENCS internal switch. If you do not isolate the traffic, both LAN traffic and transit link will flow through the same internal switch on the Cisco ENCS.

**Enable Port Tracking and Virtual NIC Update**

The configured VNICS tracks the state of the ports based on the PNICs notifications. To verify the state of the port, use `show interface` or `ethtool` commands. You can also use commands specific to the VM, that displays the interface link state.

To configure track state on GE0-0 & GE0-1:

```
configure terminal
pnic GE0-0 track-state ROUTER 1
end
```
ENCS-Left# **support show ifconfig GE0-0**

GE0-0: flags=4611<UP,BROADCAST,ALLMULTI,MULTICAST> mtu 9216
     ether 70:db:98:c3:df:28 txqueuelen 1000 (Ethernet)

To configure track state on switch port:

```bash
configure terminal
switch interface gigabitEthernet 1/3 track-state FIREWALL 4
end
```

ENCS-Left# **show vm_lifecycle deployments FIREWALL**

Name: FIREWALL
Deployment Name : FIREWALL
VM Group Name : FIREWALL
State: ALIVE
Internal State: VM_INERT_STATE
Bootup Time: -1
Image: Palo-Alto-8.1.3.tar.gz
Flavor: VM-100

VCPUs | Memory(MB) | Disk(MB) |
--- | --- | --- |
2 | 7168 | 61440 |

Low Latency: true

<table>
<thead>
<tr>
<th>VCPU</th>
<th>CPU</th>
<th>CORE</th>
<th>SOCKET</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NICID</th>
<th>VNIC</th>
<th>NETWORK</th>
<th>IP</th>
<th>MAC-ADDRESS</th>
<th>MODEL</th>
<th>PORT-FORWARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>vnic6</td>
<td>mgmt-net</td>
<td>-</td>
<td>52:54:00:2b:72:d2</td>
<td>virtio</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>vnic7</td>
<td>Untrust</td>
<td>-</td>
<td>52:54:00:eb:a3:e7</td>
<td>virtio</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>vnic8</td>
<td>HA1</td>
<td>-</td>
<td>52:54:00:f4:de:e5</td>
<td>virtio</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>vnic9</td>
<td>HA2</td>
<td>-</td>
<td>52:54:00:12:f8:21</td>
<td>virtio</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>vnic10</td>
<td>Trust</td>
<td>-</td>
<td>52:54:00:7a:6b:e9</td>
<td>virtio</td>
<td></td>
</tr>
</tbody>
</table>

ENCS-Left# **support show ifconfig vnic10**

vnic10: flags=4161<UP,BROADCAST,RUNNING,MULTICAST> mtu 9216
        inet6 fe80::fc54:ff:fe7a:6be9 prefixlen 64 scopeid 0x20<link>
        ether fe:54:00:7a:6b:e9 txqueuelen 4000 (Ethernet)

**Packet Flow for SD-Branch HA**

This section explains high-level packet flow in non-failure and failure cases.
In the non-failure case, both ENCS devices are Active, up and running:

- LAN to WAN through the ENCS1 Firewall and ENCS1 Router
- LAN to WAN through the ENCS1 Firewall and ENCS2 Router
- WAN to LAN through ENCS1 Router and ENCS1 Firewall
- WAN to LAN through ENCS2 Router and ENCS1 Firewall
Failure Case

Following are failure cases that a router must be designed and configured to adapt:

- One or more of the monitored interfaces fail. (Link Monitoring)
- One or more of the destinations specified on the firewall cannot be reached. (Path Monitoring)
- The firewall does not respond to heartbeat polls. (Heartbeat Polling and Hello messages)
# Configuration Examples and Usage Description

## ENCS-Left and ENCS-Right with Same Config

```plaintext
networks network wan-net
  bridge wan-br
!
networks network HA1
  vlan [ 126 ]
  trunk false
  bridge lan-br
!
networks network HA2
  vlan [ 127 ]
  trunk false
  bridge lan-br
!
networks network Trust
  vlan [ 128 ]
  bridge lan-br
!
networks network Untrust
  vlan [ 998 ]
  bridge lan-br
!
networks network mgmt-net
  vlan [ 100 ]
  trunk false
  bridge lan-br
!
networks network pt-2-pt
  vlan [ 996 997 ]
  bridge lan-br
!
  vlan 1
!
  vlan 100
!
  vlan 126
!
  vlan 127
!
  vlan 128
!
  vlan 996
!
  vlan 997
!
  vlan 998
!
  spanning-tree enable
  spanning-tree mode mst
  spanning-tree mst 2 priority 61440
  spanning-tree mst configuration
    name mst_LAN
    instance 1 vlan 996-998
    instance 2 vlan 100,126-128
!
```

<table>
<thead>
<tr>
<th>Description or Reasons for configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>In a HA design involving a router or Firewall, there are 3 to 6 paths required. ENCS platform has 2 WAN facing ports and 8 LAN facing ports.</td>
</tr>
<tr>
<td>• WAN facing ports are reserved for connection to WAN circuits.</td>
</tr>
<tr>
<td>• LAN facing ports are the only set of available ports for creating the 3 to 6 path required.</td>
</tr>
<tr>
<td>VLAN must be explicitly created before they are used in the interfaces.</td>
</tr>
<tr>
<td>Enable MSTP. For MST group 2 carrying “Traffic towards/from LAN”, force the External Switch to become the ROOT using the “mst &lt;group&gt; priority &lt;value&gt;” CLI. The Higher the value, lower the chance of becoming spanning-tree ROOT.</td>
</tr>
<tr>
<td>“priority” configuration is NOT required for the MST group 1 carrying “Traffic between VNFs”. There is NO loop possibility for MST group 1 VLANs.</td>
</tr>
</tbody>
</table>
## ENCS-Left and ENCS-Right with Same Config

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description or Reasons for configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfvis# show running-config switch switch interface gigabitEthernet1/1 no shutdown channel-group 1 mode auto ! interface gigabitEthernet1/2 no shutdown channel-group 1 mode auto ! switch interface port-channel1 negotiation auto no shutdown spanning-tree mst 1 cost 200000000 spanning-tree mst 2 cost 200000000 switchport mode trunk switchport trunk native vlan 1 switchport trunk allowed vlan 100,126-128,996-998 !</td>
<td>For the back-to-back ENCS connection, link redundancy is achieved using port-channel configuration. Interfaces that are belong to a port-channel group use configuration from “interface port-channel x” Goal is to prefer the direct links from ENCS to the External Switch for “Traffic towards/from LAN”. In ENCS back-to-back connection, Spanning tree cost is HIGH for MST group carrying “Traffic towards/from LAN”. This config will block one of the ENCS back-to-back interfaces for breaking the loop for MST group carrying “Traffic towards/from LAN”.</td>
</tr>
</tbody>
</table>

### Status of MST instances.

For MST instance 1, “Traffic between the VNFs”, back-to-back portchannel link is root and forwarding state. For MST instance 2, “Traffic from/towards the LAN”, links connected to External Switch are in forwarding state, path via back-to-back portchannel link is “Blocking state”. If one of the Links fail between ENCS and External switch, portchannel path for MST instance 2 will be unblocked.
ENCS-Left# show switch vlan detailed

<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>VLAN NAME</th>
<th>PORTS</th>
<th>UNTAGGED PORTS</th>
<th>CREATED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>g10,g14-6,te2,po2-4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>g13,te2,po1</td>
<td>g17</td>
<td>Manual</td>
</tr>
<tr>
<td>126</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>127</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>128</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>996</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>997</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>998</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
</tbody>
</table>

ENCS-Left# show switch spanning-tree mstp summary

spanning-tree mstp summary ist-info summary
admin-status enabled
operation-mode MSTP
Port-Cost-Method long
Loopback-guard disabled
Priority 32768
Address 70:db:98:c3:df:14
Hello-Time 2
Max-Age 20
Forward-Delay 15
Priority 32768
Address 70:db:98:c3:df:a0
Hello-Time 2
Max-Age 20
Forward-Delay 15
......

ENCS-Right# show switch vlan detail

<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>VLAN NAME</th>
<th>PORTS</th>
<th>UNTAGGED PORTS</th>
<th>CREATED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>g10,g14-6,te2,po2-4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>g13,te2,po1</td>
<td>g17</td>
<td>Manual</td>
</tr>
<tr>
<td>126</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>127</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>128</td>
<td></td>
<td>g13,te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>996</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>997</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
<tr>
<td>998</td>
<td></td>
<td>te2,po1</td>
<td>None</td>
<td>Manual</td>
</tr>
</tbody>
</table>

ENCS-Right# show switch spanning-tree mstp summary

spanning-tree mstp summary ist-info summary
admin-status enabled
operation-mode MSTP
Port-Cost-Method long
Loopback-guard disabled
Priority 32768
Address 70:db:98:c3:df:14
Hello-Time 2
Max-Age 20
Forward-Delay 15
Priority 32768
Address 70:db:98:c3:df:a0
Hello-Time 2
Max-Age 20
Forward-Delay 15
......
<table>
<thead>
<tr>
<th>INST ID</th>
<th>PORT</th>
<th>STATE</th>
<th>NBR</th>
<th>COST</th>
<th>STS ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>gi1/0</td>
<td>enabled</td>
<td>128.1</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>1</td>
<td>gi1/3</td>
<td>enabled</td>
<td>128.4</td>
<td>20000</td>
<td>forwarding designated</td>
</tr>
<tr>
<td>1</td>
<td>gi1/4</td>
<td>enabled</td>
<td>128.5</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>1</td>
<td>gi1/5</td>
<td>enabled</td>
<td>128.6</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>1</td>
<td>gi1/6</td>
<td>enabled</td>
<td>128.7</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>1</td>
<td>gi1/7</td>
<td>enabled</td>
<td>128.8</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>2</td>
<td>gi1/0</td>
<td>enabled</td>
<td>128.1</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>2</td>
<td>gi1/3</td>
<td>enabled</td>
<td>128.4</td>
<td>20000</td>
<td>forwarding root</td>
</tr>
<tr>
<td>2</td>
<td>gi1/4</td>
<td>enabled</td>
<td>128.5</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>2</td>
<td>gi1/5</td>
<td>enabled</td>
<td>128.6</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>2</td>
<td>gi1/6</td>
<td>enabled</td>
<td>128.7</td>
<td>2000000</td>
<td>disabled</td>
</tr>
<tr>
<td>2</td>
<td>gi1/7</td>
<td>enabled</td>
<td>128.8</td>
<td>2000000</td>
<td>disabled</td>
</tr>
</tbody>
</table>

From the above summary output, MST instances indicates ID and associated VLAN, and then displays all interfaces as part of VLAN instances. This behaviour differs from the way MST instances are displayed on other Cisco switching platforms.
External Switch MST Configuration

It is recommended that VLAN 996-998 is not allowed through the interfaces connecting to ENCS-Left and ENCS-Right. As a result, the external switch MSTP does not participate for VLAN 996-998.

Note

Table 30:

| VLANs carrying “Traffic between the VNFs” are NOT sent to the External Switch. |
| MST instance priority and MST link COST are kept default in the External Switch. |
| MST Priority and COST Configuration in ENCS ensure the External switch is the root and the Interfaces in the External switch connecting to ENCS are in Forwarding state. |

| VLANscarryingtrafficbetweentheVNFS|arenotusedinexternalswitchandnotconfiguredinanyinterface. |

Switch#show spanning-tree mst detail

| Bridge address f0b2.e556.e480 priority 32768 (32768 sysid 0) |
| Root address 70db.98c3.df14 priority 32768 (32768 sysid 0) |
| Port G11/0/2 path cost 0 |
| Regional Root address 70db.98c3.df14 internal cost 20000 rem hops 19 |
| Operational hello time 2, forward delay 15, max age 20, txholdcount 6 |
| Configured hello time 2, forward delay 15, max age 20, max hops 20 |

GigabitEthernet1/0/1 of MST0 is alternate blocking

Port info port id 128.1 priority 128 cost 20000
Designated root address 70db.98c3.df14 priority 32768 cost 0
Design. regional root address 70db.98c3.df14 priority 32768 cost 10000
Designated bridge address 70db.98c3.dfa0 priority 32768 port id 128.4
Timers: message expires in 5 sec, forward delay 0, forward transitions 0
Bpdus sent 27905, received 31061

GigabitEthernet1/0/2 of MST0 is root forwarding

Port info port id 128.2 priority 128 cost 20000
Designated root address 70db.98c3.df14 priority 32768 cost 0
Design. regional root address 70db.98c3.df14 priority 32768 cost 0
Designated bridge address 70db.98c3.df14 priority 32768 port id 128.4
Timers: message expires in 5 sec, forward delay 0, forward transitions 1
Bpdus sent 27904, received 31070
### MST2 vlans mapped: 100,126-128

Bridge address f0b2.e556.e480 priority 32770 (32768 sysid 2)

Root this switch for MST2

GigabitEthernet1/0/1 of MST2 is designated forwarding
Port info port id 128.1 priority 128 cost 20000
Designated root address f0b2.e556.e480 priority 32770 cost 0
Designated bridge address f0b2.e556.e480 priority 32770 port id 128.1
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 27905, received 31061

GigabitEthernet1/0/2 of MST2 is designated forwarding
Port info port id 128.2 priority 128 cost 20000
Designated root address f0b2.e556.e480 priority 32770 cost 0
Designated bridge address f0b2.e556.e480 priority 32770 port id 128.2
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 27904, received 31070

Switch#

### CIMC Secure Overlay Support

#### Table 31: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIMC Secure Overlay Support</td>
<td>NFVIS 4.4.1</td>
<td>This feature enables you to recover NFVIS when it becomes unresponsive. CIMC periodically monitors NFVIS health and when NFVIS health check fails, CIMC establishes a secure tunnel to the remote location. You can then log in to CIMC from the remote site to troubleshoot and recover NFVIS.</td>
</tr>
</tbody>
</table>

The CIMC secure overlay feature is a disaster recovery feature to recover an unresponsive NFVIS. NFVIS is managed from a remote location using a secure (IPsec) tunnel. If NFVIS hangs and does not recover by itself, you must reboot or reinstall NFVIS. With this feature, CIMC periodically monitors NFVIS health. CIMC gets NFVIS secure tunnel details when a secure tunnel is set up from NFVIS to the remote location. When NFVIS health check fails, CIMC waits for an hour by default for NFVIS to recover by itself. If NFVIS does not recover, CIMC establishes a secure tunnel to the remote site.

CIMC can access the remote site at the same NFVIS external IP address. At the remote site, you can log in to CIMC and do the necessary troubleshooting to recover NFVIS. After NFVIS is back up, the CIMC secure tunnel is broken and secure tunnel is established between NFVIS and the remote site again.

---

**Note**

For this feature to work:

- Secure IPSec tunnel should be configured on NFVIS.
- The feature should be enabled in CIMC during deployment.
The following example shows how to enable secure overlay feature on CIMC:

```
ENCS# scope ipsec-tunnel
ENCS/ipsec-tunnel# show detail
    Enable Failover: no
    NFVIS Down Timeout: 3600
    Tunnel Status: NO_TUNNEL
    Seconds to Failover: 0
ENCS/ipsec-tunnel# set ?
    cli
        CLI options
    nfvis-down-timeout
        Number of seconds CIMC waits before starting ipsec tunnel to remote
    tunnel-failover-enable
        Enables failover of ipsec tunnel when NFVIS is down
ENCS/ipsec-tunnel# set tunnel-failover-enable yes
ENCS/ipsec-tunnel# set nfvis-down-timeout 3600
ENCS/ipsec-tunnel# commit
ENCS/# show detail
    Enable Failover: yes
    NFVIS Down Timeout: 3600
    Tunnel Status: NO_TUNNEL
    Seconds to Failover: 0
ENCS/ipsec-tunnel# show
    Enable Failover NFVIS Down Timeout Tunnel Status Seconds to Failover
    -------------- ------------------ ---------------------- -------------------
    yes 3600 TUNNEL_ESTABLISHED 0
```

The CIMC tunnel status can be one of the following:

- **TUNNEL_ESTABLISHED** = all ok and remote can login to CIMC
- **TUNNEL_CONFIG_MISSING** = IPsec config was not transferred to CIMC
- **TUNNEL_ERROR** = Runtime error
- **UNKNOWN_ERROR**

## CIMC TACACS Support

### Table 32: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIMC TACACS Support</td>
<td>NFVIS 4.2.1</td>
<td>CIMC TACACS configuration support enabled on NFVIS.</td>
</tr>
</tbody>
</table>

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

On the TACACS+ server, ensure you configure Cisco attribute-value (AV) pair privilege level (priv-lvl) for Cisco Integrated Management Controller (CIMC) service for the minimum privilege level of administrators and operators.
In CIMC TACACS configuration, privilege level 14 is assigned to the administrator role, and privilege level 9 is assigned to the operator role. This means a user with privilege level 14 or higher has admin privileges when the user logs into the system, and a user with privilege level 9 or higher has all privileges of an operator at the time of login.

Privilege level below 9 has the read-only privileges.

These two are optional arguments. By default admin-priv is 15 and oper-priv is 11.

To configure a TACACS server on CIMC:

```
Server /# scope tacacs
Server /tacacs# set tacacs-server ip-address
Server /tacacs*# set tacacs-key <keystring>
Server /tacacs*# set tacacs-enable yes
Server /tacacs*# set admin-priv 12
Server /tacacs*# set oper-priv 5
Server /tacacs*# commit
Server /tacacs# show detail
```

To verify the TACACS+ server configuration:

```
Server/tacacs# show detail
tacacs Settings:
  Server domain name or IP address:  
  Enable tacacs: yes
  shared-secret key: ******
  admin-priv: 14
  oper-priv: 10
```

For more information about this feature, see TACACS+ Server.

### LTE PIM Module on Cisco Catalyst 8200 UCPE

The LTE PIM modules provide 4G LTE cellular connectivity on the Cisco Catalyst 8200 UCPE platform. The following LTE PIM modules are supported on the Cisco Catalyst 8200 UCPE platform:

- P-LTE-GB
- P-LTE-IN
- P-LTE-US
- P-LTE-VZ
- P-LTEA-EA
- P-LTEA-LA

#### Limitations

The following are the limitations when you use LTE PIM modules on Cisco Catalyst 8200 UCPE platform:

- Only one SIM card is supported at a time. If both SIMs are present, SIM in slot 0 is used. To use SIM in slot 1, slot 0 should be empty.
- IPv6 is not supported.
Remote Access to Cisco Catalyst 8200 UCPE through LTE PIM Module

Remote access to Cisco Catalyst 8200 UCPE platform through LTE PIM module is supported. This requires users to obtain cellular SIM with static IP address from service providers. After the cellular SIM with static IP address is obtained, you can use remote protocol like SSH to remotely access the Cisco Catalyst 8200 UCPE platform.

MacBook:~ $ ssh admin@192.168.100.101
admin@192.168.100.101's password:

Cisco Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS Version: 4.4.1-FC2

Copyright (c) 2015-2020 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

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

admin connected from 128.168.241.172 using ssh on C8200-NFVIS
C8200-NFVIS#

You can also access Cisco Catalyst 8200 UCPE through a portal.

