

Administration Guide for Cisco Network Function Virtualization Infrastructure Software for Collaboration Release 26.2.1

First Published: 2026-11-19

Last Modified: 2026-11-19

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

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.

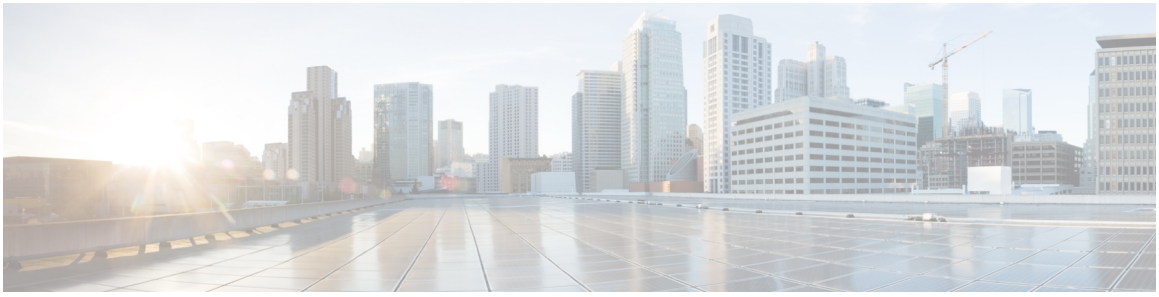
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY 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)



CONTENTS

Full Cisco Trademarks with Software License iii

CHAPTER 1 **Introduction** 1

Cisco NFVIS 1

Benefits of Cisco NFVIS 1

CHAPTER 2 **Deployment planning** 3

Architecture, topology and management interfaces 3

Hardware support & readiness 4

Host system requirements 5

NFVIS licensing readiness 6

NFVIS networking readiness 6

Open vSwitch (OVS) 6

SR-IOV concepts 7

Default system configuration 7

Application support & readiness 8

Supported applications 8

Finding application files 10

VM placement planning 10

Security readiness 10

Install readiness 10

CHAPTER 3 **NFVIS installation on supported hardware** 13

Checkpoints before starting installation 13

Install NFVIS 13

CHAPTER 4 NFVIS post-install tasks 17

Change default logon credentials 17

Set up management access 17

Set up NFVIS backup/Restore 18

CHAPTER 5 Access the NFVIS portal 19

Accessibility 19

Internationalization 20

CHAPTER 6 Cisco Smart Licensing for NFVIS 21

Prerequisites 21

Smart licensing modes 21

Smart mode 22

Smart mode configuration requirements 22

Configure smart mode using the CLI 23

Smart mode verification using the CLI 23

CSLU mode 23

Monitoring 24

How enforcement works 28

License renewal workflow 29

CHAPTER 7 NFVIS host management 31

Power management 31

Secure copy 32

Examples 32

Reset factory default 33

Before you begin 33

Reset NFVIS to factory default 33

Banner and message-of-the-day 34

Configure the banner and message of the day 34

Set the system time manually or configure NTP 35

Back up and restore VMs before an NFVIS factory reset 36

NFVIS backup and restore restrictions 36
NFVIS backup and restore behavior 37
NFVIS factory reset data retention 41

CHAPTER 8 **Virtual machine creation and management** 43
Uploading VM images to an NFVIS server 44
Register a virtual machine image 45
Customization of VM 46
VM profiles or flavors 46
VM profile creation methods 46
Create VM profile using REST API 47
Internal management networks 47
How default configuration works 48
VNF volumes 50
Restrictions for VNF volumes 50
Volume storage locations 51
Volume deployment parameters 51
Port forwarding 52
Port forwarding configuration 52
vCPU pinning 52
vCPU pinning configuration 52
Creating VM 53
VM deployment 53
VM name requirements 54
VM deployment verification 55
VM deployment parameters 56
VM bootstrap configuration options with VM deployment 59
Bootstrap configuration methods 59
VNF deployment placement 59
Placement configuration requirements 60
Configure granular RBAC for VM deployment 60
Additional capabilities 62
Deploy CUCM application 62
CDROM attachment and detachment 63

How CDROM attachment and detachment work 64
Requirement: prerequisites for CDROM attachment and detachment 64
Manage CDROM attachment and detachment 64
Recover a Cisco CUCM application using a recovery ISO 65
Recover the password for a Cisco CUCM application 66
Graceful VM stop 67
Perform a graceful stop 67

CHAPTER 9 NFVIS upgrade 69
NFVIS upgrade images 69
NFVIS supported upgrade paths 70
NFVIS pre-Upgrade checklist 70
Upgrade NFVIS 71
Upgrade NFVIS using the portal 71
Upgrade NFVIS using the CLI 72

CHAPTER 10 NFVIS serviceability, monitoring and observability 75
Log files 75
Host vs. virtual machine monitoring 77
Monitoring 77
Syslog 83
Configure syslog 90
NETConf 92
SNMP 119
SNMP GET 120
SNMP walk 120
SNMP notifications 121
Notification types 122
SNMP versions 122
SNMP MIB support 123
Configure SNMP support 126
Cisco ThousandEyes 133
Prerequisites 134
Deploy ThousandEyes container from NFVIS web interface 134

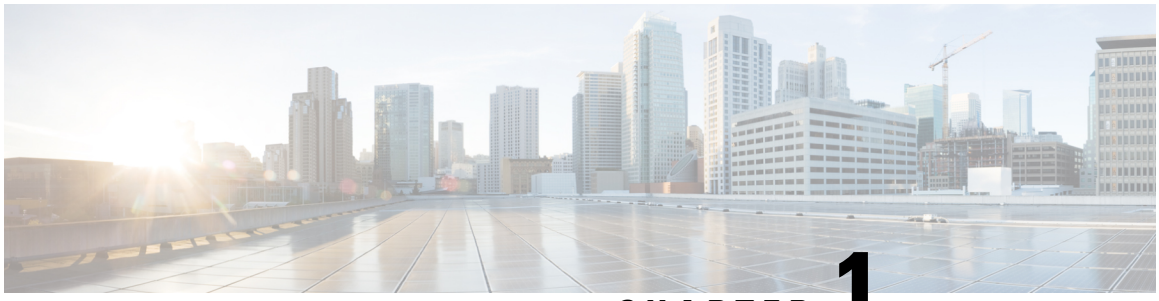
Deploy ThousandEyes on NFVIS using CLI configs 136
Troubleshooting NFVIS 144
Log and show commands 144
Configure packet capture 146

CHAPTER 11 NFVIS security considerations 147
Installation 149
Image tamper protection 149
RPM signing 149
RPM signature verification 149
NFVIS behavior during signature verification failure 149
Image integrity verification 149
How hash verification works 150
Device access 150
Device access security controls 150
Enforced password change at first login 151
Login vulnerability restrictions 151
Enforcement of strong password 151
Configure minimum length for passwords 151
Configure password lifetime 152
Previous password reuse limitation 153
NFVIS password reuse prevention 153
Login attempt frequency limits 153
Disable inactive user accounts 153
Activate an inactive user account 154
Integration with external AAA servers 154
AAA server deployment recommendations 155
Role based access control 155
NFVIS user roles and privileges 155
Roles 156
Users 157
Admin user characteristics 157
Groups 157
Group types and characteristics 157

Create groups and assign local users to the groups 158
Assign remote users to the group 158
Configure local authentication for a specific group of users 159
RADIUS 160
RADIUS implementation details 160
How RADIUS authentication works 161
Configure RADIUS 162
TACACS+ 162
TACACS+ server configuration requirements 163
How TACACS+ authentication works 163
Configure a TACACS+ server 164
Default authentication order 165
Authentication configuration 165
User specific authentication order 165
Granular role-Based access control 166
Restrict device accessibility 168
Essential ports are enabled by default 168
Authorized network access restrictions 169
NFVIS access control lists 169
Access privileged debug shell 171
Secure interfaces 172
Console 172
SSH 172
CLI session timeout 173
Session security mechanism 173
NETConf 173
REST API 173
NFVIS portal security 174
Session management 174
URL redirection 175
Portal disablement 175
HTTPS 175
Certificate management 176
Certificate paths and consumers 176

Generate and retrieve an NFVIS CSR 176
Private-key support and external CSR limitation 177
Add SANs when issuing the certificate 177
Verify and install the certificate 178
HTTPS and browser trust validation 179
SNMP security features 179
Insecure mode dependency 181
Insecure mode 181
Banners 181
Banner content requirements 182
Security considerations for banner content 182
Sample banner implementation 182
NFVIS banner configuration 182
Factory default reset 183
Infrastructure management networks 184
Infrastructure management network design considerations 184
Out-of-band management 184
Connection methods and benefits 185
Pseudo out-of-band management 185
In-band management 185
Network design considerations 185
Protecting sensitive information 186
Local password storage requirements 186
File transfer 186
File transfer implementation 187
Logging 187
Virtual machine security 188
VNC console access protection 188
How VNC console access works 188
VNC console command example 188
Encrypted VM config data variables 189
Checksum verification for remote image registration 189
How checksum verification works 189
Certification validation for remote image registration 189

Certificate specification requirements 190
VM isolation and resource provisioning 190
NFV architecture components 190
CPU isolation 190
CPU allocation by VM type 191
Memory allocation 191
How memory is allocated 192
Interface isolation 192
SR-IOV implementation details 192
Secure development lifecycle 193
Attack vector reduction 193
NFVIS SDL processes 193
Public sector security certifications 194
FIPS 140-3 194
How FIPS mode works 194
Disable FIPS mode 194
Configure FIPS mode 194
How backup and restore work in FIPS mode 196
FIPS operational status 198



CHAPTER 1

Introduction

Objective

This reference groups information about Introduction, including About Cisco NFVIS, Benefits of Cisco NFVIS.

- Cisco NFVIS, on page 1
- Benefits of Cisco NFVIS, on page 1

Cisco NFVIS

Cisco NFVIS (Network Function Virtualization Infrastructure Software) is a hypervisor product from Cisco's Enterprise Network Function Virtualization (Enterprise NFV) portfolio. It is a special-purpose Linux/KVM-based virtualization layer for deploying select application workloads on select Cisco hardware.

Cisco NFVIS-for-Collaboration is a special edition of NFVIS with a new commercial offering and separate product ID, distinct pricing, new licensing, and a slightly different admin User Interface.

Cisco NFVIS-for-Collaboration supports only select Cisco Collaboration Applications and only select Cisco Calling Appliances. For more information, see the Cisco Virtualization Guide for Cisco On-premises Calling Applications.

Releases 26.2 and beyond only support Cisco NFVIS-for-Collaboration features, workload/hardware support and administration.

Note For the rest of this document, “NFVIS” used by itself refers to NFVIS-for-Collaboration.

Benefits of Cisco NFVIS

Objective

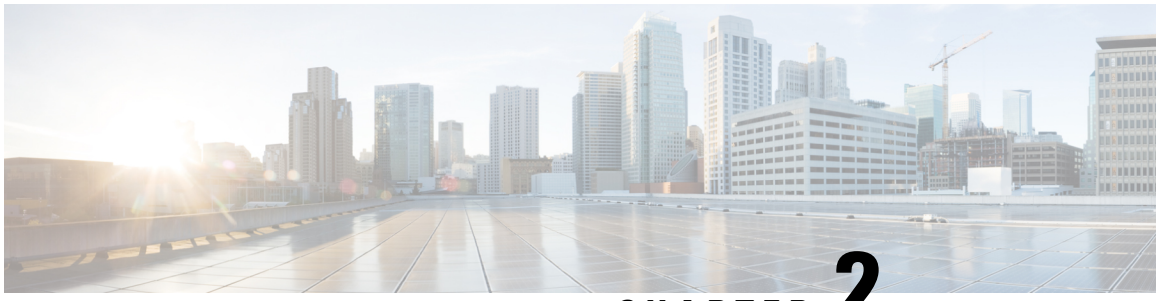
Cisco NFVIS addresses the needs of communications-centric, limited-purpose deployments: Cisco collaboration applications, calling appliance hardware and essential virtualization administration needs.

Cisco NFVIS platform benefits

Cisco NFVIS does not target general-purpose / datacenter-centric needs; on-premises deployments requiring general-purpose capabilities such as broader workload support, broader hardware support and/or advanced infrastructure management should look at other supported options in the Cisco Virtualization Guide for Cisco On-premises Calling Applications.

The Cisco NFVIS platform provides several key benefits for on-premises Collaboration deployments.

- Simple management for calling and network administrators (portal, CLI and REST APIs, all configurations exposed through YANG models).
- Cost-effective virtualization solution with predictable commercials, lifespan and support.
- On-premises-friendly, especially for air-gapped or regulated deployments.
- Simple migration from other deployments.
- Single source of support for applications, virtualization and hardware.



CHAPTER 2

Deployment planning

Objective

Consult the Release Notes for the version of NFVIS you will deploy: https://www.cisco.com/c/en/us/support/routers/enterprise-nfv-infrastructure-software/products-release-notes-list.html.

- Architecture, topology and management interfaces, on page 3
- Hardware support & readiness, on page 4
- Host system requirements, on page 5
- NFVIS licensing readiness, on page 6
- NFVIS networking readiness, on page 6
- Application support & readiness, on page 8
- Security readiness, on page 10
- Install readiness, on page 10

Architecture, topology and management interfaces

Objective

Cisco NFVIS is deployed as a standalone virtualization host for local virtual machine workloads.

Host topology and management interfaces

Additional hosts may be deployed to provide hardware/workload redundancy, additional capacity for application scale, geographic coverage or other customer requirements.

Each Cisco NFVIS host may be managed via multiple interfaces:

- NFVIS Portal (GUI), accessed through a web browser using the local ethernet management network connection
- NFVIS CLI, access through SSH. One host may access the CLI of another host.
- NFVIS NETCONF/RESTCONF APIs (see https://developer.cisco.com/docs/nfvis/, these can be used to build centralized management dashboards for multiple hosts)

Before you can use any of these management interfaces, you will need to complete installation, then configure management network access.

Hardware support & readiness

Cisco NFVIS is only supported on specific hardware platforms with specific configurations. NFVIS is only supported on the Cisco Calling Appliances listed in Table 1 below, along with the minimum required NFVIS release for each appliance model/generation. If you are doing a fresh install on an appliance, or a rebuild of a supported appliance, see their install guides below:

- Cisco Business Edition 6000 Install and Upgrade Guides at https://www.cisco.com/c/en/us/support/unified-communications/business-edition-6000/products-installation-guides-list.html
- Cisco Business Edition 7000 Install and Upgrade Guides at https://www.cisco.com/c/en/us/support/unified-communications/business-edition-7000/products-installation-guides-list.html
- Cisco Expressway CE1400V Appliance Installation Guide at https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/expressway/install_guide/CE1400V/exwy_b_cisco-expressway-ce1400v-appliance-installation-guide.html

Supported appliances require UCS BIOS and CIMC settings compatible with NFVIS. See the Supported Hardware section of the Release Notes for the NFVIS version you are deploying.

Expect installation problems and other issues if you attempt to install NFVIS on unsupported hardware. The only fix will be to get onto a supported calling appliance hardware configuration.

Table 1: Supported Calling Appliance Models

Cisco Calling Appliances	Models / Generations	Supported Base PIDs	First introduced in this NFVIS Release
Cisco Business Edition 6000	BE6000M (M7)	BE6K-M7-K9 BE6K-M7-XU	4.18.2a
Cisco Business Edition 6000	BE6000M (M6)	BE6K-M6-K9 BE6K-M6-XU	4.18.2a
Cisco Business Edition 6000	BE6000M (M5)	BE6M-M5-K9 BE6M-M5-XU	4.18.2a
Cisco Business Edition 6000	BE6000H (M5)	BE6H-M5-K9 BE6H-M5-XU	4.18.2a
Cisco Business Edition 7000	BE7000M (M7)	BE7M-M7-K9 BE7M-M7-XU	4.18.2a
Cisco Business Edition 7000	BE7000M (M6)	BE7M-M6-K9 BE7M-M6-XU	4.18.2a
Cisco Business Edition 7000	BE7000M (M5)	BE7M-M5-K9 BE7M-M5-XU	4.18.2a

Cisco Calling Appliances	Models / Generations	Supported Base PIDs	First introduced in this NFVIS Release
Cisco Business Edition 7000	BE7000H (M7)	BE7H-M7-K9 BE7H-M7-XU	4.18.2a
Cisco Business Edition 7000	BE7000H (M6)	BE7H-M6-K9 BE7H-M6-XU	4.18.2a
Cisco Business Edition 7000	BE7000H (M5)	BE7H-M5-K9 BE7H-M5-XU	4.18.2a
Expressway CE1400V	CE1400V (M7)	CE1400V-M7-K9	4.18.2a

Host system requirements

Objective

These resources are required for a standalone Cisco NFVIS. A summary quick-reference table by calling appliance model is in the Virtualization Guide for On-premises Calling at Cisco Virtualization Guide for Cisco On-premises Calling Applications.

NFVIS host resource requirements

Total Cores	NFVIS 4.18.2a and later Releases
16 or less	1 + (1 core per socket applicable to DPDK systems)
More than 16	2 cores in NUMA-0 1 core in NUMA-1 (if Multi-NUMA node system*) (1 core per socket applicable to DPDK systems)

* Indicates that Multi-NUMA node systems require an additional CPU core system reserved. This additional core is helpful in processing the cross NUMA nodes, indirectly improving the performance of Cisco NFVIS functions on the system cores.

Table 2: CPU allocation

Reserved System Memory	Up to 16 GB	Up to 32 GB	Up to 64 GB	Up to 128 GB	Greater than 128 GB	Greater than 256 GB
Reserved for NFVIS	7 GB/ 8 GB*	11 GB/ 12 GB*	11 GB/ 12 GB*	11 GB/ 12 GB*	11 GB/ 13 GB*	16 GB/ 20 GB*

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

* Indicates the memory allocation is applicable only for Multi-NUMA node systems. In case of single node systems, the memory allocation values without * is applicable.

Table 3: Memory allocation

Total System Memory	Additional memory required for DPDK support per NUMA node
Upto 63 GB	1
64 GB - 127 GB	2
128 GB - 256 GB	4

NFVIS licensing readiness

Cisco NFVIS is a paid product and requires an active subscription for entitlement to use the product, receive technical support from Cisco TAC and gain access to feature/maintenance releases.

- To ensure this required subscription is in order, Cisco Partners can consult the Cisco Business Edition 6000/7000 Ordering Guide.
- A valid, active subscription will appear in Cisco Commerce Subscriptions & Services as a Subscription ID with product name A-NFVIS-COLLAB, with a quantity of L-LIC-COLLAB Items matching the quantity of calling appliances in the deployment.

Cisco NFVIS uses Cisco Smart Licensing to manage entitlement to product usage, support, upgrades and updates.

- To license NFVIS, you require an active Cisco Smart Account. For more information on Smart Account, see, How to Request a Smart Account
- NFVIS only supports certain Smart Licensing modes.
- NFVIS enforces licensing, including a Grace Period and Compliance Mode (where admin goes read-only).
- For more details on Cisco Smart Licensing for NFVIS, see details in Chapter 6 –Cisco Smart Licensing for NFVIS.

NFVIS networking readiness

NFVIS connects virtual network functions (VNFs) to one another and to physical network ports. It does this by attaching each VNF's virtual NIC to a network configured in NFVIS. Only two ways to provide that connectivity are Open vSwitch (OVS) and SR-IOV.

Open vSwitch (OVS)

OVS is a software switch running on the NFVIS host. It forwards traffic between virtual machine virtual NICs and, through bridges, physical NICs or port channels. NFVIS uses bridges as virtual switches and networks to define which connections a virtual machine can use, including options such as VLAN settings.

For example, two virtual machines attached to the same OVS network can communicate through the bridge. A bridge connected to a physical port can also carry their traffic to or from the external network. Because NFVIS handles this switching, OVS is flexible for connecting virtual machines and applying network configuration.

SR-IOV concepts

Objective

SR-IOV lets a supported physical NIC expose multiple Virtual Functions (VFs). NFVIS can assign a VF to a virtual machine, giving it a more direct path to NIC hardware. The NIC's Physical Function (PF) provides and manages those VFs.

OVS and SR-IOV networking behavior

This direct path can improve throughput and reduce latency and host CPU work, which can suit virtual machines with demanding traffic requirements. It depends on a compatible NIC and VNF driver, and the number of available VFs is limited by the hardware. NFVIS lets you enable SR-IOV on a physical NIC and create SR-IOV networks for VNF assignment.

In short, OVS is the general-purpose virtual switching path; SR-IOV gives a virtual machine a more direct hardware networking path when performance is the priority. Some deployments use both—for example, SR-IOV for a virtual machine's external-facing interface and OVS networks for its connections to other virtual machines.

Default system configuration

Default system configuration is a network setup that

- configures networks in Cisco NFVIS to allow inbound and outbound traffic
- creates default networks and bridges that cannot be deleted.

This diagram illustrates the default network configuration:

Figure 1. Default network configuration



These networks and bridges are created by default and cannot be deleted. You can configure more as required:

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

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

Application support & readiness

Objective

This reference groups information about Application Support & Readiness, including Supported Applications, Finding Application Files, VM Placement Planning.

Supported applications

Objective

Cisco NFVIS only supports select Cisco Collaboration workloads in the Table below, which lists the first release when support for that application was added. For compatibility of applications with subsequent releases, see the document indicated by the table.

Supported applications and release compatibility

Unlisted applications/workloads are not supported for hosting on NFVIS (and are not expected to install / boot / operate properly due to lack of enablement in NFVIS for their particular software contents).

See the Release Notes for NFVIS 26.2 for which applications are supported.

Table 4: Supported Applications/Workloads

Category	Application /Workload	First i ntroduced in this NFVIS Release	Check here for Application / NFVIS compatibility matrix
On-premises Calling	Cisco Unified Communications Manager (CUCM)	4.18.2a **Note** not supported on 26.2.1 at this time.	c
On-premises Calling	Cisco Unified CM Session Management Edition (SME)	4.18.2a **Note** not supported on 26.2.1 at this time.	c
On-premises Calling	Cisco Unified CM – IM & Presence Service (IMP)	4.18.2a **Note** not supported on 26.2.1 at this time.	c
On-premises Calling	Cisco Unity Connection (CUC)	4.18.2a **Note** not supported on 26.2.1 at this time.	c

Category	Application /Workload	First i ntroduced in this NFVIS Release	Check here for Application / NFVIS compatibility matrix
On-premises Calling	Cisco Emergency Responder (CER)	4.18.2a **Note** not supported on 26.2.1 at this time.	c
On-premises Calling	Cisco Expressway (Expwy)	4.18.2a **Note** not supported on 26.2.1 at this time.	c
On-premises Meetings	Cisco Meeting Server (CMS)	26.2.1	Virtualization for Cisco Meeting Server at
On-premises Meetings	Cisco Meeting Management (CMM)	26.2.1	Virtualization for Cisco Meeting Server at
On-premises Contact Center	Unified Contact Center Express suite (UCCX)	26.2.1	Virtualization for Cisco Unified Contact Center Express at
On-premises Contact Center	Cisco Customer Collaboration Platform (CCP)	26.2.1	Cisco Customer Collaboration Platform Installation and Upgrade Guide, Release 15.0 SU1 or later at
Webex Calling	Cisco Webex Calling Dedicated Instance Enhanced Survivability Node (DI ESN)	4.18.2a **Note** not supported on 26.2.1 at this time.	Enhanced Survivability prerequisites, Virtualization specification
Webex Edge	Cisco Webex Edge Video Mesh Node (VMN)	26.2.1	Deployment Guide for Video Mesh at

Category	Application /Workload	First i ntroduced in this NFVIS Release	Check here for Application / NFVIS compatibility matrix
Webex Key Management Service (KMS)	Cisco Webex Hybrid Data Security (HDS)	26.2.1	Virtual Host Requirements, Deployment guide for Webex Hybrid Data Security

Finding application files

Objective

NFVIS requires Images and Profiles to create and deploy virtual machines. For the files each application provides for use with NFVIS, see their Installation Guides (Cisco Virtualization Guide for Cisco On-premises Calling Applications and other documents).

VM placement planning

Objective

An NFVIS host will have a system requirements holdback on physical hardware capacity for its needs. Cisco Collaboration applications also have “rules” for VM placement and minimum hardware specs to be supported.

VM placement planning considerations

Note “vcpu” term usage may differ in NFVIS monitoring screens vs. as used in application VM specs definitions and VM Placement rules.

Security readiness

At a minimum, after fresh install, you will immediately be prompted to change default login credentials

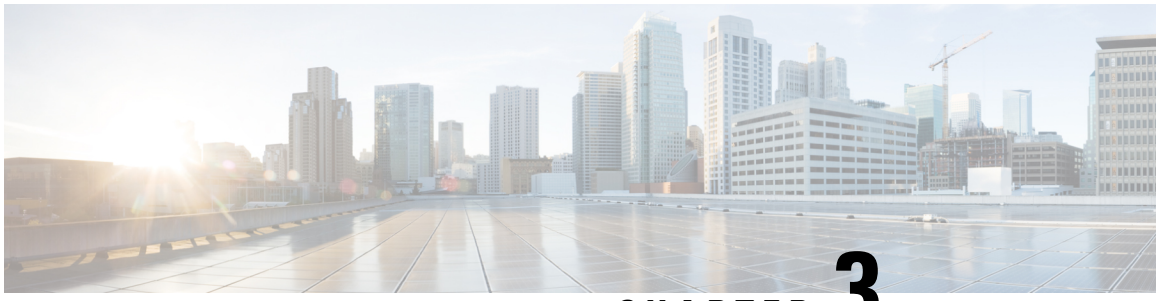
To review NFVIS credential policies, and other security-related items you may want to handle immediately after install, see NFVIS security considerations, on page 147

Install readiness

Install is dependent upon:

1. Location of the NFVIS software image on the network, relative to where the destination calling appliance is located. For example, uploading NFVIS image remotely via staging NFVIS on a server on the same LAN as the calling appliance may have drastically different durations.
2. Storage and network speed of whatever is hosting the NFVIS software image. E.g. your laptop may be slower than a server.

Depending on above factors, NFVIS fresh installation can take 30-120 minutes before the system is ready for first-time-setup and configuration. For fastest install, stage the NFVIS image on the same LAN as the calling appliance, on a server with fast storage and fast network.



CHAPTER 3

NFVIS installation on supported hardware

Objective

This reference groups information about Install NFVIS on Supported Hardware, including Checkpoints before starting installation, Installation Steps.