Show Commands

The following commands are used to display LTE PIM module's information.

The following example shows all cellular information using the show cellular command:

C8200-NFVIS# show cellular

C8200-NFVIS# show cellular
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware PRI SKU ID</td>
<td>1102526</td>
</tr>
<tr>
<td>Hardware PRI Version</td>
<td>002.079_001</td>
</tr>
<tr>
<td>Hardware Carrier</td>
<td>VERIZON</td>
</tr>
<tr>
<td>Hardware OEM PRI Version</td>
<td>000.016</td>
</tr>
<tr>
<td>Hardware Modem Status</td>
<td>Modem Online</td>
</tr>
<tr>
<td>Network Current System Time</td>
<td>2020/11/15/23/35/12</td>
</tr>
<tr>
<td>Network Network Selection Mode</td>
<td>AUTO</td>
</tr>
<tr>
<td>Network Current Service Status</td>
<td>Packet switched</td>
</tr>
<tr>
<td>Network Mobile Country Code (MCC)</td>
<td>311</td>
</tr>
<tr>
<td>Network Mobile Network Code (MNC)</td>
<td>480</td>
</tr>
<tr>
<td>Network Current Roaming Status</td>
<td>HOME</td>
</tr>
<tr>
<td>Network Network</td>
<td>Verizon</td>
</tr>
<tr>
<td>Network Packet Switch domain (PS) state</td>
<td>Attached</td>
</tr>
<tr>
<td>Network EMM State</td>
<td>Registered</td>
</tr>
<tr>
<td>Network EMM Sub state</td>
<td>Normal-Service</td>
</tr>
<tr>
<td>Network EMM Connection State</td>
<td>2</td>
</tr>
<tr>
<td>Network Tracking Area Code (TAC)</td>
<td>7936</td>
</tr>
<tr>
<td>Network Cell ID</td>
<td>7962134</td>
</tr>
<tr>
<td>Network Network MTU</td>
<td>1428</td>
</tr>
<tr>
<td>Radio-detai Radio Power Mode</td>
<td>online</td>
</tr>
<tr>
<td>Radio-details Radio Access Technology (RAT) Preference</td>
<td>AUTO</td>
</tr>
<tr>
<td>Radio-details Radio Access Technology (RAT) Selected</td>
<td>LTE</td>
</tr>
<tr>
<td>Radio-details LTE RX Channel Number</td>
<td>2075</td>
</tr>
<tr>
<td>Radio-details LTE TX Channel Number</td>
<td>20075</td>
</tr>
<tr>
<td>Radio-details LTE Band</td>
<td>4</td>
</tr>
<tr>
<td>Radio-details LTE Bandwidth</td>
<td>15 MHz</td>
</tr>
<tr>
<td>Radio-details Current RSSI</td>
<td>-73 dBm</td>
</tr>
<tr>
<td>Radio-details Current RSRP</td>
<td>-103 dBm</td>
</tr>
<tr>
<td>Radio-details Current RSRQ</td>
<td>-10 dB</td>
</tr>
<tr>
<td>Radio-details Current SNR</td>
<td>5 dB</td>
</tr>
<tr>
<td>Radio-details PCI</td>
<td>169</td>
</tr>
<tr>
<td>Profile 3, Packet Session Status</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>Call end mode</td>
<td>None</td>
</tr>
<tr>
<td>Session disconnect reason type</td>
<td>None(0)</td>
</tr>
<tr>
<td>Session disconnect reason</td>
<td>None(0)</td>
</tr>
<tr>
<td>PDP 0</td>
<td></td>
</tr>
<tr>
<td>Back off timer is NOT running</td>
<td></td>
</tr>
<tr>
<td>Back off error count</td>
<td>0</td>
</tr>
<tr>
<td>Back off timer index</td>
<td>0</td>
</tr>
<tr>
<td>Back off timer array (in minutes)</td>
<td>0 1 1 1 5 10 15 30 60</td>
</tr>
<tr>
<td>Cellular interface back-off is not enforced</td>
<td></td>
</tr>
<tr>
<td>Period of back-off</td>
<td>0 minute(s)</td>
</tr>
<tr>
<td>Profile-details 1</td>
<td></td>
</tr>
<tr>
<td>PDP-Type IPv4v6</td>
<td></td>
</tr>
<tr>
<td>APN-Name  ims</td>
<td></td>
</tr>
<tr>
<td>state INACTIVE</td>
<td></td>
</tr>
<tr>
<td>PDP-Address 0.0.0.0</td>
<td></td>
</tr>
<tr>
<td>Authentication None</td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
<tr>
<td>Profile-details 2</td>
<td></td>
</tr>
<tr>
<td>PDP-Type IPv4v6</td>
<td></td>
</tr>
<tr>
<td>APN-Name vzwadmin</td>
<td></td>
</tr>
<tr>
<td>state INACTIVE</td>
<td></td>
</tr>
<tr>
<td>PDP-Address 0.0.0.0</td>
<td></td>
</tr>
<tr>
<td>Authentication None</td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
<tr>
<td>Profile-details 3</td>
<td></td>
</tr>
<tr>
<td>PDP-Type IPv4v6</td>
<td></td>
</tr>
<tr>
<td>APN-Name we01.VZWSTATIC</td>
<td></td>
</tr>
<tr>
<td>state ACTIVE</td>
<td></td>
</tr>
<tr>
<td>PDP-Address 192.168.100.101</td>
<td></td>
</tr>
<tr>
<td>Authentication None</td>
<td></td>
</tr>
</tbody>
</table>
The following examples shows the individual cellular information using the `show cellular 1/0 [connection | details | hardware | network | profile-details | radio-details | status]` command:

**connection**: Cellular connection information

```
C8200-NFVIS# show cellular 1/0 connection
Profile 3, Packet Session Status = ACTIVE
Call end mode = None
Session disconnect reason type = None(0)
Session disconnect reason = None(0)
PDN 0
Back off timer is NOT running
Back off error count = 0
Back off timer index = 0
Cellular interface back-off is not enforced
Period of back-off = 0 minute(s)
```

**details**: Cellular basic information

```
C8200-NFVIS# show cellular 1/0 details
details IPv4 address = 192.168.100.101
details Default Gateway Address = 192.168.100.102
details Interface subnet mask = 255.255.255.252
details Interface name = int-CELL-1-0
details Interface Status = up
```
details DNS IP Address = 198.224.173.135
details Modem Status = Call_Connected

C8200-NFVIS#

**hardware**: Cellular hardware information

```
C8200-NFVIS# show cellular 1/0 hardware
hardware Modem Firmware Version = SWI9X30C_02.33.03.00
hardware Device Model ID = EM7455
hardware International Mobile Subscriber Identity (IMSI) = 311480148702353
hardware International Mobile Equipment Identity (IMEI) = 356129070591075
hardware Integrated Circuit Card ID (ICCID) = 8914800001470156497
hardware Mobile Subscriber Intergrated Services Digital Network Number (MSISDN) = 4085550087
hardware Factory Serial Number (FSN) = LF911203620410
hardware Current Modem Temperature = 37 deg C
hardware PRI SKU ID = 1102526
hardware PRI Version = 002.079_001
hardware Carrier = VERIZON
hardware OEM PRI Version = 000.016
hardware Modem Status = Modem Online
C8200-NFVIS#
```

**network**: Cellular network information

```
C8200-NFVIS# show cellular 1/0 network
network Current System Time = 2020/11/15/23/39/52
network Network Selection Mode = AUTO
network Current Service Status = Packet switched
network Mobile Country Code (MCC) = 311
network Mobile Network Code (MNC) = 480
network Current Roaming Status = HOME
network Network = Verizon
network Packet Switch domain (PS) state = Attached
network EMM State = Registered
network EMM Sub state = Normal-Service
network EMM Connection State = 2
network Tracking Area Code (TAC) = 7936
network Cell ID = 7962134
network Network MTU = 1428
C8200-NFVIS#
```

**profile-details**: Cellular profile details

```
C8200-NFVIS# show cellular 1/0 profile-details
profile-details 1
PDP-Type IPv4v6
APN-Name ims
state INACTIVE
PDP-Address 0.0.0.0
Authentication None
Username
Password
profile-details 2
PDP-Type IPv4v6
APN-Name vzwadmin
state INACTIVE
PDP-Address 0.0.0.0
Authentication None
Username
```
Profile Details 3
- PDP-Type: IPv4v6
- APN-Name: we01.VZWSTATIC
- State: ACTIVE
- PDP-Address: 192.168.100.101
- Authentication: None
- Username: 
- Password: 

Profile Details 4
- PDP-Type: IPv4v6
- APN-Name: vzwapp
- State: INACTIVE
- PDP-Address: 0.0.0.0
- Authentication: None
- Username: 
- Password: 

Profile Details 5
- PDP-Type: IPv4v6
- APN-Name: vzw800
- State: INACTIVE
- PDP-Address: 0.0.0.0
- Authentication: None
- Username: 
- Password: 

Profile Details 6
- PDP-Type: IPv4v6
- APN-Name: vzwclass6
- State: INACTIVE
- PDP-Address: 0.0.0.0
- Authentication: None
- Username: 
- Password: 

Radio Details: Cellular radio details

C8200-NFVIS# show cellular 1/0 radio-details
- Radio Power Mode: online
- Radio Access Technology (RAT) Preference: AUTO
- Radio Access Technology (RAT) Selected: LTE
- LTE RX Channel Number: 2075
- LTE TX Channel Number: 20075
- LTE Band: 4
- LTE Bandwidth: 15 MHz
- LTE Current RSSI: -72 dBm
- LTE Current RSRP: -103 dBm
- LTE Current RSRQ: -17 dB
- LTE Current SNR: 7 dB
- PCI: 169

Status: Cellular status details

C8200-NFVIS# show cellular 1/0 status
- Firmware Upgrade Status: "Not Initiated"
- DM Logging Status: Stopped

SIM
- ID | SIM Status
- 0  | OK
- 1  | Not inserted
The following example shows the Tx and Rx information on the LTE PIM module using the `show bridge-settings cellular-br` command:

```plaintext
C8200-NFVIS# show bridge-settings cellular-br
bridge-settings cellular-br
  ip-info interface cellular-br
  ip-info ipv4_address 192.168.100.101
  ip-info netmask 255.255.255.252
  ip-info link-local ipv6 address fe80::7c88:e1ff:de8c:bbc5
  ip-info link-local ipv6 prefixlen 64
  ip-info global ipv6 address ::
  ip-info global ipv6 prefixlen 0
  ip-info mac_address 6f:82:d1:8e:bf:5c
  ip-info mtu 1428
  ip-info txqueuelen 1000
  stats rx_packets 13295
  stats rx_bytes 979420
  stats rx_errors 0
  stats rx_dropped 0
  stats rx_overruns 0
  stats rx_frame 0
  stats tx_packets 11529
  stats tx_bytes 8147515
  stats tx_errors 0
  stats tx_dropped 0
  stats tx_overruns 0
  stats tx_carrier 0
  stats tx_collisions 0
  vlan tag untagged
dhcp disabled
dhcp-ipv6 disabled
slaac-ipv6 disabled
dpdk-enabled false
C8200-NFVIS#
```

The following example shows how to verify the available PIM module:

```plaintext
C8200-UCPE-1N8# show chassis compute inventory pim-modules
  POWER
  SLOT/BAY/PORT PID TYPE DESCRIPTION STATUS
  ------------------------------------------------------
  0/1/0 P-LTE-US 0x4a10 - on
C8200-UCPE-1N8#
```

```plaintext
C8200-UCPE-1N8# show chassis
chassis serial FOC241106RG
chassis pid C8200-UCPE-1N8
chassis uuid 4ce1767c-05d0-0000-2903-0f0f000a0b05
chassis bios-version C8200-UCPE_1.0.040920201359
chassis hardware-version V01
chassis compute inventory cpu units CPU0
vendor "Intel(R) Corporation"
family "Pentium 4"
thread-count 8
version "Intel(R) Atom(TM) CPU C3758 @ 2.20GHz"
speed 2200
core-count 8
status "Populated, Enabled"
signature "Type 0, Family 6, Model 95, Stepping 1"
chassis compute inventory memory units DIMM0 "BANK 0"
```
Verifying LTE PIM Module Network Connectivity

After Cisco Catalyst 8200 UCPE boots up with a supported LTE PIM and a carrier SIM card, you can verify if the LTE PIM module is connected to the carrier network using the `show cellular 1/0 details` command.

The following example shows how to verify if the LTE PIM module is connected:

```
C8200-NFVIS# show cellular 1/0 details
details IPv4 address = 101.144.109.57 <-- IP address obtained
details Default Gateway Address = 101.144.109.58
details Interface subnet mask = 255.255.255.252
details Interface name = int-CELL-1-0
details Interface Status = up
details DNS IP Address = 101.224.173.135
details Modem Status = Call_Connected <-- Cellular call connected
C8200-NFVIS#
```

For a successful network connection:

1. The LTE PIM module should be able to obtain an IP address.
   - Example here is 101.144.109.57.
   - If the SIM card has a dynamic IP address, this IP address would change over time. This works the same way as dynamic IP address assignment with wired network.

2. Modem status should show as `Call_Connected`.

3. Ping to any public IP network should be successful (8.8.8.8).

If any of the above are incorrect, see the troubleshooting section below.
LTE PIM Module LED

For information on LTE PIM module's LED colors and behaviors, see LTE LED Details.

Online Insertion and Removal (OIR) of LTE PIM Module

You can replace the LTE PIM module without affecting the Cisco Catalyst 8200 UCPE platform system operations with the online insertion and removal (OIR) operation. OIR commands are issued before removing and after installing an LTE PIM. When you perform OIR, use an identical or another variation of LTE PIM module to replace the original one.

Command

```
chassis compute inventory pim-modules units 0/1/0 [reload | start | stop]
```

Removing the LTE PIM module without executing the `chassis compute inventory pim-modules units 0/1/0 stop` command is not supported.

Examples

```
C8200-NFVIS# chassis compute inventory pim-modules units 0/1/0 stop
Are you sure you want to stop this module? [no,yes] yes
C8200-NFVIS# chassis compute inventory pim-modules units 0/1/0 start
Are you sure you want to start this module? [no,yes] yes
C8200-NFVIS# show cellular 1/0 details
details IPv4 address = 0.0.0.0
details Default Gateway Address = 0.0.0.0
details Interface subnet mask = 0.0.0.0
details Interface name = int-CELL-1-0
details Interface Status = down
details DNS IP Address = 0.0.0.0
details Modem Status = Disconnected
C8200-NFVIS# show cellular 1/0 details
details IPv4 address = 192.168.100.101
```

Note

```
C8200-NFVIS# show chassis | begin SLOT
SLOT/BAY/PORT PRODUCT ID SERIAL REVISION NUMBER MAC ADDRESS
-------------------------------------------------------------------------
0/2/0 NIM-LTEA-EA - 0x1 - 00:5d:73:48:e4:80
C8200-NFVIS# show chassis | begin SLOT
SLOT/BAY/PORT PID TYPE DESCRIPTION STATUS
-------------------------------------------------------
0/1/0 P-LTEA-EA 0x4710 - off
```

C8200-NFVIS# chassis compute inventory pim-modules units 0/1/0 start
Are you sure you want to start this module? [no,yes] yes
C8200-NFVIS# chassis compute inventory pim-modules units 0/1/0 stop
C8200-NFVIS# show cellular 1/0 details
details IPv4 address = 192.168.100.101
details Default Gateway Address = 192.168.100.102
details Interface subnet mask = 255.255.255.252
details Interface name = int-CELL-1-0
details Interface Status = up
details DNS IP Address = 198.224.173.135
details Modem Status = Call_Connected
C8200-NFVIS#
C8200-NFVIS# show chassis | begin SLOT
SLOT/BAY/PORT PRODUCT ID SERIAL REVISION NUMBER MAC ADDRESS
-------------------------------------------------------------------------
0/2/0 NIM-LTEA-EA - 0x1 - 00:5d:73:48:e4:80
C8200-NFVIS#
C8200-NFVIS# show chassis | begin SLOT
SLOT/BAY/PORT PID TYPE DESCRIPTION STATUS
-------------------------------------------------------------------------
0/1/0 P-LTEA-EA 0x4710 - on
C8200-NFVIS#

Turning LTE PIM Module’s Radio On/Off

The following examples show how to turn the LTE PIM’s radio off or on using the cellular 1/0 modem-radio-off and cellular 1/0 modem-radio-on commands:

C8200-NFVIS#
C8200-NFVIS# cellular 1/0 modem-radio-off
C8200-NFVIS#
C8200-NFVIS# cellular 1/0 modem-radio-on

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
Resetting Modem on the LTE PIM Module

The following example shows how to reset the LTE PIM's modem using the `cellular 1/0 modem-reset` command:

```plaintext
C8200-NFVIS# show cellular 1/0 hardware
hardware Modem Firmware Version = SWI9X30C_02.33.03.00
hardware Device Model ID = EM7455
hardware International Mobile Subscriber Identity (IMSI) = 311480148702353
hardware International Mobile Equipment Identity (IMEI) = 356129070591075
hardware Integrated Circuit Card ID (ICCID) = 8914800001470156497
hardware Mobile Subscriber Intergrated Services Digital Network Number (MSISDN) = 4083180087
hardware Factory Serial Number (FSN) = LF911203620410
hardware Current Modem Temperature = 36 deg C
hardware PRI SKU ID = 1102526
hardware PRI Version = 002.079_001
hardware Carrier = VERIZON
hardware OEM PRI Version = 000.016
hardware Modem Status = Modem Online
C8200-NFVIS# cellular 1/0 modem-reset
C8200-NFVIS# show cellular 1/0 hardware
hardware Modem Firmware Version = None
hardware Device Model ID = None
hardware International Mobile Subscriber Identity (IMSI) = None
hardware International Mobile Equipment Identity (IMEI) = None
hardware Integrated Circuit Card ID (ICCID) = None
hardware Mobile Subscriber Intergrated Services Digital Network Number (MSISDN) = None
hardware Factory Serial Number (FSN) = None
hardware Current Modem Temperature = None
hardware PRI SKU ID = None
hardware PRI Version = None
hardware Carrier = None
hardware OEM PRI Version = None
hardware Modem Status = None <-- modem is down
C8200-NFVIS# show cellular 1/0 hardware
hardware Modem Firmware Version = SWI9X30C_02.33.03.00
hardware Device Model ID = EM7455
hardware International Mobile Subscriber Identity (IMSI) = 311480148702353
hardware International Mobile Equipment Identity (IMEI) = 356129070591075
hardware Integrated Circuit Card ID (ICCID) = 8914800001470156497
hardware Mobile Subscriber Intergrated Services Digital Network Number (MSISDN) = 4083180087
hardware Factory Serial Number (FSN) = LF911203620410
hardware Current Modem Temperature = 36 deg C
hardware PRI SKU ID = 1102526
hardware PRI Version = 002.079_001
hardware Carrier = VERIZON
hardware OEM PRI Version = 000.016
hardware Modem Status = Modem Online <-- modem back up
C8200-NFVIS#
```
Modifying Profile on the LTE PIM Module

The following examples show how to create and delete LTE PIM's cellular profile using the `cellular profile create [apn-name | authentication | interface-id | pdp-type | profile-id]` and `cellular profile delete [apn-name | authentication | interface-id | pdp-type | profile-id]` commands:

```
C8200-NFVIS# show cellular 1/0 profile-details | begin "profile-details 7"
C8200-NFVIS# <-- cellular profile does not exist
C8200-NFVIS#

C8200-NFVIS# cellular profile create apn-name test-apn authentication None interface-id 1/0 pdp-type IPv4 profile-id 7
resp Profile Management Cmd Executed successfully
C8200-NFVIS#
C8200-NFVIS#

C8200-NFVIS# show cellular 1/0 profile-details | begin "profile-details 7"
profile-details 7 <-- profile created
  PDP-Type IPv4
  APN-Name test-apn
  state INACTIVE
  PDP-Address 0.0.0.0
  Authentication None
  Username
  Password
C8200-NFVIS#
C8200-NFVIS#

C8200-NFVIS# cellular profile delete apn-name test-apn authentication None interface-id 1/0 pdp-type IPv4 profile-id 7
resp Profile Management Cmd Executed successfully
C8200-NFVIS#
C8200-NFVIS#
C8200-NFVIS# show cellular 1/0 profile-details | begin "profile-details 7"
C8200-NFVIS# <-- profile deleted
C8200-NFVIS#
```

DM Log Collection on LTE PIM Module

Diagnostic Monitor (DM) is a Qualcomm proprietary protocol. Diagnostic software tools, such as Sierra Wireless SwiLog and Qualcomm QXDM, are based on DM protocol. These tools can be used to capture data transactions between the modem and the network over the RF interface, which makes them useful tools for troubleshooting 3G and 4G data connectivity or performance issues.