- Checkpoints before starting installation, on page 13
- Install NFVIS, on page 13

Checkpoints before starting installation

Objective

Supported Hardware.

NFVIS installation prerequisites

1. Supported Hardware
2. NFVIS software installation image from link below:
   - Installation images can be found on software.cisco.com/download/home, search for Enterprise NFV Infrastructure Software, use images from NFV Infrastructure Software for version 26.2.
   - Cisco NFVIS installation images are available as .iso files. Images are digitally signed to ensure cryptographic integrity and authenticity. In addition, all software packages in the image are verified during the install.
   - 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.

Install NFVIS

This task installs Cisco NFVIS by configuring boot order, mapping virtual media, and completing the installation process through CIMC. RAID configuration / disk groups must be complete before installing NFVIS. See the Installation Guides of supported calling appliances for supported RAID configurations.

Procedure

Step 1 Log in to CIMC

NFVIS must be installed onto the calling appliance using CIMC and virtual Keyboard Video Mouse (vKVM). Calling appliances do not support installation via other methods from general-purpose servers, such as via USB, removable media or dedicated boot drives.



Warning Warning: if you are doing a reinstall, take a fresh backup of all configured application VMs. A reinstall will erase any previously configured VMs.

a) Ensure your RAID configuration is supported for your calling appliance model/generation. If not, delete any previous virtual drives and reconfigure.
b) Launch CIMC's virtual Keyboard Video Mouse (vKVM) by selecting **Launch KVM** from the CIMC homepage.

   You can choose Java or HTML based KVM. It is recommended to use HTML based KVM. Ensure that the pop-up blocker is disabled as KVM Console will open in a separate window.

c) Use vKVM to mount the NFVIS installation image as virtual media for hardware boot:
d) Select **Virtual Media** and then **Activate Virtual Devices**.
e) Select **Virtual Media** again and then **Map CD/DVD**.

   Browse and select the NFVIS ISO image. Click **Open** and **Map Drive** to mount the image.

f) Select **Virtual Media** again to ensure the NFVIS ISO image is now mapped to CD/DVD.
g) Configure boot order:

   1. From the **CIMC Compute**, select **BIOS**.
   2. Select **Configure Boot Order** and the **Configure Boot Order** dialog box appears.
   3. Select **Advanced**.
   4. The **Add Boot Device** page appears. Select **Add Virtual Media**, and the **Add Virtual Media** dialog box appears.
   5. Enter a name and select **KVM Mapped DVD**. Set state to **Enabled** and order as 1, and **Save Changes**.
   6. The **Add Boot Device** page appears again, select **Add Local HDD**, and **Add Virtual Media** dialog box appears.
   7. Enter a name, set state to **Enabled** and order as 2, and **Save Changes**.
   8. Click **Close**.

Step 2 Power cycle server to start the installation:

a) From CIMC homepage, select **Host Power**.
b) Reboot the server by selecting the **Power Off** option.
c) After the server is down, select the **Power On** option.

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

Step 3 Complete the first login after NFVIS installation.

a) After the installation is complete, wait for the system to reboot automatically from the hard drive.
b) Log into the system when the command prompt NFVIS login is displayed after the reboot.
c) Use admin as the login name and Admin123# as the default password.

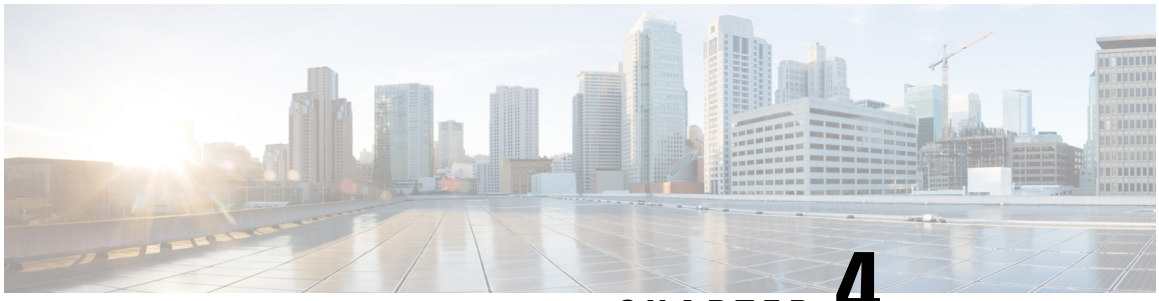


Note The system prompts you to change the default password at the first login attempt. You must set a strong password as per the on-screen instructions to proceed with the application. You cannot run API commands or proceed with any tasks unless you change the default password at the first login. The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

Step 4 Verify the installation using the System API, CLI, or by viewing the system information from the Cisco NFVIS portal:

Table 5: Installation verification

Step	Phase	Description
1	Pre-install	Preparing installation, verifying packages, and detecting hardware.
2	Partitioning	Partitioning disks and preparing storage layout.
3	Package Install	Installing packages with live package counter and progress tracking
4	Configuration	Configuring system during %post --nochroot stage
5	Finalization	Preparing reboot during %post --chroot stage
6	First Boot	Rebooting system and starting NFVIS services



CHAPTER 4

NFVIS post-install tasks

Objective

This reference groups information about NFVIS Post-Install Tasks, including Change Default Logon Credentials, Setup Management Access, Setup NFVIS Backup/Restore.

- Change default logon credentials, on page 17
- Set up management access, on page 17
- Set up NFVIS backup/Restore, on page 18

Change default logon credentials

Procedure

Change default logon credentials.

When NFVIS installation is complete, you will see the NFVIS CLI `nfvis login:` prompt. Log in with the default username **`admin`** and password **`admin123#`**. You will be immediately prompted to change the password.

Note The system prompts you to change the default password at the very first login attempt. You must set a strong password as per the on-screen instructions to proceed with using NFVIS. Until you change the default password, you cannot run API commands (they will return a 401 unauthorized error) or proceed with any other NFVIS tasks.

Set up management access

During and immediately after installation, your CLI session has been via CIMC's vKVM. Before you can manage NFVIS through CLI via SSH, or NFVIS Portal or NFVIS API, you must first assign it a management IP address from the CLI.

Before you do this, familiarize yourself with the following:

- Default host configuration immediately after install. Run these commands:
- **show version**
- **show platform**
- Default virtual networking configuration immediately after install. Run these commands:
- **show networks** (noting different OVS and SR-IOV networks)
- **show running-config** (noting system settings, networks network, **bridges bridge** and pnic sections, as these are the ones modified during initial configuration.



Note Saving a configuration change is done with the **commit** command, you can do this after each configuration or after several.

Procedure

Step 1 From the console, login and enter configuration mode via configuration terminal.

Step 2 Set system hostname, via **system settings hostname***hostname* (this will change what you see in the command prompt).

Step 3 Set management IP address, **system settings mgmt ip address** *ip-address ip-subnet-mask*

Step 4 Set default gateway, **system settings default-gw** *ip-address*

Step 5 Set source interface to originating traffic from NFVIS, **system settings source-interface** *ip-address*. After **commit**, you can rerun **show running-config** to see your changes.

Step 6 (Optional) (Optional, only if you need to change the default physical port used for NFVIS management). Set physical port for management via **bridges bridge** *bridge-name*, then **no port** *old port ID*, **port** *new port ID*.

Step 7 Configure the upstream network access switch for the NFVIS management port.

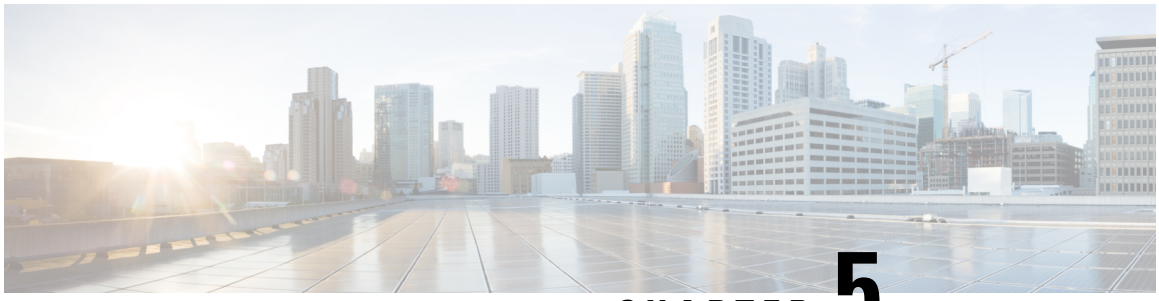
Step 8 Verify connectivity by accessing NFVIS CLI via Secure Shell client, or NFVIS Portal at https://**`NFVIS Management IP or FQDN`** via web browser. You should also be able to use NFVIS API as well.

Set up NFVIS backup/Restore

Procedure

Set up NFVIS backup/Restore.

When you are done with all post-install tasks, setup NFVIS Backup and take a fresh backup of the host. See Back up and restore VMs before an NFVIS factory reset, on page 36.



CHAPTER 5

Access the NFVIS portal

Access the NFVIS portal through a web browser using the local Ethernet management network connection.

Procedure

Step 1 Connect your laptop to the local Ethernet management network. In your web browser address bar, enter https://**NFVIS IP address**.

We recommend using Google Chrome.

Step 2 Log in to the NFVIS portal using the username `admin` and the newly generated password.

You have successfully accessed the NFVIS portal. The NFVIS dashboard appears and provides a summary of activities on the device.

- Accessibility, on page 19
- Internationalization, on page 20

Accessibility

Objective

The NFVIS GUI includes accessibility enhancements designed to support users with disabilities and improve overall interface usability through keyboard navigation, screen reader compatibility, visual clarity improvements, and assistive technology integration.

Impact of accessibility enhancements

Impact of accessibility enhancements

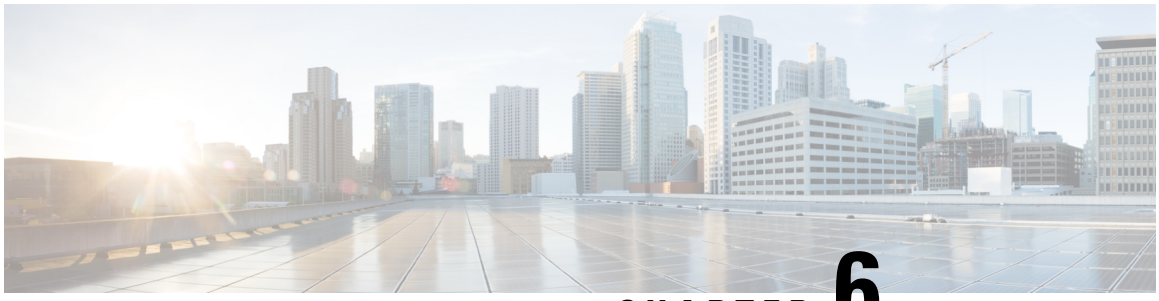
- Keyboard Accessibility: Improved portal usability for customers who rely on keyboard-only navigation.
- Inclusive Experience: Enhanced support for screen readers and other assistive technologies like ARIA roles and descriptive labels.
- Visual Clarity: Improved readability of interface elements through better color contrast alignment.

- Workflow Efficiency: Streamlined forms and workflows through clearer labeling and logical structure.
- Equivalent Information Access: Provided meaningful text alternatives for icons and visual elements, ensuring all users receive the same information.
- User Independence: Empowered customers to complete tasks more efficiently and independently across the portal.

Internationalization

Objective

NFVIS Portal also supports Internationalization for Japanese (4.18) and French Canadian (26.2) onwards.



CHAPTER 6

Cisco Smart Licensing for NFVIS

Objective

This reference groups information about Cisco Smart Licensing for NFVIS, including Prerequisites, Smart Licensing Modes, Monitoring, Enforcement Behavior, License Renewal Workflow.

- Prerequisites, on page 21
- Smart licensing modes, on page 21
- Monitoring, on page 24
- How enforcement works, on page 28
- License renewal workflow, on page 29

Prerequisites

Objective

Lists the requirements needed before implementing Cisco NFVIS on Cisco BE6K/BE7K/CE1400V appliances. This reference identifies the essential prerequisites that must be met before implementing Cisco NFVIS support for Cisco BE6K/BE7K/CE1400V appliances.

NFVIS Smart Licensing prerequisites

You require an active Cisco Smart Account. For more information on Smart Account, see, How to Request a Smart Account. You need to purchase Cisco BE6K/BE7K/CE1400V appliances along with Cisco NFVIS Smart License.

Smart licensing modes

Objective

Describes how to configure Cisco Smart Licensing for Cisco NFVIS, which operates in two distinct modes: Smart Mode and CSLU Mode for license management and activation.

Smart and CSLU licensing modes

Smart Licensing for Cisco NFVIS operates in two modes:

1. Smart Mode
2. CSLU Mode

Smart mode

Smart mode is a licensing operation mode that

- enables Cisco calling appliances to connect directly to Cisco Smart Software Manager (CSSM) using Cisco NFVIS
- requires specific configuration inputs including CSSM tokens and transport settings, and
- reports license usage periodically to the CSSM server after successful trust handshake.

Smart mode configuration requirements

Objective

Lists the requirements for configuring Smart Mode.

Smart mode configuration requirements

1. A unique token for customer virtual account on CSSM.

Generate a unique token using your Cisco Virtual Account on the Licensing Server (CSSM). For more information, see How can I create a token to register my device?

1. Provide the generated token as the input to the Cisco calling appliances. The CSSM token is your password for the user account on CSSM. This input is mandatory for the Smart Mode to function and has no default values.
2. Transport Mode

The Transport Mode specifies the type of connection existing between the device and CSSM (whether the device is directly connected/ Connected through CSLU). For the Smart Mode, the Transport Mode is called as Smart Transport. The default value for the transport mode is CSLU Mode.

1. Transport URL

The Transport URL specifies the URL to reach a CSSM/CSLU server from the device. The default transport URL for Smart Mode is https://smartreceiver.cisco.com/licservice/license. The default transport URL for CSLU mode is available on online search engines.

1. Time interval between two licence usage announcements

The Resource Usage Measurement (RUM) is sent periodically from the device to CSSM/CSLU which presents the License Usage pertaining to the NFVIS License Entitlement to the CSSM server. You can specify the periodic time interval as an input. The default value is 30 days.

When the device has provided with the above listed inputs for the smart mode of operation, the device initiates the Trust Handshake with the CSSM server. Once Trust Handshake is successful, the device starts reporting license usage according to the time interval specified.

Configure smart mode using the CLI

Procedure

Configure smart mode using the CLI.

```
configure terminal
license smart transport smart   smart-url
license smart transport smart   token
license smart transport smart   usage
```

Smart mode verification using the CLI

Objective

Use the **show license opdata** command to verify the success of smart mode.

Smart mode operational status output

```
show license opdata
license opdata operational-status OK
license opdata usage-reporting last-ack-received None
license opdata usage-reporting usage-interval 1
license opdata usage-reporting next-ack-deadline "Oct 01 20:32:19 2023 UTC"
license opdata usage-reporting last-report-push None
license opdata usage-reporting next-report-push None
```

CSLU mode

CSLU mode is a Smart Licensing transport mode that

- provides reachability between Cisco BEx7K/CE1400V appliances based out of UCS-C series servers and a Cisco Smart Licensing Utility (CSLU) application
- operates as the default transport mode without requiring token input, and
- reports license usage at 30-day intervals by default.



Note Cisco NFVIS supports SSM On-Prem in CSLU transport mode.

CSLU mode has these characteristics:

- Ensure that you have logged into CSLU using your Cisco Smart Account on the CSSM.
- The license reporting in the CSLU mode happens at an interval of 30 days by default.
- The CSLU is the default Transport Mode and no token is required as an input.

- The default Transport URL for CSLU mode is http://CSLU-local:8182/CSLU/v1/pi
- If you require any change in the inputs other than the defaults, configure the inputs for CSLU mode.
- The CSLU Mode does not require any inputs and the CSLU Mode is the default mode of operation for the Cisco NFVIS on Cisco BEx7K/CE1400V appliances based out of UCS-C series servers.

Note

- There are no specific operations in terms of Cisco NFVIS Licensing with regards to the device reboot and upgrade. The Cisco NFVIS continues to operate with respect to the interactions established through the modes of operations.
- The licensing configuration and Smart Licensing Database are not cleared from the device in case of a Factory Default Reset.

  License usage reporting to CSSM/CSLU resumes when the device regains network connectivity through a reconfig operation after a Factory Default Reset.
- To initiate an RMA for a device, manually release the licenses associated with the device from the CSSM, or use the Cisco NFVIS action command **license smart release**. The command frees up the license entitlements used by this device on CSSM. If the device continues to operate with Cisco NFVIS even after the license release, it will, after a set time interval, resend the usage report (RUM) to CSSM. This report indicates the correct number of license units consumed, which is reserved again on CSSM.
- While performing a device data backup, Cisco NFVIS backs up the config to a file and sends the forced License Usage report to CSSM.

Monitoring

Objective

This reference provides show commands used to monitor Cisco NFVIS Smart Licensing for Cisco calling appliances along with some configuration examples.

Smart Licensing monitoring commands

Show commands for license monitoring

- Use the **show running-config license** command in the Cisco NFVIS CLI to view the license information:

```
show running-config license
license smart
 transport smart
 usage interval 1
 token
"SSNiXjw3R0wUU/A2WDdH8LvRF6v+PSUXpZZqINCKQYujbbJMi8C3WpjtWnq8RU2ljisfd9MF0qS+F9X8/uD3iLbwGqjMQ4R0z77ojeAwbD
NMLNM0pR/+IF\nhNRgo4762u2ob6DjSr2YGcez8s7TMKQfjswUURDXYxYn1AG8VgSjs9RfPpFVi4Ly4BUFJg=="
```

- Use the **show license accounts** command in the Cisco NFVIS CLI to view the Cisco smart account information that the device is linked with:

```
show license accounts
license accounts VIRTUAL ACCOUNT:NFVIS-VA-1
                 SMART ACCOUNT:InternalTestDemoAccount8.cisco.com
```

- Use the **show license license-units-consumed** to view the units of entitlement used by the device on the CSSM:

```
show license license-units-consumed
license license-units-consumed 2
```

- Use the **show license opdata** command to view the licensing operational data:

```
show license opdata
license opdata operational-status OK
license opdata usage-reporting last-ack-received None
license opdata usage-reporting usage-interval 1
license opdata usage-reporting next-ack-deadline "Oct 01 20:32:19 2023
UTC"
license opdata usage-reporting last-report-push None
license opdata usage-reporting next-report-push None
```

- Use the **show license status** command to view the status of your license:

Show more

```
show license status
Current tenant: nfvis-single-tenant

               Smart Licensing is ENABLED

               Smart Licensing Policy:
                 Status: ENABLED

               Transport:
                 Type: SmartTransport
                 URL:
https://smartreceiver-stage.cisco.com/licservice/license
                 Proxy:
                  Not configured

               Trust Code Installed: Jul 05 06:20:56 2023 UTC
                 Last Attempt Date: Jul 05 06:19:31 2023 UTC
                 Last Attempt Status: JobStatus.success
                 Last Attempt Result: success
                 Attempt In Progress: False
                 Next Attempt Date: None


                 Usage Reporting:
                 Last ACK received: None
                 Next ACK deadline: Oct 01 20:32:19 2023 UTC
                 Reporting push interval: 1
                 Next ACK push check: None
```

```
    Next report push: None
    Last report push: None
    Last report file write: None


  Policy:
    Policy in use: None
    Policy name: Default Policy
    Reporting ACK required: yes
    Subscription:
     First report requirement (days): 90
     Ongoing reporting frequency (days): 90
     On change reporting (days): 90
    Perpetual:
     First report requirement (days): 365
     Ongoing reporting frequency (days): None
     On change reporting (days): 90
    Enforced:
     First report requirement (days): None
     Ongoing reporting frequency (days): None
     On change reporting (days): None
    Export:
     First report requirement (days): None
     Ongoing reporting frequency (days): None
     On change reporting (days): None
```

- Use the command **show license summary** to view the summary of your license

Show more

```
show license summary
license summary
                #### Current tenant: nfvis-single-tenant

                Smart Licensing is ENABLED

                Smart Licensing Policy:
                  Status: ENABLED

                Transport:
                  Type: SmartTransport
                  URL:
https://smartreceiver-stage.cisco.com/licservice/license
                  Proxy:
                   Not configured

                Trust Code Installed: Jul 05 06:20:56 2023 UTC
                  Last Attempt Date: Jul 05 06:19:31 2023 UTC
                  Last Attempt Status: JobStatus.success
                  Last Attempt Result: success
                  Attempt In Progress: False
                  Next Attempt Date: None
```

```
    Usage Reporting:
    Last ACK received: None
    Next ACK deadline: Oct 01 20:32:19 2023 UTC
    Reporting push interval: 1
    Next ACK push check: None
    Next report push: None
    Last report push: None
    Last report file write: None
  Policy:
    Policy in use: None
    Policy name: Default Policy
    Reporting ACK required: yes
    Subscription:
     First report requirement (days): 90
     Ongoing reporting frequency (days): 90
     On change reporting (days): 90
    Perpetual:
     First report requirement (days): 365
     Ongoing reporting frequency (days): None
     On change reporting (days): 90
    Enforced:
     First report requirement (days): None
     Ongoing reporting frequency (days): None
     On change reporting (days): None
    Export:
     First report requirement (days): None
     Ongoing reporting frequency (days): None
     On change reporting (days): None


  License Usage
  =============
    Name: None
    Short-name-tag: NFVIS_CORE_UCSC
    Name-tag:
regid.2023-04.com.cisco.NFVIS_CORE_UCSC,1.0_81ac6671-e3e7-489e-ae14-9610ae3ccdaf

    Description: None
    Count: 2
    Version: None
    Status: IN USE
    Enforcement_type: NOT ENFORCED
    Feature: None
    Feature description: None
```

- Use the **show license usage** command to view your license usage:

Show more

```
show license usage
license usage
              License Usage
              =============
                Name: None
                Short-name-tag: NFVIS_CORE_UCSC
                Name-tag:
regid.2023-04.com.cisco.NFVIS_CORE_UCSC,1.0_81ac6671-e3e7-489e-ae14-9610ae3ccdaf

                Description: None
                Count: 2
                Version: None
                Status: IN USE
                Enforcement_type: NOT ENFORCED
                Feature: None
                Feature description: None
```

- The license tech support is a part of Cisco NFVIS tech support.

How enforcement works

Summary

At Day-0 (NFVIS connecting to Licensing server)

Workflow

Three triggers for entering Grace are below:

Process details

At Day-0 (NFVIS connecting to Licensing server)

- License requested in PENDING status
- Sync Overdue Timer started.
- Sync Overdue Timer expiry entitlement will move to Out of Compliance (OOC).
- Day-0 enforcements post Sync Overdue Timer expiry
- Day-0 enforcement kicks in after 30 days if no connectivity to License server.
  - NFVIS becomes read-only except NFVIS network settings and licensing configs. List of blocked commands are listed below after Day-N
  - Host action commands are exempted from enforcements
  - Banner in the login prompt displays number of days remaining for Timer expiry
  - Same Banner on the local portal

    In normal case where Sync overdue timer has not expired and customer is able to use license till the term of the license. After term expiration, we enter a grace period.

Day-N enforcements post Grace Period Expiry

- NFVIS becomes read-only. Configuration paths that are blocked are listed below in next bullet.
- Host action commands are exempted from enforcements
- Banner in the login prompt displays number of days remaining for Grace expiry
- Same Banner on the local portal
- Device enforces post Grace expiry

Configuration blocked during Day-0 /Day-N enforcement

- Networking: /networks/network, /bridges/bridge, /pnics/pnic and its /speed, /duplex, /promiscuous, and /track-state subpaths; /routers/router/bgp, /route-distributions, /system/routes, and /host-firewall/default-action
- Security and access: /security:security, /security_servers, /security_servers/aaa/groups/group/user, /security_servers/aaa/groups/group/policy, and /system:system/security/selinux/mode
- System and services: /banner-motd, /netsnmp, /monitor/session, /pnp, /nfvis-ip:ip/hosts, /system/portal, /system/time, /system:system/system:mode, /system:system/storage, and /system/usb-mount
- Maintenance: /system/fwupdate/image-name, /system/fwupdate/apply-image, /system/upgrade/image-name, and /system/upgrade/apply-image

Following configuration paths are exempted from enforcements

- /system:system/settings
- /network
- /license:license/smar

Three triggers for entering Grace are given below:

- License over-consumption (OOC)
- Subscription expiry
- Sync-overdue Timer expiry (For additional Licenses requested at Day-N)

License renewal workflow

Objective

Use these action commands that can help you release, sync and trust Cisco NFVIS licences.

Cisco NFVIS license action commands

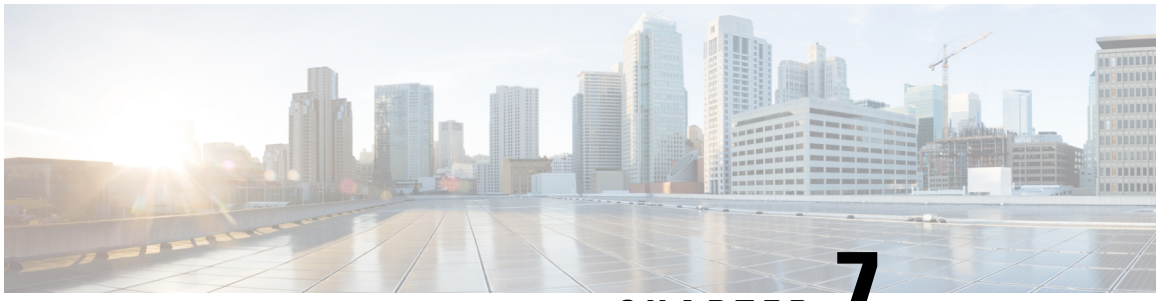
This command reference includes action commands for managing Cisco NFVIS licenses:

- **license smart release**: Use this command to send a "License Usage 0" message to CSSM. This action prompts the CSSM to release the license entitlement associated with your Cisco UCS C Rack servers. If the device continues to operate with Cisco NFVIS after the license release, a usage report, also known

as RUM, is sent to CSSM after a predetermined time interval. This report will indicate the actual number of license units consumed, which will then be reserved again on CSSM. Following a license release, a notification and system logging warning is issued, urging you to cease using Cisco NFVIS as the license has been released. You receive this notification every 8 hours during the 24 hours after a license release. If you continue to use Cisco NFVIS beyond this 24-hour period, a RUM report will be generated reflecting the appropriate number of license units consumed by Cisco NFVIS. This report is sent to the Licensing server based on the set periodic interval, after which the licensing server will reserve the correct number of license units for the device again.

- **license smart sync**: Report a license usage to CSSM using this action command.
- **license smart trust**: Initiate the establishment of trust between the device and the CSSM. This action can prove beneficial in various scenarios, including but not limited to these:

1. If you delete the product instance from CSSM but wish to continue with Cisco NFVIS Licensing.
2. If you transfer your licenses from one Virtual Account to another.
3. If there is an asynchrony between the licensing state on CSSM and on the device.
4. If the licensing certificates on the device reach their expiry date.



CHAPTER 7

NFVIS host management

Objective

The NFVIS host system provides a set of commands to manage power states, maintain file systems, and perform secure data transfers. These operations are performed directly on the NFVIS host to ensure system stability and efficient data management.

- Power management, on page 31
- Secure copy, on page 32
- Reset factory default, on page 33
- Banner and message-of-the-day, on page 34
- Set the system time manually or configure NTP, on page 35
- Back up and restore VMs before an NFVIS factory reset, on page 36
- NFVIS backup and restore restrictions, on page 36
- NFVIS backup and restore behavior, on page 37
- NFVIS factory reset data retention, on page 41

Power management

Objective

Use these commands to control the power state of the NFVIS host. A notification and syslog entry are generated for each operation to indicate that the action was performed.

Power management

Action	Command
Power cycle the system	`nfvis# hostaction powercycle`
Reboot the system	`nfvis# hostaction reboot`
Shut down the system	`nfvis# hostaction shutdown`

Secure copy

Objective

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.

SCP syntax and usage notes

Syntax: **scp***sourcedestination*

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

SCP between two NFVIS devices is not supported.

Examples

Objective

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

SCP file transfer examples

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

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

```
nfvis# 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/copie
```

Reset factory default

Procedure

Reset factory default.

Factory default reset allows you to restore the host server to its original configuration state for troubleshooting purposes.

Factory default reset is available on all NFVIS supported hardware platforms. You can reset the host server to factory default with three different options that provide varying levels of data preservation.

Before you begin

Contact Cisco Technical Support before performing factory default reset.

Follow these steps to reset to factory default:

Reset NFVIS to factory default

The system resets to factory default configuration based on the selected option and reboots successfully.

Procedure

Step 1 Choose the appropriate factory default reset option based on your requirements.

Table 6: Factory default reset options

Option	Description
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.

Step 2 Enter the factory default reset command with your chosen option.

Example:

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

Step 3 Enter Yes when prompted with the factory default warning message or no to cancel.

The system resets to factory default configuration based on the selected option and reboots successfully.

Banner and message-of-the-day

Objective

Configure custom banners and messages to provide information to users when they access the Cisco NFVIS portal.

Banner types and messages of the day

Cisco 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.

Configure the banner and message of the day

Procedure

Enter global configuration mode and configure the banner and message.

```
nfvis# configure terminal
nfvis# banner-motd banner "This is a banner" motd "This is the message of the day"
nfvis# 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.

The custom banner is displayed on the login page and the message of the day appears on the portal's home page after users log in.

Set the system time manually or configure NTP

Procedure

Step 1 Set the system time manually.

This task allows you to configure the system time on Cisco NFVIS either manually or by synchronizing with an external NTP server to ensure accurate timekeeping.

Accurate system time is essential for logging, security, and synchronization with other network devices. You can set the time manually for isolated systems or use NTP for automatic synchronization with time servers.

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

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

Step 2 Set the system time using NTP IPv4.

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

Step 3 Verify the system time configuration using the show system time command in privileged EXEC mode.

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

REMOTE               REFID   ST    T         WHEN       POLL       REACH         DELAY         OFFSET
        JITTER

==============================================================================

*calo-timeserver     .GPS.    1         u           4   64          1         69.423        2749736
        0.000

 * sys.peer and synced, o pps.peer, # selected, + candidate,

- outlyer, . excess, x falseticker, space reject
```

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

The system time is configured either manually or synchronized with NTP servers. The verification command displays the current time configuration and NTP server status.

Back up and restore VMs before an NFVIS factory reset

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.

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

To avoid losing VMs during a host restore because of insufficient disk space, complete these steps.

Procedure

Step 1 Stop the VMs that are associated with Cisco NFVIS.

Step 2 Perform individual image backups of the VMs using the **vmExportAction** command.

Step 3 After the backups succeed, delete the VMs and their images from Cisco NFVIS.

Step 4 After deleting the VMs and images, perform a host-level backup with the configuration-only option by using **hostaction backup configuration-only file-path** *extdatastore2:sample-dir/sample* .

Step 5 Copy the backup files to a file server.

Step 6 Perform a factory reset by using the **factory-default-reset** command.

Step 7 Copy the backup from the file server to NFVIS, and restore the host-level backup file by using **hostaction restore file-path** *extdatastore2:sample-dir/sample.bkup* .

Step 8 If the restore fails because of disk storage issues, restore the configuration-only backup. After the restore succeeds, restore the VMs and their images by using **vmImportAction importPath** */mnt/extdatastore1/tiny_backup.vmbkp* .

NFVIS backup and restore restrictions

Observe these restrictions when you back up or restore 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.
- The time required to back up a VM depends on the selected backup option.
- 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 **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.

NFVIS backup and restore behavior

Objective

This reference describes how NFVIS backup and restore options affect configurations, VMs, and images.

Backup commands and status

The following examples shows backup options and status:

```
nfvis# hostaction backup configuration-and-vms file-path intdatastore:sample
nfvis# hostaction backup configuration-only file-path extdatastore2:sample-dir/sample
nfvis# show hostaction backup status
hostaction backup status 2020-07-16T07:02:44-00:00
destination intdatastore:backup_20200704.bkup
status      BACKUP-SUCCESS
size        "2798.0 MB"
components FIREWALL
  status      BACKUP-SUCCESS
  last update 2020-07-16T07:07:38-00:00
  size        "20.49 MB"
  details     ""
components Linux
  status      BACKUP-SUCCESS
  last update 2020-07-16T07:07:36-00:00
  size        "0.01 MB"
  details     ""
components NFS
  status      BACKUP-SUCCESS
  last update 2020-07-16T07:06:44-00:00
  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 2020-07-16T07:07:35-00:00
  size        "579.89 MB"
  details     ""
components VM_Images_Flavors
  status      BACKUP-SUCCESS
  last update 2020-07-16T07:06:45-00:00
  size        "2197.73 MB"
  details     ""
```

Restore options and outcomes

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.

Condition	Restore configuration-only	Restore configuration-and-vms
Restore system configurations	Yes	Yes
Restore upgrade configurations	yes, requires same upgrade files in system if the host backup was taken has such upgrade files. No, if host where backup was taken did not have any upgrade files registered. Restore will fail.	Yes, requires same upgrade files in system if the host backup was taken has such upgrade files. No, if host where backup was taken did not have any upgrade files registered. Restore will fail.

Condition	Restore configuration-only	Restore configuration-and-vms
Restore registered images and flavors	Yes, if images sources are still available (URL link is still valid, or uploaded files are still in the same locations). 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.	Yes, restore from backup file.
Restore deployments	No	Yes, restore from backup file.

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.

Condition	dpdk-disabled while backup	dpdk-enable while backup
dpdk-disabled while restore	Yes (system is dpdk-disabled)	Yes (system will beconverted to dpdk enabled, and VM vnic will be convereted inf needed)
dpdk-enabled while restore	No support	Yes (system is dpdk-enabled)

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

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

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

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

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     ""
```

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.

NFVIS factory reset data retention

Objective

This reference lists data retained or deleted by each NFVIS factory-default-reset option.

Backup, restore, and factory-default-reset

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:

Data	factory-default-reset all	factory-default-reset all-except-images	factory-default-reset all-except-images-connectivity
files under intdatastore	Retain	Retain	Retain
files under intdatastore/uploads/	Delete	Delete	Delete
files under extdatastore${1,2}	Delete	Retain	Retain
files under extdatastore${1,2}/uploads/	Delete	Delete	Delete
files under USB	Retain	Retain	Retain
files under NFS mounted datastore	Retain	Retain	Retain
Deployments	Delete	Delete	Delete
Registered Images and Flavors	Delete	Retain	Retain

Failure to restore

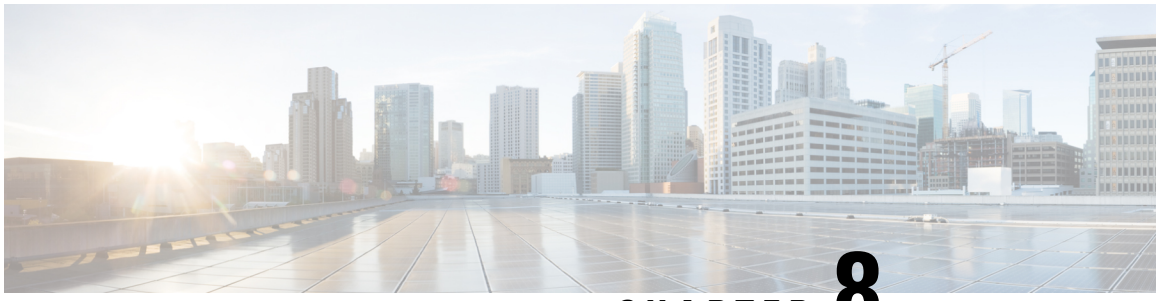
NFVIS configurations fail to restore if there is no sufficient disk space. Restore requires temporary disk space to save un-compressed files. You can move, copy or upload the backup file to a larger datastore and run system restore.

```
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"
```

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
```



CHAPTER 8

Virtual machine creation and management

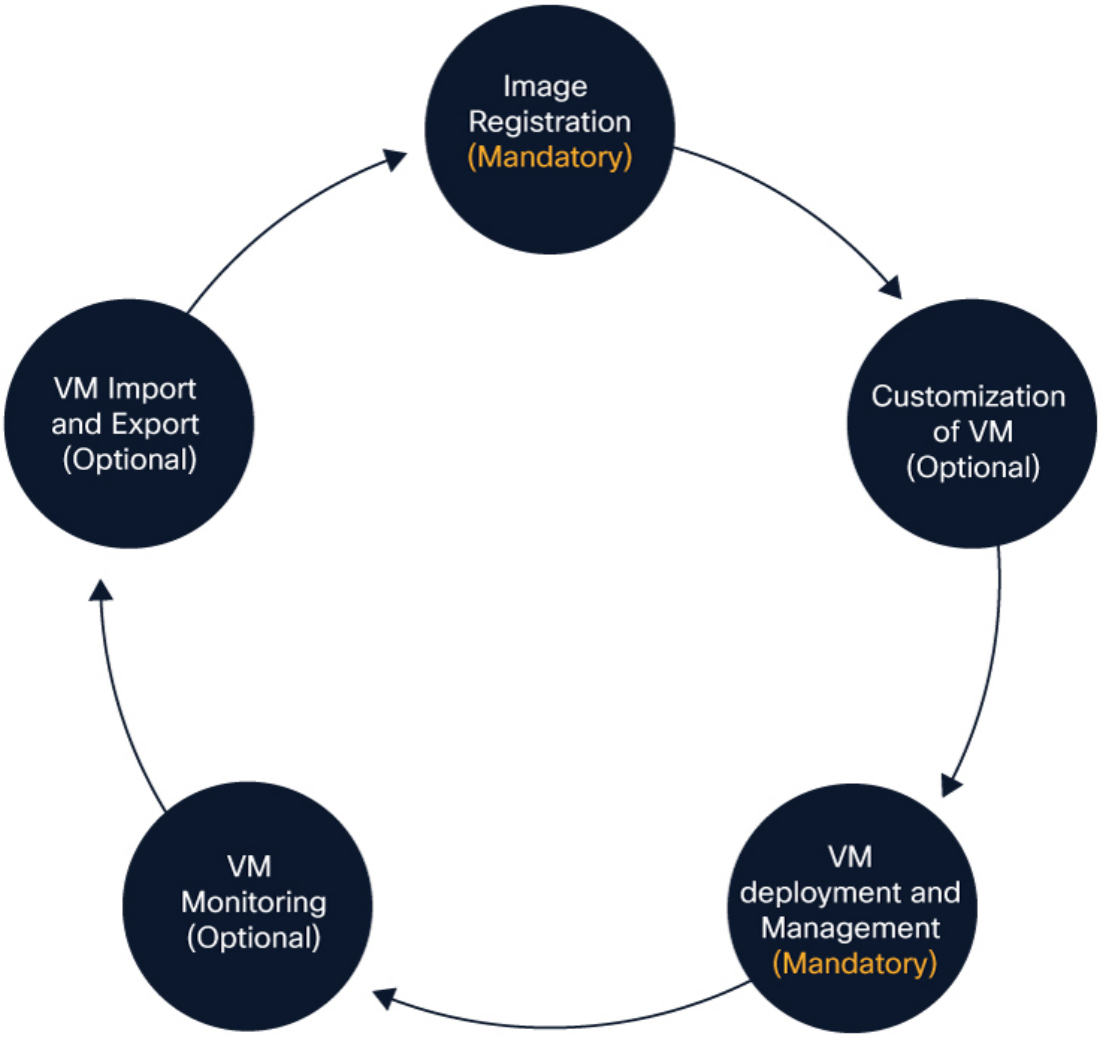
Objective

VM life cycle management is a process that

VM deployment workflow

- refers to the entire process of registering, deploying, updating, monitoring VMs
- gets VMs service chained as per your requirements, and
- can be performed using a set of REST APIs or NETCONF commands or the Cisco NFVIS portal.

VM Deployment Workflow Diagram:



- Uploading VM images to an NFVIS server, on page 44
- Register a virtual machine image, on page 45
- Customization of VM, on page 46
- Creating VM, on page 53
- Additional capabilities, on page 62

Uploading VM images to an NFVIS server

Uploading VM images to an NFVIS server is a file transfer process that copies VM images to the default location (/data/intdatastore/uploads) on the host server.

VM image upload methods

You can upload VM images to an NFVIS server in these ways:

- Copy the images from your local system to the NFVIS server—Use the Image Upload option from the Cisco NFVIS portal.
- Copy using the scp command (scp username@external_server:/path/image.tar.gz intdatastore:image.tar.gz).

Register a virtual machine image

Use this task to register a Virtual Machine (VM) image hosted on a remote server (HTTPS or SCP) for use in Cisco NFVIS. This process allows NFVIS to download the ISO image and OVA file, parse the metadata, and automatically create deployment profiles and flavors.

Table 7: Protocols

Protocol	Authentication	Notes
HTTPS	Not supported	No Username and password
SCP	Required	Username and password

Procedure

Step 1 From the Cisco NFVIS portal, choose **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.

Step 2 Select the remote image registration option.

Register Image.

Select **Remote Image Registration**.

Step 3 Configure image properties.

Enter a descriptive name in the **Image name** field.

Step 4 Configure the remote server connection.

Select HTTPS or SCP from the **Protocol** drop-down list.

Enter the server hostname or IP address in the **IP Address** field.

Enter the directory path to the ISO file in the **Image File Path** field.

Step 5 Configure authentication.

If you selected FTP, SFTP, or SCP as the protocol, enter the username and password for the remote server.

Note Authentication is not supported for HTTP or HTTPS protocols.

Step 6 Upload the OVA metadata.

Select the **Metadata** check box.

Enter the path to the OVA file in the **OVA File Path** field.

Example: `/home/user/images/cucm_15.0_vmv17_v1.2.sha512.ova`



Note The OVA file must reside on the same remote server as the ISO file.

Step 7 Configure additional options.

Select the **Dedicated Cores** check box if dedicated CPU cores are required for the virtual machine.

Step 8 Submit the registration request.

Click **Submit** to submit the registration request.

Step 9 Monitor the registration progress.

Track the registration progress in the **Image Repository**. Downloading large files can take several minutes.

NFVIS downloads the ISO image and OVA file, parses the OVA metadata, registers the image, creates profiles and flavors, and makes the image available for deployment.

Customization of VM

Objective

This reference groups information about Customization of VM, including VM profiles or flavors, VM profile creation methods.

VM profiles or flavors

A VM profile or flavor is a configuration template that

- defines VMs in terms of number of parameters for how to run the VM
- specifies parameters such as number of vCPUs, RAM, disk size and so on, and
- provides standardized VM configurations based on requirements.

VM profile creation methods

Objective

VM profiles are created using different methods depending on the image package type:

VM profile creation methods

- Flavors are created as part of image registration if you use the tar.gz image packages for registering a VM.

- For other image packages such as .qcow2, iso, and raw, you must define custom flavors based on your requirements.
- See Install Guides of supported applications for what types of Images and Profiles are supported.

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.

Create VM profile using REST API

Procedure

Step 1 Create VM profile using REST API.

Create a VM profile to specify the resource allocation and configuration parameters for virtual machines in your environment.

VM profiles define the hardware specifications for virtual machines including memory allocation, disk space, and CPU configuration. Use REST API calls to programmatically create these profiles for consistent VM deployment.

Step 2 Execute the following cURL command to create a VM profile using the 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>

    <memory_mb>4096</memory_mb>

    <root_disk_mb>12288</root_disk_mb>

    <swap_disk_mb>0</swap_disk_mb>

    <vcpus>2</vcpus>

</flavor>'
```

The VM profile is successfully created with the specified configuration parameters and can be used for virtual machine deployment.

Internal management networks

An internal management network is a separate network within the Cisco NFVIS infrastructure that

- handles communication between the Cisco NFVIS and other management-related functions like port forwarding
- is used for VM monitoring and recovery, and

- is created by Cisco NFVIS as a network named int-mgmt-net by default.

How default configuration works

Summary

Here is an example of the default int-mgmt-net configuration:

Workflow

You can update the default configuration before deploying VMs/VNFs. When the default config is deleted, Cisco NFVIS recreates the internal management network upon rebooting the host.

Process details

```
vm_lifecycle networks network int-mgmt-net
 subnet int-mgmt-net-subnet
  address 10.20.0.0
  netmask 255.255.255.0
  gateway 10.20.0.1
 !
!
```

VNF deployment monitoring: Cisco NFVIS lets you configure the VNF deployment to be monitored. To monitor the VM deployment, an interface is attached to the internal management network (int-mgmt-net). The interface attached contains the NIC ID 0 by default.

Internal Management Network in the VM Deployment Payload	IP ADDRESS token/variable in Day-0 config	Behavior

Example configuration using IP address:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`<ip_address>10.20.0.21</ip_address>`
`</interface>`
	Token ${NICID_0_IP_ADDRESS}" is either present or absent.	If the IP payload is available, the payload is assigned to the interface.
		If there is a token/variable, the IP ADDRESS is replaced with the token.
		When the IP ADDRESS in the IP payload is unavailable or incorrect, the deployment fails.

Example configuration when no IP ADDRESS is configured:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`</interface>`
	${NICID_0_IP_ADDRESS} token is present in the day-0 config.	Cisco NFVIS auto assigns the IP ADDRESS and the token/variable is replaced in the day-0 config with the same IP address and the token/variable is replaced in the day-0 config with the same IP address.

Internal Management Network in the VM Deployment Payload	IP ADDRESS token/variable in Day-0 config	Behavior

Example configuration when no IP ADDRESS is configured with the NIC ID being absent:
`<interface>`
`<nicid>0</nicid>`
`<network>int-mgmt-net</network>`
`</interface>`
	Not present
	By default, the interface is configured as DHCP. The token is not replaced.
		Note
		Starting from Cisco NFVIS Release 4.12.1, configuring an interface as a DHCP server is supported.
		The IP ADDRESS is auto allocated using the internal DHCP server. Cisco NFVIS utilizes the IP address for monitoring the deployment.

Here is a sample day-0 config with the IP address and the token/variable:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="${SSH_USERNAME}"/>
    <Property oe:key="com.cisco.c8000v.login-password.1" oe:value="${SSH_PASSWORD}"/>
    <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
!!!GigabitEthernet1-nicid(0)-int-mgmt-interface-don't change ip address or don't shutdown
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1"
oe:value="${NICID_0_IP_ADDRESS}/${NICID_0_CIDR_PREFIX}"/>         ===> IP address set via
token
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="${TECH_PACKAGE}"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="interface
GigabitEthernet1"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip address
${NICID_0_IP_ADDRESS} ${NICID_0_NETMASK}"/>   ===> IP address set via token
    <Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0012" oe:value="ip route vrf Mgmt-intf
0.0.0.0 0.0.0.0 ${NICID_0_GATEWAY}"/>
  </PropertySection>
</Environment>
```

Here is a sample day-0 config using DHCP server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="True"/>
    <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="${SSH_USERNAME}"/>
```

```
    <Property oe:key="com.cisco.c8000v.login-password.1" oe:value="${SSH_PASSWORD}"/>
    <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
!!!GigabitEthernet1-nicid(0)-int-mgmt-interface-don't change ip address or don't shutdown
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value=""/>
                ===> IP address left blank here
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="${TECH_PACKAGE}"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="vrf definition Mgmt-intf"/>

    <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="address-family ipv4"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="address-family ipv6"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="exit-address-family"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="interface
GigabitEthernet1"/>
   <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="vrf forwarding Mgmt-intf"/>

   <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip address dhcp setroute"/>
              ===> DHCP specified vs IP address
    <Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="no shut"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="exit"/>
  </PropertySection>
</Environment>
```

VNF volumes

Objective

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

Objective

Example: payload for creating volumes

VNF volume restrictions and payload examples

- 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.

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>
- <storage_location>datastore2</storage_location>
- <storage_location>datastore3</storage_location>

Volume deployment parameters

Objective

Cisco NFVIS supports the fields for volume deployment shown in this table.

Volume properties and allowed values

Property	Allowed Values	Description
NAME	string, { length 1..255 }	NAME of the volume
volid	uint16	Volumes will be presented to the VM sorted by volume ID.
bus	virtio, ide, scsi	The bus type
size	unit 16	Size of the Volume
sizeunit	MiB,GiB,TiB,PiB	Size units. MiB/GiB/TiB/PiB/EiB
format	qcow2, raw	Format of the disk to be created.
device_type	disk	Type of the device being attached to the VM.
storage_location	local, datastore1, datastore2, datastore3, nfs:NFS_MOUNT_NAME	Storage location NAME **Note** Starting from NFVIS 4.1 release, external datastore storage location is supported.

Port forwarding

Port forwarding is a network configuration mechanism that

- redirects incoming traffic from WAN to access the internal management network of the VM
- uses the WAN bridge interface (WAN-br) by default for traffic redirection, and
- allows modification of the bridge interface using the source_bridge tag in the deployment payload.

Port forwarding configuration

Objective

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 in this example:

Port forwarding configuration examples

```
<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 this payload, the traffic coming from the WAN side is redirected through the management interface (MGMT) instead of the default WAN bridge (WAN-br) interface.

vCPU pinning

vCPU pinning is a resource allocation mechanism that

- assigns VMs to specific virtual CPUs for dedicated processing
- enables low latency configuration through dedicated core allocation
- reuses the Node.js CPU allocation API to interface with NFVIS CPU allocation scripts.

vCPU pinning configuration

Objective

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.

Container pinning and volume details

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.



Note

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

Example: Container volume configuration

To mount a container using a volume or a specific path, use this 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>
```

Creating VM

Objective

This reference groups information about Creating VM, including VM deployment, VM Deployment using granular RBAC.

VM deployment

VM deployment is a virtualization process that

- creates and configures virtual machines through API or CLI methods
- requires prior registration of VM images before deployment can begin, and
- accepts mandatory and optional parameters to customize the deployment configuration.

VM name requirements

Objective

The VM name must meet these requirements:

VM name 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.

Note Ensure that you have registered a VM image before attempting to deploy it. For more details, see Image Registration.

Example: deploy VMs using REST API

This example demonstrates how to deploy a VM using REST API with a sample payload and how to verify the deployment status.

This is 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>
    <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>
        <variable>
          <name>TECH_PACKAGE</name>
          <val>ax</val>
        </variable>
      </configuration>
    </config_data>
  </vm_group>
</deployment>'
```

VM deployment verification

Objective

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

VM deployment operational data

show vm_lifecycle opdata tenants tenant admin deployments @@VAR0@@/@@VAR1@@/@@VAR2@@

```
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
```

Note 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

Objective

VNFs can be deployed using multiple mandatory and option parameters.

VM deployment parameter definitions

Parameter	Notes	Example
Image	Mandatory. The image needs to be registered and active when it is being referred in deployment.	<IMAGE>ISR_IMAGE</IMAGE>
bootup_time	This parameter is no longer mandatory from 3.12 and later, provided that it is specified in IMAGE properties. Accepted Values: For unmonitored VMs: -1 For monitored VMs: Number of seconds	<bootup_time>500</bootup_time>
vim_vm_name	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.	
kpi_data	Mandatory for monitored VMs.	
rules	Mandatory for monitored VMs.	
config_data	Mandatory if the day-0 configuration has variables that have tokens assigned to them.	
Encrypted config Variable	Optional. Only one value for a Variable is allowed to be encrypted.	<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.	<placement> <type>zone_host</type> <host>nfs:nfs_storage</host> </placement>

Parameter	Notes	Example
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)	
port_forwarding	Optional. If port forwarding is included, all elements under it are mandatory.	<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>
Ngio interface	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.	<VARIABLE> <name>ngio</name> <val>enable</val> </VARIABLE>
interface model	Optional. If the model is not specified for an interface, the default model is used. For Windows, the default model is rtl8139.	[illegible]
VNC	Optional. If the VNC password is not specified, there is no default password.	<VNC><password>vnc_password</password></VNC>

VM bootstrap configuration options with VM deployment

Objective

VM bootstrap configuration options with VM deployment are methods that

VM bootstrap configuration options

- enable inclusion of bootstrap configuration (day zero configuration) of a VM in the VM deployment payload
- provide flexibility in how configuration files are handled during deployment, and
- support different approaches based on deployment requirements and infrastructure setup.

Bootstrap configuration methods

Objective

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

Bootstrap configuration methods

- 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.

VNF deployment placement

VNF deployment placement is a parameter mechanism that

- allows you to specify where a VNF should be deployed using the placement tag for parameter deployment
- supports various placement parameters with accepted values, and
- enables precise control over VNF location within the Cisco NFVIS environment.

Placement configuration requirements

Objective

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.

VNF placement configuration examples

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>
  <flavor>windows</flavor>
  <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>'
```

Configure granular RBAC for VM deployment

The system administrator can define a set of groups and assign VNFs to these groups. When you create a user, you can assign that user to one of these groups to control VNF access.

Procedure

Step 1 Create a VM with a resource group.