To start DM log collection

The following example shows how to start DM log collection using the `dm-log enable` command:

```
C8200-NFVIS# config terminal
Entering configuration mode terminal
C8200-NFVIS(config)# controller cellular 1/0
C8200-NFVIS(config-cellular-1/0)# dm-log enable
C8200-NFVIS(config-cellular-1/0)# commit
Commit complete.
C8200-NFVIS(config-cellular-1/0)# end
C8200-NFVIS#
```

---

**Note**

DM log rotation is enabled by default.

To stop DM log collection

The following example shows how to stop DM log collection using the `no dm-log enable` command:

```
C8200-NFVIS(config-cellular-1/0)# no dm-log enable
C8200-NFVIS(config-cellular-1/0)# commit
Commit complete.
C8200-NFVIS(config-cellular-1/0)# end
C8200-NFVIS#
```
C8200-NFVIS# config terminal
Entering configuration mode terminal
C8200-NFVIS(config)# controller cellular 1/0
C8200-NFVIS(config-cellular-1/0)# no dm-log enable
Commit complete.
C8200-NFVIS(config-cellular-1/0)# end
C8200-NFVIS#

**To view DM log collection status**

The following example shows how to view the status of DM log collection using the `show cellular 1/0 status` command:

C8200-NFVIS# show cellular 1/0 status
status Firmware Upgrade Status = "Not Initiated"
status DM Logging Status = Started <-- DM log enabled
SIM
ID   SIM STATUS
----------
0     OK
1     Not inserted

C8200-NFVIS#

C8200-NFVIS# show cellular 1/0 status
status Firmware Upgrade Status = "Not Initiated"
status DM Logging Status = Stopped <-- DM log not enabled
SIM
ID   SIM STATUS
----------
0     OK
1     Not inserted

C8200-NFVIS#

**To retrieve the collected DM log file**

Download the system's tech-support file.

**To configure DM log non-default filter**

- SCP the DM log filter onto C8200-UCPE-1N8.
  
  `scp <username>@<server_ip>:<dm_log_filter_file> intdatastore:<dm_log_filter_file>`

  Example:
  `scp admin@192.19.145.241:/home/gpsmask.sqf intdatastore:gpsmask.sqf`

- Configure specific DM log filter.
  
  `dm-log filter-path /data/intdatastore/uploads/<dm_log_filter_file>`

**Note**

To enable DM log collection with DM log filter, need to issue “commit” twice:

1st time: after configure DM log filter
2nd time: after enable DM log collection

Example:
C8200-NFVIS# config terminal
   Entering configuration mode terminal
C8200-NFVIS(config)# controller cellular 1/0
C8200-NFVIS(config-cellular-1/0)# dm-log filter-path
   /data/intdatastore/uploads/gpsmask.sqf
C8200-NFVIS(config-cellular-1/0)# commit <-- need to execute before enable DM log
   Commit complete.
C8200-NFVIS(config-cellular-1/0)# dm-log enable
C8200-NFVIS(config-cellular-1/0)# commit <-- need to execute again
   Commit complete.
C8200-NFVIS(config-cellular-1/0)# end
C8200-NFVIS#

When you use an invalid filter or mistype the filter's name, a DM log file of 0 byte size will be created.

**To remove DM log filter**

The following example shows how to remove DM log filter using the *no dm-log filter-path* command:

C8200-NFVIS# config term
   Entering configuration mode terminal
C8200-NFVIS(config)# controller cellular 1/0
C8200-NFVIS(config-cellular-1/0)# no dm-log filter-path
C8200-NFVIS(config-cellular-1/0)# commit
   Commit complete.
C8200-NFVIS(config-cellular-1/0)# end
C8200-NFVIS#

**Firmware Upgrade LTE PIM Module's Modem**

From time to time, the LTE PIM's modem requires firmware upgrade to obtain fixes for cellular modem's issues. Firmware upgrade on the LTE PIM module can be done with the following procedure.

1. Obtain firmware and/or OEM PRI files and put them into a server the C8200-UCPE-1N8 can SCP from.


   Use only Cisco certified firmware. Using a firmware version not certified by Cisco may impact the wireless service provider network adversely.

2. SCP the firmware or OEM PRI files from the external source to /data/intdatastore/uploads folder of the C8200-UCPE-1N8.

   Example:

   ```
   scp admin@192.19.145.241:/fw_folder/WP76xx_02.28.03.01_ATT_002.071_000.spk
   intdatastore:wp7603_02.28.03.01_ATT.spk
   ```
3. Firmware upgrade the LTE PIM's modem.

```
cellular 1/0 upgrade-firmware path /data/intdatastore/uploads/
```

**Note**

- Firmware upgrade process will take up to 15 minutes. During this time the modem will be unusable. Please do not remove power or reload the C8200-UCPE-1N8 during the firmware upgrade process.

- Firmware upgrade status can be seen with `show cellular 1/0 status` command.

4. Delete the firmware or OEM PRI files from the /data/intdatastore/uploads folder of the C8200-UCPE-1N8.

```
system file-delete file name /data/intdatastore/uploads/<fw_file>
```

**Note**

- There can only be 1 firmware/OEM PRI file in the /data/intdatastore/uploads folder for each firmware upgrade process.

- For EM74xx modems' firmware upgrade, only have 1 .CWE and 1 .NVU files in the folder.

- For WP76xx modems' firmware upgrade, only have 1 .SPK file in the folder.

- For EM74xx modems' and WP76xx modems' OEM PRI upgrade, only have 1 .NVU file in the folder.

---

### Enabling Debug to Troubleshooting LTE PIM Module

Enable debugs to troubleshoot cellular issues can be done with the following procedures:

1. Find out the current system logging level.

```
show system logging-level
```

```
C8200-NFVIS# show system logging-level
system logging-level configuration info
system logging-level operational warning
C8200-NFVIS#
```

2. Set logging level for both configuration and operational to debug.

```
system set-log level debug logtype all
```

```
C8200-NFVIS# system set-log level debug logtype all
C8200-NFVIS#
C8200-NFVIS# show system logging-level
system logging-level configuration debug
system logging-level operational debug
C8200-NFVIS#
```

3. Enable debug for cellular logging traces.

```
cellular debug eng-mode mode enable
```

```
C8200-NFVIS#
C8200-NFVIS# cellular debug eng-mode mode enable
C8200-NFVIS#
```

4. Disable debug for cellular logging traces.
cellular debug eng-mode mode disable

C8200-NFVIS#
cellular debug eng-mode mode disable
C8200-NFVIS#

5. Put system logging level back

system set-log level <critical | debug | error | info | warning> logtype <all | configuration | operational>

C8200-NFVIS#
system set-log level info logtype configuration
C8200-NFVIS#
system set-log level warning logtype operational
C8200-NFVIS#

To retrieve the debug log file

Download the system tech-support file.

Troubleshooting

Check the following use cases if the LTE PIM does not have cellular network connectivity:

- Verify that the LTE PIM module is inserted and recognized by the Cisco Catalyst 8200 UCPE. The following example shows how to verify the same using the command `show chassis` command:

```
C8200-NFVIS# show chassis
chassis serial  FGL2352LLC8
chassis pid    C8200-UCPE-1N8
chassis uuid   4c710d27-b960-0000-2903-0f0f000a0b05
chassis bios-version C8200-UCPE_1.01.042820201140
chassis hardware-version V01
chassis compute inventory cpu units CPU0
  vendor    "Intel(R) Corporation"
  family    "Pentium 4"
  thread-count 8
  version    "Intel(R) Atom(TM) CPU C3758 @ 2.20GHz"
  speed      2200
  core-count 8
  status     "Populated, Enabled"
  signature  "Type 0, Family 6, Model 95, Stepping 1"
chassis compute inventory memory units DIMM0 "BANK 0"
  capacity  8192
  channel-speed 2666
  channel-type DDR4
  type-detail "Synchronous Registered (Buffered)"
  manufacturer Smart
  serial      30304754
  asset-tag   "BANK 0 DIMM0 AssetTag"
  part-number SR5721G812APGDAME2
  data-width  72
chassis compute inventory memory units DIMM1 "BANK 0"
  manufacturer "NO DIMM"
  serial      "NO DIMM"
  asset-tag   "NO DIMM"
  part-number "NO DIMM"
```

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
CISCO  eMMC  HS-SD/MMC  31.79

<table>
<thead>
<tr>
<th>SLOT/BAY/PORT</th>
<th>PRODUCT ID</th>
<th>SERIAL</th>
<th>REVISION</th>
<th>NUMBER</th>
<th>MAC ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/2/0</td>
<td>NIM-LTEA-EA</td>
<td>0x1</td>
<td></td>
<td></td>
<td>00:5d:73:48:e4:80</td>
</tr>
</tbody>
</table>

**POWER**

<table>
<thead>
<tr>
<th>SLOT/BAY/PORT</th>
<th>PID</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/1/0</td>
<td>P-LTEA-EA</td>
<td>0x4710</td>
<td>on</td>
<td></td>
</tr>
</tbody>
</table>

<--- LTE PIM module is recognized and power on.

**MAX**

<table>
<thead>
<tr>
<th>PRODUCT</th>
<th>INPUT</th>
<th>OUTPUT</th>
<th>ID</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>90</td>
<td></td>
<td>--------------</td>
</tr>
</tbody>
</table>

C8200-NFVIS#

If the LTE PIM module is not recognized, try to reinsert the LTE PIM module.

- Verify if the SIM card is detected by the LTE PIM module. The following example shows how to verify the same using the `show cellular 1/0 status` command:

C8200-NFVIS# show cellular 1/0 status
status Firmware Upgrade Status = "Not Initiated"
status DM Logging Status = Stopped
SIM ID SIM STATUS
----------------------
0 OK
1 Not inserted

<-- SIM card is detected.

C8200-NFVIS#

If the SIM card is not detected, perform OIR of the LTE PIM module and re-seat the SIM card.

- Verify if the LTE PIM module is attached to the cellular network. The following example shows how to verify the same using the `show cellular 1/0 network` command:

C8200-NFVIS# show cellular 1/0 network
network Current System Time = 2020/12/20/6/56/2
network Network Selection Mode = AUTO
network Current Service Status = Packet switched
network Mobile Country Code (MCC) = 311
network Mobile Network Code (MNC) = 480
network Current Roaming Status = HOME
network Network = Verizon
network Packet Switch domain (PS) state = Attached

<-- LTE PIM module is attached to the Verizon network.

If the LTE PIM module is not attached to the cellular network, continue on to the next bullet.
• Verify the LTE module has cellular signal or reception. The following example shows how to verify the same using the `show cellular 1/0 radio-details` command:

```
C8200-NFVIS# show cellular 1/0 radio-details
radio-details Radio Power Mode = online
radio-details Radio Access Technology(RAT) Preference = AUTO
radio-details Radio Access Technology(RAT) Selected = LTE
radio-details LTE RX Channel Number = 2075
radio-details LTE TX Channel Number = 20075
radio-details LTE Band = 4
radio-details LTE Bandwidth = 15 MHz
radio-details Current RSSI = -71 dBm
radio-details Current RSRP = -105 dBm
radio-details Current RSRQ = -11 dB
radio-details Current SNR = 7 dB
radio-details PCI = 169
C8200-NFVIS#
```

<table>
<thead>
<tr>
<th>Table 33: Reference Radio Signal Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSSI (dBm)</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>Excellent</td>
</tr>
<tr>
<td>Good</td>
</tr>
<tr>
<td>Fair</td>
</tr>
<tr>
<td>Poor</td>
</tr>
</tbody>
</table>

• RSSI – Represents the entire received power including the wanted power from the serving cell as well as all co-channel power and other sources of noise.

• RSRP – The average power received from a single reference signal and its typical range is around -44dBm (good) to -140dBm (bad).

• RSRQ – Indicates quality of the received signal and its range is typically -19.5dB(bad) to -3dB (good).

• SNR – The signal-to-noise ratio of the given signal

If the cellular signal or reception is not good, try to place the cellular antenna at a location where it can pick up good signal or reception.

• Verify the LTE module has correct cellular profile configuration. The following example shows how to verify the same using the `show cellular 1/0 profile-details` command:

```
C8200-NFVIS# show cellular 1/0 profile-details
profile-details 1
PDP-Type IPv4v6
APN-Name ims
state INACTIVE
PDP-Address 0.0.0.0
Authentication None
Username
Password
profile-details 2
PDP-Type IPv4v6
```

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x
Before you configure, check with your cellular network carrier for the correct values.

- Verify cellular back-off feature is running or not. When this feature is running, it means that the LTE PIM module is performing necessary back-off operations in order to work smoothly with the provider network. The network provider side initiates this back-off request and the reason for such action depends on the code.

The following example shows how to verify if the LTE PIM module's cellular back-off feature is running or not, use the `show cellular 1/0 connection` command:

```bash
C8200-NFVIS# show cellular 1/0 connection
Profile 3, Packet Session Status = ACTIVE
Call end mode = None
Session disconnect reason type = None(0)
Session disconnect reason = None(0)
PON 0
Back off timer is NOT running  <-- cellular back-off feature is not running
Back off error count = 0
Back off timer index = 0
Back off timer array (in minutes) = 0 1 1 1 5 10 15 30 60
Cellular interface back-off is not enforced
Period of back-off = 0 minute(s)
C8200-NFVIS#
```
If the LTE PIM module has cellular back-off feature running, verify that the cellular profile is configured correctly. If it is configured correctly and you continue to see cellular back-off feature running, check with your cellular network carrier for the reason of back-off.

For more information on cellular back-off feature, see Cellular Back-Off.

• Verify that there is a routing table for LTE PIM module's cellular interface or cellular-br. The following example shows how to verify the same using the `support show router` command:

```
C8200-NFVIS# support show route
default via 101.144.109.58 dev cellular-br metric 10
  default route is through the cellular-br
10.20.0.0/24 dev int-mgmt-net-br proto kernel scope link src 10.20.0.1
166.144.109.56/30 dev cellular-br proto kernel scope link src 166.144.109.57
192.168.1.0/24 dev lan-br proto kernel scope link src 192.168.1.1
192.168.10.0/24 dev csxbr proto kernel scope link src 192.168.10.11
C8200-NFVIS#
```

If there are multiple default routes, ensure that the cellular-br metric is lower than the other default route.

• Troubleshooting your LTE PIM for cellular network connectivity can also be done through the LEDs located at the front of the LTE PIM module. For more information on the LED behaviors, see 4G LTE-Advanced LEDs.

### External Storage for Cisco ENCS 5400 and Cisco Cloud Services Platforms

#### Table 34: Feature History

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for External Storage for Cisco Cloud Services Platforms</td>
<td>NFVIS 4.6.1</td>
<td>External disks are supported for Cisco Cloud Services Platforms (CSP).</td>
</tr>
</tbody>
</table>

#### Restrictions for External Storage for Cisco ENCS 5400 and Cisco Cloud Services Platforms

• A maximum of two RAID groups can be configured for Cisco ENCS 5400 and a maximum of three RAID groups for CSP platforms.

#### Information about External Storage for Cisco ENCS 5400 and Cisco Cloud Services Platforms

For details on supported storage type, number of storage devices and, RAID modes on each hardware, see the table below:

<table>
<thead>
<tr>
<th>Device</th>
<th>Storage Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCS 5400</td>
<td>Cisco 5400 Enterprise Network Compute System Hardware Installation Guide</td>
</tr>
</tbody>
</table>
RAID controller is optional on ENCS 5400.

RAID configurations are performed from Cisco IMC for each hardware platform. For UCS-E and CSP devices, all RAID configurations should be performed before installing the NFVIS software. For ENCS 5400, because the OS installation is done on internal SSD, RAID configurations can also be done after the NFVIS software is installed.

This table provides information about the number of RAID groups supported on each platform and the NFVIS release in which the support was introduced:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Release introduced</th>
<th>Number of RAID groups supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCS 5400</td>
<td>NFVIS 3.8 release</td>
<td>Two RAID groups: First group—extdatastore1 Other group—extdatastore2</td>
</tr>
<tr>
<td>Cisco CSP</td>
<td>NFVIS 4.6 release</td>
<td>Three RAID groups: First group—intdatastore (the OS is installed in this group) Second group—extdatastore1 Third group—extdatastore2</td>
</tr>
<tr>
<td>UCS-E Single-Wide series</td>
<td>NFVIS 3.5 release</td>
<td>One RAID group for fresh installation</td>
</tr>
<tr>
<td>UCS-E Double-Wide series</td>
<td>NFVIS 3.x release and previous releases</td>
<td>One RAID group</td>
</tr>
<tr>
<td></td>
<td>NFVIS 4.1 release</td>
<td>Two RAID groups</td>
</tr>
</tbody>
</table>

Power off the system before you remove or insert disks in ENCS 5400.