```
nfvis(config)# vm_lifecycle tenants tenant admin deployments deployment sample resource_group
localgroup2 vm_group test image centos flavor small vim_vm_name sample bootup_time -1 recovery_wait_time
 0 recovery_policy recovery_type AUTO action_on_recovery REBOOT_ONLY monitor_on_error false
nfvis(config-vm_group-test)# commit
```

Step 2 Move the VM to another resource group.

```
nfvis(config)# vm_lifecycle tenants tenant admin deployments deployment sample

nfvis(config-deployment-sample)# no resource_group

nfvis(config-deployment-sample)# commit

Commit complete.

nfvis(config-deployment-sample)# resource_group localgroup

nfvis(config-deployment-sample)# commit

Commit complete.
```

Step 3 View the VMs and their resource group assignments.

```
nfvis# show vm_lifecycle deployments

 Name           Monitored  State           Internal-State           GroupInfo

-------------------------------------------------------------------------------

centos1        No         ALIVE           VM_INERT_STATE           localgroup

cetos_global   No         ALIVE           VM_INERT_STATE                                Denotes global


sample         No         ALIVE           VM_INERT_STATE           localgroup
```

Step 4 View the resource groups, users, and VM mappings.

Example:

```
nfvis# show group-summary

 Group_Name             Policies                    Users                    Vms

-------------------------------------------------------------------------------------------------

localgroup         resource-access-control        local_oper            centos1.centos1

                                                  sample.test

-------------------------------------------------------------------------------------------------

localgroup2        resource-access-control         test_admin

-------------------------------------------------------------------------------------------------

                           NA                       admin            cetos_global.cetos_global

                                                   localAdmin
```

The VNF is deployed with granular RBAC configuration, and you can view the resource group assignments and user mappings for proper access control.

Additional capabilities

Objective

This reference groups information about Additional Capabilities, including Deploy CUCM application, CDROM attachment and detachment, Recover the password for a Cisco CUCM application, Graceful VM stop.

Deploy CUCM application

This procedure follows successful registration of the CUCM application image in NFVIS. The deployment creates a functional CUCM virtual machine with automated initial configuration.

Before you begin, ensure that you have:

- A registered CUCM application image (ISO and OVA) in NFVIS.
- A Bootstrap ISO with Day0 configuration for the CUCM application.

For instructions on creating the Bootstrap ISO with Day0 configuration, see the CUCM documentation: Touchless Installation Task Flow.

Procedure

Step 1 Open the deployment page in the NFVIS portal.

From the NFVIS portal, click **Configuration** > **Deploy**.

Step 2 Select a VM type.

Select a VM type from the drop-down list.

If your application type is not listed, select **OTHER**.

Step 3 Select the CUCM image.

From the **Image** drop-down list, select the desired image. Images are populated based on the VM type selected.

For example, `Bootable_UCSInstall_UCOS_15.0.1.ISO`.

Step 4 Select a flavor.

From the **Flavor** drop-down list, select the appropriate flavor based on your sizing requirements. Flavors are automatically created from the OVA during image registration.

Step 5 Configure the network connections.

In the **Network Design** section, locate the **VM icon** on the canvas.

Add additional networks as required.

Drag a line from the VM icon to the desired network to connect the VM to the appropriate networks for your deployment.

Step 6 Select the Bootstrap ISO option.

Select the **Bootstrap ISO** check box.

Step 7 Upload the Day0 configuration ISO.

Click **Upload** and select your Day0 configuration ISO.

Step 8 Deploy the virtual machine.

Click **Deploy**.

The system validates the configuration. If validation errors occur, they are displayed for correction.

Step 9 Monitor the deployment and complete the installation.

Track the deployment progress and wait until the VM state changes to ACTIVE.

After the VM is ACTIVE, click the **Terminal** button in the VM row. Monitor the installation progress in the console and follow the application-specific installation prompts.

Table 8: Virtual machine deployment statuses

Status	Description	Action
DEPLOYING	VM is being created.	Wait for completion.
ACTIVE	VM is running.	Access the console and verify operation.
ERROR	Deployment failed.	Check logs and troubleshoot.
SHUTDOWN	VM is powered off.	Power on the VM if needed.

Note The deployment steps for Cisco Unity Connection and Cisco IM and Presence Service (IM&P) on NFVIS are the same as those for Cisco Unified Communications Manager (CUCM). These steps include:

Image registration: Register the ISO and OVA metadata using the same process as CUCM.

Virtual machine deployment: Deploy the VM with a Bootstrap ISO using the same process as CUCM.

Virtual machine management: Perform power operations, console access, and network updates as for CUCM.

For application-specific information, refer to the respective Cisco documentation for Cisco Unity Connection and Cisco IM and Presence Service.

The CUCM virtual machine is deployed on NFVIS with the VM status showing as ACTIVE, and the Day0 configuration from the Bootstrap ISO is applied automatically during installation.

CDROM attachment and detachment

CDROM attachment and detachment is a VM management capability that

- allows you to connect or disconnect ISO images to a VM

- supports various use cases, including system recovery, operating system installation, and software deployment, and
- enables the VM to access content from a virtual CDROM drive.

How CDROM attachment and detachment work

Summary

Attaching a CDROM to a VM is a two-step process:

Workflow

1. Image Registration: First, you must register the desired ISO image using the existing image registration workflows provided by NFVIS. This makes the ISO available for selection. For information about ISO image registration, see Image Registration.
2. CDROM Action: After registration, you can proceed to attach the registered ISO to the target VM by utilizing the dedicated CDROM actions available in the **Manage Deployments** page.

Process details

To detach an ISO image, you would use the corresponding detachment action in the **Manage Deployments** page. Both attachment and detachment operations require the VM to be in a shutdown state.

Requirement: prerequisites for CDROM attachment and detachment

To successfully perform CDROM attachment or detachment operations, the VM must be in a shutdown state. These operations cannot be initiated or completed on an active VM.

Manage CDROM attachment and detachment

Attach or detach ISO images to and from virtual machines for recovery and maintenance operations.

CDROM attachment and detachment operations allow you to mount ISO images to virtual machines for recovery purposes or maintenance tasks. These operations can only be performed when the VM is in a shutdown state.

- Ensure the ISO image you intend to attach is registered using the image registration workflows. For more information, see Image Registration.
- Verify that the VM is in a shutdown state. CDROM attach and detach operations cannot be performed on an active VM.

Follow these steps to manage CDROM attachment and detachment:

Procedure

Step 1 From the **Manage Deployments** page, click the **CD-ROM** icon.

Step 2 Choose the operation you want to perform.

To attach an ISO image, click **Attach**.

To detach an ISO image, click **Detach**.

Step 3 Select the appropriate image or disk.

Table 9: CDROM attachment and detachment options

If...	Then...
You are attaching a CDROM	From the **Images** drop-down list, choose the desired recovery ISO from the list of registered ISO images.
You are detaching a CDROM	From the **Disk Name** drop-down list, choose the disk name that corresponds to the recovery ISO you want to detach.

Step 4 Click **Submit**.

A confirmation message appears indicating the CDROM operation is complete.

After attaching a CDROM, to use the attached recovery ISO, you must start the VM and access its boot menu to select the ISO disk for booting. After detaching a CDROM, you can start the VM for normal operation. To confirm detachment, click **CD-ROM** > **Detach**. The ISO image should no longer be listed in the **Disk Name** drop-down list.

Recover a Cisco CUCM application using a recovery ISO

This task allows you to recover or repair a Cisco CUCM application when the virtual machine requires restoration from a corrupted or damaged state.

Use this procedure when a Cisco CUCM application needs recovery or repair. The process involves using recovery ISO media to restore the application to a functional state through the NFVIS portal interface.

Ensure you have access to the NFVIS portal and the appropriate recovery ISO file for your CUCM application version.

Follow these steps to recover a Cisco CUCM application using a recovery ISO:

Procedure

Step 1 From the NFVIS portal, click **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.

Step 2 Upload the Recovery ISO.

a) If Recovery ISO registration is not allowed, upload the respective VM OVA and register it.
b) Name the image descriptively. For example, CUCM-15-recovery-ISO.

Step 3 Shut down the VM.

a) Click **Configuration** > **Virtual Machine** > **Manage**.
b) Select the VM.
c) Click **Switch Power** to shut down the VM.
d) Wait until the VM state changes to SHUTDOWN.

Step 4 Attach the Recovery ISO to the **CD-ROM**.

a) With the VM in the SHUTDOWN state, click the **CD-ROM** button.

b) Click **ATTACH**.
c) Select the **Recovery ISO** from the drop-down list.
d) Click **Submit**.

The Recovery ISO is attached to the VM.

Step 5 Power on and boot from the Recovery ISO.

a) Click **Switch Power** to start the VM.
b) Click **Terminal** to access the console.
c) Enter the Boot Menu by pressing the appropriate key during startup, typically F12 or ESC.
d) Select the **Recovery ISO device** from the boot menu.
e) Follow the application-specific recovery procedure in the console.

The VM boots from the Recovery ISO.

Step 6 Detach the **CD-ROM** after recovery.

a) Shut down the VM.
b) Click the **CD-ROM** button.
c) Select the **DETACH** action.
d) Select the **Recovery ISO disk**.
e) Click **Submit**.

Step 7 Power on the VM for normal operation.

The Cisco CUCM application is recovered and restored to a functional state. The VM boots normally and the application is ready for use.

Recover the password for a Cisco CUCM application

Use this procedure to reset the password for a Cisco CUCM application while the VM is in an active state.

Cisco NFVIS allows you to eject and insert ISO media, which is a required step in the password recovery process for CUCM applications. This procedure covers only the NFVIS media operations. For the complete application-specific password recovery procedure, refer to the Reset or Change CUCM OS Admin and Security Password guide in the official CUCM documentation.

Follow these steps to recover the password for a Cisco CUCM application:

Procedure

Step 1 From the NFVIS portal, choose **Configuration** > **Virtual Machine** > **Manage**.

Step 2 Verify the VM state.

Ensure the VM is running and in an ACTIVE state as password recovery is performed on a running VM.

Step 3 **Eject** the current ISO.

a) Select the VM.
b) Click the **CD-ROM** button.
c) Select **Eject**.

d) Select the ISO image to eject.
e) Click **Submit**.

Step 4 **Insert** the required ISO.

a) Click the **CD-ROM** button.
b) Select **Insert**.
c) Select the required ISO from the drop-down list.
d) Select the appropriate **Device ID** for the **CD-ROM**.
e) Click **Submit**.

Step 5 Complete the password recovery.

a) Access the VM console by clicking the **Terminal** button.
b) Follow the application-specific password recovery steps provided in the CUCM documentation.

The ISO media is successfully swapped, and the application-specific password recovery process is initiated.

Graceful VM stop

A graceful stop is a VM power management option that

- sends an ACPI signal to the operating system within the VM
- allows services and applications to terminate in an orderly manner before the VM powers off, and
- contrasts with forceful shutdown, which immediately cuts power to the VM.

Key features and behavior

- Default Behavior: When initiating a VM shutdown, the **Graceful stop** checkbox is selected by default for VMs deployed with type as OTHER.
- Timeout: The graceful stop operation has a timeout period of 20 minutes.
- Error State: If the VM does not shut down gracefully within the 20-minute timeout, it will transition to an error state.
- Forceful Fallback: If a graceful stop fails or the VM enters an error state, you can uncheck the **Graceful stop** option to perform a forceful shutdown.

To shut down a VM gracefully, see Perform a graceful stop.

Perform a graceful stop

Use a graceful stop for routine VM shutdowns to allow running processes to complete and help prevent data loss.

For VMs deployed with the type set to OTHER, the **Graceful stop** check box is selected by default. The graceful stop operation can take up to 20 minutes.

Procedure

Step 1 From the **Manage Deployments** page, click the **Switch Power** icon for the VM that you want to stop.

For VMs deployed with the type set to OTHER, the **Graceful stop** check box is selected by default.

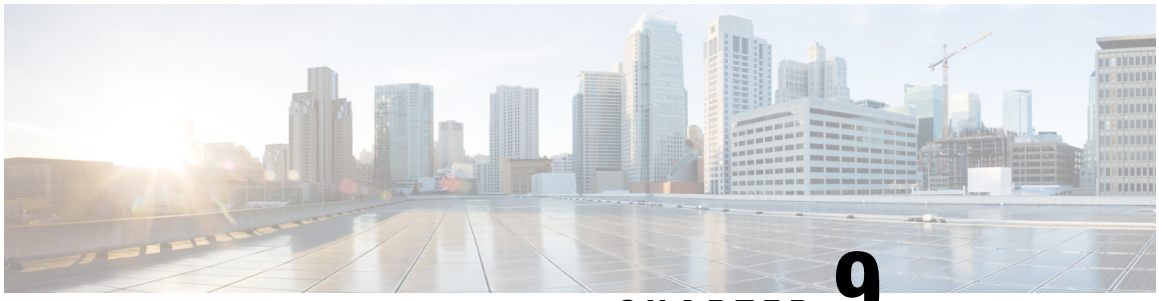
Step 2 Click **OK** to confirm the VM shutdown.

The graceful shutdown process begins in the background.

Step 3 Monitor the VM status in the management interface.

If the VM does not shut down within 20 minutes, it moves to an error state. To perform a forceful shutdown, repeat this procedure and clear the **Graceful stop** check box.

The VM status changes from active to shutdown, indicating that the graceful stop succeeded.



CHAPTER 9

NFVIS upgrade

Objective

Upgrade Cisco NFVIS to a newer version to ensure the NFVIS platform is running the desired software version and includes any new features or fixes.

NFVIS release types and version numbering

NFVIS Release Types

NFVIS release numbering is X.Y.Z, where

- X is major feature release
- Y is minor feature release within the major
- Z is maintenance release with the major/minor release train.

- NFVIS upgrade images, on page 69
- NFVIS supported upgrade paths, on page 70
- NFVIS pre-Upgrade checklist, on page 70
- Upgrade NFVIS, on page 71

NFVIS upgrade images

Objective

Upgrade images can be found in the same place as installation images: software.cisco.com, Enterprise NFV Infrastructure Software.

NFVIS upgrade image details

The Cisco NFVIS upgrade image is available as a .iso file. The upgrade image is digitally signed to ensure cryptographic integrity and authenticity. In addition, all software packages in the image are verified during the upgrade.

NFVIS supported upgrade paths

Objective

This matrix shows the supported upgrade paths from older Cisco NFVIS versions to 26.2 versions in this guide, including the required image types for each upgrade scenario.

NFVIS upgrade matrix

Table 10: Upgrade matrix for upgrading Cisco NFVIS

Running Version	Supported Upgrade Version	Supported Upgrade Image Type(s)
4.18.2a and subsequent 4.18.z	26.2.1	iso

Cisco NFVIS supports version upgrades from version X to versions X+1, X+2 and X+3.

- Version downgrades using an .ISO file are not supported.
- Version upgrades from 4.18 to 26.1 using an .ISO file are not supported.

In case of an error while upgrading, Cisco NFVIS reverts to the source version.

NFVIS pre-Upgrade checklist

1. Plan for a maintenance outage. You should only upgrade NFVIS during a maintenance window as downtime is required.

   - The upgrade process typically takes approximately 20-30 minutes to complete.
   - During the upgrade process, the NFVIS software is not available for use, and all VMs in a running state will be powered off automatically gracefully.
   - While the NFVIS upgrade is in progress, ensure that the system is not powered off. If the system is powered off during the NFVIS upgrade process, the system may become inoperable, and you may need to reinstall the system.
   - After the upgrade completes, the NFVIS software will be accessible, and all VMs will be restored to state prior to upgrade.

1. Determine your desired destination NFVIS version and verify its compatibility with your hardware and application VMs.
2. Take a fresh backup of the NFVIS software
3. Take a fresh backup of all your applications (see each application's technical documentation for supported methods).
4. Verify NFVIS has enough free disk space to upload the NFVIS upgrade image and complete the upgrade.
5. Locate the NFVIS upgrade image for your desired destination release.

Upgrade NFVIS

Objective

This reference groups information about Upgrade NFVIS, including Using Portal, Using CLI.

Upgrade NFVIS using the portal

Complete the NFVIS pre-Upgrade checklist before you begin.

Use the NFVIS portal to upload an upgrade image from your local computer or a remote SCP server.

Procedure

Step 1 From the NFVIS portal, click **Operations** > **Upgrade**.

The **System Upgrade** section has **LOCAL** and **REMOTE** tabs. The **Upgrade Image** section lists registered upgrade images.

Step 2 Choose how to upload the upgrade image.

- To upload an image from the computer that runs your web browser, select the **LOCAL** tab. Click **Select File (.iso)** and browse to the upgrade image.
- To upload an image from a remote SCP server, select the **REMOTE** tab. For the required fields, see Table 1.

Step 3 For a remote upload, enter the server and image details, then click **Submit**.

Table 11: Upload an upgrade image from a remote server

Field	Description
Image Name	Enter a descriptive name for the upgrade image.
Protocol	Select **HTTP** from the drop-down list. Remote registration currently supports only HTTP.
IP Address	Enter the SCP server IP address.
Port	Enter the port number. This field is optional and defaults to 80.
Image File Path	Enter the path to the upgrade image on the SCP server.
Image Checksum	Enter the SHA512 checksum value.

Step 4 Wait for NFVIS to upload and register the upgrade image.

For remote uploads, a progress bar is not available. The upload and registration process runs in the background.

When the process completes, the image appears in the **Upgrade Image List** with a status of Valid.

Step 5 Verify that the upgrade image is registered.

In the **Upgrade Image List**, verify that the image shows:

- **Status**: Valid
- **Version**: The correct upgrade version
- **Upload Date**: The registration timestamp

Step 6 Schedule the upgrade.

a) In the **Upgrade Image** section, click the **+** icon.
b) Select the registered upgrade image from the list.
c) Choose when to run the upgrade.

To upgrade now, set the schedule time to 0 hours. To upgrade later, specify the number of hours from now when the upgrade should begin.

d) Click **Submit**.

The **Status** column shows the upgrade status.

Step 7 After the upgrade completes, verify the NFVIS version and VM operation.

a) From the NFVIS portal, click **Platform** > **About Platform** and verify the new version.
b) Verify that all VMs are powered on and operational.

NFVIS is upgraded to the selected version, and the VMs are powered on and operational.

Upgrade NFVIS using the CLI

Complete the NFVIS pre-Upgrade checklist before you begin.

Use this procedure to transfer the upgrade image from an external SCP server to the NFVIS internal datastore and upgrade NFVIS from the CLI.

Procedure

Step 1 Copy the NFVIS upgrade image from the external SCP server to the NFVIS internal datastore.

Use this command syntax:

scp *source-path destination-path*

Specify the source path as the exact path of the image on the external SCP server in the format *user@remotehostip:file*. Specify the destination as **intdatastore**:*image-filename*.

For example:

```
nfvis# scp admin@192.0.2.9:/NFS/2022-01-23/13/nfvis/iso/Cisco_NFVIS-4.8.0-13-20220123_020232.iso
intdatastore:Cisco_NFVIS-4.8.0-13-20220123_020232.iso
```

NFVIS uploads the upgrade image.

Step 2 Register the upgrade image.

a) Enter configuration mode using **config terminal**.
b) Register the image using **system upgrade image-name** *image-filename* **location** *image-path*.

   Set *image-path* to the full path in the internal datastore, for example, */data/intdatastore/uploads/Cisco_NFVIS-4.8.0-13-20220123_020232.iso*.

c) Commit the configuration using **commit**.

Step 3 Verify that the upgrade image is registered.

Run either **show system upgrade** or **show system upgrade reg-info**. Verify that the output shows:

- `STATUS`: Valid
- `PACKAGE VERSION`: The correct upgrade version
- `UPLOAD DATE`: The registration timestamp

Step 4 Schedule the upgrade using **system upgrade apply-image**.

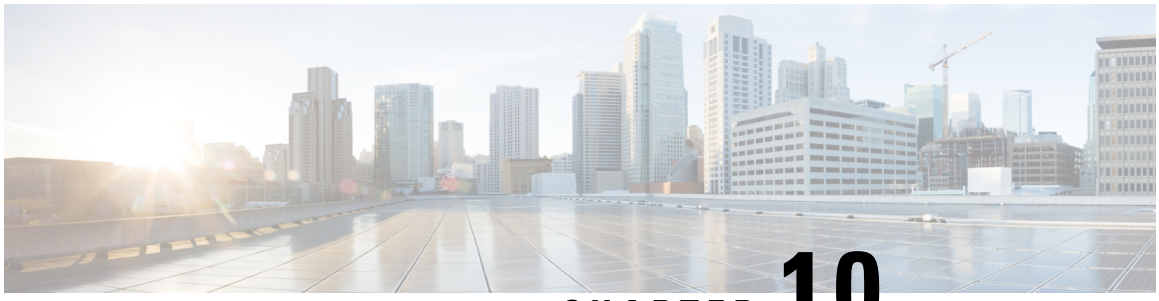
To upgrade now, set the schedule time to 0 hours. To upgrade later, specify the number of hours from now when the upgrade should begin.

After scheduling the upgrade, run **show system upgrade** and verify that the status is SCHEDULED.

Step 5 Verify the upgrade and check that the VMs are operational.

a) After the upgrade completes, run **show system upgrade apply-image** and verify that the status is success.
b) Verify that all VMs are powered on and operational.

NFVIS is upgraded to the selected version, and all VMs are powered on and operational.



CHAPTER 10

NFVIS serviceability, monitoring and observability

Objective

This reference groups information about NFVIS Serviceability, Monitoring and Observability, including Log Files, Host vs. Virtual Machine Monitoring, Cisco ThousandEyes, Troubleshooting NFVIS.

- Log files, on page 75
- Host vs. virtual machine monitoring, on page 77
- Cisco ThousandEyes, on page 133
- Troubleshooting NFVIS, on page 144

Log files

Objective

System logs are diagnostic files that

Log file types and configuration

- provide information for troubleshooting system issues
- consist of configuration logs and operational logs with different types of entries, and
- support configurable log levels to control the verbosity of logged information.

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 these types:

Table 12: System log levels

Log Level	Purpose
DEBUG	Information, typically of interest only when diagnosing problems.
INFO	Confirmation that things are working as expected.
WARNING	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.
ERROR	Due to a serious problem, the software application is not able to perform some function.
CRITICAL	A serious ERROR, indicating that the program itself may not be able to continue running.

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 periordically 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
```

System log configuration verification

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 13: System log APIs and commands

System Log APIs	System Log Commands
• /api/operations/system/set-log • /api/operational/system/logging-level	• **system set-log logtype** [all/configuration/operational] **level** [CRITICAL/DEBUG/ERROR/INFO/WARNING] • **show system logging-level**

Host vs. virtual machine monitoring

Objective

This reference groups information about Host vs. Virtual Machine Monitoring, including Monitoring, Syslog, NETConf, SNMP.

Monitoring

System monitoring is a capability that

- provides system monitoring commands and APIs to monitor the host and the VMs deployed on NFVIS
- collects statistics on CPU utilization, memory, disk and ports with metrics collected periodically and displayed for a specified duration, and
- enables the user to view historical data on the system's operation with metrics shown as graphs on the portal.

For larger durations average values are displayed.

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, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, 30D with min as minutes, h and H as hours, d and D as days.

Note When a pNIC is actively connected to a vNIC through SRIOV connection, the port usage metrics are displayed only for the last 5 minutes (last 30 values) irrespective of the time interval provided in the CLI to view the port usage.

Example

This 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.

Monitor 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.

CPU-state	Description
non-idle	100 – idle-CPU-percentage
interrupt	Indicates the percentage of the processor time spent in servicing interrupts
nice	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.
system	The system CPU state shows the amount of CPU time used by the kernel.
user	The user CPU state shows CPU time used by user space processes
wait	Idle time while waiting for an I/O operation to complete

The non-idle state is what the user usually needs to monitor. Use this 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 this CLI and API:

```
nfvis# show system-monitoring host cpu table cpu-usage duration
```

```
/api/operational/system-monitoring/host/cpu/table/cpu-usage/<duration>?deep
```

Monitor 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 collected 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 minutess for packets/sec and kilobits/sec values.

Monitor host memory

Statistics for the physical memory utilization are displayed for these categories:

Field	Description
buffered-MB	Memory used for buffering I/O
cached-MB	Memory used for caching file system access
free-MB	Memory available for use
used-MB	Memory in use by the system
SLAB-recl-MB	Memory used for SLAB-allocation of kernel objects, that can be reclaimed
SLAB-unrecl-MB	Memory used for SLAB-allocation of kernel objects, that cannot be reclaimed

Use this 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 this CLI and API:

```
nfvis# show system-monitoring host memory table mem-usage duration
```

```
/api/operational/system-monitoring/host/memory/table/mem-usage/<duration>?deep
```

Monitor host disks

Statistics for disk operations and disk space can be obtained for the list of disks and disk partitions on the NFVIS host.

Monitor host disks operations

These disk performance statistics are displayed for each disk and disk partition:

Field	Description
IO-time-ms	Average time spent doing I/O operations in milliseconds
IO-time-weighted-ms	Measure of both I/O completion time and the backlog that may be accumulating
merged-reads-per-sec	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.
merged-writes-per-sec	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.
bytes-read-per-sec	Bytes read per second
bytes-written-per-sec	Bytes written per second
reads-per-sec	Number of read operations per second
writes-per-sec	Number of write operations per second
time-per-read-ms	The average time a read operation takes to complete
time-per-write-ms	The average time a write operation takes to complete
pending-ops	The queue size of pending I/O operations

Use this 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
```

Monitor host disk space

This data related to file system usage, that is how much space on a mounted partition is used and how much is available is collected.

Field	Description
free-GB	Gigabytes available
used-GB	Gigabytes in use
reserved-GB	Gigabytes reserved for the root user

Use this 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
```

Monitor host ports

These statistics for network traffic and errors on interfaces are displayed:

Field	Description
name	Interface name
total-packets-per-sec	Total (received and transmitted) packet rate
rx-packets-per-sec	Packets received per second
tx-packets-per-sec	Packets transmitted per second
total-errors-per-sec	Total (received and transmitted) error rate
rx-errors-per-sec	Error rate for received packets
tx-errors-per-sec	Error rate for transmitted packets

Use this 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 this CLI and API:

```
nfvis# show system-monitoring host port table
```

```
/api/operational/system-monitoring/host/port/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 these fields:

Field	Description
total-percentage	Average CPU utilization across all the logical CPUs used by the VM
ID	Logical CPU ID
vcpu-percentage	CPU utilization percentage for the specified logical CPU ID

Use this 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

These statistics are collected for VNF memory utilization:

Field	Description
total-MB	Total memory of the VNF in MB
RSS-MB	Resident Set Size (RSS) of the VNF in MB 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 this 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

These disk performance statistics are collected for each disk used by the VM:

Field	Description
bytes-read-per-sec	Bytes read from the disk per second
bytes-written-per-sec	Bytes written to the disk per second
reads-per-sec	Number of read operations per second

Field	Description
writes-per-sec	Number of write operations per second

Use this 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

These network interface statistics are collected for VMs deployed on NFVIS:

Field	Description
total-packets-per-sec	Total packets received and transmitted per second
rx-packets-per-sec	Packets received per second
tx-packets-per-sec	Packets transmitted per second
total-errors-per-sec	Total error rate for packet reception and transmission
rx-errors-per-sec	Error rate for receiving packets
tx-errors-per-sec	Error rate for transmitting packets

Use this 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
```

Syslog

Objective

Syslog messages provide NFVIS configuration and operational information and can be sent to remote servers for centralized collection.