To display the number of external disks on the system, use the `show system ext-disks` command.

```
nfvis# show system ext-disks
NAME
------------------
extdatastore1
```

To display the disk space on an external disk, use the `show system disk-space` command.

```
nfvis# show system disk-space
ASSOCIATED
PHYSICAL TOTAL SIZE SIZE USE
DISK NAME DISK SIZE USED AVAILABLE PERCENT
```
### External Storage for Cisco ENCS 5400 and Cisco Cloud Services Platforms

<table>
<thead>
<tr>
<th>Volume</th>
<th>Driver</th>
<th>Size</th>
<th>Free</th>
<th>Used</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>lv_data</td>
<td>sde2</td>
<td>99G</td>
<td>4.3G</td>
<td>94G</td>
<td>5%</td>
</tr>
<tr>
<td>lv_var</td>
<td>sde2</td>
<td>3.9G</td>
<td>245M</td>
<td>3.4G</td>
<td>7%</td>
</tr>
<tr>
<td>lv_root</td>
<td>sde2</td>
<td>7.8G</td>
<td>1.9G</td>
<td>5.5G</td>
<td>26%</td>
</tr>
<tr>
<td>extdatastore1</td>
<td>sda</td>
<td>917G</td>
<td>77M</td>
<td>871G</td>
<td>1%</td>
</tr>
</tbody>
</table>

*Note: For more information on the drives and how to create them, see Managing Storage Adapters*
NFVIS Logging

- Configuring System Logs, on page 247

Configuring System Logs

NFVIS generates log files for troubleshooting issues. The configuration log and the operational log are the two main system log files. The configuration log has information related to configurations and actions performed on the system such as creation of networks. The operational log has information related to system operation such as statistics collection and monitoring.

Log entries can be one of the following types:

<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>

By default, the configuration log has a log-level of INFO. All logs of type INFO, WARNING, ERROR and CRITICAL are logged.

By default, the operational log has a log-level of WARNING. All logs of type WARNING, ERROR and CRITICAL are logged.

The log-level for these log files can be changed using the `system set-log` command:

`system set-log level error logtype configuration`

The change to the log level is not persistent across a reboot. After a reboot, the default log levels are used.
The current log files are kept in the /var/log directory in the system:

- show log - To display the list of available log files
- show log {filename} - To display the contents of a specific log file

Log Rotation

There is a size limit for the log files, under /var/log/ directory. When the log files reach the size limit, the location of logs is rotated to another place. The space limit for the total size of all rotated log files is 2 GB. The older log files are dropped automatically on reaching the space limit. You can also execute a command to trigger the log rotation procedure. The log files are monitored periodically and if a log file gets too big, it is rotated to another place.

There is a size limit for the log files stored in the /var/log directory. The size of the log files is monitored periodically every fifteen minutes and if a log file gets too big, it is rotated to the /data/intdatastore/logs directory. The space limit for the total size of all the rotated log files is 2 GB. The older log files are dropped automatically on reaching the space limit. You can also execute the logrotate command to trigger the log rotation procedure.

nfvis# logrotate

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 [all/configuration/operational] level [critical/debug/error/info/warning]</td>
</tr>
<tr>
<td>/api/operational/system/logging-level</td>
<td>• show system logging-level</td>
</tr>
</tbody>
</table>
CHAPTER 11

NFVIS Monitoring

- Syslog, on page 249
- NETCONF Event Notifications, on page 251
- SNMP Support on NFVIS, on page 252
- System Monitoring, on page 264

Syslog

The Syslog feature allows event notifications from NFVIS to be sent to remote syslog servers for centralized log and event collection. The syslog messages are based on the occurrence of specific events on the device and provide configuration and operational information such as creation of users, changes to the interface status, and failed login attempts. Syslog data is critical to recording day-to-day events as well as notifying operational staff of critical system alerts.

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

Syslog Message Format

Syslog messages have the following format:

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

Sample Syslog messages:

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

Note

To refer to the complete list of syslog messages, see Syslog Messages, on page 319
**Configure a Remote Syslog Server**

To send syslogs to an external server, configure its IP address or DNS name along with the protocol to send syslogs and the port number on the syslog server.

To configure a remote Syslog server:

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

A maximum of 4 remote syslog servers can be configured. The remote syslog server can be specified using its IP address or DNS name. The default protocol for sending syslogs is UDP with a default port of 514. For TCP, the default port is 601.

**Configure Syslog Severity**

The syslog severity describes the importance of the syslog message.

To configure syslog severity:

```
configure terminal
system settings logging severity <debug | informational | notice | warning| error| critical | alert | emergency>
```

**Table 35: Syslog Severity Levels**

<table>
<thead>
<tr>
<th>Severity Level</th>
<th>Description</th>
<th>Numeric Encoding for Severity in the Syslog Message Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td>Debug-level messages</td>
<td>7</td>
</tr>
<tr>
<td>informational</td>
<td>Informational messages</td>
<td>6</td>
</tr>
<tr>
<td>notice</td>
<td>Normal but significant condition</td>
<td>5</td>
</tr>
<tr>
<td>warning</td>
<td>Warning conditions</td>
<td>4</td>
</tr>
<tr>
<td>error</td>
<td>Error conditions</td>
<td>3</td>
</tr>
<tr>
<td>critical</td>
<td>Critical conditions</td>
<td>2</td>
</tr>
<tr>
<td>alert</td>
<td>Take action immediately</td>
<td>1</td>
</tr>
<tr>
<td>emergency</td>
<td>System is unusable</td>
<td>0</td>
</tr>
</tbody>
</table>
By default, the logging severity of syslog is informational which means all syslogs at informational severity and higher will be logged. Configuring a value for severity will result in syslogs at the configured severity and syslogs which are more severe than the configured severity.

**Configure Syslog Facility**

The syslog facility can be used to logically separate and store syslog messages on the remote syslog server. For example, syslogs from a particular NFVIS can be assigned a facility of local0 and can be stored and processed in a different directory location on the syslog server. This is useful to separate it from syslogs with a facility of local1 from another device.

To configure syslog facility:

```
configure terminal
system settings logging facility local5
```

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

By default, NFVIS sends syslogs with the facility of local7

**Syslog Support APIs and Commands**

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

**NETCONF 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. For more information see, Event Notifications, on page 285.
SNMP Support on NFVIS

Introduction about SNMP

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

The SNMP framework has three parts:

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

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

SNMP Operations

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

- **SNMP Get** - The SNMP GET operation is performed by a Network Management Server (NMS) to retrieve SNMP object variables.
- **SNMP Set** - The SNMP SET operation is performed by a Network Management Server (NMS) to modify the value of an object variable.
- **SNMP Notifications** - A key feature of SNMP is its capability to generate unsolicited notifications from an SNMP agent.

SNMP Get

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

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

The command for SNMP GET is:
SNMP Walk

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

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

The command for SNMP walk with SNMP v2 is:

```
snmpwalk -v2c -c [community-name] [nfvis-box-ip] [tag-name, example ifSpeed], [index value]
```

```
snmpwalk -v2c -c myUser 172.19.147.115 1.3.6.1.2.1.1
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Cisco NFVIS
```

```
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.1836
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (43545580) 5 days, 0:57:35.80
```

```
SNMPv2-MIB::sysContact.0 = STRING:
```

```
SNMPv2-MIB::sysName.0 = STRING:
```

```
SNMPv2-MIB::sysLocation.0 = STRING:
```

```
SNMPv2-MIB::sysServices.0 = INTEGER: 70
```

```
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

```
IF-MIB::ifIndex.1 = INTEGER: 1
```

```
IF-MIB::ifIndex.2 = INTEGER: 2
```

```
IF-MIB::ifIndex.3 = INTEGER: 3
```

```
IF-MIB::ifIndex.4 = INTEGER: 4
```

```
IF-MIB::ifIndex.5 = INTEGER: 5
```

```
IF-MIB::ifIndex.6 = INTEGER: 6
```

```
IF-MIB::ifIndex.7 = INTEGER: 7
```

```
IF-MIB::ifIndex.8 = INTEGER: 8
```

```
IF-MIB::ifIndex.9 = INTEGER: 9
```

```
IF-MIB::ifIndex.10 = INTEGER: 10
```

```
IF-MIB::ifIndex.11 = INTEGER: 11
```

```
IF-MIB::ifDescr.1 = STRING: GE0-0
```

```
IF-MIB::ifDescr.2 = STRING: GE0-1
```

```
IF-MIB::ifDescr.3 = STRING: MGMT
```

```
IF-MIB::ifDescr.4 = STRING: gigabitEthernet1/0
```

```
IF-MIB::ifDescr.5 = STRING: gigabitEthernet1/1
```

```
IF-MIB::ifDescr.6 = STRING: gigabitEthernet1/2
```

```
IF-MIB::ifDescr.7 = STRING: gigabitEthernet1/3
```

```
IF-MIB::ifDescr.8 = STRING: gigabitEthernet1/4
```

```
IF-MIB::ifDescr.9 = STRING: gigabitEthernet1/5
```

```
IF-MIB::ifDescr.10 = STRING: gigabitEthernet1/6
```

```
IF-MIB::ifDescr.11 = STRING: gigabitEthernet1/7
```

```
SNMPv2-SMI::mib-2.47.1.1.12.0 = STRING: "Cisco NFVIS"
```

```
SNMPv2-SMI::mib-2.47.1.1.13.0 = OID: SNMPv2-SMI::enterprises.9.1.1836
```

```
SNMPv2-SMI::mib-2.47.1.1.1.4.0 = INTEGER: 0
```

```
SNMPv2-SMI::mib-2.47.1.1.1.5.0 = INTEGER: 3
```

```
SNMPv2-SMI::mib-2.47.1.1.1.6.0 = INTEGER: -1
```

```
SNMPv2-SMI::mib-2.47.1.1.1.7.0 = STRING: "ENCS5412/K9"
```

```
SNMPv2-SMI::mib-2.47.1.1.1.8.0 = STRING: "M3"
```

```
SNMPv2-SMI::mib-2.47.1.1.1.9.0 = ""
```

```
SNMPv2-SMI::mib-2.47.1.1.1.10.0 = STRING: "3.7.0-817"
```

```
SNMPv2-SMI::mib-2.47.1.1.1.11.0 = STRING: "FGL203012P2"
```

```
SNMPv2-SMI::mib-2.47.1.1.1.12.0 = STRING: "Cisco Systems, Inc."
```

```
SNMPv2-SMI::mib-2.47.1.1.1.13.0 = ""
```

```
...```

The following is a sample configuration of SNMP walk with SNMP v3:
snmpwalk -v 3 -u user3 -a sha -A changePassphrase -x aes -X changePassphrase -l authPriv -n snmp 172.16.1.101 system
SNMPv2-MIB::sysDescr.0 = STRING: Cisco ENCS 5412, 12-core Intel, 8 GB, 8-port PoE LAN, 2 HDD, Network Compute System
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.2377
SNMPv2-MIB::sysUpTimeInstance = TimeTicks: (16944068) 1 day, 23:04:00.68
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 70
SNMPv2-MIB::sysORLastChange.0 = TimeTicks: (0) 0:00:00.00

SNMP Notifications

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

Note

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

SNMP Versions

Cisco enterprise NFVIS supports the following versions of SNMP:

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

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

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

The security features provided in SNMPv3 are as follows:

- Message integrity—Ensuring that a packet has not been tampered with in transit.
- Authentication—Determining that the message is from a valid source.
- Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.
Both SNMP v1 and SNMP v2c use a community-based form of security. The community of managers able to access the agent MIB is defined by an IP address Access Control List and password.

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

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

**SNMP MIB Support**

*Table 36: Feature History*

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP VM Monitoring MIB</td>
<td>NFVIS 4.4.1</td>
<td>SNMP VM monitoring MIBs support added.</td>
</tr>
</tbody>
</table>

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

IF-MIB (1.3.6.1.2.1.31):
- ifDescr
- ifType
- ifPhysAddress
- ifSpeed
- ifOperStatus
- ifAdminStatus
- ifMtu
- ifName
- ifHighSpeed
- ifPromiscuousMode
- ifConnectorPresent
- ifInErrors
- ifInDiscards
- ifInOctets
- ifOutErrors
- ifOutDiscards
- ifOutOctets
• ifOutUcastPkts
• ifHCInOctets
• ifHCInUcastPkts
• ifHCOutOctets
• ifHCOutUcastPkts
• ifInBroadcastPkts
• ifOutBroadcastPkts
• ifInMulticastPkts
• ifOutMulticastPkts
• ifHCInBroadcastPkts
• ifHCOutBroadcastPkts
• ifHCInMulticastPkts
• ifHCOutMulticastPkts

Entity MIB (1.3.6.1.2.1.47):
• entPhysicalIndex
• entPhysicalDescr
• entPhysicalVendorType
• entPhysicalContainedIn
• entPhysicalClass
• entPhysicalParentRelPos
• entPhysicalName
• entPhysicalHardwareRev
• entPhysicalFirmwareRev
• entPhysicalSoftwareRev
• entPhysicalSerialNum
• entPhysicalMfgName
• entPhysicalModelName
• entPhysicalAlias
• entPhysicalAssetID
• entPhysicalIsFRU

Cisco Process MIB (1.3.6.1.4.1.9.9.109):
• cpmCPUTotalPhysicalIndex (.2)
• cpmCPUTotal5secRev (.6.x)*
• cpmCPUTotal1minRev (.7.x)*
• cpmCPUTotal5minRev (.8.x)*
• cpmCPUMonInterval (.9)
• cpmCPUMemoryUsed (.12)
• cpmCPUMemoryFree (.13)
• cpmCPUMemoryKernelReserved (.14)
• cpmCPUMemoryHCUsed (.17)
• cpmCPUMemoryHCFree (.19)
• cpmCPUMemoryHCKernelReserved (.21)
• cpmCPULoadAvg1min (.24)
• cpmCPULoadAvg5min (.25)
• cpmCPULoadAvg15min (.26)

* Support data for per CPU core from NFVIS 3.12.3 release.

---

Cisco Environmental MIB (1.3.6.1.4.1.9.9.13):

• Voltage Sensor:
  • ciscoEnvMonVoltageStatusDescr
  • ciscoEnvMonVoltageStatusValue

• Temperature Sensor:
  • ciscoEnvMonTemperatureStatusDescr
  • ciscoEnvMonTemperatureStatusValue

• Fan Sensor
  • ciscoEnvMonFanStatusDescr
  • ciscoEnvMonFanState
Sensor support for the following hardware platforms:

- ENCS 5400 series: all
- ENCS 5100 series: none
- UCS-E: voltage, temperature
- UCS-C: all
- CSP: CSP-2100, CSP-5228, CSP-5436 and CSP5444 (Beta)

Cisco Environmental Monitor MIB notification starting from NFVIS 3.12.3 release:

- ciscoEnvMonEnableShutdownNotification
- ciscoEnvMonEnableVoltageNotification
- ciscoEnvMonEnableTemperatureNotification
- ciscoEnvMonEnableFanNotification
- ciscoEnvMonEnableRedundantSupplyNotification
- ciscoEnvMonEnableStatChangeNotif

VM-MIB (1.3.6.1.2.1.236) starting from NFVIS 4.4 release:

- vmHypervisor:
  - vmHvSoftware
  - vmHvVersion
  - vmHvUpTime

- vmTable:
  - vmName
  - vmUUID
  - vmOperState
  - vmOSType
  - vmCurCpuNumber
  - vmMemUnit
  - vmCurMem
  - vmCpuTime

- vmCpuTable:
  - vmCpuCoreTime
Configuring SNMP Support

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

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

To create SNMP community:

```
configure terminal
snmp community <community_name> community-access <access>
```

SNMP community name string supports [A-Za-z0-9_-] and maximum length of 32. NFVIS supports only `readOnly` access.

To create SNMP Group:

```
configure terminal
snmp group <group_name> <context> <version> <security_level> notify <notify_list> read <read_list> write <write_list>
```

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_name</td>
<td>Group name string. Supporting string is [A-Za-z0-9_-] and maximum length is 32.</td>
</tr>
<tr>
<td>context</td>
<td>Context string, default is snmp. Maximum length is 32. Minimum length is 0 (empty context).</td>
</tr>
<tr>
<td>version</td>
<td>1, 2 or 3 for SNMP v1, v2c and v3.</td>
</tr>
<tr>
<td>security_level</td>
<td>authPriv, authNoPriv, noAuthNoPriv</td>
</tr>
<tr>
<td>Note</td>
<td>SNMP v1 and v2c uses noAuthNoPriv only.</td>
</tr>
<tr>
<td>notify_list/read_list/write_list</td>
<td>It can be any string. read_list and notify_list is required to support data retrieval by SNMP tools. write_list can be skipped because NFVIS SNMP does not support SNMP write access.</td>
</tr>
</tbody>
</table>

To create SNMP v3 user:

When security level is authPriv

```
configure terminal
snmp user <user_name> user-version 3 user-group <group_name> auth-protocol <auth> priv-protocol <priv> passphrase <passphrase_string>
```

When security level is authNoPriv:
configure terminal
snmp user <user_name> user-version 3 user-group <group_name> auth-protocol <auth> passphrase <passphrase_string>

When security level is noAuthNoPriv

configure terminal
snmp user <user_name> user-version 3 user-group <group_name>

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
<td>User name string. Supporting string is [A-Za-z0-9-_] and maximum length is 32. This name has to be the same as community_name.</td>
</tr>
<tr>
<td>version</td>
<td>1 and 2 for SNMP v1 and v2c.</td>
</tr>
<tr>
<td>group_name</td>
<td>Group name string. This name has to be same as the group name configured in the NFVIS.</td>
</tr>
<tr>
<td>auth</td>
<td>md5 or sha</td>
</tr>
<tr>
<td>priv</td>
<td>aes or des</td>
</tr>
<tr>
<td>passphrase_string</td>
<td>Passphrase string. Supporting string is [A-Za-z0-9-_#@%$*&amp;!]. NFVIS does not support different passphrases for auth and priv.</td>
</tr>
</tbody>
</table>

Note

Do not use auth-key and priv-key. The auth and priv passphrases are encrypted after configuration and saved in NFVIS.

To enable SNMP traps:

configure terminal
snmp enable traps <trap_event>

trap_event can be linkup or linkdown

To create SNMP trap host:

configure terminal
snmp host <host_name> host-ip-address <ip_address> host-port <port> host-user-name <user_name> host-version <version> host-security-level noAuthNoPriv

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_name</td>
<td>User name string. Supporting string is [A-Za-z0-9-_] and maximum length is 32. This is not FQDN host name, but an alias to IP address of traps.</td>
</tr>
<tr>
<td>ip_address</td>
<td>IP address of traps server.</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Default is 162. Change to other port number based on your own setup.</td>
</tr>
<tr>
<td>user_name</td>
<td>User name string. Must be the same as user_name configured in NFVIS.</td>
</tr>
<tr>
<td>version</td>
<td>1, 2 or 3 for SNMP v1, v2c or v3.</td>
</tr>
<tr>
<td>security_level</td>
<td>authPriv, authNoPriv, noAuthNoPriv</td>
</tr>
</tbody>
</table>

#### Note
- SNMP v1 and v2c uses noAuthNoPriv only.

### SNMP Configuration Examples

The following example shows SNMP v1 and v2 configuration:

```bash
configure terminal
snmp community public community-access readOnly
snmp group testgroup snmp 2 noAuthNoPriv read read-access write write-access notify notify-access
snmp user public user-group testgroup user-version 2
snmp host host2 host-ip-address 2.2.2.2 host-port 162 host-user-name public host-version 2 host-security-level noAuthNoPriv
snmp enable traps linkup
snmp enable traps linkDown
```

The following example shows SNMP v3 configuration:

```bash
configure terminal
snmp group testgroup3 snmp 3 authPriv notify test write test read test
snmp user user3 user-version 3 user-group testgroup3 auth-protocol sha priv-protocol aes passphrase changePassphrase
snmp host host3 host-ip-address 3.3.3.3 host-port 162 host-user-name user3 host-version 3 host-security-level authPriv host-port 162
```

To change the security level:

```bash
configure terminal
snmp group testgroup4 snmp 3 authNoPriv notify test write test read test
snmp user user4 user-version 3 user-group testgroup4 auth-protocol md5 passphrase changePassphrase
snmp host host4 host-ip-address 4.4.4.4 host-port 162
```

---

Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x

Page 261
To change default context SNMP:

```bash
configure terminal
snmp group testgroup5 devop 3 authPriv notify test write test read test
snmp user user5 user-version 3 user-group testgroup5 auth-protocol md5 priv-protocol des
passphrase changePassphrase
```

To use empty context and noAuthNoPriv

```bash
configure terminal
snmp group testgroup6 "" 3 noAuthNoPriv read test write test notify test
snmp user user6 user-version 3 user-group testgroup6
```

---

**Note**

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

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

Do not use auth-key and priv-key to configure SNMP v3 passphrase. These keys are generated differently between different NFVIS systems for the same passphrase.