Syslog message format and examples

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

To configure remote syslog settings, see Configure syslog.

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
```

Syslog event notifications include:

Table 14: Syslog events

Event	Trigger Condition	Syslog Messages
NETWORK_CREATE	create a new network	nfvis %SYS-6-NETWORK_CREATE: Network my-net created successfully
NETWORK_UPDATE	modify an existing network	nfvis %SYS-6-NETWORK_UPDATE: Network my-net updated successfully
NETWORK_DELETE	delete a network	nfvis %SYS-6-NETWORK_DELETE: Network my-net deleted successfully
BRIDGE_CREATE	create a new bridge	nfvis %SYS-6-BRIDGE_CREATE: Bridge created successfully: my-bridge
BRIDGE_UPDATE	modify an existing bridge	nfvis %SYS-6-BRIDGE_UPDATE: Updated bridge successfully: my-bridge
BRIDGE_DELETE	delete a bridge	nfvis %SYS-6-BRIDGE_DELETE: Bridge deleted successfully: my-bridge
WAN_DHCP_RENEW	dhcp renew from wan interface	nfvis %SYS-6-WAN_DHCP_RENEW: wan-br DHCP IP address is being renewed

Event	Trigger Condition	Syslog Messages
BRIDGE_DHCP_RENEW	bridge dhcp renew	nfvis %SYS-6-BRIDGE_DHCP_RENEW: Bridge DHCP IP address is being renewed
MGMT_DHCP_RENEW	dhcp renew from MGMT interface	nfvis %SYS-6-MGMT_DHCP_RENEW: wan-br DHCP IP address is being renewed
INTF_STATUS_CHANGE	interface status change	nfvis %SYS-6-INTF_STATUS_CHANGE: Interface eth0, changed state to up
UPGRADE_REGISTER	upgrade package registration	nfvis %SYS-6-UPGRADE_REGISTER: Upgrade package registration successful: Cisco_NFVIS_Upgrade-3.6.1-6.9.8-2017.04.02_04.28.11.nfvispkg
UPGRADE_APPLY	upgrade process	nfvis %SYS-6-UPGRADE_APPLY: Upgrade Process: In Progress
RBAC_USER_CREATE	create a new user	nfvis %SYS-6-RBAC_USER_CREATE: Created user admin as administrators successfully
RBAC_USER_PASSWORD_UPDATE	change user's password	nfvis %SYS-6-RBAC_USER_PASSWORD_UPDATE: Set admin password successfully
RBAC_USER_ROLE_UPDATE	change user's role	nfvis %SYS-6-RBAC_USER_ROLE_UPDATE: Modified user: somebody successfully
RBAC_USER_DELETE	delete a user	nfvis %SYS-6-RBAC_USER_DELETE: Deleted rbac user successfully: somebody
RBAC_USERS_INACTIVATED	disable the user	nfvis %SYS-6-RBAC_USERS_INACTIVATED: Following users have been marked as INACTIVE, [user1, user2]. Please take necessary action.

Event	Trigger Condition	Syslog Messages
RBAC_USER_ACTIVATED	activate the user	nfvis %SYS-6-RBAC_USER_ACTIVATED: Modified user user1 successfully.
RBAC_PWD_EXPIRED	password expired	nfvis %SYS-6-RBAC_PWD_EXPIRED: User user1's password is older than 60 days. Please reset password.
RBAC_LOGIN_FAILURE	invalid user login	nfvis %SYS-3-RBAC_LOGIN_FAILURE: Login with invalid username from maapi failed
SECURITY_SERVER_CREATE	create server config	nfvis %SYS-6-SECURITY_SERVER_CREATE: TACACS+ server config created successfully.
SECURITY_SERVER_UPDATE	update server config	nfvis %SYS-6-SECURITY_SERVER_UPDATE: TACACS+ server configuration updated successfully.
SECURITY_SERVER_DELETE	delete server config	nfvis %SYS-6-SECURITY_SERVER_DELETE: TACACS+ server deleted successfully.
AAA_TYPE_CREATE	create AAA authentication type	nfvis %SYS-6-AAA_TYPE_CREATE: AAA authentication type TACACS created successfully.
AAA_TYPE_UPDATE	update AAA authentication type	nfvis %SYS-6-AAA_TYPE_UPDATE: AAA authentication type TACACS+ updated successfully. **AAA authentication** updated to use TACACS+ server
RECREATE_CERTIFICATE	recreate self-sign certificate	nfvis %SYS-6-RECREATE_CERTIFICATE: Self Signed Certificate re-created. Application connection may become temporarily unavailable.
CERT_CSR_CREATE	create a CSR file	nfvis %SYS-6-CERT_CSR_CREATE: signing-request created /data/intdatastore/download/nfvis.csr

Event	Trigger Condition	Syslog Messages
CERT_SWITCH_CERT	switch to use different certificate	nfvis %SYS-6-CERT_SWITCH_CERT: switch certificate from ca-signed to self-signed.
CERT_CA_CERT_INSTALL	install CA signed certificate	nfvis %SYS-6-CERT_CA_CERT_INSTALL: ca-signed certificate file:// installed
REBOOT	system reboot	nfvis %SYS-6-REBOOT: System will be rebooted
SHUTDOWN	system shutdown	nfvis %SYS-6-SHUTDOWN: System will be shutdown
LOGGING_FAILURE	logging failure	nfvis %SYS-6-LOGGING_FAILURE: Unable to write to log file nfvis_config.log. Log message: log_config.CONFIG_LOGGER: File not found.
DISK_SPACE_ALMOST_FULL	disk space almost full	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%.
ROTATED_LOGS_DELETE	delete rotated logfiles when accumulated rotated log files reach 2GB	nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days
TIME_UPDATE	Change system time manually	nfvis %SYS-6-TIME_UPDATE: Manual time updated successfully Manual time is now set to 2018-04-26 11:43:00
TIMEZONE_UPDATE	Change system timezone	nfvis %SYS-6-TIMEZONE_UPDATE: Timezone updated successfully. Timezone is now set to US/Eastern
FILE_COPY_STATUS	copy status of file	nfvis %SYS-6-FILE_COPY_STATUS: hostaction.py Copied Successfully.
CREATE_IMAGE	create image	nfvis %SYS-6-CREATE_IMAGE: Image creation successful: TinyLinux.tar.gz

Event	Trigger Condition	Syslog Messages
DELETE_IMAGE	delete image	nfvis %SYS-6-DELETE_IMAGE: Image deletion successful: TinyLinux.tar.gz
CREATE_FLAVOR	create flavor	nfvis %SYS-6-CREATE_FLAVOR: Profile creation successful: small
DELETE_FLAVOR	delete flavor	nfvis %SYS-6-DELETE_FLAVOR: Profile deletion successful: small
VM_DEPLOYED	vm deployment	nfvis %SYS-6-VM_DEPLOYED: VM deployment successful:
VM_ALIVE	vm alive	nfvis %SYS-6-VM_ALIVE: VM active successful:
SERVICE_ALIVE	service alive	nfvis %SYS-6-SERVICE_ALIVE: Service group deployment completed successfully!
VM_UNDEPLOYED	vm undeployed	nfvis %SYS-6-VM_UNDEPLOYED: VM undeployment successful: SERVICE_UNDEPLOYED service undeployed nfvis %SYS-6-SERVICE_UNDEPLOYED: Service group undeployment completed successfully
VM_UPDATED (update flavor)	vm updated	nfvis %SYS-6-VM_UPDATED: VM update successful: VM is resized with flavor [ISRv-medium].
VM_UPDATED (vnic add / delete / update)	vm updated	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]
SERVICE_UPDATED	service updated	nfvis %SYS-6-SERVICE_UPDATED: Service group update completed successfully

Event	Trigger Condition	Syslog Messages
VM_STOPPED	vm stopped	nfvis %SYS-6-VM_STOPPED: VM stop successful: SystemAdmin_ROUTER_[illegible]
VM_STARTED	vm started	nfvis %SYS-6-VM_STARTED: VM start successful: SystemAdmin_ROUTER_[illegible]
VM_REBOOTED	vm rebooted	nfvis %SYS-6-VM_REBOOTED: VM reboot successful: SystemAdmin_ROUTER_[illegible]
VM_RECOVERY_INIT	vm recovery initiation	nfvis %SYS-6-VM_RECOVERY_INIT: VM recovery initiation successful: SystemAdmin_ROUTER_[illegible]
VM_RECOVERY_REBOOT	vm recovery reboot	nfvis %SYS-6-VM_RECOVERY_REBOOT: VM recovery reboot successful: SystemAdmin_ROUTER_[illegible]
VM_RECOVERY_COMPLETE	vm recovery complete	nfvis %SYS-6-VM_RECOVERY_COMPLETE: VM recovery successful: SystemAdmin_ROUTER_[illegible]
VM_MONITOR_UNSET	vm monitoring unset	nfvis %SYS-6-VM_MONITOR_UNSET: Unsetting VM monitoring successful: SystemAdmin_ROUTER_[illegible]
VM_MONITOR_SET	vm monitoring set	nfvis %SYS-6-VM_MONITOR_SET: Setting VM monitoring successful: SystemAdmin_ROUTER_[illegible]
ROTATED_LOGS_DELETE (When logs older than 30 days are present)	delete rotated logs	nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days
ROTATED_LOGS_DELETE (When Log file size exceed 2GB, older logs are deleted)	delete rotated logs	nfvis %SYS-6-ROTATED_LOGS_DELETE: Rotated logs had exceeded 2G, older logs have been deleted to make space

Event	Trigger Condition	Syslog Messages
OVS_DPDK_SUCCESS	Enable DPDK	nfvis %SYS-3-OVS_DPDK_SUCCESS: OVS-DPDK enabled
OVS_DPDK_FAILURE	DPDK failure	nfvis %SYS-3-OVS_DPDK_FAILURE: Unable to allocate CP
BACKUP_INIT	Backup configuration initiation	nfvis %SYS-6-BACKUP_INIT: Starting backup: configuration-xxx
BACKUP_SUCCESS	Backup configuration successful	nfvis %SYS-6-BACKUP_SUCCESS: Backup configuration-xxx completed successfully
BACKUP_FAILURE	Backup configuration failure	nfvis %SYS-3-BACKUP_FAILURE: Backup configuration-xxx failed
RESTORE_INIT	Restore initiation	nfvis %SYS-6-RESTORE_INIT: Restore started
RESTORE_ SUCCESS	Successful restore	nfvis %SYS-6-RESTORE_ SUCCESS: Restore successful
RESTORE_FAILURE	Failure to restore	nfvis %SYS-3-RESTORE_FAILURE: Restore failed - internal error

Configure syslog

Configure syslog settings to send NFVIS event notifications to remote servers for centralized log and event collection.

For information about syslog messages and event types, see Syslog.

Procedure

Step 1 Configure a remote syslog server.

Specify the server IP address or DNS name, transport protocol, and port number. For example:

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

Note You can configure a maximum of four remote syslog servers. Specify a remote syslog server by its IP address or DNS name. By default, NFVIS uses UDP on port 514. For TCP, the default port is 601.

Step 2 Configure the syslog severity.

Severity describes the importance of a syslog message. Use this command syntax:

```
configure terminal
system settings logging severity severity-level
```

Table 15: Syslog severity levels

Severity Level	Description	Numeric Encoding for Severity in the Syslog Message Format
debug	Debug-level messages	7
informational	Informational messages	6
notice	Normal but significant condition	5
warning	Warning conditions	4
error	Error conditions	3
critical	Critical conditions	2
alert	Take action immediately	1
emergency	System is unusable	0

Note By default, the logging severity is informational, so NFVIS logs syslogs with informational severity and higher. When you configure a severity, NFVIS logs syslogs at that severity and syslogs with higher severity.

Step 3 Configure the syslog facility.

The facility logically separates and stores syslog messages on the remote syslog server. For example, you can assign syslogs from one NFVIS device to local0 and syslogs from another device to local1 so that the server stores and processes them in different directories.

Example:

```
configure terminal
system settings logging facility local5
```

Note You can configure a facility from local0 through local7.

By default, NFVIS sends syslogs with the local7 facility.

NFVIS sends syslog messages to the configured remote syslog servers for centralized log and event collection.

NETConf

A NETCONF event notification is a monitoring mechanism that

- enables Cisco NFVIS to generate alerts for key events
- allows NETCONF clients to subscribe and monitor configuration activation progress, and
- provides status change information for the system and VMs.

Event notification types

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 these 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 refer below

nfvisEvent

Table 16: NFVIS event notification output examples

Event Type	Notification Trigger	Notification Output Example
WAN_DHCP_RENEW	DHCP renew operation is performed.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T18:06:46.142089+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Wan DHCP IP address is being renewed</status_message> <details>NA</details> <event_type>WAN_DHCP_RENEW</event_type> </nfvisEvent> </notification>
BRIDGE_DHCP_RENEW	Bridge DHCP renew operation is performed.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2018-04-26T09:47:06.066264+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Bridge DHCP IP address is being renewed</status_message> <details>NA</details> <event_type>BRIDGE_DHCP_RENEW</event_type> </nfvisEvent> </notification>
INTF_STATUS_CHANGE	Interface status is changed.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T18:12:09.963556+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <event_type>INTF_STATUS_CHANGE</event_type> <intf_name>eth7</intf_name> <intf_prv_op>up</intf_prv_op> <intf_op>down</intf_op> <intf_prv_link>down</intf_prv_link> <intf_link>down</intf_link> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
NETWORK_CREATE	A network is created.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-09-22T12:41:04.564298+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_message>Network created succesfully</status_message> <event_type>NETWORK_CREATE</event_type> <network_name>testn1</network_name> <network_bridge>test-net-br</network_bridge> <network_sriov>false</network_sriov> <network_vlan/> <network_trunk/> </nfvisEvent> </notification>
NETWORK_UPDATE	A network is updated.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-09-22T12:42:03.391986+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_message>Network updated succesfully</status_message> <event_type>NETWORK_UPDATE</event_type> <network_name>testn1</network_name> <network_bridge/> <network_sriov/> <network_vlan/> <network_trunk/> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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 succesfully</status_message> <event_type>NETWORK_DELETE</event_type> <network_name>testn1</network_name> <network_bridge/> <network_sriov/> <network_vlan/> <network_trunk/> </nfvisEvent> </notification>
UPGRADE_REGISTER	System upgrade is registered.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T15:57:50.434636+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Upgrade package registration successful: Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg</status_message> <event_type>UPGRADE_REGISTER</event_type> </nfvisEvent> </notification>
UPGRADE_APPLY	System upgrade is applied.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T16:02:43.885516+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Upgrade Process: In Progress</status_message> <event_type>UPGRADE_APPLY</event_type> </nfvisEvent> </notification>

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

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

Event Type	Notification Trigger	Notification Output Example
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>

Event Type	Notification Trigger	Notification Output Example
PNIC_SRIOV_ENABLE	Enable of disable SR-IOV 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 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_UPDATE	Set switchmode to veb/vepa	<?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>

Event Type	Notification Trigger	Notification Output Example
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 <eventTime>2020-03-27T19:58:52.333682+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>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>

Event Type	Notification Trigger	Notification Output Example
[illegible]	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>

Event Type	Notification Trigger	Notification Output Example
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>

Event Type	Notification Trigger	Notification Output Example
[illegible]		<?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 unconfigured and a manual reboot is required to disable it</status_message> <details>NA</details> <event_type>FIPS_UNCONFIGURED_REBOOT_TO_DISABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

vmlcEvent

Table 17: VM life cycle event notification output examples

Event Type	Notification Trigger	Notification Output Example
CREATE_IMAGE	The VM image is registered.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:12:30.76+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Image creation completed successfully.</status_message> <image>isrv-universalk9.16.03.01.tar.gz</image> <vm_source/> <vm_target/> <event> <type>CREATE_IMAGE</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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_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_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_target/> <event> <type>DELETE_FLAVOR</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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: [SystemAdmin-Tena_ROUTER_0_df6733c1-0768-4ae6-8dbe-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/> <event> <type>VM_DEPLOYED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_ALIVE	The state of a monitored VM becomes ACTIVE.	<?xml 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: [SystemAdmin_Tena_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/> <event> <type>VM_ALIVE</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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> <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/> <event> <type>VM_UNDEPLOYED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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_target/> <event> <type>SERVICE_UPDATED</type> </event> </vmlcEvent> </notification>
VM_STOPPED	The VM is stopped per VM action requrest.	<?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 [SystemAdmin_Tena_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/> <event> <type>VM_STOPPED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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 [SystemAdmin_Tena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223eedb036c].</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/> <event> <type>VM_STARTED</type> </event> </vmlcEvent> </notification>
VM_REBOOTED	The VM is rebooted per VM action request.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:36:56.5+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Rebooted VM [SystemAdmin_Tena_ROUTER_0_f17fc494-8535-4b05-b88d-f0fd2effdc7d]</status_message> <depname>1479342258</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>827e871a-30d5-4f5f-a05a-263b7ee3a734</depid> <vm_group>ROUTER</vm_group> <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> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
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 [SystemAdminTena_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] trigge red. Processing Auto healing. Proceeding with Recovery.</status_message> <depname>1479328919</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_RECOVERY_INIT</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_RECOVERY_REBOOT	Recovery reboot starts for the monitored VM, which is not reachable.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T16:27:53.979+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VM [SystemAdminTena_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] is being rebooted. </statu s_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></vm_target> <event> <type>VM_RECOVERY_REBOOT</type> </event> </vmlcEvent> </notification>

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

Event Type	Notification Trigger	Notification Output Example
VM_MONITOR_UNSET	Monitoring is disabled per VM action request.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-18T13:36:43.613+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Unset monitor completed successfully</status_message> <depname>1479413090</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>742dd335-330c-4bf0-a75d-a44003c645c5</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_MONITOR_UNSET</type> </event> </vmlcEvent> </notification>
VM_MONITOR_SET	Monitoring is enabled per VM action request.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-18T13:40:15.276+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Set monitor completed successfully</status_message> <depname>1479413090</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>742dd335-330c-4bf0-a75d-a44003c645c5</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>23ec3793-37ab-4ec2-a978-a10e08585fdd</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_MONITOR_SET</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_UPDATED	VM's flavor is changed.	

Event Type	Notification Trigger	Notification Output Example
		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-12-08T00:50:39.034+00:00</eventTime> <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VM is resized with flavor [ISRv-medium].</status_message> <user_name>admin</user_name> <depname>1512766000</depname> <tenant>admin</tenant> <tenant_id>adminUUID</tenant_id> <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid> <vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname> <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>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:70:06:4a</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> <interface> <nicid>2</nicid> <type>virtual</type> <port_id>vnic2</port_id> <network>lan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:c7:30:1c</mac_address> <netmask>255.255.255.0</netmask> <gateway>192.168.1.1</gateway> </interface> </interfaces> </vm_source> <event> <type>VM_UPDATED</type> </event> </vmlcEvent></notification>]]>]]>

Event Type	Notification Trigger	Notification Output Example
		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-12-08T00:50:39.06+00:00</eventTime> <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Service group update completed successfully</status_message> <user_name>admin</user_name> <depname>1512766000</depname> <tenant>admin</tenant> <tenant_id>adminUUID</tenant_id> <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid> <event> <type>SERVICE_UPDATED</type> </event> </vmlcEvent>

Event Type	Notification Trigger	Notification Output Example
VM_UPDATED	VNIC is added, deleted or updated.	

Event Type	Notification Trigger	Notification Output Example
		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-12-08T02:10:56.184+00:00</eventTime> <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Added 1 interface: [managed, net=my-net-1, nicid=3] Updated 2 interface: [managed, net=lan-net, nicid=1],[managed, net=wan-net, nicid=2]</status_message> <user_name>admin</user_name> <depname>1512766000</depname> <tenant>admin</tenant> <tenant_id>adminUUID</tenant_id> <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid> <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:c7:30:1c</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> <interface> <nicid>3</nicid> <type>virtual</type>

Event Type	Notification Trigger	Notification Output Example
		<port_id>vnic3</port_id> <network>my-net-1</network> <subnet>N/A</subnet> <mac_address>52:54:00:66:b5:c1</mac_address> </interface> </interfaces> </vm_source> <event> <type>VM_UPDATED</type> </event> </vmlcEvent>

SNMP

Simple Network Management Protocol (SNMP) is an application-layer protocol that

- provides a message format for communication between SNMP managers and agents, and
- 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 operations are network management procedures that

- retrieve data from SNMP object variables using GET operations
- modify SNMP object variable values using SET operations, and
- send unsolicited notifications from SNMP agents to management servers.

SNMP operation types

SNMP applications perform three types of 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

Objective

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

SNMP GET command syntax

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

The command for SNMP GET is:

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

SNMP walk

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

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

The command for SNMP walk with SNMP v2 is:

snmpwalk -v2c -c [*community-name*] [*NFVIS-box-ip*]

Show more

```
snmpwalk -v2c -c myUser 172.19.147.115 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Cisco NFVIS
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.12.3.1.3.1291
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (43545580) 5 days, 0:57:35.80
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 70
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
```

```
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
IF-MIB::ifIndex.11 = INTEGER: 11
IF-MIB::ifDescr.1 = STRING: GE0-0
IF-MIB::ifDescr.2 = STRING: GE0-1
IF-MIB::ifDescr.3 = STRING: MGMT
IF-MIB::ifDescr.4 = STRING: gigabitEthernet1/0
IF-MIB::ifDescr.5 = STRING: gigabitEthernet1/1
IF-MIB::ifDescr.6 = STRING: gigabitEthernet1/2
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-2.47.1.1.1.1.2.0 = STRING: "Cisco NFVIS"
SNMPv2-SMI::mib-2.47.1.1.1.1.3.0 = OID: SNMPv2-SMI::enterprises.9.1.1836
SNMPv2-SMI::mib-2.47.1.1.1.1.4.0 = INTEGER: 0
SNMPv2-SMI::mib-2.47.1.1.1.1.5.0 = INTEGER: 3
SNMPv2-SMI::mib-2.47.1.1.1.1.6.0 = INTEGER: -1
SNMPv2-SMI::mib-2.47.1.1.1.1.7.0 = STRING: "ENCS5412/K9"
SNMPv2-SMI::mib-2.47.1.1.1.1.8.0 = STRING: "M3"
SNMPv2-SMI::mib-2.47.1.1.1.1.9.0 = ""
SNMPv2-SMI::mib-2.47.1.1.1.1.10.0 = STRING: "3.7.0-817"
SNMPv2-SMI::mib-2.47.1.1.1.1.11.0 = STRING: "FGL203012P2"
SNMPv2-SMI::mib-2.47.1.1.1.1.12.0 = STRING: "Cisco Systems, Inc."
SNMPv2-SMI::mib-2.47.1.1.1.1.13.0 = ""
...
```

This 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
DISMAN-EVENT-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

Objective

SNMP notifications are asynchronous messages that

SNMP notification behavior

- generate from an SNMP agent without requiring requests from the SNMP manager
- can be generated as traps or inform requests to alert managers about network conditions, and
- indicate events such as improper user authentication, restarts, connection closures, loss of connection to neighbor routers, or other significant events.

Notification types

Objective

SNMP notifications include these types:

SNMP notification types

- Traps: Messages alerting the SNMP manager to a condition on the network.
- Inform requests (informs): Traps that include a request for confirmation of receipt from the SNMP manager.

SNMP versions

Objective

SNMP versions are network management protocol variants that

- provide different levels of security and functionality for managing network devices
- support community-based or user-based security models, and
- enable monitoring and configuration of network infrastructure components.

Supported SNMP versions

Cisco NFVIS supports these 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

Objective

This reference provides the MIBs that are supported for SNMP on NFVIS.

Supported SNMP MIBs

CISCO-MIB

CISCO-MIB OID 1.3.6.1.4.1.9.2.1.3. hostname

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)

CISCO environmental monitor MIB notification

- ciscoEnvMonEnableShutdownNotification
- ciscoEnvMonEnableVoltageNotification
- ciscoEnvMonEnableTemperatureNotification
- ciscoEnvMonEnableFanNotification
- ciscoEnvMonEnableRedundantSupplyNotification
- ciscoEnvMonEnableStatChangeNotif

VM-MIB (1.3.6.1.2.1.236)

- vmHypervisor:
- vmHvSoftware
- vmHvVersion
- vmHvUpTime
- vmTable:
- vmName
- vmUUID
- vmOperState
- vmOSType
- vmCurCpuNumber

- vmMemUnit
- vmCurMem
- vmCpuTime
- vmCpuTable:
- vmCpuCoreTime
- vmCpuAffinityTable
- vmCpuAffinity

Configure SNMP support

Use these steps to configure SNMP to enable network monitoring and management through SNMP protocols. This configuration allows administrators to monitor device status, collect statistics, and receive trap notifications.

For SNMP v1 and v2c, use the same community and user name. Use the same SNMP version for a user and group.

For complete configuration examples, verification commands and output, configuration limits, APIs, and security-mode behavior, see SNMP configuration reference.

Procedure

Step 1 Create an SNMP community.

Example:

```
configure terminal
snmp community community_name community-access access
```

The community name supports letters, numbers, underscores, and hyphens, and can contain a maximum of 32 characters. NFVIS supports only **readOnly** access.

Step 2 Create an SNMP group.

Example:

```
configure terminal
snmp group group_name context version security_level notify notify_list read read_list write
write_list
```

Table 18: SNMP group command variables

Variable	Description
group_name	Group name string. It supports letters, numbers, underscores, and hyphens, and can contain a maximum of 32 characters.
context	Context string. The default is SNMP. The maximum length is 32 characters. The minimum length is 0 (empty context).
version	1, 2, or 3 for SNMP v1, v2c, and v3.

Variable	Description
security_level	authPriv, authNoPriv, or noAuthNoPriv. **Note** SNMP v1 and v2c use noAuthNoPriv only.
notify_list, read_list, write_list	Each value can be any string. read_list and notify_list are required to support data retrieval by SNMP tools. You can omit write_list because NFVIS SNMP does not support SNMP write access.

Step 3 Create an SNMP v3 user for the selected security level.

Enter **configure terminal**, then use the command form that matches the security level:

- For authPriv without a separate encryption passphrase:

  snmp user *user_name* **user-version 3 user-group** *group_name* **auth-protocol** *auth* **priv-protocol** *priv* **passphrase** *passphrase_string*

- For authPriv with a separate encryption passphrase:

  snmp user *user_name* **user-version 3 user-group** *group_name* **auth-protocol** *auth* **priv-protocol** *priv* **passphrase** *passphrase_string* **encryption-passphrase** *encryption_passphrase*

- For authNoPriv:

  snmp user *user_name* **user-version 3 user-group** *group_name* **auth-protocol** *auth* **passphrase** *passphrase_string*

- For noAuthNoPriv:

  snmp user *user_name* **user-version 3 user-group** *group_name*

Table 19: SNMP user command variables

Variable	Description
user_name	User name string. It supports letters, numbers, underscores, and hyphens, and can contain a maximum of 32 characters. For SNMP v1 and v2c, the user name must match the community name.
version	1 and 2 for SNMP v1 and v2c. The SNMP v3 commands in this task use version 3.
group_name	Name of the group configured in NFVIS.
auth	md5 or sha.
priv	aes or des.
passphrase_string	Passphrase string. Supported characters are letters, numbers, and \-_#@%$*&! and space.
encryption_passphrase	Passphrase string. Supported characters are letters, numbers, and \-_#@%$*&! and space. Configure passphrase_string before you configure encryption_passphrase.

Note Do not use auth-key and priv-key. The auth and priv passphrases are encrypted after configuration and saved in NFVIS.

Step 4 Enable SNMP traps.

Example:

```
configure terminal
snmp enable traps trap_event
```

trap_event can be linkup or linkdown.

Step 5 Create an SNMP trap host.

Example:

```
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 security_level
```

Use noAuthNoPriv for SNMP v1 and v2c. For SNMP v3, use the security level configured for the user and group.

Table 20: SNMP trap host command variables

Variable	Description
host_name	User name string. It supports letters, numbers, underscores, and hyphens, and can contain a maximum of 32 characters. This is an alias for the trap server IP address, not an FQDN host name.
ip_address	IP address of the trap server.
port	Default is 162. Change the port number based on your setup.
user_name	User name configured in NFVIS.
version	1, 2, or 3 for SNMP v1, v2c, or v3.
security_level	authPriv, authNoPriv, or noAuthNoPriv. SNMP v1 and v2c use noAuthNoPriv only.

SNMP communities, groups, users, traps, and hosts are configured. The system can now support SNMP monitoring, statistics collection, and trap notifications based on the configured parameters.

SNMP configuration reference

Objective

Use this reference for SNMP configuration examples, verification commands and output, configuration limits, supported APIs, and security mode requirements.

SNMP v3 configuration

This reference provides standard configuration syntax for SNMP versions 1, 2, and 3 on NFVIS systems. These examples show how to configure SNMP monitoring and trap notifications.

Use this syntax to configure an SNMP v3 group and user with authentication and privacy protocols.