---

**Note**

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

---

**Note**

NFVIS 3.12.1 release supports the following special characters: `-_#@%$&!` and whitespace. Backslash (`\`) is not supported.

---

**Verify the Configuration for SNMP Support**

Use the `show snmp agent` command to verify the snmp agent description and ID.

```bash
nfvis# show snmp agent
snmp agent sysDescr "Cisco NFVIS"
snmp agent sysOID 1.3.6.1.4.1.9.12.3.1.3.1291
```

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

```bash
nfvis# show snmp traps
TRAP   TRAP
NAME   STATE
```
Use the `show snmp stats` command to verify the snmp stats.

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

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

```
nfvis# show running-config snmp
snmp agent enabled true
snmp agent engineID 00:00:00:09:11:22:33:44:55:66:77:88
snmp enable traps linkUp
snmp community pub_comm
community-access readOnly
!
snmp community tachen
community-access readOnly
!
snmp group tachen snmp 2 noAuthNoPriv
   read  test
   write test
   notify test
!
snmp group testgroup snmp 2 noAuthNoPriv
   read  read-access
   write write-access
   notify notify-access
!
snmp user public
   user-version  2
   user-group  2
   auth-protocol md5
   priv-protocol des
!
snmp user tachen
   user-version  2
   user-group  tachen
!
snmp host host2
   host-port  162
   host-ip-address  2.2.2.2
   host-version  2
   host-security-level noAuthNoPriv
   host-user-name  public
```
Upper limit for SNMP configurations

Upper limit for SNMP configurations:

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

SNMP Support APIs and Commands

<table>
<thead>
<tr>
<th>APIs</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/config/snmp/agent</td>
<td>agent</td>
</tr>
<tr>
<td>/api/config/snmp/communities</td>
<td>community</td>
</tr>
<tr>
<td>/api/config/snmp/enable/traps</td>
<td>trap-type</td>
</tr>
<tr>
<td>/api/config/snmp-hosts</td>
<td>host</td>
</tr>
<tr>
<td>/api/config/snmp/user</td>
<td>user</td>
</tr>
<tr>
<td>/api/config/snmp/groups</td>
<td>group</td>
</tr>
</tbody>
</table>

System Monitoring

NFVIS provides system monitoring commands and APIs to monitor the host and the VMs deployed on NFVIS. These commands are useful to collect statistics on CPU utilization, memory, disk and ports. The metrics related to these resources are collected periodically and displayed for a specified duration. For larger durations average values are displayed.

System monitoring enables the user to view historical data on the system's operation. These metrics are also shown as graphs on the portal.

Collection of System Monitoring Statistics

System monitoring statistics are displayed for the requested duration. The default duration is five minutes.

The supported duration values are 1min, 5min, 15min, 30min, 1h, 6h, 6H, 1d, 1D, 5d, 5D, 30d, 30D with min as minutes, h and H as hours, d and D as days.

Example

The following is a sample output of system monitoring statistics:

```
nfvis# show system-monitoring host cpu stats cpu-usage 1h state non-idle
system-monitoring host cpu stats cpu-usage 1h state non-idle
collect-start-date-time 2019-12-20T11:27:20-00:00
collect-interval-seconds 10
cpu
  id      0
```
usage-percentage "[7.67, 5.52, 4.89, 5.77, 5.03, 5.93, 10.07, 5.49, ...

The time at which the data collection started is displayed as **collect-start-date-time**.
The sampling interval at which data is collected is shown as **collect-interval-seconds**.
The data for the requested metric like host CPU statistics is displayed as an array. The first data point in the array was collected at the specified **collect-start-date-time** and each subsequent value at an interval specified by **collect-interval-seconds**.

In the sample output, CPU id 0 has a utilization of 7.67% on 2019-12-20 at 11:27:20 as specified by **collect-start-date-time**. 10 seconds later, it had a utilization of 5.52% since the **collect-interval-seconds** is 10. The third value of cpu-utilization is 4.89% at 10 seconds after the second value of 5.52% and so on.

The sampling interval shown as **collect-interval-seconds** changes based on the specified duration. For higher durations, the collected statistics are averaged at a higher interval to keep the number of results reasonable.

## Host System Monitoring

NFVIS provides system monitoring commands and APIs to monitor the host’s CPU utilization, memory, disk and ports.

### Monitoring the Host CPU Usage

The percentage of time spent by the CPU in various states, such as executing user code, executing system code, waiting for IO operations, etc. is displayed for the specified duration.

<table>
<thead>
<tr>
<th><strong>cpu-state</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>non-idle</td>
<td>100 – idle-cpu-percentage</td>
</tr>
<tr>
<td>interrupt</td>
<td>Indicates the percentage of the processor time spent in servicing interrupts</td>
</tr>
<tr>
<td>nice</td>
<td>The nice CPU state is a subset of the user state and shows the CPU time used by processes that have a lower priority than other tasks.</td>
</tr>
<tr>
<td>system</td>
<td>The system CPU state shows the amount of CPU time used by the kernel.</td>
</tr>
<tr>
<td>user</td>
<td>The user CPU state shows CPU time used by user space processes</td>
</tr>
<tr>
<td>wait</td>
<td>Idle time while waiting for an I/O operation to complete</td>
</tr>
</tbody>
</table>

The non-idle state is what the user usually needs to monitor. Use the following CLI or API for monitoring CPU usage:

```
nfvis# show system-monitoring host cpu stats cpu-usage <duration> state <cpu-state>
```

```
/api/operational/system-monitoring/host/cpu/stats/cpu-usage/<duration>,<cpu-state>?deep
```
The data is also available in an aggregate form for the minimum, maximum, and average CPU utilization using the following CLI and API:

nfvis# show system-monitoring host cpu table cpu-usage <duration>

/api/operational/system-monitoring/host/cpu/table/cpu-usage/<duration>?deep

**Monitoring the Host Port Statistics**

The statistics collection for non-switch ports is handled by the collectd daemon on all platforms. The input and output rate calculation per port is enabled and the rate calculations are done by the collectd daemon.

Use the `show system-monitoring host port stats` command to display the outputs of the calculations done by collectd for packets/sec, errors/sec and now kilobits/sec. Use the `system-monitoring host port table` command to display the outputs of the collectd stats average for last 5 minutes for packets/sec and kilobits/sec values.

**Monitoring Host Memory**

Statistics for the physical memory utilization are displayed for the following categories:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>buffered-MB</td>
<td>Memory used for buffering I/O</td>
</tr>
<tr>
<td>cached-MB</td>
<td>Memory used for caching file system access</td>
</tr>
<tr>
<td>free-MB</td>
<td>Memory available for use</td>
</tr>
<tr>
<td>used-MB</td>
<td>Memory in use by the system</td>
</tr>
<tr>
<td>slab-recl-MB</td>
<td>Memory used for SLAB-allocation of kernel objects, that can be reclaimed</td>
</tr>
<tr>
<td>slab-unrecl-MB</td>
<td>Memory used for SLAB-allocation of kernel objects, that can’t be reclaimed</td>
</tr>
</tbody>
</table>

Use the following CLI or API for monitoring host memory:

nfvis# show system-monitoring host memory stats mem-usage <duration>

/api/operational/system-monitoring/host/memory/stats/mem-usage/<duration>?deep

The data is also available in an aggregate form for the minimum, maximum, and average memory utilization using the following CLI and API:

nfvis# show system-monitoring host memory table mem-usage <duration>

/api/operational/system-monitoring/host/memory/table/mem-usage/<duration>?deep

**Monitoring Host Disks**

Statistics for disk operations and disk space can be obtained for the list of disks and disk partitions on the NFVIS host.
Monitoring Host Disks Operations

The following disk performance statistics are displayed for each disk and disk partition:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>io-time-ms</td>
<td>Average time spent doing I/O operations in milliseconds</td>
</tr>
<tr>
<td>io-time-weighted-ms</td>
<td>Measure of both I/O completion time and the backlog that may be accumulating</td>
</tr>
<tr>
<td>merged-reads-per-sec</td>
<td>The number of read operations that could be merged into already queued operations, that is one physical disk access served two or more logical operations. The higher the merged reads, the better the performance.</td>
</tr>
<tr>
<td>merged-writes-per-sec</td>
<td>The number of write operations that could be merged into other already queued operations, that is one physical disk access served two or more logical operations. The higher the merged reads, the better the performance.</td>
</tr>
<tr>
<td>bytes-read-per-sec</td>
<td>Bytes read per second</td>
</tr>
<tr>
<td>bytes-written-per-sec</td>
<td>Bytes written per second</td>
</tr>
<tr>
<td>reads-per-sec</td>
<td>Number of read operations per second</td>
</tr>
<tr>
<td>writes-per-sec</td>
<td>Number of write operations per second</td>
</tr>
<tr>
<td>time-per-read-ms</td>
<td>The average time a read operation takes to complete</td>
</tr>
<tr>
<td>time-per-write-ms</td>
<td>The average time a write operation takes to complete</td>
</tr>
<tr>
<td>pending-ops</td>
<td>The queue size of pending I/O operations</td>
</tr>
</tbody>
</table>

Use the following CLI or API for monitoring host disks:

```
nfvis# show system-monitoring host disk stats disk-operations <duration>
```

```
/api/operational/system-monitoring/host/disk/stats/disk-operations/<duration>?deep
```

Monitoring Host Disk Space

The following data related to file system usage, that is how much space on a mounted partition is used and how much is available is collected:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>free-GB</td>
<td>Gigabytes available</td>
</tr>
<tr>
<td>used-GB</td>
<td>Gigabytes in use</td>
</tr>
</tbody>
</table>
Use the following CLI or API for monitoring host disk space:

```
nfvis# show system-monitoring host disk stats disk-space <duration>

/api/operational/system-monitoring/host/disk/stats/disk-space/<duration>?deep
```

**Monitoring Host Ports**

The following statistics for network traffic and errors on interfaces are displayed:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Interface name</td>
</tr>
<tr>
<td>total-packets-per-sec</td>
<td>Total (received and transmitted) packet rate</td>
</tr>
<tr>
<td>rx-packets-per-sec</td>
<td>Packets received per second</td>
</tr>
<tr>
<td>tx-packets-per-sec</td>
<td>Packets transmitted per second</td>
</tr>
<tr>
<td>total-errors-per-sec</td>
<td>Total (received and transmitted) error rate</td>
</tr>
<tr>
<td>rx-errors-per-sec</td>
<td>Error rate for received packets</td>
</tr>
<tr>
<td>tx-errors-per-sec</td>
<td>Error rate for transmitted packets</td>
</tr>
</tbody>
</table>

Use the following CLI or API for monitoring host ports:

```
nfvis# show system-monitoring host port stats port-usage <duration>

/api/operational/system-monitoring/host/port/stats/port-usage/<duration>?deep
```

The data is also available in an aggregate form for the minimum, maximum, and average port utilization using the following CLI and API:

```
nfvis# show system-monitoring host port table

/api/operational/system-monitoring/host/table/port-usage/<duration>,<name>?deep
```

**VNF System monitoring**

NFVIS provides system monitoring commands and APIs to get statistics on the virtualized guests deployed on NFVIS. These statistics provide data on the VM’s CPU utilization, memory, disk and network interfaces.

**Monitoring the VNF CPU Usage**

The CPU utilization of a VM is displayed for the specified duration using the following fields:
### NFVIS Monitoring

#### VNF System monitoring

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total-percentage</td>
<td>Average CPU utilization across all the logical CPUs used by the VM</td>
</tr>
<tr>
<td>id</td>
<td>Logical CPU ID</td>
</tr>
<tr>
<td>vcpu-percentage</td>
<td>CPU utilization percentage for the specified logical CPU id</td>
</tr>
</tbody>
</table>

Use the following CLI or API to monitor the CPU usage of the VNF:

```
nfvis# show system-monitoring vnf vcpu stats vcpu-usage <duration>
```

```
/api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/<duration>/vnf/<vnf-name>?deep
```

### Monitoring VNF memory

The following statistics are collected for VNF memory utilization:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total-MB</td>
<td>Total memory of the VNF in MB</td>
</tr>
<tr>
<td>rss-MB</td>
<td>Resident Set Size (RSS) of the VNF in MB</td>
</tr>
</tbody>
</table>

The Resident Set Size (RSS) is the portion of memory occupied by a process, that is held in the RAM. The rest of the occupied memory exists in the swap space or file system, because some parts of the occupied memory are paged out, or some parts of the executable are not loaded.

Use the following CLI or API to monitor VNF memory:

```
nfvis# show system-monitoring vnf memory stats mem-usage <duration>
```

```
/api/operational/system-monitoring/vnf/memory/stats/mem-usage/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/memory/stats/mem-usage/<duration>/vnf/<vnf-name>?deep
```

### Monitoring VNF Disks

The following disk performance statistics are collected for each disk used by the VM:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes-read-per-sec</td>
<td>Bytes read from the disk per second</td>
</tr>
<tr>
<td>bytes-written-per-sec</td>
<td>Bytes written to the disk per second</td>
</tr>
</tbody>
</table>
### NFVIS Monitoring

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reads-per-sec</td>
<td>Number of read operations per second</td>
</tr>
<tr>
<td>writes-per-sec</td>
<td>Number of write operations per second</td>
</tr>
</tbody>
</table>

Use the following CLI or API to monitor VNF disks:

```
nfvis# show system-monitoring vnf disk stats <duration>
```

```
/api/operational/system-monitoring/vnf/disk/stats/disk-operations/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/disk/stats/disk-operations/<duration>/vnf/<vnf-name>?deep
```

### Monitoring VNF Ports

The following network interface statistics are collected for VMs deployed on NFVIS:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total-packets-per-sec</td>
<td>Total packets received and transmitted per second</td>
</tr>
<tr>
<td>rx-packets-per-sec</td>
<td>Packets received per second</td>
</tr>
<tr>
<td>tx-packets-per-sec</td>
<td>Packets transmitted per second</td>
</tr>
<tr>
<td>total-errors-per-sec</td>
<td>Total error rate for packet reception and transmission</td>
</tr>
<tr>
<td>rx-errors-per-sec</td>
<td>Error rate for receiving packets</td>
</tr>
<tr>
<td>tx-errors-per-sec</td>
<td>Error rate for transmitting packets</td>
</tr>
</tbody>
</table>

Use the following CLI or API to monitor VNF ports:

```
nfvis# show system-monitoring vnf port stats port-usage <duration>
```

```
/api/operational/system-monitoring/vnf/port/stats/port-usage/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/port/stats/port-usage/<duration>/vnf/<vnf-name>?deep
```

### ENCS Switch Monitoring

**Table 37: Feature History**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCS Switch Monitoring</td>
<td>NFVIS 4.5.1</td>
<td>This feature allows you to calculate the data rate for ENCS switch ports based on the data collected from the ENCS switch.</td>
</tr>
</tbody>
</table>
For ENCS switch ports, the data rate is calculated based on the data collected from the ENCS switch using periodic polling every 10 seconds. Input and output rate in Kbps is calculated based on octets collected from the switch every 10 seconds.

The formula used for the calculation is as follows:

\[
\text{Avg rate} = (\text{Avg rate} - \text{Current interval rate}) \times (\alpha) + \text{Current Interval rate}.
\]
\[
\text{Alpha} = \frac{\text{Multiplier}}{\text{Scale}}
\]
\[
\text{Multiplier} = \text{scale} - \frac{(\text{scale} \times \text{compute interval})}{\text{Load interval}}
\]

Where compute interval is the polling interval and Load_interval is the interface load interval = 300 sec and scale = 1024.

Because the data is obtained directly from the switch the kbps rate includes Frame Check Sequence (FCS) bytes.

The bandwidth calculation is extended to the ENCS switch port channels using the same formula. Input and output rate in kbps is displayed separately for each gigabit Ethernet port as well as for the corresponding port-channel group the port is associated with.

Use the `show switch interface counters` command to view the data rate calculations.
CHAPTER 12

Troubleshoot and Debug Cisco NFVIS

- System Shell Access, on page 273
- Log and Show Commands, on page 276
- SPAN Session or Port Mirroring, on page 278
- Configuring Packet Capture, on page 282

System Shell Access

NFVIS Linux Shell Access When System is in Working Condition and External Authentication is Not Enabled

To debug the system error, you will have to enter Linux shell. You can enter the Linux shell after challenge-response:

1. Get the challenge string from NFVIS.

   nfvis# system shell-access

   ****** If you are not able to retrieve a response string at this time, please enter 'DONE.'
   (single quotes are excluded) as your response to exit. ******

  *******************************************************************************
   Welcome to NFV Infrastructure Software's Interactive Debug.
   *******************************************************************************

   Challenge String (Please copy everything between the asterick lines exclusively):

  𬘫�/wkcAAARvRLJIV0VQIM1NDA4L0s5AgAAAnfC507uspnBW+RkdMMjEyNDgx
   VFa1WfpAAA=
   DONE.
   ******************************************************************************

2. In any Cisco server, run a script to get the response based on the challenge string from the previous step, "-C1" followed by the first line of the challenge string, "-C2" followed by the second line. This also requires users Cisco password.
ct_sign_client -C1 "N/H//wkAAABORlZJU0VOQ1M1NDA4L0s5AgAAAAAHNvfc507uspnBN+RkdMMjEyNDgx" -C2 "VFa1WfpdAAA=" -cec
Welcome, user!!

Please wait... This may take a few minutes.

yminglei password:
Response String (Please copy everything between the asterisk lines exclusively):
*********************************************************************
AQAAAAEAAADfWfpdCHltaW5nbGVpDE1pbmdsZWkgWWFuZwAAAYgBAAIBAQIABAAA FFFFEB72
AXwDAFSEAcDdT10aNjb1N5c3RlbXM7TIU9TkZWNV7Tz10aNjb1N5c3RlbXM7 FFFFC15
AAGlREZBT1ERgYAJ0NPUnc2NvU3lzdGVtctzPTU1B12UztFPUnc2NvU3lzdGVtctz FFFFEA6D
dU7cwcAQAIAAEBQCAABBoAAAQELAQCoS95OepGr7hYQlakihMpm299ogu77HP FFFFEAB4
DOKpyU3A4IA1XM90Mtcse7Gr8+GLHNo2ta1ntDztntXroz37t10HLCg1owXcvAAs FFFFEA48
BERRoAgrdzy+ZeGv1byKLSUCyr9buWOxX8LPdxgByfWbi4j'oZ72s57XghYSxvxwE FFFFEA96
YJXyJ5dB0aLoSI5rU67EhMk2x2aVDXOW6w2SO6gXIFR/vyFbpuC10H616oca FFFFEAE4
BJ+77+jL31uji0ERMPuLe/EfVt4ilJWn+aazG6t755VY9v4jzhboouX3asAfgeW FFFFEAECF
sfwzg24DlV3m3YumaU1a8qs6UDuQNEJwMj7lp16E/KOHOU2ebQMaDAABBqv6w== FFFFEAD5
DONE.
*********************************************************************

3. Cut and paste the string back to NFVIS. You will get a prompt for NFVIS password. When you enter the password, you will enter the linux shell.

Please input your response when ready:
AQAAAAEAAADfWfpdCHltaW5nbGVpDE1pbmdsZWkgWWFuZwAAAYgBAAIBAQIABAAA FFFFEB72
AXwDAFSEAcDdT10aNjb1N5c3RlbXM7TIU9TkZWNV7Tz10aNjb1N5c3RlbXM7 FFFFEA6D
dU7cwcAQAIAAEBQCAABBoAAAQELAQCoS95OepGr7hYQlakihMpm299ogu77HP FFFFEAB4
DOKpyU3A4IA1XM90Mtcse7Gr8+GLHNo2ta1ntDztntXroz37t10HLCg1owXcvAAs FFFFEA48
BERRoAgrdzy+ZeGv1byKLSUCyr9buWOxX8LPdxgByfWbi4j'oZ72s57XghYSxvxwE FFFFEA96
YJXyJ5dB0aLoSI5rU67EhMk2x2aVDXOW6w2SO6gXIFR/vyFbpuC10H616oca FFFFEAE4
BJ+77+jL31uji0ERMPuLe/EfVt4ilJWn+aazG6t755VY9v4jzhboouX3asAfgeW FFFFEAECF
sfwzg24DlV3m3YumaU1a8qs6UDuQNEJwMj7lp16E/KOHOU2ebQMaDAABBqv6w== FFFFEAD5
DONE.
You have successfully entered all text. Please wait...
Starting NFVIS Debug session....
Starting NFVIS Debug session....
Password: [admin@nfvis admin]$

NFVIS System Shell Access When External Authentication Server is Enabled

When TACACS+ is enabled on NFVIS an additional step is needed to get shell access:

1. Login to NFVIS system authenticated by TACACS+ as tacacs_user.

2. Create a local user account with the same name. You can use a different password from the tacacs password. This is used for authentication when you are in linux shell, using sudo.

    nfvis#rbac authentication user create-user name <tacacs_user> password <local password> role administrator

3. Issue system shell-access command to get the challenge string, run the ct_sign_client string to get the response string. Cut and paste the string back to NFVIS. Use the local password you created before getting into the shell system. Always use the local password after you enter linux shell.
System Shell Access When NFVIS is Down

When major NFVIS services fail, NFVIS is inaccessible through normal login. To collect the logs and debug the issue, only an admin user with the local password can go through the console with assistance from Cisco Technical Assistance Center (TAC) to answer the challenge and response authentication and enter the Linux shell.

When the system is down, if you login from the console, a challenge string is automatically printed. Cisco TAC can get a response from the cisco server, cut and paste the challenge string back to the failed NFVIS service. In the end, enter the local password.

```
from console connection
login: admin
FAILED: cdb_connect(cs, st, addr, addrlen), Error: system call failed (24): Connection timed out, in function run, line 2485
NFVIS service is FAIL
NFVIS service is down, admin user can login system shell after challenge-response authentication

******************************************************************************
 Welcome to NFV Infrastructure Software's Interactive Debug.
******************************************************************************
 Challenge String (Please copy everything between the asterick lines exclusively):
******************************************************************************
e/D//wkAAABORlZJU0NTUD0NDQAY1dDAAAADB6NNeLuplaPx4YV1pQMj1XNjBO
Q1Fdud2dAAA=
 DONE.
******************************************************************************
Please input your response when ready:
AQAAAAIABAABWYvpICHlta5sW5bGpDE1pmds2WkgWFuZwAAAygBAAIAQIAABAAA
FFFEF71AwXfDAF8EAc0DTj1aXNjb1N5c3RlbXM7T1U9TkZWSMV7Ti1aXNjb1N5c3RlbXMF
FFFEFEC15AgiREZBnj1INgYAJONOPUNpc2NVU3izGvctztVT10R1JUzPPUNpc2NVU3iz
FFFEF8A92GVtcwcaAQAIABE8QABAAoAQEAQdQABAAoAQEBQABvMV080b/LN2h0i/ClyxPzfhe7X1M9cs/TS
FFFEFEC32gLPEZl1f1qsB3jPf7trj8Qz2eV1QDH1tVXq+zT81MTh5/hVqj38wjtJo/T
FFFEFEEA6HOG9J2jgyYTkmnt9XN6i/W5kvt43N075YsJ78esleg31xeppv0XL5cfWQ3
FFFEFEEA50BYyGfPsvTysMo2aMC2aLmY9Ve/DvHRb75xwYps1dQqKQUpJLu1f0POYpF2gL5U
FFFEFEEA50X4eTPC067shWmKRmK6YFKYv26AbMJzdxtek9XH1ijs3MptILmF/ugw1pKhdDxE1
FFFEFEE97vptPhasoNMpCX/xWNjVl1zvEsoINPfGwFbJ71BTtqINGTeY/1DAABQevr6w== FFFFE97C
DONE.

Password: <--- enter admin's local password
[admin@nfvis admin]$  
```

Cisco IMC Shell Access

NFVIS is supported on ENCS 5000 series platform with Cisco IMC as its controller. To debug issues related to ENCS 5000 series platform, you must get into the CIMC shell. If you login from the Cisco IMC console, a challenge string is automatically printed. Cisco TAC can get a response from the cisco server.

```
login to CIMC, run connect debug-shell"
ENCS5408-FGL22691HW# connect debug-shell
CIMC Debug Firmware Utility Shell [ cli ]
[ help ]# sldp
******************************************************************************
 Welcome to Cisco Integrated Management Controller's Interactive Debug. Please provide your credentials in order to proceed.
```
**Troubleshoot and Debug Cisco NFVIS**

**Log and Show Commands**

**Support Commands and Show Commands**

The following commands translate to corresponding linux commands like virsh, ovs and ip:
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show system status</strong></td>
<td>To display system defaults and services status.</td>
</tr>
<tr>
<td><strong>show system disk-space</strong></td>
<td>To display information about the system disk space.</td>
</tr>
<tr>
<td><strong>show system memory</strong></td>
<td>To display information about the system memory. If DPDK is enabled, check if HugePage is available to use.</td>
</tr>
<tr>
<td><strong>show resources cpu-info</strong></td>
<td>To get information on the resource assignment.</td>
</tr>
<tr>
<td><strong>VM</strong></td>
<td></td>
</tr>
<tr>
<td><strong>support virsh all-info</strong></td>
<td>To display the output of all supported VM and index by number.</td>
</tr>
<tr>
<td><strong>support virsh dumpxml &lt;num&gt;</strong></td>
<td>To display all information about one VM index by &lt;num&gt;.</td>
</tr>
<tr>
<td><strong>support virsh domiflist &lt;num&gt;</strong></td>
<td>To display the list of interfaces on VM index by &lt;num&gt; and MAC address of the VNICs.</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
</tr>
<tr>
<td><strong>support show ifconfig</strong></td>
<td>To display the configuration details of all network interfaces or a specific interface.</td>
</tr>
<tr>
<td><strong>support virsh net-list</strong></td>
<td>To display all the networks in the host</td>
</tr>
<tr>
<td><strong>support virsh net-dumpxml &lt;network name&gt;</strong></td>
<td>To display the network information about one network and bridge attachment.</td>
</tr>
<tr>
<td><strong>support virsh iface-list</strong></td>
<td>To display a list of interfaces on the host.</td>
</tr>
<tr>
<td><strong>Bridge</strong></td>
<td></td>
</tr>
<tr>
<td><strong>support ovs vsctl show:::</strong></td>
<td>To display an overview of the bridge, port and vlan tag.</td>
</tr>
<tr>
<td><strong>support ovs appctl fdb-show &lt;bridge-name&gt;</strong></td>
<td>To display information about the ports of a bridge.</td>
</tr>
<tr>
<td><strong>support ovs all-info</strong></td>
<td>To display the output of all supported ovs commands</td>
</tr>
<tr>
<td><strong>Firewall</strong></td>
<td></td>
</tr>
<tr>
<td><strong>support show firewall get-all-rule</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Log Files**

The tech-support includes all the logs. Download tech-support and record the time of the occurrence of error.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show log</code></td>
<td>To display a list of available log files or content of a specific log file.</td>
</tr>
<tr>
<td><code>show log nfvis_syslog.log</code></td>
<td>To display syslogs.</td>
</tr>
<tr>
<td><code>show log nfvis_config.log</code></td>
<td>To display system configuration related logs.</td>
</tr>
<tr>
<td><code>show log esc/escmanager.log</code></td>
<td>To display VM deployment related logs.</td>
</tr>
<tr>
<td><code>show log switch_config.log</code></td>
<td>To display the built-in switch configuration logs.</td>
</tr>
</tbody>
</table>

**SPAN Session or Port Mirroring**

**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.

Use VM Group Name for `source vm-vnic` and `destination vm-vnic` commands. Use the `show vm_lifecycle deployments all | include Group` command to get Group Name.

```
ENCS# show vm_lifecycle deployments all | include Group
VM Group Name : deployment-ASAv
VM Group Name : deployment-ISRv
VM Group Name : deployment-Linux
ENCS#
```

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.

Note

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
Example: SPAN Session Traffic on a LAN SRIOV

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

Note

This scenario is applicable only to the Cisco ENCS.

VM deployment for VF-VF scenario:

CentOS_SRIOV, C3, and C5 are CentOS VMs with SRIOV support.

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

SPAN destination and source:

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

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

```bash
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_all true
  statistics "tx_bytes=0, tx_packets=0"
```

nfvis# show monitor session 11
NUMBER STATUS
------------------------
11 CREATE_SUCCESS

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:

```
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.

### Example: Debug Built-in Switch Issues

To monitor traffic problems related to built-in switch on an internal interface:

The regular traffic flow between int-LAN and GE1/0 is:

```
GE0-0-- vnic1---(VM)--vnic2--intLAN--GE1/0
```

The NFVIS portal has the capability to capture packets. In the network diagram, right click on any vertical line and a window pops up where you can select the duration of the capture. The packet capture starts on the selected interface link. At the end of the capture, a file is downloaded to your local machine. SPAN sessions are supported on both NFVIS host and the built-in switch.

The following is an example of SPAN in built-in switch:

1. From NFVIS system shell-access, get the password which can be used later.

   ```
   cd /opt/switch-confd/
   python decrypt_switch.py
   
   <it will print out a string, it will be the password you need to use later>
   8H7)gR348V4Byq4mwjiNt
   ```

2. From Cisco IMC complete the challenge-response authentication:

   ```
   # connect debug-shell
   # sldp
   login <hit return>
   it will print out the challenge string
   enter the respond string
   # switch-con ge
   user-name:cisco
   password: <enter the string we get from nfvis system shell>
   
   User Name:cisco
   Password:***************<this is the password you get from step 1 above>
   ```
3. To configure SPAN specify the source and distribution interface and direction of the packet flow. For example, if you want to mirror XG2 output packet to Ge0, connect an external packet capture tool in GE1/0 and you will see all packets flow from internal XG2. In the following example, the traffic between int_LAN and GE1/0 go through internal interface XE2 and traffic for XE2 interface is monitored:

nfvis(config)#monitor session 1 source interface XG 2 out
nfvis(config)#monitor session 1 destination interface GigabitEthernet 0
remember to unconfig it once you finish debugging.
nfvis(config)#no monitor session 1 destination
nfvis(config)#no monitor session 1 source interface XG 2

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>
Appendix

• Event Notifications, on page 285
• Syslog Messages, on page 319

Event Notifications

nfvisEvent

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAN_DHCP_RENEW</td>
<td>DHCP renew operation is performed.</td>
<td>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;&lt;notification xmlns=&quot;urn:ietf:params:xml:ns:netconf:notification:1.0&quot;&gt; &lt;eventTime&gt;2017-04-26T18:06:46.142089+00:00&lt;/eventTime&gt; &lt;nfvisEvent xmlns=&quot;http://www.cisco.com/nfvis/notifier&quot;&gt; &lt;user_id&gt;NA&lt;/user_id&gt; &lt;config_change&gt;false&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;Wan DHCP IP address is being renewed&lt;/status_message&gt; &lt;details&gt;NA&lt;/details&gt; &lt;event_type&gt;WAN_DHCP_RENEW&lt;/event_type&gt; &lt;/nfvisEvent&gt; &lt;/notification&gt;</td>
</tr>
</tbody>
</table>

DHCPrenew operation is performed.
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| 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>
<style>SUCCESS</style>
<style_message>Network created successfully</style_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>
<style>SUCCESS</style>
<style_message>Network updated successfully</style_message>
<event_type>NETWORK_UPDATE</event_type>
<network_name>testn1</network_name>
<network_bridge/>
<network_sriov/>
<network_vlan/>
<network_trunk/>
</nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| NETWORK_DELETE     | A network is deleted. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-09-22T12:42:03.391986+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_message>Network deleted successfully</status_message>
<event_type>NETWORK_DELETE</event_type>
<network_name>testn1</network_name>
<network_bridge/>
<network_sriov/>
<network_vlan/>
<network_trunk/>
</nfvisEvent>
</notification> |
| UPGRADE_REGISTER   | System upgrade is registered. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T15:57:50.434636+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Upgrade package registration successful:
Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg</status_message>
<event_type>UPGRADE_REGISTER</event_type>
</nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| UPGRADE_APPLY           | System upgrade is applied. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T16:02:43.885516+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Upgrade Process: In Progress</status_message>
<event_type>UPGRADE_APPLY</event_type>
</nfvisEvent>
</notification> |
| ROTATED_LOGS_DELETE     | Rotated logs older than 30 days are deleted by the system. | <?xml version="1.0" encoding="UTF-8"?>
<rpc-reply
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="1">
<ok/>
</rpc-reply>
<?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-04-26T17:38:10.321152+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>NA</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Deleted rotated logs from archive older than 30 days</status_message>
<details>NA</details>
<event_type>ROTATED_LOGS_DELETE</event_type>
</nfvisEvent>
</notification> |
## Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
**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"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-04-26T17:42:10.321152+00:00</eventTime>
  <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
    <user_id>NA</user_id>
    <config_change>true</config_change>
    <transaction_id>0</transaction_id>
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>Rotated logs had exceeded 2G, older logs have been deleted to make space</status_message>
    <details>NA</details>
    <event_type>ROTATED_LOGS_DELETE</event_type>
  </nfvisEvent>
</notification>
```  