```
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 encryption-passphrase
encryptPassphrase
! Configure the SNMP host to enable SNMP v3 traps
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
 host-security-level authPriv host-port 162
!
```

SNMP v1 and v2 configuration syntax

Use this syntax to configure SNMP v1 and v2 community strings and trap enablement.

```
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
```

SNMP v3 configuration with MD5 authentication

Use this syntax to update an existing SNMP v3 configuration to use MD5 authentication.

```
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 md5
priv-protocol aes passphrase changePassphrase
! Configure the SNMP host to enable SNMP v3 traps
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
 host-security-level authPriv host-port 162
!
```

SNMP security-level modification

Use this syntax to update an existing SNMP v3 configuration to use MD5 authentication and enable traps.

```
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
! Configure the SNMP host to enable SNMP v3 traps
snmp host host4 host-ip-address 4.4.4.4 host-version 3 host-user-name user4
 host-security-level authNoPriv host-port 162
!
snmp enable traps linkUp
snmp enable traps linkDown
```

SNMP context configuration

Use these examples to configure specific SNMP contexts.

```
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
!
```

Empty context and noAuthNoPriv configuration

Use this syntax to configure an SNMP v3 group and user using an empty context string.

```
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.

SNMP agent status

This reference provides command syntax for verifying SNMP operations, configuration limits, API endpoints, and security compliance requirements for NFVIS.

Use the **show snmp agent** command to verify the SNMP agent description and ID:

```
nfvis# show snmp agent
snmp agent sysDescr "Cisco NFVIS "
snmp agent sysOID 1.3.6.1.4.1.9.12.3.1.3.1291
```

SNMP trap state

Use the **show snmp traps** command to verify the state of SNMP traps:

```
nfvis# show snmp traps

TRAP      TRAP
NAME      STATE
--------------------
linkDown  disabled
linkUp    enabled
```

SNMP statistics

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
```

SNMP running configuration

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
!
```

SNMP configuration limits

Upper limit for SNMP configurations:

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

SNMP APIs and commands

Table 21: SNMP support APIs and commands

API	Command
/api/config/snmp/agent	**agent**
/api/config/snmp/communities	**community**
/api/config/snmp/enable/traps	**trap-type**
/api/config/snmp/hosts	**host**
/api/config/snmp/user	**user**
/api/config/snmp/groups	**group**

SNMP security mode behavior

In Cisco NFVIS Release 26.2.1, insecure options will no longer be enabled by default and will require explicit user action. Enabling such options will be done through a dedicated command (for example, **system mode insecure**).

A secure configuration enforces standard security controls and safeguards to protect the system from unauthorized access and potential threats. An insecure configuration relaxes or bypasses these protections, which may be required for specific use cases but increases the risk to the system.

When this command is invoked, users will be presented with a clear warning outlining the security implications. They must explicitly confirm their intent to proceed, acknowledging the risks associated with enabling insecure configurations.

NFVIS generates warning messages for SNMP v1, SNMP v2, and SNMP v3 configurations that do not use authentication.

Table 22: SNMP security settings by mode

Sl. No	SNMP Type	SNMP Version	Security Protocol	System Mode
1	SNMP HOST	V3	noAuthNoPriv	insecure
1	SNMP GROUP	V3	noAuthNoPriv	insecure
1	SNMP USER	V3	auth-protocol: md5,sha; priv-protocol: des	insecure
2	SNMP HOST	V3	authNoPriv	insecure
2	SNMP GROUP	V3	authNoPriv	insecure
2	SNMP USER	V3	auth-protocol: md5,sha; priv-protocol: des	insecure
3	SNMP HOST	V3	authPriv	secure
3	SNMP GROUP	V3	authPriv	secure
3	SNMP USER	V3	auth-protocol: sha256; priv-protocol: aes	secure
4	All type SNMP	V1	All type	insecure
5	All type SNMP	V2	All type	insecure

Cisco ThousandEyes

Objective

Use this reference to understand the capabilities and benefits of Cisco NFVIS ThousandEyes support for network monitoring and performance visibility.

ThousandEyes support in NFVIS

Cisco NFVIS ThousandEyes support is a pre-integrated solution that

- allows deploying ThousandEyes Enterprise Agents on Cisco NFVIS
- enables running ThousandEyes network monitoring and testing capabilities directly on your Cisco NFVIS infrastructure, and
- provides visibility into the performance of the underlying network infrastructure.

Cisco NFVIS also supports ThousandEyes Docker Upgrade using existing VM lifecycle workflow. The VM lifecycle allows seamless image updates on the existing ThousandEyes deployment. Additionally, ThousandEyes has released NFVIS specific Docker images with install type as NFVIS. You must specify the nfvis.docker ThousandEyes image download URL in the update deployment request. NFVIS will then download the image and complete the upgrade of to the specified image.

This support provides these benefits:

- Gain end-to-end visibility into the performance of your network infrastructure, including cloud providers, WAN links, and internal data center networks within Cisco NFVIS providing end-to-end visibility into network performance.
- Reduce mean time to resolution (MTTR) for network issues, improve network reliability, and optimize application performance.

A ThousandEyes deployment on Cisco NFVIS is a network monitoring implementation that

- can be installed using the NFVIS web interface,
- supports command-line installation options, and
- provides network monitoring capabilities on the NFVIS platform.

Prerequisites

Objective

Ensure that your Cisco NFVIS environment meets all prerequisite requirements before enabling ThousandEyes support.

ThousandEyes deployment requirements

- Ensure that the minimum software version for Cisco NFVIS devices is Cisco NFVIS Release 4.18.2a.
- Ensure that your devices are meeting the minimum hardware requirements. For more information on the minimum hardware requirements see, Enterprise Agent System Requirements in the ThousandEyes documentation.
- Configure nameservers on your Cisco NFVIS devices to enable the download of the ThousandEyes Docker image from downloads.thousandeyes.com.

Deploy ThousandEyes container from NFVIS web interface

Use this task to deploy or upgrade a ThousandEyes container from the Cisco NFVIS web interface.

Before deploying, download the latest ThousandEyes container image from the **Add New Enterprise Agent** dialog, then upload and register the image on Cisco NFVIS. For details, see Uploading VM images to an NFVIS server and Register a remote virtual machine image.

Procedure

Step 1 In Cisco NFVIS portal, click **Configuration** > **Deploy**.

Step 2 In the **Select VM or Container** menu, select **TE**.

The ThousandEyes node appears in the topology.

Step 3 Configure the fields in the **VM Details** window.

Table 23: VM Details fields

Field	Description
VM Name	Enter the container name.
Image	Choose the registered `thousandeyes.docker` image from the drop-down list, or click the plus button to add an image.
Profile	Choose the profile for the ThousandEyes agent from the drop-down list, or click the plus button to add a profile. For more information, see Enterprise Agent System Requirements for an agent without the browser-bot version.
Group Name	(Optional) Choose a group name from the drop-down list.
Deployment Disk	Choose a deployment disk from the drop-down list. **Note** Do not choose NFS Store as a deployment disk.
Add Bootstrap Config	Add bootstrap configuration data in the expanded **Add Bootstrap Config** pane. Select the **Is External File Path** check box if the bootstrap file path is from an external source.
Add Config Options	Add configuration in the expanded **Add Config Options** pane. Choose the **Config Option Name**, then enter the **Config Option Value** after the equals sign in the prepopulated value. For example, `TEAGENT_INET=4`.
Add Volume	The volume required to enable ThousandEyes on Cisco NFVIS is prepopulated with the minimum required size. You can add or remove volumes based on your requirements.
Remove Volume	You can remove volumes based on your requirements.
Exclude Disks from Export	Not supported by ThousandEyes.

Step 4 Click **Deploy**.

Step 5 Upgrade the ThousandEyes Agent container from the NFVIS web interface by using the **Upgrade** icon on the **Manage Deployments** page.

To upgrade a registered image:

a. In the Cisco NFVIS portal, select **Configuration** > **Virtual Machine** > **Manage**.

b. Click the **Upgrade** icon.

c. Choose an image from the drop-down list of previously registered images.

d. Click **Submit** to upgrade the existing configuration to the selected image.

To upgrade using a new image:

a. In the Cisco NFVIS portal, select **Configuration** > **Virtual Machine** > **Manage**.

b. Click the **Upgrade** icon.

c. Click **New Image**.

d. Choose a protocol.

Table 24: Image source fields

Source	Description
HTTPS	The server path is prefilled. Enter the image name.
Docker	The repository name is prefilled. Enter the Docker image tag.

e. Click **Submit** to register the new image.

The ThousandEyes container is deployed on Cisco NFVIS and is ready for monitoring network performance. If you performed an upgrade, the existing configuration is updated with the selected image.

Deploy ThousandEyes on NFVIS using CLI configs

Use these steps to deploy or upgrade ThousandEyes on NFVIS using CLI configuration examples, NETCONF payloads, or RESTCONF requests.

For software, hardware, and network prerequisites, see ThousandEyes prerequisites. The examples include image registration and container configuration.

ThousandEyes Enterprise Agents can be deployed on NFVIS platforms to provide network monitoring capabilities. This deployment can be accomplished through multiple methods depending on your operational preferences and automation requirements

Procedure

Step 1 Register the ThousandEyes Docker Image

Note The URL in the `src` field is provided by the **Add New Enterprise Agent** dialog.

Example:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.33.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.33.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
!
```

Step 2 Configure Resource Details as a Flavor

Example:

```
vm_lifecycle flavors flavor thousandeyes-flavor vcpus 2 memory_mb 1024 root_disk_mb
 20480
!
```

Step 3 Deploy the ThousandEyes Agent Container

Example:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name TE_DEMO
   locator vim_id container
   image thousandeyes-enterprise-agent-0.33.0
   flavor thousandeyes-flavor
   bootup_time -1
   config_data configuration bootstrap_config
    data "{ \"env_variables\" : { \"TEAGENT_ACCOUNT_TOKEN\" :
\"${TEAGENT_ACCOUNT_TOKEN}\", \"TEAGENT_INET\" : \"${TEAGENT_INET}\"} }"
    template_engine VELOCITY
    variable TEAGENT_ACCOUNT_TOKEN
     val [ 53rettywagbuouhw06hu65767rtyuyyui ]
    !
    variable TEAGENT_INET
     val [ 4 ]
   !
  !
 !
```

Step 4 Commit the configuration

Step 5 To upgrade the ThousandEyes Agent container through the CLI, choose one of the following options.

Option 1: Register and Update the Image

Register the New Docker Image:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.34.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
!
```

Update the Container Deployment Configuration to use the newly registered image:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name TE_DEMO
   locator vim_id container
   image thousandeyes-enterprise-agent-0.34.0
 !
```

Commit the configuration. NFVIS will handle downloading the image and upgrading the container.

Option 2: Direct Image URL Update - Skip the image registration step by directly providing the ThousandEyes agent container image URL. NFVIS will automatically download the image, register it with Docker, and upgrade the container to the specified image:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name TE_DEMO
   locator vim_id container
   image
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker

 !
```

Step 6 Configure proxy settings if your NFVIS is behind a proxy

For more information, see Configuring an Enterprise Agent to Use a Proxy Server for other ThousandEyes Agent environment variables.

Image Registration with Proxy:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.34.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
 properties property http_proxy
  value [ http://proxy.com ]
 !
 properties property https_proxy
  value [ http://proxy.com ]
 !
 properties property no_proxy
  value [ .cisco.com,10.1.1.1 ]
 !
!
```

Deployment Configuration with Proxy:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name TE_DEMO
   locator vim_id container
   image thousandeyes-enterprise-agent-0.33.0
   flavor thousandeyes-flavor1
   bootup_time -1
   config_data configuration bootstrap_config
    data "{ \"env_variables\" : { \"TEAGENT_ACCOUNT_TOKEN\" :
\"${TEAGENT_ACCOUNT_TOKEN}\", \"TEAGENT_INET\" : \"${TEAGENT_INET}\", \"http_proxy\"
```

```
 : \"${http_proxy}\", \"https_proxy\" : \"${https_proxy}\", \"no_proxy\" :
\"${no_proxy}\"} }"
   template_engine VELOCITY
   variable TEAGENT_ACCOUNT_TOKEN
    val [ your token goes here ]
   !
   variable TEAGENT_INET
    val [ 4 ]
   !
   variable http_proxy
    val [ http://proxy.com:80/ ]
   !
   variable https_proxy
    val [ http://proxy.com:80/ ]
   !
   variable no_proxy
    val [ .cisco.com,10.1.1.1 ]
   !
  !
 !
```

Step 7 Deploy ThousandEyes Container on NFVIS Using NETCONF

These NETCONF payloads are examples that can be sent to NFVIS via a NETCONF client to manage the lifecycle of the Cisco ThousandEyes Container



Note Ensure you merge these payloads into the existing configuration to avoid deleting other configurations. Refer to your NETCONF client documentation for merging procedures.

Image Registration

From Docker Hub:

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <images>
    <image>
      <name>te_latest</name>
      <src>docker://thousandeyes/enterprise-agent:latest-agent</src>
      <credentials>
        <username>${DOCKER_USERNAME_HERE}</username> <!--Credentials are optional-->
        <password>${DOCKER_PAT}</password>
      </credentials>
      <properties>
        <property>
          <name>http_proxy</name> <!--Proxies are optional and depend on your topology-->
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>https_proxy</name>
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>vnf_type</name>
          <value>THOUSANDEYES</value>
        </property>
```

```
      </properties>
      <locator>
        <vim_id>container</vim_id>
      </locator>
    </image>
  </images>
</vm_lifecycle>
```

From ThousandEyes Webserver:

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <images>
    <image>
      <name>te_latest</name>

<src>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker</src>

      <properties>
        <property>
          <name>http_proxy</name> <!--Proxies are optional and depend on your topology-->
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>https_proxy</name>
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>vnf_type</name>
          <value>THOUSANDEYES</value>
        </property>
      </properties>
      <locator>
        <vim_id>container</vim_id>
      </locator>
    </image>
  </images>
</vm_lifecycle>
```

Flavor Registration

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <flavors>
    <flavor>
      <name>TE</name>
      <vcpus>2</vcpus>
      <memory_mb>1024</memory_mb>
      <root_disk_mb>8192</root_disk_mb>
    </flavor>
  </flavors>
</vm_lifecycle>
```

Deployment

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>te_nfvis</name>
          <vm_group>
            <name>te_nfvis</name>
            <image>te_latest</image>
```

```
            <flavor>TE</flavor>
            <bootup_time>-1</bootup_time>
            <config_data>
              <configuration>
                <dst>bootstrap_config</dst>
                <data>{
    "env_variables" : {
        "TEAGENT_ACCOUNT_TOKEN" : "${TEAGENT_ACCOUNT_TOKEN}",
        "TEAGENT_INET" : "${TEAGENT_INET}",
        "http_proxy" : "${http_proxy}",
        "https_proxy" : "${https_proxy}",
        "no_proxy" : "${no_proxy}"
    }
}</data>
                 <variable>
                    <name>TEAGENT_INET</name>
                    <val>4</val>
                </variable>
                <variable>
                    <name>TEAGENT_ACCOUNT_TOKEN</name>
                    <val>53rettywagbuouhw06hu65767rtyuyyui</val>
                </variable>
                <variable>
                    <name>https_proxy</name>
                    <val>http://example.com:80/</val>
                </variable>
                <variable>
                    <name>http_proxy</name>
                    <val>http://example.com:80/</val>
                </variable>
                <variable>
                    <name>no_proxy</name>
                    <val>.example.com,1.2.3.4</val>
                </variable>
              </configuration>
            </config_data>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</vm_lifecycle>
```

ThousandEyes Container Deployment Upgrade on NFVIS Using NETCONF:

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>te_nfvis</name> <!--The deployment name and vm_group name MUST match the existing
deployment that you wish to upgrade-->
          <vm_group>
            <name>te_nfvis</name>

<image>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.35.0-nfvis.docker</image>
 <!--Only image tag needs to be updated with URL or docker tag pointing to the new TE image-->
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</vm_lifecycle>
```

Step 8 Deploy ThousandEyes Container on NFVIS Using RESTCONF

The following are sample RESTCONF payloads that can be sent to NFVIS using curl or an equivalent client to manage the lifecycle of the Cisco ThousandEyes Container:

Image Registration

From Docker Hub:

```
curl -k -v -u admin:Cisco123# -X POST
'https://172.29.91.33/restconf/data/vmlc:vm_lifecycle/images' \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>
        <name>thousandeyes-enterprise-agent-0.34.0</name>
        <src>docker://thousandeyes/enterprise-agent:0.34.0-agent</src>
    <locator>
        <vim_id>container</vim_id>
    </locator>
    <properties>
        <property>
            <name>vnf_type</name>
            <value>THOUSANDEYES</value>
        </property>
        <property>
            <name>https_proxy</name> <!--Proxies are optional and depend on your topology-->
            <value>http://proxy.com:80/</value>
        </property>
        <property>
            <name>no_proxy</name>
            <value>.cisco.com,10.1.1.1</value>
        </property>
    </properties>
</image>'
```

From ThousandEyes Webserver:

```
curl -k -v -u admin:password -X POST
'https://nfvis_host_ip/restconf/data/vmlc:vm_lifecycle/images' \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>
        <name>thousandeyes-enterprise-agent-0.34.0</name>

<src>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.33.0-nfvis.docker</src>

    <locator>
        <vim_id>container</vim_id>
    </locator>
    <properties>
        <property>
            <name>vnf_type</name>
            <value>THOUSANDEYES</value>
        </property>
        <property>
            <name>https_proxy</name> <!--Proxies are optional and depend on your topology-->
            <value>http://proxy.com:80/</value>
        </property>
        <property>
            <name>no_proxy</name>
            <value>.cisco.com,10.1.1.1</value>
        </property>
    </properties>
</image>'
```

Flavor Creation

```
curl -k -v -u admin:password -X POST
'https://nfvis_host_ip/restconf/data/vmlc:vm_lifecycle/flavors' \
--header 'Content-Type: application/yang-data+xml' \
--data '<flavor>
    <name>thousandeyes-flavor</name>
    <vcpus>2</vcpus>
    <memory_mb>1024</memory_mb>
    <root_disk_mb>20480</root_disk_mb>
</flavor>'
```

ThousandEyes Agent Container Deployment

```
curl -k -v -u admin:password -X POST
'https://<nfvis_host_ip>/restconf/data/vmlc:vm_lifecycle/tenants/tenant=admin/deployments'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<deployment>
         <name>TE_DEMO</name>
         <vm_group>
           <name>TE_DEMO</name>
           <vim_vm_name>TE_DEMO</vim_vm_name>
           <locator>
             <vim_id>container</vim_id>
           </locator>
           <image>thousandeyes-enterprise-agent-0.33.0</image>
           <flavor>thousandeyes-flavor</flavor>
           <bootup_time>-1</bootup_time>
           <config_data>
             <configuration>
             <dst>bootstrap_config</dst>
             <data>{ "env_variables" : { "TEAGENT_ACCOUNT_TOKEN" : "${TEAGENT_ACCOUNT_TOKEN}",
"TEAGENT_INET" : "${TEAGENT_INET}" } }</data>
             <template_engine>VELOCITY</template_engine>
             <variable>
               <name>TEAGENT_ACCOUNT_TOKEN</name>
               <val>53rettywagbuouhw06hu65767rtyuyyui</val>
             </variable>
             <variable>
               <name>TEAGENT_INET</name>
               <val>4</val>
             </variable>
             </configuration>
           </config_data>
         </vm_group>
       </deployment>
    '
```

Step 9 Upgrade ThousandEyes Container Deployment on NFVIS Using RESTCONF

To upgrade the existing ThousandEyes container deployment with a newer container image, you need to update the deployment configuration with the new image details. Here is a sample RESTCONF command using a Docker Hub image:

```
curl -k -v -u admin:password -X PUT
'https://<nfvis_host_ip>/restconf/data/vmlc:vm_lifecycle/tenants/tenant=admin/deployments/deployment=TE_DEMO/vm_group=TE_DEMO/image'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>docker://thousandeyes/enterprise-agent:0.34.0-agent</image>'
```

Note If an upgrade fails due to issues such as an image download error, verification failure, or any other problem, the container will continue running with the old image. However, the NFVIS configuration will reflect the latest settings. This behavior is similar to how VM deployment updates are handled.

The ThousandEyes Enterprise Agent is deployed on NFVIS and ready to monitor network performance. The container is registered with ThousandEyes and begins collecting network metrics according to its configuration. If you upgraded the container, the existing configuration is updated with the selected image.

Troubleshooting NFVIS

Objective

This reference groups information about Troubleshooting NFVIS, including Log and show commands, Configure packet capture.

Log and show commands

Objective

Lists the support commands, show commands, and log files available for troubleshooting and monitoring system status.

Support and show commands

This reference provides information about the log and show commands available for Cisco NFVIS. These commands translate to corresponding Linux commands like virsh, ovs, and ip. The tech-support includes all the logs, and you can download tech-support and record the time of the occurrence of error.

Support commands and show commands

These commands translate to corresponding linux commands like virsh, ovs and ip:

Table 25: NFVIS support and show commands

Category	Command	Description
System	**show system status**	To display system defaults and services status.
	show system disk-space	To display information about the system disk space.
	show system memory	To display information about the system memory. If DPDK is enabled, check if HugePage is available to use.
	show resources cpu-info	To get information on the resource assignment.

Category	Command	Description
VM	**support virsh all-info**	To display the output of all supported VM and index by number.
	support virsh dumpxml *num*	To display all information for the VM at the specified index.
	support virsh domiflist *num*	To display the interfaces and VNIC MAC addresses for the VM at the specified index.
	support flush cache memory	To clear cache memory and free up some system memory for seamless performance of Cisco NFVIS. Clearing caches using the support flush cache command can help resolve issues related to outdated or corrupted cache data. For example, clearing the cache in a web browser can help resolve issues such as slow page load times of Cisco NFVIS portal.
Network	**support show ifconfig**	To display the configuration details of all network interfaces or a specific interface.
	support virsh net-list	To display all the networks in the host
	support virsh net-dumpxml *network name*	To display the network information about one network and bridge attachment.
	support virsh iface-list	To display a list of interfaces on the host.
Bridge	**support ovs vsctl show**	To display an overview of the bridge, port and vlan tag.
	support ovs appctl fdb-show *bridge-name*	To display information about the ports of a bridge.
	support ovs all-info	To display the output of all supported ovs commands
Firewall	**support show firewall get-all-rule**	To display all firewall rules.

Log files

The tech-support includes all the logs. Download tech-support and record the time of the occurrence of error.

Table 26: NFVIS log commands

Command	Description
show log	To display a list of available log files or content of a specific log file.
show log nfvis_syslog.log	To display syslogs.
show log nfvis_config.log	To display system configuration related logs.
show log esc/escmanager.log	To display VM deployment related logs.

Configure packet capture

Configure packet capture to analyze all packets being transmitted and received over physical and virtual network interface controllers.

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.

- 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.

Packets are stored in the /data/intdatastore/pktcaptures folder on the host server.

Procedure

Step 1 Configure packet capture on a physical port.

Example:

configure terminal

tcpdump port `eth0`

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

Step 2 Configure packet capture on a vNIC.

Example:

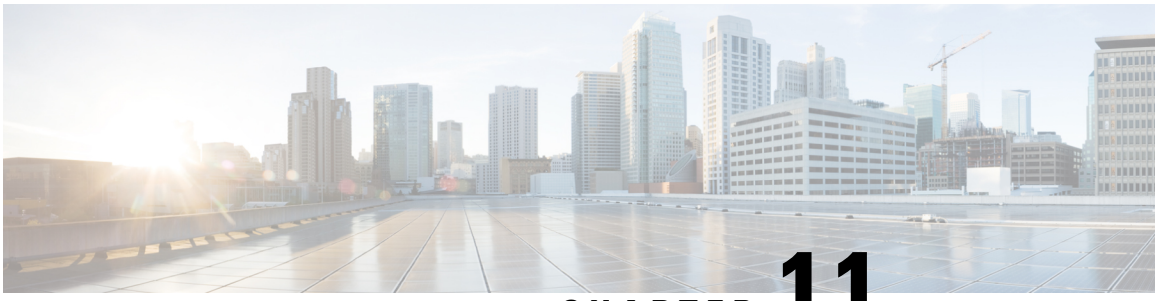
configure terminal

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

These errors are logged in the nfvis_config.log. By default, warnings and errors are logged.

Packet capture is configured and packets are captured to the specified location for analysis.



CHAPTER 11

NFVIS security considerations

Objective

NFVIS Embedded Security

NFVIS security layers and controls

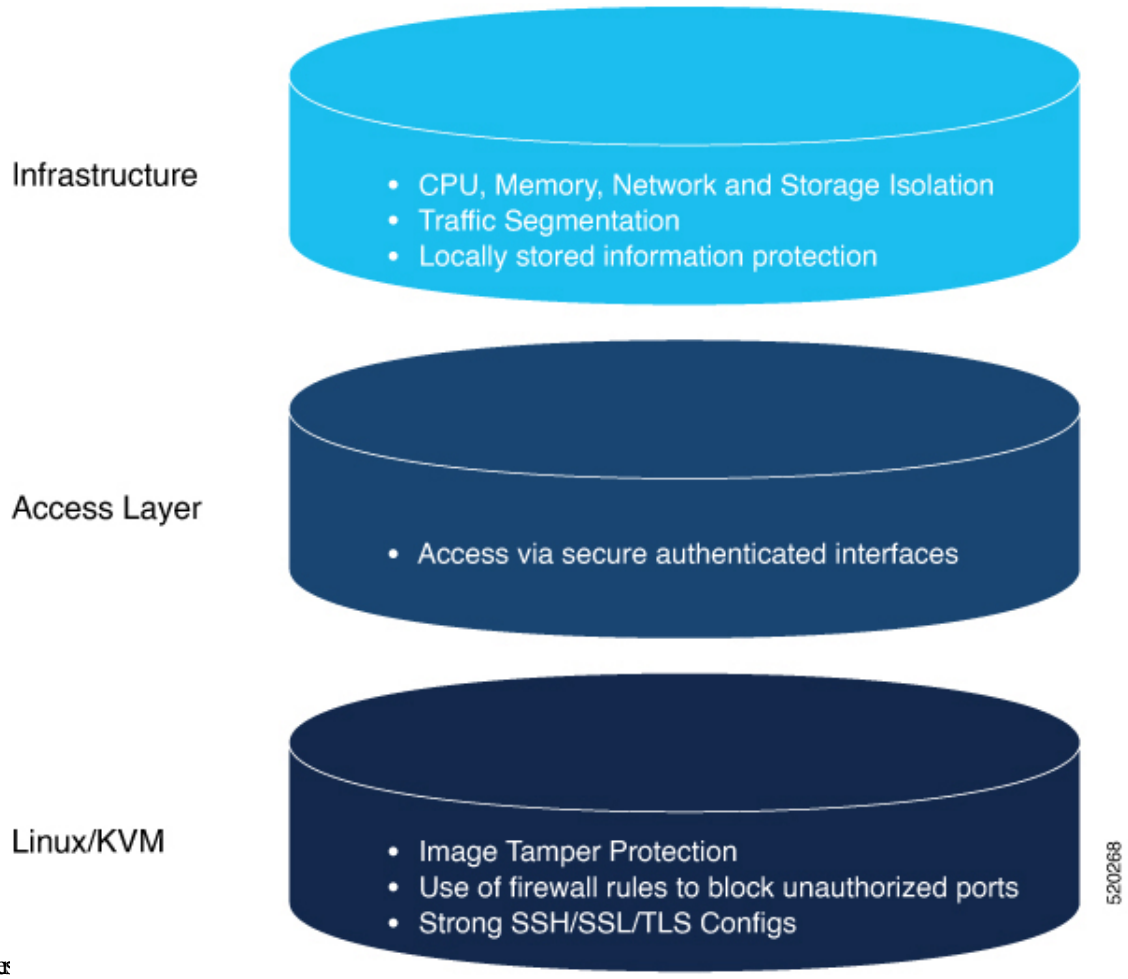
Security considerations are security features and components in NFVIS that

- provide a high-level overview of security related components for planning deployment-specific security strategies
- include recommendations on security best practices for enforcing core network security elements, and
- embed security from installation through all software layers including credential management, integrity and tamper protection, session management, and secure device access.

NFVIS software layers

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.

Figure 1. Software



layer

- Installation, on page 149
- Device access, on page 150
- Secure interfaces, on page 172
- NFVIS portal security, on page 174
- Certificate management, on page 176
- SNMP security features, on page 179
- Banners, on page 181
- Factory default reset, on page 183
- Infrastructure management networks, on page 184
- Protecting sensitive information, on page 186
- Logging, on page 187
- Virtual machine security, on page 188
- Secure development lifecycle, on page 193
- Public sector security certifications, on page 194

Installation

Objective

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

Objective

NFVIS supports RPM signing and signature verification for all RPM packages in the ISO and upgrade images.

RPM signing

RPM signing is a cryptographic security mechanism that

- ensures cryptographic integrity and authenticity of all RPM packages in the Cisco NFVIS ISO and upgrade images
- guarantees that the RPM packages have not been tampered with, and
- verifies that the RPM packages are from NFVIS using private keys created and securely maintained by Cisco.

RPM signature verification

RPM signature verification is a security mechanism that

- verifies the signature of all RPM packages before an installation or upgrade
- aborts the upgrade if signature verification fails, and
- ensures package integrity during NFVIS software operations.

NFVIS behavior during signature verification failure

NFVIS protects software installation and upgrade operations by

- verifying the signature of every RPM package before installation or upgrade, and
- aborting the operation if signature verification fails.

Image integrity verification

Image integrity verification is a security mechanism that

- ensures the integrity of all additional non-RPM files available in the Cisco NFVIS ISO image using published hash values

- provides hash verification for both Cisco NFVIS ISO images and upgrade images, and
- complements RPM signing and signature verification for complete image security.

How hash verification works

Summary

RPM signing and signature verification apply only to RPM packages available in Cisco NFVIS ISO and upgrade images. Cisco publishes a hash with the ISO image and a hash with the upgrade image to verify the integrity of additional non-RPM files.

Workflow

Run **/usr/bin/sha512sum** on the image and compare the result with the hash published by Cisco.

Process details

```
% /usr/bin/sha512sum ImageFile
c2122783efc18b039246ae1bcd4eec4e5e027526967b5b809da5632d462dfa6724a9b20ec318c74548c6bd7e9b8217ce96b5ece93dcdd74fda5e01bb382ad607
  <ImageFile>
```

Device access

Device access is a security framework that

- provides different access mechanisms including console as well as remote access based on protocols such as HTTPS and SSH
- ensures that access is only granted to authenticated users and they can perform just the authorized actions
- restricts device accessibility, user capabilities, and permitted methods of access to prevent unauthorized access to NFVIS.

Device access security controls

Objective

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. Device access is logged for auditing and NFVIS ensures the confidentiality of locally stored sensitive data.

Device access control recommendations

It is critical to establish the appropriate controls in order to prevent unauthorized access to NFVIS.

Enforced password change at first login

Enforced password change at first login is a security feature that

- prevents security incidents by requiring users to change default credentials after initial login
- forces NFVIS users to update their password when first accessing the system with default credentials (username: admin and password Admin123#), and
- protects systems from attacks that exploit unchanged default login credentials.

Login vulnerability restrictions

You can prevent the vulnerability to dictionary and Denial of Service (DoS) attacks by using the following features.

Enforcement of strong password

Objective

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 these rules.

Password composition requirements

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.

Configure minimum length for passwords

Configuring a minimum password length strengthens security by making brute-force attacks more difficult. Password length significantly affects the search space for attackers attempting to guess user passwords.

The admin user can set the minimum password length for all users. The value must be from 7 through 128 characters. The default minimum is 7 characters.

Procedure

Set the minimum password length using the CLI or API.

Set the value to an integer from 7 through 128. The following CLI command sets it to 9 characters:

```
nfvis(config)# rbac authentication min-pwd-length 9
```

To configure the value using the API, use this endpoint:

```
/api/config/rbac/authentication/min-pwd-length
```

The minimum password length is configured for all user accounts, enhancing system security by requiring stronger passwords.

What to do next

To set how long users can keep their passwords before changing them, see Configure password lifetime.

Configure 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.

Procedure

Step 1 Configure password lifetime using CLI.

Example:

configure terminal

rbac authentication password-lifetime enforce true min-days *2* **max-days** *30*

commit

Step 2 Configure password lifetime using API.

Example:

```
/api/config/rbac/authentication/password-lifetime/
```

Password lifetime settings are configured with the specified minimum and maximum values, and the enforcement rule is applied to all users except the admin user.

Previous password reuse limitation

Previous password reuse limitation is a security mechanism that

- prevents users from reusing recently used passwords
- makes password expiry meaningful by blocking immediate reversion to old passwords, and
- maintains password history to enforce security policies.

NFVIS password reuse prevention

Objective

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.

Login attempt frequency limits

Restrict the frequency of login attempts to prevent brute force attacks and avoid denial of service conditions.

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.

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

Procedure

Configure account inactivity enforcement using the CLI or API.

The following CLI example enables enforcement and sets the inactivity period to 30 days:

```
configure terminal
rbac authentication account-inactivity enforce true inactivity-days 30
commit
```

To configure account inactivity through the API, use this endpoint:

```
/api/config/rbac/authentication/account-inactivity/
```

User accounts that remain inactive for the specified number of days are automatically marked as inactive and cannot login to the system.

What to do next

To allow an inactive user to log in again, see Activate an inactive user account.

Activate an inactive user account

This task allows the admin user to reactivate a user account that has been deactivated or is in an inactive state.

User accounts may become inactive due to various reasons such as security policies or administrative actions. When a user account needs to be reactivated, the admin user can perform this task using either CLI commands or API calls.

Choose one of the following methods to activate the inactive user account:

Procedure

Activate the user account using the CLI or API.

To activate the example user account `guest_user` using the CLI, enter:

```
configure terminal
rbac authentication users user guest_user activate
commit
```

To activate a user account using the API, use this endpoint:

```
/api/operations/rbac/authentication/users/user/username/activate
```

The previously inactive user account is now activated and the user can access the system with their credentials.

Integration with external AAA servers

Integration with external AAA servers is a security framework that

- authenticates users through password credentials when they login to NFVIS via ssh or the Web UI
- authorizes users to perform specific operations based on their access requirements, and
- supports RADIUS and TACACS protocols to mediate network access through centralized AAA servers.

AAA server deployment recommendations

Objective

It is recommended that a centralized AAA server be deployed to enforce per-user, AAA-based login authentication for NFVIS 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.

Role based access control

Role-based access control (RBAC) is a method of restricting network access that

- limits access based on the roles of individual users within an enterprise
- lets users access just the information they need
- prevents them from accessing information that does not pertain to them, and
- uses an employee's role in the enterprise to determine the permissions granted.

NFVIS user roles and privileges

Objective

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.

NFVIS user roles and privileges

An employee's role in the enterprise should be used to determine the permissions granted, in order to ensure that employees with lower privileges cannot access sensitive information or perform critical tasks.

The user roles and privileges are defined in NFVIS:

User Role	Privilege
Administrators	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 "administrators".
Operators	Can start and stop a VM, and view all information.
Auditors	They are the least privileged users. They have read-only permission and therefore, cannot modify any configuration.

Benefits of using RBAC to restrict unnecessary network access based on people's roles within an organization:

- 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.

```
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 is permitted for each role. If a user is found to have unnecessary access to a certain system, change the user's role.

Roles

Objective

The three roles defined by the system are:

User role permissions

- Administrator: An administrator user has complete access to configure, update or delete a resource.
- Operator: An operator user can only view and operate a resource.
- Auditor: An auditor user can only view a resource and cannot perform any action on it.

Note All three roles have a read-access to all host level configurations, VMs, and images.

Users

A user is an account that

- is created with a role definition that is consistent across all groups
- has read access to NFVIS configurations, filesystems, logs, VMs and images
- can be a member of only one group, and
- may have remote authentication mapping for TACACS and RADIUS based on privilege level.

Admin user characteristics

Objective

The admin user is a special user account with unique properties:

Admin user account properties

- The admin user cannot be deleted or modified.
- The admin user permanently has the administrator role in the default global group.
- The admin user functions as a member of every group and can execute administrative privileges for every group.
- The admin user cannot be assigned to a specific user group.

Groups

A group is a collection of users that

- provides access control to resources based on membership
- defines resource assignment boundaries where a resource can belong to one specific group, and
- ensures all members have access to the resources assigned to that group, with privileges defined by the user's role.

Group types and characteristics

Objective

The global group is a special group assigned to users who are not members of any other group. A user in the global group can access all resources on the system, at the privilege level.

Group types and resource access

Resources can be assigned to the global group or a specific group.

When creating a group, the `resource-access-control` policy should be enabled, to have resource restrictions.

Create groups and assign local users to the groups

Groups organize users and apply policies for role-based access control (RBAC). You can create a group with or without a policy, assign local users to it, and view the users with their roles and group membership.

Procedure

Step 1 Create a group with the resource-access-control policy.

```
aaa groups group rac_group policy resource-access-control
commit
Commit complete.
```

Step 2 Create a local user and assign the user to the group.

```
rbac authentication users create-user name local_admin_3 password Cisco123# role
administrators group rac_group
```

Step 3 View the RBAC users and their role and group information.

```
show running-config rbac authentication users
rbac authentication users user admin role administrators
!
rbac authentication users user local_admin_1 role administrators
!
rbac authentication users user local_admin_2 role administrators
!
rbac authentication users user local_admin_3 role administrators groups group rac_group
!
!
rbac authentication users user local_oper_1 role operators
!
rbac authentication users user local_test_1 role operators
!
```

The group is created and the local user is assigned to it. The running configuration shows the user's role and group.

Assign remote users to the group

Assign remote users to a resource control group so they can manage deployments based on their defined role in the remote server.

NFVIS depends on a remote server for user authentication and authorization. When remote users login to NFVIS, they can only manage the deployment that belongs to its own resource control group, based on their defined role in the remote server. Any remote user that is not mentioned in the resource group, is treated as a global group user and that user can operate the system and manage the deployments based on their defined role.

Procedure

Assign remote users to a resource control group and verify the group membership.

In this example, add three TACACS+ or RADIUS users to the `tac_group` resource control group:

```
aaa groups group tac_group user remote_admin1
user remote_admin2
user remote_operator3
policy resource-access-control
commit
end
```

Verify the group configuration:

```
show running-config aaa groups group tac_group
aaa groups group tac_group
policy resource-access-control
user remote_admin1
user remote_admin2
user remote_operator3
!
!
```

Verify the users:

```
nfvis# show rbac authentication users
NAME
-------
admin
```



Note In this example, `remote_admin1`, `remote_admin2`, and `remote_operator3` are TACACS+ or RADIUS users.

The remote users are assigned to the resource control group and can manage deployments according to their defined roles.

Configure local authentication for a specific group of users

This task enables you to create a group and add specific users to it, allowing these users to bypass the default authentication order and only go through local authentication instead of the default TACACS then local authentication sequence.

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.

Restrictions for local authentication for a specific group of users:

- One user can be assigned to a maximum of one group.
- If a local user assigned to the local authentication group has the same user name as a remote (TACACS+/RADIUS) user, then only the local user's credentials are taken into consideration. The remote user's credentials are considered even if the local user's authentication fails.

Procedure

Step 1 Create a group with the local-authentication-only policy.

Example:

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

Step 2 Assign a user to the group using one of these methods:

Assign an existing user to the group:

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

Create a new user and assign to the group simultaneously:

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

To remove a user from a group, use:

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

The group is created with the local-authentication-only policy, and the specified users are assigned to the group. These users bypass external TACACS authentication and use local authentication.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client-server system that

- secures networks against unauthorized access
- uses Cisco routers as RADIUS clients that send authentication requests to a central RADIUS server containing all user authentication and network service access information, and
- operates as a fully open protocol distributed in source code format that can be modified to work with any security system currently available on the market.

RADIUS implementation details

Objective

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

RADIUS server configuration and failover

Note You can configure upto 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.

How RADIUS authentication works

Summary

Summary

Workflow

The key components involved in RADIUS authentication are:

Process details

- User: Provides credentials and receives authentication responses
- Access server: Prompts for credentials and relays authentication requests
- RADIUS server: Validates credentials and provides authorization data

Workflow

These stages describe how RADIUS authentication works:

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 these responses from the RADIUS server:
   - ACCEPT—The user is authenticated.
   - CHALLENGE—A CHALLENGE is issued by the RADIUS server. The CHALLENGE collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new PASSWORD.
   - REJECT—The user is not authenticated and is prompted to reenter the username and PASSWORD, or access is denied.

Result

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 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, and connection parameters, including the host or client IP address, access list, and user timeouts.

Configure RADIUS

NFVIS supports RADIUS shared-secret encryption. Configure either a shared secret or an encrypted shared secret at a time. Use an encrypted secret when the secret contains special characters. NFVIS encrypts both shared-secret and encrypted-shared-secret configurations and supports TLS and non-TLS RADIUS communication.

The secret must be 1 through 127 characters and contain only characters from this set: `[-_a-zA-Z0-9 ./\<>%!*$€#{}()+@]*.`

Procedure

Step 1 Configure RADIUS support using one of the following methods:

For non-TLS RADIUS communication, enter:

```
configure terminal
radius-server host 1.2.3.4 shared-secret abc admin-priv 15 oper-priv 11
commit
```

For TLS RADIUS communication, enter:

```
configure terminal
radius-server host 1.2.3.4 shared-secret efkbkuabcwuabvauvvwqbd use-tls true pskidentity
 identity admin-priv 15 oper-priv 11
commit
```

Step 2 Verify the RADIUS configuration.

Use the following command:

```
show running-config radius-server
```

The shared secret is displayed in encrypted form.

```
radius-server host 1.2.3.4
shared-secret $8$lZwgccBvs9x1oQpx6f1s8/JkZ4rLdQ95VMUjRrhD9Z8=
admin-priv    15
oper-priv     11
```

RADIUS authentication is configured with the specified server, secret, and privilege levels.

TACACS+

TACACS+ is a security application that

- provides centralized validation of users attempting to gain access to a router or network access server
- maintains services in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation, and
- requires TACACS+ server configuration before the configured TACACS+ features on your network access server are available.

TACACS+ server configuration requirements

Objective

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

TACACS+ server availability and limits

Note You can configure upto 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.

How TACACS+ authentication works

Summary

Summary

Workflow

The key components involved in TACACS operation are:

Process details

- User: Attempts to log in to NFVIS using credentials
- NFVIS: Sends user credentials to TACACS+ server and processes responses
- TACACS+ server: Authenticates users and provides authorization responses

Workflow

The TACACS operation involves these stages:

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:
   - ACCEPT—The user is authenticated and service can begin. If NFVIS is configured to require authorization, authorization begins at this time.
   - 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.
   - 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.
   - 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.

Configure a TACACS+ server

Configure TACACS+ server authentication to enable centralized user authentication and authorization with customizable privilege levels for different user roles.

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. The following pattern is supported for encrypted-shared-key: [-_a-zA-Z0-9./\\<>%!*$€#{}()+].

NFVIS encrypts both shared-secret and encrypted-shared-secret configurations.

NFVIS supports both TLS and non-TLS client-server communication for TACACS+.

The following constraints apply:

- Secret length must be between 1 and 127 characters.
- Secret must only contain characters from the set: [-_a-zA-Z0-9 ./\<>%!*$€#{}()+@]*

Procedure

Step 1 Configure a TACACS+ server without TLS.

```
configure terminal
tacacs-server host 1.2.3.4 shared-secret asdfghh 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 privilge 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.

Step 2 Configure a TACACS+ server with TLS.

```
configure terminal
tacacs-server host 1.2.3.4 shared-secret mfqwudvkdwnbkyuDLndkw use-tls true pskidentity
 identity admin-priv 14 oper-priv 9
commit
```

Step 3 Verify the TACACS+ configuration.

Use the following command:

```
show running-config tacacs-server
```

The shared secret is displayed in encrypted form.

```
tacacs-server host 1.2.3.4
shared-secret $8$JkMZFgA3DkbjAHOrmdBr3U2cLg2qY1FuHAIJiIp7nSw=
admin-priv    14
oper-priv     9
```

The TACACS+ server is configured with the specified host, secret, and privilege levels. Users can authenticate through the server with the corresponding role-based privileges.

Default authentication order

Default authentication order is a security mechanism that

- supports both TACACS+ and RADIUS but allows only one authentication method to be enabled at a time
- requires method lists to be defined for TACACS+ and RADIUS authentication through AAA commands, and
- uses local authentication as fallback when TACACS+ or RADIUS is not accessible.

Authentication configuration

Objective

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.

AAA authentication methods and behavior

```
nfvis(config)# aaa authentication ?
Possible completions:
  radius   Use RADIUS for AAA
  tacacs   Use TACACS+ for AAA
  users    List of local users
```



Note

- 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 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.

User specific authentication order

Objective

The system follows this authentication sequence for user-specific authentication:

User-specific authentication sequence

- 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.

Granular role-Based access control

Objective

This feature adds a new resource group policy that manages the VM and VNF and allows you to assign users to a group to control VNF access, during VNF deployment. For more information, see below

Resource group operations and role permissions

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
User Management							
Create, update, delete new logins and assign roles	Y	Y	N	N	N	N	N
List all logins	Y	Y	Y	Y	Y	Y	Y
Ability to reset password on any login	Y	Y	N	N	Y	N	N
Create and delete custom VNF groups and assign users to VNF groups	Y	Y	N	N	N	N	N
VNF Management							
Create	Y	Y	N	N	Y	N	N
Delete or update VNFs	Y	Y	N	N	Specific VNFs	N	N
Update VNF group name	Y	Y	N	N	N	N	N
Start and stop VNFs	Y	Y	Y	N	Specific VNFs	Specific VNFs	N
VNF snapshots	Y	Y	Y	N	Specific VNFs	Specific VNFs	N

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
Restore from snapshot	Y	Y	N	N	Specific VNFs	N	N
List all VNF instances	Y	Y	Y	Y	Specific VNFs	Specific VNFs	Specific VNFs
Access the VNF console	Y	Y	Y	N	Specific VNFs	Specific VNFs	N
Assign VNF to a custom group	Y	Y	N	N	N	N	N
Device Level Operations							
View and monitor device level statistics, including logs	Y	Y	Y	Y	Y	Y	Y
Factory reset	Y	Y	N	N	N	N	N
System-wide device configurations such as DNS	Y	Y	N	N	N	N	N
Device backup and restore	Y	N	N	N	N	N	N
Platform upgrades	Y	Y	N	N	N	N	N
Change BIOS password, change CIMC password, PNIC-breakout, power cycle, host reboot, and hostaction shutdown	Y	Y	N	N	N	N	N
Clear BIOS password, SSH keys, and stats files	Y	Y	N	N	N	N	N
Clear cache, mgmt-dhcp-renew, wan-dhcp-renew, and PnP actions	Y	Y	Y	N	N	N	N
System certificate modifications, ISCSI settings, and file delete operation	Y	Y	N	N	N	N	N
Tech support and logs	Y	Y	Y	Only through portal	Y	Y	Only through portal

Operations/Roles	"admin" Super User	Administrator	Operator	Auditor	VNF Group Administrator	VNF Group Operator	VNF Group Auditor
Create VFs for SR-IOV, and enable DPDK	Y	Y	N	N	N	N	N
Image Repository Changes							
Upload new VNF images	Y	Y	N	N	Y	N	N
Upload new NFVIS upgrade files	Y	Y	N	N	Y	N	N
File transfer to the device (certificates)	Y	Y	N	N	Y	N	N
File transfer from the device	Y	Y	N	N	Y	N	N
File-copy and file-download	Y	Y	N	N	Y	N	N
File-delete	Y	Y	N	N	N	N	N
View contents	Y	Y	Y	Y	Y	Y	Y
Clustering Operations							
Set up a cluster (both management and storage)	Y	Y	N	N	N	N	N

Restrict device accessibility

Objective

Disable unused services to prevent security vulnerabilities from default configurations and passwords that attackers can exploit to gain unauthorized access to system information.

Restricting device access

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.

Essential ports are enabled 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.

Open Port	Service	Description
22/TCP	SSH	Secure Socket Shell for remote command-line access to NFVIS
80/TCP	HTTP	Hypertext Transfer Protocol for the NFVIS portal access. All HTTP traffic received by NFVIS is redirected to port 443 for HTTPS
443/TCP	HTTPS	Hypertext Transfer Protocol Secure for secure NFVIS portal access
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, SNMP.

Authorized network access restrictions

Authorized network access restriction is a security mechanism that

- permits only authorized originators to attempt device management access
- limits access to services they are authorized to use, and
- reduces the risk of unauthorized access and exposure to attacks such as brute force, dictionary, or DoS attacks.

NFVIS access control lists

Objective

NFVIS can be configured such that access is restricted to known, trusted sources and expected management traffic profiles. 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.