**REBOOT** | system reboot | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2018-04-26T09:37:47.387525+00:00</eventTime>
  <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
    <user_id>admin</user_id>
    <config_change>false</config_change>
    <transaction_id>0</transaction_id>
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>System will be rebooted</status_message>
    <details>NA</details>
    <event_type>REBOOT</event_type>
    <hostname>NFVIS</hostname>
  </nfvisEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| SHUTDOWN         | system shutdown                      | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
  <nfvisEvent
  xmlns="http://www.cisco.com/nfvis/notifier">
    <user_id>admin</user_id>
    <config_change>false</config_change>
    <transaction_id>0</transaction_id>
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>System will be shutdown</status_message>
    <details>NA</details>
    <event_type>SHUTDOWN</event_type>
    <hostname>NFVIS</hostname>
  </nfvisEvent>
</notification> |
| SECURE_OVERLAY_CREATING | create secure overlay | <notification
<eventTime> 2018-11-02T04:23:02.641317+00:00
<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 initial creation. Active local bridge: wan-br</status_message>
    <details>NA</details>
    <event_type>SECURE_OVERLAY_CREATING</event_type>
    <hostname>NFVIS</hostname>
  </nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| SECURE_OVERLAY_UP       | Secure Overlay is UP | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Secure Overlay mgmthub up. Active bridge: wan-br</status_message>
<details>Secure overlay initial creation</details>
<event_type>SECURE_OVERLAY_UP</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification> |
| WAN_DHCP_SWITCHOVER     | WAN bridge toggle   | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
<nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
<user_id>admin</user_id>
<config_change>true</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Switch over to bridge wan2-br for auto DHCP enablement successful</status_message>
<details>NA</details>
<event_type>WAN_DHCP_SWITCHOVER</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| WAN_DHCP_TOGGLE_END  | WAN bridge toggle    | ```xml version="1.0" encoding="UTF-8"?><notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
|                      |                      | <eventTime>2018-04-26T09:47:06.066264+00:00</eventTime>
|                      |                      |   <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
|                      |                      |     <user_id>admin</user_id>
|                      |                      |     <config_change>true</config_change>
|                      |                      |     <transaction_id>0</transaction_id>
|                      |                      |     <status>SUCCESS</status>
|                      |                      |     <status_code>0</status_code>
|                      |                      |     <status_message>Disabling bridge toggle for auto DHCP enablement.</status_message>
|                      |                      |     <details>NA</details>
|                      |                      |     <event_type>WAN_DHCP_TOGGLE_END</event_type>
|                      |                      |     <severity>INFO</severity>
|                      |                      |     <hostname>NFVIS</hostname>
|                      |                      |   </nfvisEvent>
|                      |                      | </notification>                                                                                                                                             |
| ROUTE_DISTRIBUTION_START | To start route distribution | ```xml version="1.0" encoding="UTF-8"?><notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
|                      |                      | <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
|                      |                      |   <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier">
|                      |                      |     <user_id>admin</user_id>
|                      |                      |     <config_change>true</config_change>
|                      |                      |     <transaction_id>0</transaction_id>
|                      |                      |     <status>SUCCESS</status>
|                      |                      |     <status_code>0</status_code>
|                      |                      |     <status_message>Route Distribution initial creation. Neighbor Address: 172.25.221.106</status_message>
|                      |                      |     <details>NA</details>
|                      |                      |     <event_type>ROUTE_DISTRIBUTION_START</event_type>
|                      |                      |     <severity>INFO</severity>
|                      |                      |     <hostname>NFVIS</hostname>
|                      |                      |   </nfvisEvent>
<p>|                      |                      | &lt;/notification&gt;                                                                                                                                             |</p>
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE DISTRIBUTION DOWN</td>
<td>Route distribution is down</td>
<td>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;notification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xmlns=&quot;urn:ietf:params:xml:ns:netconf:notification:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;eventTime&gt;2019-03-15T21:46:28.034133+00:00&lt;/eventTime&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;nfvisEvent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xmlns=&quot;<a href="http://www.cisco.com/nfvis/notifier%22%3E">http://www.cisco.com/nfvis/notifier&quot;&gt;</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;user_id&gt;admin&lt;/user_id&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;config_change&gt;true&lt;/config_change&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;transaction_id&gt;0&lt;/transaction_id&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status&gt;SUCCESS&lt;/status&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status_code&gt;0&lt;/status_code&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status_message&gt;Neighbor Address: 172.25.221.106&lt;/status_message&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;details&gt;NA&lt;/details&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;event_type&gt;ROUTE DISTRIBUTION DOWN&lt;/event_type&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;severity&gt;INFO&lt;/severity&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;hostname&gt;NFVIS&lt;/hostname&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/nfvisEvent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/notification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROUTE DISTRIBUTION ERROR</th>
<th>Route distribution in error</th>
<th>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;notification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xmlns=&quot;urn:ietf:params:xml:ns:netconf:notification:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;eventTime&gt;2019-03-15T21:46:28.034133+00:00&lt;/eventTime&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;nfvisEvent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xmlns=&quot;<a href="http://www.cisco.com/nfvis/notifier%22%3E">http://www.cisco.com/nfvis/notifier&quot;&gt;</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;user_id&gt;admin&lt;/user_id&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;config_change&gt;true&lt;/config_change&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;transaction_id&gt;0&lt;/transaction_id&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status&gt;SUCCESS&lt;/status&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status_code&gt;0&lt;/status_code&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;status_message&gt;Neighbor Address: 172.25.221.106&lt;/status_message&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;details&gt;NA&lt;/details&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;event_type&gt;ROUTE DISTRIBUTION_ERROR&lt;/event_type&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;severity&gt;INFO&lt;/severity&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;hostname&gt;NFVIS&lt;/hostname&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/nfvisEvent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/notification</td>
</tr>
<tr>
<td>Event Type</td>
<td>Notification Trigger</td>
<td>Notification Output Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ROUTE_DISTRIBUTION_DELETE  | Route distribution deleted | ```<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent
 xmlns="http://www.cisco.com/nfvis/notifier">
 <user_id>admin</user_id>
 <config_change>true</config_change>
 <transaction_id>0</transaction_id>
 <status>SUCCESS</status>
 <status_code>0</status_code>
 <status_message>All Neighbor Addresses deleted</status_message>
 <details>NA</details>
 <event_type>ROUTE_DISTRIBUTION_DELETE</event_type>
 <severity>INFO</severity>
 <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
| ROUTE_DISTRIBUTION_UP      | Route distribution up      | ```<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
<nfvisEvent
 xmlns="http://www.cisco.com/nfvis/notifier">
 <user_id>admin</user_id>
 <config_change>true</config_change>
 <transaction_id>0</transaction_id>
 <status>SUCCESS</status>
 <status_code>0</status_code>
 <status_message>Neighbor Address: 172.25.221.106</status_message>
 <details>NA</details>
 <event_type>ROUTE_DISTRIBUTION_UP</event_type>
 <severity>INFO</severity>
 <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>``` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| OVS_DPDK_SUCCESS    | Enable DPDK          | `<xml version="1.0" encoding="UTF-8"?>
  <notification
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
    <nfvisEvent
      xmlns="http://www.cisco.com/nfvis/notifier">
      <user_id>admin</user_id>
      <config_change>false</config_change>
      <transaction_id>0</transaction_id>
      <status>SUCCESS</status>
      <status_code>0</status_code>
      <status_message>OVS-DPDK enabled</status_message>
      <details>NA</details>
    </nfvisEvent>
  </notification>` |
| OVS_DPDK_FAILURE    | DPDK failure         | `<xml version="1.0" encoding="UTF-8"?>
  <notification
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
    <nfvisEvent
      xmlns="http://www.cisco.com/nfvis/notifier">
      <user_id>admin</user_id>
      <config_change>false</config_change>
      <transaction_id>0</transaction_id>
      <status>SUCCESS</status>
      <status_code>0</status_code>
      <status_message>Unable to allocate CPU</status_message>
      <details>NA</details>
    </nfvisEvent>
  </notification>` |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| PNIC_SRIOV_ENABLE   | Enable SR-IOV to 2 vFs                | `<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>Physical Interface: eth0-1
Number of VFs 2</status_message>
<details>NA</details>
<event_type>PNIC_SRIOV_ENABLE</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification>` |
| PNIC_SRIOV_DISABLE  | Disable SR-IOV                        | `<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>Physical Interface: eth0-1
Number of VFs 0</status_message>
<details>NA</details>
<event_type>PNIC_SRIOV_DISABLE</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>
<tbody>
<tr>
<td>Event Type</td>
<td>Notification Trigger</td>
<td>Notification Output Example</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| PNIC_SRIOV_UPDATE   | Set switchmode to veb/vepa in error           | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime>
  <nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
    <user_id>admin</user_id>
    <config_change>true</config_change>
    <transaction_id>0</transaction_id>
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>Physical Interface: eth0-1 Switchmode vepa</status_message>
    <details>NA</details>
    <event_type>PNIC_SRIOV_UPDATE</event_type>
    <severity>INFO</severity>
    <hostname>NFVIS</hostname>
  </nfvisEvent>
</notification> |
| PROMISC_ENABLED     | Enable promiscuous mode                       | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2020-03-27T19:58:52.333682+00:00</eventTime>
  <nfvisEvent
xmlns="http://www.cisco.com/nfvis/notifier">
    <user_id>admin</user_id>
    <config_change>false</config_change>
    <transaction_id>0</transaction_id>
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>GE0-0: promiscuous mode enabled</status_message>
    <details>GE0-0: promiscuous mode enabled</details>
    <event_type>PROMISC_ENABLED</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>
| FIPS_CONFIGURED_REBOOT_TO_ENABLE | When FIPS mode is configured but a reboot is required to enable it. | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2020-06-07T21:26:06.231166+00:00</eventTime>
<nfvisEvent
<user_id>admin</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Fips-mode configured and a manual reboot is required to enable it</status_message>
<details>NA</details>
<event_type>FIPS_CONFIGURED_REBOOT_TO_ENABLE</event_type>
<severity>INFO</severity>
<hostname>NFVIS</hostname>
</nfvisEvent>
</notification> |
| FIPS_ENABLED        | FIPS mode is enabled.                                      | <?xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2020-06-07T21:26:06.231166+00:00</eventTime>
<nfvisEvent
<user_id>admin</user_id>
<config_change>false</config_change>
<transaction_id>0</transaction_id>
<status>SUCCESS</status>
<status_code>0</status_code>
<status_message>Fips-mode successfully enabled</status_message>
<details>NA</details>
<event_type>FIPS_ENABLED</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>
| FIPS_FAILED     |                      | `<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2020-06-07T21:26:06.231166+00:00</eventTime>
<nfvisEvent
  <user_id>admin</user_id>
  <config_change>false</config_change>
  <transaction_id>0</transaction_id>
  <status>SUCCESS</status>
  <status_code>0</status_code>
  <status_message>Fips-mode configuration is unsuccessful</status_message>
  <details>NA</details>
  <event_type>FIPS_FAILED</event_type>
  <severity>INFO</severity>
  <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>` |
| FIPS_DISABLED   | FIPS mode is disabled. | `<xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2020-06-07T21:26:06.231166+00:00</eventTime>
<nfvisEvent
  <user_id>admin</user_id>
  <config_change>false</config_change>
  <transaction_id>0</transaction_id>
  <status>SUCCESS</status>
  <status_code>0</status_code>
  <status_message>Fips-mode successfully disabled</status_message>
  <details>NA</details>
  <event_type>FIPS_DISABLED</event_type>
  <severity>INFO</severity>
  <hostname>NFVIS</hostname>
</nfvisEvent>
</notification>` |
### 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>2016-11-17T12:30:76+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>
</vmlcEvent>
</notification>` |

### 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>2016-11-17T12:30:76+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>
</vmlcEvent>
</notification>` |

---

**Cisco Enterprise Network Function Virtualization Infrastructure Software Configuration Guide, Release 4.x**

302
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| DELETE_IMAGE  | The VM image is unregistered | <?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:14:51.169+00:00</eventTime>
<vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Image deletion completed successfully.</status_message>
<image>isrv-universalk9.16.03.01.tar.gz</image>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
<type>DELETE_IMAGE</type>
</event>
</vmlcEvent>
</notification> |
| CREATE_FLAVOR | A flavor is created. | <?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:12:29.685+00:00</eventTime>
<vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Flavor creation completed successfully.</status_message>
<flavor>ISRv-small</flavor>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
<type>CREATE_FLAVOR</type>
</event>
</vmlcEvent>
</notification> |
| DELETE_FLAVOR | A flavor is deleted. | <?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:14:51.425+00:00</eventTime>
<vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Flavor deletion completed successfully.</status_message>
<flavor>ISRv-small</flavor>
<vm_source></vm_source>
<vm_target></vm_target>
<event>
<type>DELETE_FLAVOR</type>
</event>
</vmlcEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_DEPLOYED | The VM is deployed.  | <?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:19:16.927+00:00</eventTime>
<vmlcEvent
 xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>VIM Driver: VM successfully created, VM Name:
[SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c]</status_message>
<depname>1479341445</depname>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
<interfaces>
<intface>
<nicid>0</nicid>
<port_id>vnet0</port_id>
<network>int-mgmt-net</network>
<subnet>N/A</subnet>
<ip_address>10.20.0.2</ip_address>
<mac_address>52:54:00:31:c5:7f</mac_address>
<netmask>255.255.255.0</netmask>
<gateway>10.20.0.1</gateway>
</interface>
<intface>
<nicid>1</nicid>
<port_id>vnet1</port_id>
<network>wan-net</network>
<subnet>N/A</subnet>
<mac_address>52:54:00:59:52:41</mac_address>
<netmask>255.255.255.0</netmask>
<gateway>172.19.181.152</gateway>
</interface>
</interfaces>
</vm_source>
</vm_target>
</event>
</vmlcEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| 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>
    <depname>1479341445</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
      <interfaces>
        <interface>
          <nicid>0</nicid>
          <port_id>vnet0</port_id>
          <network>int-mgmt-net</network>
          <subnet>N/A</subnet>
          <ip_address>10.20.0.2</ip_address>
          <mac_address>52:54:00:31:c5:7f</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>10.20.0.1</gateway>
        </interface>
        <interface>
          <nicid>1</nicid>
          <port_id>vnet1</port_id>
          <network>wan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:59:52:41</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
      </interfaces>
    </vm_source>
    <vm_target></vm_target>
    <event>
      <type>VM_ALIVE</type>
    </event>
  </vmlcEvent>
</notification> |
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_UNDEPLOYED    | The VM is undeployed | <?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>
<depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target>
</vm_target>
</vmlcEvent>
</notification>
<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 [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c].</status_message>
<depname>1479341445</depname>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<svcid>NULL</svcid>
<depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target></vm_target>
<event>
<type>VM_STOPPED</type>
</event>
</vmlcEvent>
</notification> |
<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 [SystemAdminTenTa_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c].</status_message>
    <depname>1479341445</depname>
    <tenant>admin</tenant>
    <tenant_id>AdminTenantId</tenant_id>
    <svcid>NUL</svcid>
    <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
    </vm_source>
    <vm_target></vm_target>
    <event>
      <type>VM_STARTED</type>
    </event>
  </vmlcEvent>
</notification> |
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
**VM_REBOOTED** | The VM is rebooted per VM action request. | ```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T17:36:56.5+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-f0fd2efffd7d]</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>
<vm_source>
<vmid>d918a3b1-f2a9-4065-9d8e-2135b0a37d87</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target>
<event>
<type>VM_REBOOTED</type>
</event>
</vm_target>
</vmlcEvent>
</notification>```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_RECOVERY_INIT | A monitored VM is not reachable. | <xml version="1.0" encoding="UTF-8"?>
<notification
xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2016-11-17T16:27:51.627+00:00</eventTime>
<vmlcEvent
xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Recovery event for VM [SystemAdminTenantry_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] triggered. Processing Auto healing. Proceeding with Recovery.</status_message>
<depid>1479328919</depid>
<tenant_id>AdminTenantId</tenant_id>
<svcid>NULL</svcid>
<depid>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target>
<event>
<type>VM_RECOVERY_INIT</type>
</event>
</vm_target>
</vmlcEvent>
</notification>
### Event Type | Notification Trigger | Notification Output Example
--- | --- | ---
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 [SystemAdminTenant_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] is being rebooted. _
</status_message>
<depname>1479328919</depname>
<tenant>admin</tenant>
<tenant_id>AdminTenantId</tenant_id>
<svcid>NULL</svcid>
<depid>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid>
<hostid>NFVIS</hostid>
<hostname>NFVIS</hostname>
</vm_source>
<vm_target>
<event>
<type>VM_RECOVERY_REBOOT</type>
</event>
</vm_target>
</vmlcEvent>
</notification>```
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</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>
      <interfaces>
        <interface>
          <nicid>0</nicid>
          <port_id>vnet0</port_id>
          <network>int-mgmt-net</network>
          <subnet>N/A</subnet>
          <ip_address>10.20.0.2</ip_address>
          <mac_address>52:54:00:7b:3f:de</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>10.20.0.1</gateway>
        </interface>
        <interface>
          <nicid>1</nicid>
          <port_id>vnet1</port_id>
          <network>wan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:96:8a:4d</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
      </interfaces>
    </vm_source>
    <vm_target>
      <vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid>
      <hostid>NFVIS</hostid>
      <hostname>NFVIS</hostname>
      <interfaces>
        <interface>
          <nicid>0</nicid>
          <port_id>vnet0</port_id>
          <network>int-mgmt-net</network>
          <subnet>N/A</subnet>
          <ip_address>10.20.0.2</ip_address>
          <mac_address>52:54:00:7b:3f:de</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>10.20.0.1</gateway>
        </interface>
        <interface>
          <nicid>1</nicid>
          <port_id>vnet1</port_id>
          <network>wan-net</network>
          <subnet>N/A</subnet>
          <mac_address>52:54:00:96:8a:4d</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
      </interfaces>
    </vm_target>
    <event>
      <type>VM_RECOVERY_COMPLETE</type>
    </event>
  </vmlcEvent>
</notification>
```
<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>
<svidc>NULL</svidc>
<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>
<svidc>NULL</svidc>
<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-b84d-c0a2c8a481b1</vmid>
</vm_source>
<event>
?type>VM_UPDATED</type>
</event>
<vmlcEvent
xmlns='http://www.cisco.com/nfvis/vm_lifecycle'>
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>VM is resized with flavor
[ISRv-medium].</status_message>
<user_name>admin</user_name>
<depname>1512766000</depname>
<tenant>admin</tenant>
<tenant_id>adminUUID</tenant_id>
<depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid>
<vm_group>ROUTER</vm_group>
<vm_source>
<vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid>
</vm_source>
<event>
?type>VM_UPDATED</type>
</event>
</notification>
### 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>2017-12-08T00:50:39.06+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-c8c65b9ef70f</depid>
    <event>
      <type>SERVICE_UPDATED</type>
    </event>
  </vmlcEvent>
</notification>
```
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Trigger</th>
<th>Notification Output Example</th>
</tr>
</thead>
</table>
| VM_UPDATED | VNIC is added, deleted or updated. | }
<?xml version="1.0" encoding="UTF-8"?>
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-12-08T02:10:56.184+00:00</eventTime>
  <vmlcEvent
    xmlns='http://www.cisco.com/nfvis/vm_lifecycle'>
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Added 1 interface: [managed, net=my-net-1, nicid=3]
    Updated 2 interface: [managed, net=lan-net, nicid=1],[managed,
    net=wan-net, nicid=2]</status_message>
    <user_name>admin</user_name>
    <depid>1512766000</depid>
    <tenant>admin</tenant>
    <tenant_id>adminUUID</tenant_id>
    <vm_group>ROUTER</vm_group>
    <vm_source>
      <vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname>
      <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>52:54:00:73:30:1c</mac_address>
          <netmask>255.255.255.0</netmask>
          <gateway>172.19.181.152</gateway>
        </interface>
        <interface>
          <nicid>3</nicid>
          <type>virtual</type>
          <port_id>vnic3</port_id>
      </interfaces>
    </vm_source>
  </vmlcEvent>
</notification>
### Syslog Messages

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Notification Output Example</th>
<th>Syslog Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK_CREATE</td>
<td><code>&lt;network&gt;my-net-1&lt;/network&gt; &lt;subnet&gt;N/A&lt;/subnet&gt; &lt;mac_address&gt;52:54:00:66:b5:c1&lt;/mac_address&gt; &lt;/interface&gt; &lt;/interfaces&gt; &lt;/vm_source&gt; &lt;event&gt; &lt;type&gt;VM_UPDATED&lt;/type&gt; &lt;/event&gt; &lt;/vmlcEvent&gt;</code></td>
<td>nfvis %SYS-6-NETWORK_CREATE: Network my-net created successfully</td>
</tr>
<tr>
<td>NETWORK_UPDATE</td>
<td>nfvis %SYS-6-NETWORK_UPDATE: Network my-net updated successfully</td>
<td></td>
</tr>
<tr>
<td>NETWORK_DELETE</td>
<td>nfvis %SYS-6-NETWORK_DELETE: Network my-net deleted successfully</td>
<td></td>
</tr>
<tr>
<td>BRIDGE_CREATE</td>
<td>nfvis %SYS-6-BRIDGE_CREATE: Bridge created successfully: my-bridge</td>
<td></td>
</tr>
<tr>
<td>BRIDGE_UPDATE</td>
<td>nfvis %SYS-6-BRIDGE_UPDATE: Updated bridge successfully: my-bridge</td>
<td></td>
</tr>
<tr>
<td>BRIDGE_DELETE</td>
<td>nfvis %SYS-6-BRIDGE_DELETE: Bridge deleted successfully: my-bridge</td>
<td></td>
</tr>
<tr>
<td>WAN_DHCP_RENEW</td>
<td>nfvis %SYS-6-WAN_DHCP_RENEW: wan-br DHCP IP address is being renewed</td>
<td></td>
</tr>
<tr>
<td>BRIDGE_DHCP_RENEW</td>
<td>nfvis %SYS-6-BRIDGE_DHCP_RENEW: Bridge DHCP IP address is being renewed</td>
<td></td>
</tr>
<tr>
<td>MGMT_DHCP_RENEW</td>
<td>nfvis %SYS-6-MGMT_DHCP_RENEW: wan-br DHCP IP address is being renewed</td>
<td></td>
</tr>
<tr>
<td>INTF_STATUS_CHANGE</td>
<td>nfvis %SYS-6-INTF_STATUS_CHANGE: Interface eth0, changed state to up</td>
<td></td>
</tr>
<tr>
<td>UPGRADE_REGISTER</td>
<td>nfvis %SYS-6-UPGRADE_REGISTER: Upgrade package registration successful:</td>
<td>Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg</td>
</tr>
<tr>
<td>UPGRADE_APPLY</td>
<td>nfvis %SYS-6-UPGRADE_APPLY: Upgrade Process: In Progress</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RBAC_USER_CREATE</td>
<td>create a new user</td>
<td>nfvis %SYS-6-RBAC_USER_CREATE: Created user admin as administrators successfully</td>
</tr>
<tr>
<td>RBAC_USER_PASSWORD_UPDATE</td>
<td>change user's password</td>
<td>nfvis %SYS-6-RBAC_USER_PASSWORD_UPDATE: Set admin password successfully</td>
</tr>
<tr>
<td>RBAC_USER_ROLE_UPDATE</td>
<td>change user's role</td>
<td>nfvis %SYS-6-RBAC_USER_ROLE_UPDATE: Modified user: somebody successfully</td>
</tr>
<tr>
<td>RBAC_USER_DELETE</td>
<td>delete a user</td>
<td>nfvis %SYS-6-RBAC_USER_DELETE: Deleted rbac user successfully: somebody</td>
</tr>
<tr>
<td>RBAC_USERS_INACTIVATED</td>
<td>disable the user</td>
<td>nfvis %SYS-6-RBAC_USERS_INACTIVATED: Following users have been marked as INACTIVE, [user1, user2]. Please take necessary action.</td>
</tr>
<tr>
<td>RBAC_USER_ACTIVATED</td>
<td>activate the user</td>
<td>nfvis %SYS-6-RBAC_USER_ACTIVATED: Modified user user1 successfully.</td>
</tr>
<tr>
<td>RBAC_PWD_EXPIRED</td>
<td>password expired</td>
<td>nfvis %SYS-6-RBAC_PWD_EXPIRED: User user1's password is older than 60 days. Please reset password.</td>
</tr>
<tr>
<td>RBAC_LOGIN_FAILURE</td>
<td>invalid user login</td>
<td>nfvis %SYS-3-RBAC_LOGIN_FAILURE: Login with invalid username from maapi failed</td>
</tr>
<tr>
<td>SECURITY_SERVER_CREATE</td>
<td>create server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_CREATE: TACACS+ server config created successfully.</td>
</tr>
<tr>
<td>SECURITY_SERVER_UPDATE</td>
<td>update server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_UPDATE: TACACS+ server configuration updated successfully.</td>
</tr>
<tr>
<td>SECURITY_SERVER_DELETE</td>
<td>delete server config</td>
<td>nfvis %SYS-6-SECURITY_SERVER_DELETE: TACACS+ server deleted successfully.</td>
</tr>
<tr>
<td>AAA_TYPE_CREATE</td>
<td>create AAA authentication type</td>
<td>nfvis %SYS-6-AAA_TYPE_CREATE: AAA authentication type TACACS created successfully.</td>
</tr>
<tr>
<td>AAA_TYPE_UPDATE</td>
<td>update AAA authentication type</td>
<td>nfvis %SYS-6-AAA_TYPE_UPDATE: AAA authentication type TACACS+ updated successfully. AAA authentication updated to use TACACS+ server</td>
</tr>
<tr>
<td>RECREATE_CERTIFICATE</td>
<td>recreate self-sign certificate</td>
<td>nfvis %SYS-6-RECREATE_CERTIFICATE: Self Signed Certificate re-created. Application connection may become temporarily unavailable.</td>
</tr>
<tr>
<td>CERT_CSR_CREATE</td>
<td>create a CSR file</td>
<td>nfvis %SYS-6-CERT_CSR_CREATE: signing-request created /data/intdatastore/download/nfvis.csr</td>
</tr>
<tr>
<td>CERT_SWITCH_CERT</td>
<td>switch to use different certificate</td>
<td>nfvis %SYS-6-CERT_SWITCH_CERT: switch certificate from ca-signed to self-signed.</td>
</tr>
<tr>
<td>CERT_CA_CERT_INSTALL</td>
<td>install CA signed certificate</td>
<td>nfvis %SYS-6-CERT_CA_CERT_INSTALL: ca-signed certificate file:// installed</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REBOOT</td>
<td>system reboot</td>
<td>nfvis %SYS-6-REBOOT: System will be rebooted</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>system shutdown</td>
<td>nfvis %SYS-6-SHUTDOWN: System will be shutdown</td>
</tr>
<tr>
<td>LOGGING_FAILURE</td>
<td>logging failure</td>
<td>nfvis %SYS-6-LOGGING_FAILURE: Unable to write to log file nfvis_config.log. Log message: log_config.CONFIG_LOGGER: File not found.</td>
</tr>
<tr>
<td>DISK_SPACE_ALMOST_FULL</td>
<td>disk space almost full</td>
<td>nfvis %SYS-6-DISK_SPACE_ALMOST_FULL: 'lv_data' currently occupies 95% of available disk space, which is more than or equal to the threshold of 90%.</td>
</tr>
<tr>
<td>ROTATED_LOGS_DELETE</td>
<td>delete rotated logfiles when accumulated rotated log files reach 2GB</td>
<td>nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days</td>
</tr>
<tr>
<td>TIME_UPDATE</td>
<td>Change system time manually</td>
<td>nfvis %SYS-6-TIME_UPDATE: Manual time updated successfully Manual time is now set to 2018-04-26 11:43:00</td>
</tr>
<tr>
<td>TIMEZONE_UPDATE</td>
<td>Change system timezone</td>
<td>nfvis %SYS-6-TIMEZONE_UPDATE: Timezone updated successfully. Timezone is now set to US/Eastern</td>
</tr>
<tr>
<td>FILE_COPY_STATUS</td>
<td>copy status of file</td>
<td>nfvis %SYS-6-FILE_COPY_STATUS: hostaction.py Copied Successfully.</td>
</tr>
<tr>
<td>CREATE_IMAGE</td>
<td>create image</td>
<td>nfvis %SYS-6-CREATE_IMAGE: Image creation successful: TinyLinux.tar.gz</td>
</tr>
<tr>
<td>DELETE_IMAGE</td>
<td>delete image</td>
<td>nfvis %SYS-6-DELETE_IMAGE: Image deletion successful: TinyLinux.tar.gz</td>
</tr>
<tr>
<td>CREATE_FLAVOR</td>
<td>create flavor</td>
<td>nfvis %SYS-6-CREATE_FLAVOR: Profile creation successful: small</td>
</tr>
<tr>
<td>DELETE_FLAVOR</td>
<td>delete flavor</td>
<td>nfvis %SYS-6-DELETE_FLAVOR: Profile deletion successful: small</td>
</tr>
<tr>
<td>VM_DEPLOYED</td>
<td>vm deployment</td>
<td>nfvis %SYS-6-VM_DEPLOYED: VM deployment successful: SystemAdmin: ROUTER_0_di6733e10784e68eb223ca036c</td>
</tr>
<tr>
<td>VM_ALIVE</td>
<td>vm alive</td>
<td>nfvis %SYS-6-VM_ALIVE: VM active successful: SystemAdmin: ROUTER_0_di6733e10784e68eb223ca036c</td>
</tr>
<tr>
<td>SERVICE_ALIVE</td>
<td>service alive</td>
<td>nfvis %SYS-6-SERVICE_ALIVE: Service group deployment completed successfully!</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>VM_UNDEPLOYED</td>
<td>vm undeployed</td>
<td><code>nfvis%SYS-6-VM_UNDEPLOYED: VM undeployment successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c SERVICE_UNDEPLOYED service undeployed nfvis%SYS-6-SERVICE_UNDEPLOYED: Service group undeployment completed successfully</code></td>
</tr>
<tr>
<td>VM_UPDATED (update flavor)</td>
<td>vm updated</td>
<td><code>nfvis%SYS-6-VM_UPDATED: VM update successful: VM is resized with flavor [ISRv-medium].</code></td>
</tr>
<tr>
<td>VM_UPDATED (vnic add / delete / update)</td>
<td>vm updated</td>
<td><code>nfvis%SYS-6-VM_UPDATED: VM update successful: Added 1 interface: [managed, net=my-net-1, nicid=3] Updated 2 interface: [managed, net=lan-net, nicid=1],[managed, net=wan-net, nicid=2]</code></td>
</tr>
<tr>
<td>SERVICE_UPDATED</td>
<td>service updated</td>
<td><code>nfvis%SYS-6-SERVICE_UPDATED: Service group update completed successfully</code></td>
</tr>
<tr>
<td>VM_STOPPED</td>
<td>vm stopped</td>
<td><code>nfvis%SYS-6-VM_STOPPED: VM stop successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_STARTED</td>
<td>vm started</td>
<td><code>nfvis%SYS-6-VM_STARTED: VM start successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_REBOOTED</td>
<td>vm rebooted</td>
<td><code>nfvis%SYS-6-VM_REBOOTED: VM reboot successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_RECOVERY_INIT</td>
<td>vm recovery initiation</td>
<td><code>nfvis%SYS-6-VM_RECOVERY_INIT: VM recovery initiation successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_RECOVERY_REBOOT</td>
<td>vm recovery reboot</td>
<td><code>nfvis%SYS-6-VM_RECOVERY_REBOOT: VM recovery reboot successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_RECOVERY_COMPLETE</td>
<td>vm recovery complete</td>
<td><code>nfvis%SYS-6-VM_RECOVERY_COMPLETE: VM recovery successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_MONITOR_UNSET</td>
<td>vm monitoring unset</td>
<td><code>nfvis%SYS-6-VM_MONITOR_UNSET: Unsetting VM monitoring successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>VM_MONITOR_SET</td>
<td>vm monitoring set</td>
<td><code>nfvis%SYS-6-VM_MONITOR_SET: Setting VM monitoring successful: SystemAdminTena_ROUTER_0_df6733c1-0768-4ac6-8dc6-223eab9c</code></td>
</tr>
<tr>
<td>ROTATED_LOGS_DELETE (When logs older than 30 days are present)</td>
<td>delete rotated logs</td>
<td><code>nfvis%SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days</code></td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ROTATED LOGS DELETE</td>
<td>delete rotated logs</td>
<td>nfvis %SYS-6-ROTATED_LOGS_DELETE: Rotated logs had exceeded 2G, older logs have been deleted to make space</td>
</tr>
<tr>
<td>CMC_PASSWORD_UPDATE</td>
<td>cimc password update operation</td>
<td>nfvis %SYS-6-CMC_PASSWORD_UPDATE: CMC password change is successful</td>
</tr>
<tr>
<td>BIOS_PASSWORD_UPDATE</td>
<td>bios password update operation</td>
<td>nfvis %SYS-6-BIOS_PASSWORD_UPDATE: BIOS password change is successful</td>
</tr>
<tr>
<td>SECURE_OVERLAY_CREATING</td>
<td>create secure overlay</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_CREATING: Secure Overlay mgmt hub initial creation. Active local bridge: wan-br</td>
</tr>
<tr>
<td>SECURE_OVERLAY_UP</td>
<td>secure overlay is up</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_UP: Secure Overlay mgmt hub up. Active bridge: wan-br Secure Overlay up after network interruption</td>
</tr>
<tr>
<td>SECURE_OVERLAY_DELETE</td>
<td>secure overlay is deleted</td>
<td>nfvis %SYS-6-SECURE_OVERLAY_DELETE: Secure Overlay deleted</td>
</tr>
<tr>
<td>SECURE_OVERLAY_ERROR</td>
<td>error in secure overlay</td>
<td>nfvis %SYS-3-SECURE_OVERLAY_ERROR: Secure Overlay mgmt hub creation in error. Active bridge: wan-br Secure overlay initial creation nfvis %SYS-3-SECURE_OVERLAY_ERROR: Secure Overlay mgmt hub creation in error. Active bridge: wan-br Cannot ping remote system ip address 10.0.0.1</td>
</tr>
<tr>
<td>WAN_DHCP_SWITCHOVER</td>
<td>WAN bridge toggle</td>
<td>nfvis %SYS-6-WAN_DHCP_SWITCHOVER: Switch over to bridge wan-br for auto DHCP enablement successful</td>
</tr>
<tr>
<td>WAN_DHCP_TOGGLE_END</td>
<td>WAN bridge toggle</td>
<td>nfvis %SYS-6-WAN_DHCP_TOGGLE_END: Disabling bridge toggle for auto DHCP enablement.</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_DOWN</td>
<td>Route distribution down</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_DOWN: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_START</td>
<td>Route distribution start</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_START: Route Distribution initial creation. Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_ERROR</td>
<td>Route distribution in error state</td>
<td>nfvis %SYS-3-ROUTE_DISTRIBUTION_ERROR: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_DELETE</td>
<td>Route distribution deleted</td>
<td>nfvis %SYS-6-ROUTE_DISTRIBUTION_DELETE: All Neighbor Addresses deleted</td>
</tr>
<tr>
<td>ROUTE_DISTRIBUTION_UP</td>
<td>Route distribution up</td>
<td>nfvis %SYS-3-ROUTE_DISTRIBUTION_UP: Neighbor Address: 172.25.221.106</td>
</tr>
<tr>
<td>Event</td>
<td>Trigger Condition</td>
<td>Syslog Messages</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OVS_DPDK_SUCCESS</td>
<td>Enable DPDK</td>
<td>nfvis %SYS-3-OVS_DPDK_SUCCESS: OVS-DPDK enabled</td>
</tr>
<tr>
<td>OVS_DPDK_FAILURE</td>
<td>DPDK failure</td>
<td>nfvis %SYS-3-OVS_DPDK_FAILURE: Unable to allocate CP</td>
</tr>
<tr>
<td>BACKUP_INIT</td>
<td>Backup configuration initiation</td>
<td>nfvis %SYS-6-BACKUP_INIT: Starting backup: configuration-xxx</td>
</tr>
<tr>
<td>BACKUP_SUCCESS</td>
<td>Backup configuration successful</td>
<td>nfvis %SYS-6-BACKUP_SUCCESS: Backup configuration-xxx completed successfully</td>
</tr>
<tr>
<td>BACKUP_FAILURE</td>
<td>Backup configuration failure</td>
<td>nfvis %SYS-3-BACKUP_FAILURE: Backup configuration-xxx failed</td>
</tr>
<tr>
<td>RESTORE_INIT</td>
<td>Restore initiation</td>
<td>nfvis %SYS-6-RESTORE_INIT: Restore started</td>
</tr>
<tr>
<td>RESTORE_SUCCESS</td>
<td>Successful restore</td>
<td>nfvis %SYS-6-RESTORE_SUCCESS: Restore successful</td>
</tr>
<tr>
<td>RESTORE_FAILURE</td>
<td>Failure to restore</td>
<td>nfvis %SYS-3-RESTORE_FAILURE: Restore failed - internal error</td>
</tr>
</tbody>
</table>
## Glossary

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS</td>
<td>BIOS is firmware used to perform hardware initialization during the booting process, and to provide runtime services for operating systems and programs. The BIOS firmware comes pre-installed on a personal computer's system board, and it is the first software to run when powered on.</td>
</tr>
<tr>
<td>Cisco IMC</td>
<td>The Cisco Integrated Management Controller (IMC) is the management service for the C-Series servers. CIMC runs within the server. You can use a web-based GUI or SSH-based CLI to access, configure, administer, and monitor the server.</td>
</tr>
<tr>
<td>DPDK</td>
<td>The Data Plane Development Kit (DPDK) is a set of data plane libraries and network interface controller drivers for fast packet processing.</td>
</tr>
<tr>
<td>DTLS</td>
<td>Datagram Transport Layer Security (DTLS) is a communications protocol that provides security for datagram-based applications by allowing them to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.</td>
</tr>
<tr>
<td>IPSec</td>
<td>Internet Protocol Security (IPsec) is a secure network protocol suite that authenticates and encrypts the packets of data to provide secure encrypted communication between two computers over an Internet Protocol network</td>
</tr>
<tr>
<td>LACP</td>
<td>Link Aggregation Control Protocol (LACP) is a protocol for the collective handling of multiple physical ports that can be seen as a single channel for network traffic purposes.</td>
</tr>
<tr>
<td>LLDP</td>
<td>Link Layer Discovery Protocol (LLDP) is a vendor independent link layer protocol used by network devices for advertising their identity, capabilities to neighbors on a LAN segment.</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base (MIB) is a database of the objects that can be managed on a device. The managed objects, or variables, can be set or read to provide information on the network devices and interfaces.</td>
</tr>
<tr>
<td>NETCONF</td>
<td>A Network Configuration Protocol (NETCONF) is a protocol defined by the IETF to install, edit, and delete the configuration of network devices.</td>
</tr>
<tr>
<td>NGIO</td>
<td>Next Generation Input/Output (NGIO)</td>
</tr>
<tr>
<td>Terms</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PnP</td>
<td>Plug and Play (PnP) increases speed and reduces complexity of device deployments.</td>
</tr>
<tr>
<td>Port Channels</td>
<td>Port channels combine individual links into a group to create a single logical link that provides the aggregate bandwidth of up to eight physical links.</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Authentication Dial-In User Service (RADIUS) is a networking protocol, operating on port 1812 that provides centralized Authentication, Authorization, and Accounting management for users who connect and use a network service.</td>
</tr>
<tr>
<td>REST API</td>
<td>Representational state transfer (REST) suggests to create an object of the data requested by the client and send the values of the object in response to the user.</td>
</tr>
<tr>
<td>Service Chaining</td>
<td>Service chaining allows data traffic to be rerouted through one or more services, such as firewall, load balancer, and intrusion detection and prevention (IDP) devices.</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol (SNMP) is a framework used for managing devices on the internet. It provides a set of operations for monitoring and managing the internet.</td>
</tr>
<tr>
<td>SPAN</td>
<td>Switched Port Analyzer (SPAN) feature, which is sometimes called port mirroring or port monitoring, selects network traffic for analysis by a network analyzer.</td>
</tr>
<tr>
<td>Spanning Tree</td>
<td>Spanning Tree Protocol (STP) is a Layer 2 protocol that runs on bridges and switches. The main purpose of STP is to ensure that you do not create loops when you have redundant paths in your network.</td>
</tr>
<tr>
<td>SR-IOV</td>
<td>Single Root I/O Virtualization (SR-IOV) allows multiple VMs running a variety of guest operating systems to share a single PCIe network adapter within a host server. SR-IOV allows a VM to move data directly to and from the network adapter, bypassing the hypervisor for increased network throughput and lower server CPU burden.</td>
</tr>
<tr>
<td>TACACS</td>
<td>Terminal Access Controller Access-Control System (TACACS) refers to a family of related protocols handling remote authentication and related services for networked access control through a centralized server.</td>
</tr>
<tr>
<td>UEFI</td>
<td>The Unified Extensible Firmware Interface (UEFI) is a specification that defines a software interface between an operating system and platform firmware.</td>
</tr>
<tr>
<td>virtio</td>
<td>Virtual input/output (virtio) is a virtualization standard for network and disk device drivers where just the guest's device driver &quot;knows&quot; it is running in a virtual environment, and cooperates with the hypervisor.</td>
</tr>
<tr>
<td>VM</td>
<td>A virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination.</td>
</tr>
<tr>
<td>VNF</td>
<td>Virtual Network Functions (VNFs), the software version of network appliances such as a router, firewall, load-balancer etc.</td>
</tr>
</tbody>
</table>