IP receive ACL parameters and examples

These parameters are configured in an IP receive Access Control List (IP-receive-ACL):

Parameter	Value	Description
Source network/Netmask	Network/netmask. For example: 0.0.0.0/0 172.39.162.0/24	This field specifies the IP address/network from which the traffic originates
Service	HTTPS ICMP NETCONF scpd SNMP SSH	Type of traffic from the specified source.
Action	accept drop reject	Action to be taken on the traffic from the source network. With accept, new connection attempts will be granted. With reject, connection attempts will not be accepted. If the rule is for a TCP based service such as HTTPS, NETCONF, SCP, SSH, the source will get a TCP reset (RST) packet. For non-TCP rules such as SNMP and ICMP, the packet will be dropped. With drop, all packets will be dropped immediately, there is no information sent to the source.
Priority	A numeric value	The priority is used to enforce an order on the rules. Rules with a higher numeric value for priority will be added further down in the chain. If you want to make sure that a rule will be added after another one, use a low priority number for the first and a higher priority number for the next.

The 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:

```
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.

```
nfvis(config)# no system settings ip-receive-acl 171.70.63.0/24
```

For more details see, Configure the IP receive ACL

Access privileged debug shell

The super-user account on NFVIS is disabled by default to prevent unrestricted, potentially adverse system-wide changes, and NFVIS does not expose the system shell to users. However, for difficult-to-debug issues on NFVIS, the Cisco Technical Assistance Center (TAC) or development team might require shell access to the customer's NFVIS. NFVIS uses a secure unlock infrastructure to restrict privileged debug access on field devices to authorized Cisco employees. A challenge-response authentication mechanism between NFVIS and the Cisco-maintained Interactive Debug server provides access to the Linux shell for this interactive debugging. The admin user's password is also required, in addition to the challenge-response entry, to confirm the customer's consent.

Follow these steps to access the shell for interactive debugging:

Procedure

Step 1 Initiate the shell access procedure using the hidden command.

```
nfvis# system shell-access
```

Step 2 Copy the challenge string displayed on the screen.

```
Challenge String (Copy everything between the asterisk lines exclusively):
********************************************************************************
SPH//wkAAABORlZJU0VOQ1M1NDA4L0s5AQAAABt+dcx+hB0V06r9RkdMMjEzNTgw
RlHq7BxeAAA=
DONE.
********************************************************************************
```

Step 3 Provide the challenge string to the Cisco member for verification.

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.

Step 4 Enter the response string when prompted.

Input your response when ready:

Step 5 Enter the admin password when prompted.

Step 6 Access the shell if the password is valid.

The development or TAC team uses the shell to proceed with debugging.

Step 7 Type **Exit** to exit shell access when debugging is complete.

You have successfully accessed the privileged debug shell and can perform interactive debugging with Cisco support team assistance.

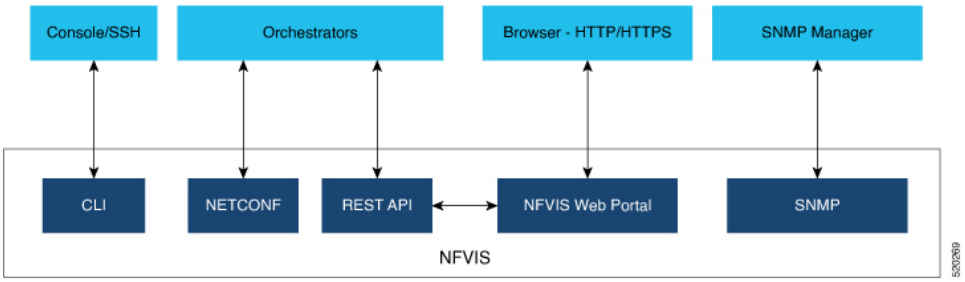
Secure interfaces

Objective

Describes the interfaces available for NFVIS management access and outlines security best practices to implement for each interface to ensure secure system administration.

NFVIS management 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

A console port is an asynchronous serial port that

- allows you to connect to the NFVIS CLI for initial configuration
- can be accessed with either physical access to the NFVIS or remote access through the use of a terminal server, and
- requires access lists on the terminal server to allow access only from the required source addresses when console port access is required via a terminal server.

SSH

SSH is a secure network protocol that

- provides users with secure remote access to the NFVIS CLI
- ensures the integrity and confidentiality of NFVIS management traffic, and
- uses version 2 with strong encryption, hash, and key exchange algorithms recommended by the Security and Trust Organization within Cisco.

NFVIS uses SSH version 2, which is Cisco's and the Internet's de facto standard protocol for interactive logins. 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.

CLI session timeout

A CLI session timeout is a security mechanism that

- automatically logs out users from CLI sessions after 15 minutes of inactivity
- limits the risk of internal attacks, such as one user trying to use another user's session, and
- reduces network security risks when users leave logged-in sessions unattended.

Session security mechanism

Objective

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.

CLI session timeout behavior

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 that

- is developed and standardized by the IETF for the automated configuration of network devices
- uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages, and
- exchanges protocol messages 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.

REST API

A REST API is a configuration interface that

- allows requesting systems to access and manipulate NFVIS configuration using a uniform and predefined set of stateless operations
- operates over HTTPS for secure communication, and

- limits concurrent sessions to 100 to prevent denial of service attacks.

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 portal security

The NFVIS portal is a web-based graphical user interface that

- displays information about NFVIS
- presents users with an easy means to configure and monitor NFVIS over HTTPS, and
- eliminates the need to know the NFVIS CLI and API.

Session management

Objective

The stateless nature of HTTP and HTTPS requires a method of uniquely tracking users through the use of unique session IDs and cookies.

Session encryption, auditing, and timeout

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 these 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 are currently authenticated. To prevent this, NFVIS uses CSRF tokens to validate every REST API that is invoked during each session.

URL redirection

Objective

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.

HTTP redirection and security headers

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 these headers configured:

- X-Content-Type-Options
- X-XSS-Protection
- Content-Security-Policy
- X-Frame-Options
- Strict-Transport-Security
- Cache-Control

Portal disablement

Objective

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:

Disabling portal access

```
Configure terminal
System portal access disabled
commit
```

HTTPS

Objective

All 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.

HTTPS certificates and TLS security

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. 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).

Certificate management

Objective

NFVIS uses a server certificate for HTTPS management access for Portal. The default certificate is self-signed; administrators can replace it with a CA-signed certificate using the NFVIS CSR and certificate-install actions. The built-in CSR action accepts subject fields but has no SAN input. NFVIS also has no supported action to import an externally generated private key. The supported flow is to keep the NFVIS-generated key on the appliance and ask the CA to add SANs to the issued certificate, if the CA's policy allows it.

Certificate paths and consumers

Objective

At startup, NFVIS generates a self-signed RSA certificate and key under /etc/pki/self-signed/, then installs copies as /etc/pki/nginx/server.crt and /etc/pki/nginx/server.key. Nginx uses those active files for HTTPS. The VNC console's WebSocket proxy also uses the active Nginx certificate and key.

Certificate paths and consumers

The self-signed certificate is regenerated when the NFVIS hostname changes. The generator sets the common name to that hostname and does not add SAN extensions. Modern browsers check SANs for hostname identity, so the default certificate may show a name error as well as an untrusted-issuer warning.

The **system certificate use-cert** action switches the active certificate between the self-signed pair and the CA-signed pair. Switching to self-signed copies files from /etc/pki/self-signed/. Switching to CA-signed uses /

etc/pki/third-party/server.key and /etc/pki/third-party/server.crt.

Generate and retrieve an NFVIS CSR

The certificate signing request (CSR) command accepts common name, country code, locality, organization, organization unit name, and state fields. The common name is required.

Procedure

Step 1 Generate the CSR on NFVIS.

Replace `NFVIS-HOSTNAME` with the DNS name clients use to reach NFVIS.

```
nfvis# system certificate signing-request common-name NFVIS-HOSTNAME country-code US
organization nfvis-ap
```

NFVIS saves the CSR to `/data/intdatastore/download/nfvis.csr` and generates the matching private key at `/etc/pki/third-party/server.key`. The private key remains on NFVIS and is not included in the CSR.

Step 2 Retrieve only the CSR from a workstation using SCP on port 22222.

Use the `-O` option with OpenSSH clients that fail to work with the NFVIS transfer endpoint because they use SFTP by default:

```
scp -O -P 22222 admin@NFVIS-IP:/data/intdatastore/download/nfvis.csr .
```

The CSR is generated on NFVIS and copied to the workstation. The matching private key remains on NFVIS.

What to do next

To request subject alternative names for the issued certificate, see Add SANs when issuing the certificate.

Private-key support and external CSR limitation

Objective

NFVIS does not provide a supported CLI or API action to import a private key generated elsewhere. The install-cert action installs a certificate file only; it does not import a key. The use-cert cert-type ca-signed action expects the matching key to already exist at /etc/pki/third-party/server.key.

External private key and CSR limitations

The separate system certificate input action is for a PnP root certificate. It does not import a server private key. Manually copying an external key into NFVIS is outside the supported workflow. In the current implementation, activation checks that the key and certificate files exist and that the certificate parses, but it does not verify that the key matches the certificate. Compare their public-key digests before activation.

If the customer must use an externally generated key and CSR, the current NFVIS workflow does not support that request. It would require a product enhancement to securely import the key and generate a SAN-bearing CSR from it.

Add SANs when issuing the certificate

The NFVIS CSR action does not accept subject alternative name (SAN) fields and generates a subject-only request. If SANs must be included in the CSR itself, the current NFVIS command does not support that.

When you submit the CSR to your organization's CA, specify the required DNS names and IP addresses in the CA enrollment form or certificate profile. The CA may issue a certificate with SANs that are not encoded in the CSR if its policy allows it. The issued certificate must retain the public key from the NFVIS CSR so that it matches the private key stored on NFVIS.

Procedure

Step 1 For a local test CA, define the certificate extensions and SAN values.

Replace `NFVIS-HOSTNAME` and `NFVIS-IP` with the hostname and management IP address clients use to reach NFVIS:

```
[server_cert]
basicConstraints = critical,CA:FALSE
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer

[alt_names]
DNS.1 = NFVIS-HOSTNAME
IP.1 = NFVIS-IP
```

Step 2 Sign the NFVIS CSR with the test CA certificate and private key.

This local CA procedure is for testing. Keep the CA private key off NFVIS. For production, use your organization's approved CA process and profile.

openssl x509 -req -in *nfvis.csr* **-CA** *test-ca.crt* **-CAkey** *test-ca.key* **-CAcreateserial -out** *nfvis-test.crt* **-days** *30* **-sha256 -extfile** *nfvis-san.conf* **-extensions** *server_cert*

The CA can issue a certificate with the requested SANs if its policy allows, and the certificate public key matches the private key generated on NFVIS.

Verify and install the certificate

Objective

Check the issued certificate's subject, issuer, validity period, and SANs:

Certificate verification commands

openssl x509 -in nfvis-test.crt -noout -subject -issuer -dates -ext subjectAltName

To verify the certificate matches the CSR's public key, compare the two SHA-256 digests. They must match:

```
openssl req -in nfvis.csr -pubkey -noout |
openssl pkey -pubin -outform DER | openssl dgst -sha256
openssl x509 -in nfvis-test.crt -pubkey -noout |
openssl pkey -pubin -outform DER | openssl dgst -sha256
```

Upload the issued certificate to NFVIS:

scp -O -P 22222 nfvis-test.crt admin@@@VAR0@@:/data/intdatastore/uploads/

Install and activate it from the NFVIS CLI:

```
nfvis# system certificate install-cert path
file:///data/intdatastore/uploads/nfvis-test.crt
nfvis(config)# system certificate use-cert cert-type ca-signed
```

The install action accepts a certificate path. The CA-signed activation action uses the private key already at /etc/pki/third-party/server.key. Confirm the certificate matches that key before activation; the current action code does not compare them.

HTTPS and browser trust validation

Objective

For a locally issued test certificate, verify the server using the local test CA certificate.

Validate HTTPS and browser trust

```
openssl s_client \
-connect NFVIS-IP:443 \
-servername NFVIS-HOSTNAME \
-CAfile test-ca.crt \
-verify_hostname NFVIS-HOSTNAME \
-brief </dev/null
```

If OpenSSL reports successful verification, the presented chain and hostname validate against `test-ca.crt`. A browser may still show a warning until that test CA certificate is added to the workstation's trusted root store. The browser URL must also match a SAN: use a DNS name listed in the certificate or an IP address listed as an IP SAN.

For production, use a CA already trusted by clients, or distribute the organization's private CA root through its managed trust process. Importing a test CA makes certificates issued by that CA trusted by the workstation, so remove the test CA trust after testing.

SNMP security features

Objective

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.

SNMP version security options

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.

This table summarizes the security options within SNMP:

Model	Level	Authentication	Encyption	Outcome
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5-96 or HMAC-SHA-96 algorithms.
v3	authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	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)

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.

SNMP on NFVIS supports V1, V2, and V3 — but only SNMPv3 with authPriv (sha256/AES) operates in secure mode; all other combinations (V1, V2, and V3 with noAuthNoPriv or authNoPriv using md5/SHA/DES) are classified as insecure.

Insecure mode dependency

Objective

Insecure mode dependency.

SNMP requirements in insecure mode

```
nfvis# show system mode
system mode status secure
nfvis(config)# snmp user test user-version 1 user-group test_group
auth-protocol md5 priv-protocol des passphrase qwertyuiop
encryption-passphrase qwertyuiop
nfvis(config-user-test)# commit
Aborted: 'snmp user test user-version': SNMP Version 1 is insecure. Enable insecure mode
to configure this
```

Insecure mode

Objective

Insecure mode.

Insecure mode commands and behavior

```
nfvis# show system mode
system mode status insecure
nfvis(config)# snmp user test user-version 1 user-group test_group
auth-protocol md5 priv-protocol des passphrase qwertyuiop
encryption-passphrase qwertyuiop
nfvis(config-user-test)# commit
Commit complete.
```

Banners

Objective

We recommend 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.

Legal notification banner considerations

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.

Banner content requirements

Objective

Discuss this issue with your own legal counsel to ensure that the notification banner meets company, local, and international legal requirements.

Legal banner content requirements

- 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.

Legal notification requirements are complex and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary. This is often critical to securing appropriate action in the event of a security breach.

Security considerations for banner content

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.

Sample banner implementation

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
```



Note Present a legal notification banner approved by company legal counsel.

NFVIS banner configuration

Objective

We recommend 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.

Banner and MOTD configuration command

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. 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 your banner and message of the day.

Factory default reset

A factory default reset is a device reset feature that

- removes all customer-specific data that has been added to the device since the time of its shipping
- erases configurations, log files, VM images, connectivity information, and user login credentials, and
- provides one command to reset the device to factory-original settings.

Factory default reset is useful in these 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 these options within Factory default reset:

Factory Reset Option	Data Erased	Data Retained
all	All configuration, uploaded image files, VMs and logs. Connectivity to the device will be lost.	The admin account is retained and the password will be changed to the factory default password.
all-except-images	All configuration except image configuration, VMs, and uploaded image files. Connectivity to the device will be lost.	Image configuration, registered images and logs The admin account is retained and the password will be changed to the factory default password.

Factory Reset Option	Data Erased	Data Retained
all-except-images-connectivity	All configuration except image, network and connectivity configuration, VMs, and uploaded image files. Connectivity to the device is available.	Images, network and connectivity related configuration, registered images, and logs. The admin account is retained and the previously configured admin password will be preserved.
manufacturing	All configuration except image configuration, VMs, uploaded image files, and logs. Connectivity to the device will be lost.	Image related configuration and registered images The admin account is retained and the password will be changed to the factory default password.

The user must choose the appropriate option carefully based on the purpose of the Factory Default reset.

Infrastructure management networks

An infrastructure management network is a network that

- carries the control and management plane traffic (such as NTP, SSH, SNMP, syslog, etc.) for the infrastructure devices
- provides visibility into and control over the network through device access via console and Ethernet interfaces, and
- enables remote manageability even under high load and high traffic conditions.

Infrastructure management network design considerations

This control and management plane traffic is critical to network operations. 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.

Infrastructure management network implementation approaches are:

Out-of-band management

Out-of-band management (OOB) is a network management approach that

- uses a network which is completely independent and physically disparate from the data network that it helps to manage
- is sometimes referred to as a Data Communications Network (DCN), and
- provides greater control over device management by restricting management packets to designated interfaces.

Connection methods and benefits

Objective

Network devices can connect to the OOB network in different ways:

OOB connection methods and benefits

- 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.
- 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.

Benefits include:

- More security for devices by restricting management packets to designated interfaces
- 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
- Prevention of management packet floods from reaching the CPU.

Pseudo out-of-band management

A pseudo out-of-band management network is a network configuration that

- uses the same physical infrastructure as the data network but provides logical separation through the virtual separation of traffic, by using VLANs
- enables NFVIS to create VLANs and virtual bridges to help identify different sources of traffic and separate traffic between VMs, and
- isolates the virtual machine network's data traffic and the management network through separate bridges and VLANs, thus providing traffic segmentation between the VMs and the host.

In-band management

An in-band management network is a management approach that uses the same physical and logical paths as the data traffic.

Network design considerations

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.

Protecting sensitive information

Protecting sensitive information is a security mechanism that

- stores passwords and secrets locally on NFVIS as hashes to prevent recovery of original credentials
- maintains passwords through centralized AAA servers while providing local fallback capabilities, and
- follows widely accepted industry norms for password protection.

Local password storage requirements

Objective

NFVIS requires locally-stored passwords for specific cases even when centralized AAA servers are deployed:

Cases requiring locally stored passwords

- Local fallback when AAA servers are not available
- Special-use usernames

File transfer

File transfer is a network operation that

- enables the secure copying of VM image and NFVIS upgrade files to NFVIS devices
- uses Secure Copy (SCP) protocol to ensure security and authentication during transfer, and
- relies on SSH for secure authentication and transport mechanisms.

File transfer implementation

Objective

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.

SCP syntax and transfer behavior

The syntax for the SCP command is:

```
scp source destination
```

NFVIS uses port 22222 for the 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 (@@VAR0@@)/(@@VAR1@@) service scpd priority
 (@@VAR2@@) 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

Logging is a security mechanism that

- records NFVIS access and configuration changes as audit logs
- provides forensic analysis capabilities for unauthorized access attempts and configuration issues, and
- enables real-time identification of anomalous activities that may indicate attacks.

NFVIS access and configuration changes are logged as audit logs to record this 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.

Virtual machine security

Objective

This section describes security features related to the registration, deployment and operation of Virtual Machines on NFVIS.

VNC console access protection

VNC console access protection is a security mechanism that

- allows users to create Virtual Network Computing (VNC) sessions to access a deployed VM's remote desktop
- dynamically opens a port for 60 seconds to which users can connect using their web browser, and
- assigns port numbers dynamically to allow only one-time access to the VNC console.

How VNC console access works

Summary

NFVIS dynamically opens a port when a user creates a VNC session. 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.

Workflow

NFVIS dynamically opens a port when a user creates a VNC session.

VNC console command example

Objective

Pointing your browser to https://<NFVIS ip>:6005/vnc_auto.html will connect to the ROUTER VM's VNC console.

VNC console command example

```
nfvis# vncconsole start deployment-name 1510614035 vm-name ROUTER
vncconsole-url :6005/vnc_auto.html
```

Encrypted VM config data variables

An encrypted VM config data variable is a security mechanism that

- allows users to flag config data variables as sensitive during VM deployment
- encrypts sensitive values using AES-CFB-128 encryption before storage or passing to internal subsystems, and
- prevents passwords and keys from appearing as clear text in log files and internal database records.

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.

Checksum verification for remote image registration

Checksum verification for remote image registration is a security mechanism that

- verifies the integrity of downloaded VNF images from external sources
- ensures files are not corrupted during network transmission or modified by malicious third parties, and
- uses SHA256 or SHA512 algorithms to validate image checksums before installation.

How checksum verification works

Summary

When registering a remotely located VNF image, the user specifies its location for download from external sources such as NFS servers or remote HTTPS servers.

Workflow

NFVIS supports checksum and checksum_algorithm options that allow users to provide the expected checksum and specify the checksum algorithm (SHA256 or SHA512) for verifying the downloaded image. Image creation fails if the checksum does not match the expected value.

Certification validation for remote image registration

Certification validation for remote image registration is a security mechanism that

- verifies SSL certificates when downloading VNF images from remote HTTPS servers
- requires users to specify either the path to the certificate file or PEM format certificate contents, and
- ensures secure downloads during the image registration process in NFVIS.

Certificate specification requirements

Objective

When registering a VNF image located on an HTTPS server, the image must 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.

VM isolation and resource provisioning

VM isolation and resource provisioning is a network function virtualization capability that

- partitions physical hardware resources into separate virtual environments for multiple VNFs
- ensures individual VM domains are isolated as separate, distinct, and secure environments that do not contend with each other for shared resources, and
- prevents VMs from using more resources than provisioned to avoid Denial of Service conditions.

NFV architecture components

Objective

The Network Function Virtualization (NFV) architecture consists of:

NFV architecture components and resource protection

- 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. Resources are partitioned as needed from the physical environment to the many virtual environments.

As a result, CPU, memory, network and storage are protected.

CPU isolation

CPU isolation is a resource management mechanism that

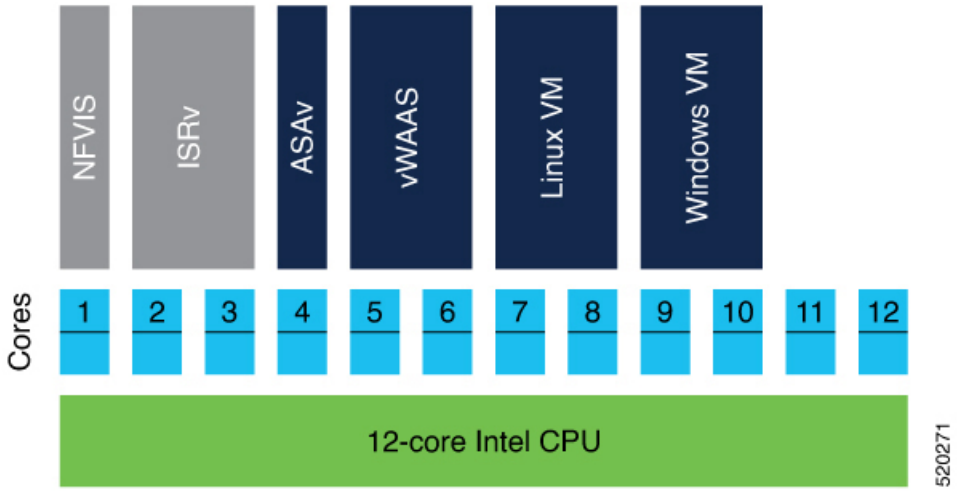
- reserves cores for infrastructure software running on the host
- makes the remaining cores available for VM deployment, and
- guarantees that VM performance does not affect NFVIS host performance.

CPU allocation by VM type

Objective

NFVIS handles CPU allocation differently based on VM latency requirements.

CPU allocation methods by VM type



The system uses two distinct allocation methods:

- Low-latency VMs: NFVIS explicitly assigns dedicated cores to low latency VMs. 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

Memory allocation is a resource management mechanism that

- reserves memory for NFVIS Infrastructure and previously deployed VMs before allowing new VM deployment
- ensures sufficient memory availability through validation checks, and
- prevents memory oversubscription for VMs.

How memory is allocated

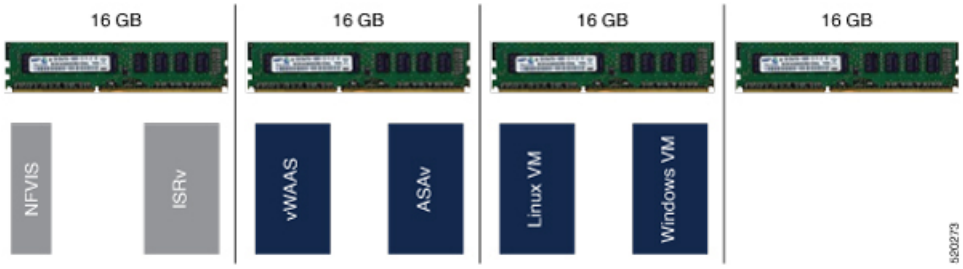
Summary

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.

Workflow

VMs are not allowed to directly access the host file system and storage.

Process details



Interface isolation

Interface isolation is a virtualization capability that

- allows the isolation of PCI Express (PCIe) resources such as an Ethernet port using Single Root I/O Virtualization (SR-IOV),
- enables a single Ethernet port to appear as multiple, separate, physical devices known as Virtual Functions, and
- provides data protection between guests on the same physical server as the data is managed and controlled by the hardware.

SR-IOV implementation details

Objective

All Virtual Function devices on an adapter share the same physical network port. A guest can use one or more of these Virtual Functions, with each Virtual Function appearing to the guest as a network card, in the same way as a normal network card would appear to an operating system.

Virtual Function performance and usage

Virtual Functions provide the following performance characteristics:

- Near-native performance
- Better performance than para-virtualized drivers and emulated access

NFVIS VNFs can use SR-IOV networks to connect to WAN and LAN Backplane ports. Each VM owns a virtual interface and its related resources achieving data protection among VMs.

Secure development lifecycle

A secure development lifecycle is a software development process that

- reduces vulnerabilities and enhances the security and resilience of Cisco solutions
- applies industry-leading practices and technology to build trustworthy solutions, and
- results in fewer field-discovered product security incidents.

Attack vector reduction

Attack vector reduction is a security strategy that

- limits software packages to only those essential for NFVIS functionality to reduce potential security vulnerabilities
- registers all third-party software in a central Cisco database for organized security response, and
- ensures periodic patching of software packages for known Common Vulnerabilities and Exposures (CVEs) in every release.

This approach provides multiple security advantages. 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.

NFVIS SDL processes

Objective

Every NFVIS release goes through these processes:

NFVIS security development lifecycle

- 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
- Using Application Security tools such as IBM AppScan, Nessus, and other Cisco internal tools.

Public sector security certifications

Objective

This reference groups information about Public Sector Security Certifications, including FIPS 140-3.

FIPS 140-3

FIPS mode on NFVIS is a security compliance mechanism that

- implements Federal Information Processing Standards (FIPS) Publication 140-3 for United States federal government and contractor use
- attempts to prevent the use of non-FIPS compatible algorithms on the device, and
- requires manual configuration to ensure only FIPS approved algorithms are used.

How FIPS mode works

Summary

In FIPS mode, certain functions may fail silently if they attempt to use non-compliant algorithms. You must ensure that the device is configured to use only FIPS-approved algorithms. FIPS mode is enabled by default on BExK platforms.

Workflow

In FIPS mode, certain functions may fail silently if they attempt to use non-compliant algorithms.

Disable FIPS mode

To disable FIPS mode, first perform a factory reset. The same sequence applies when you re-enable FIPS mode.

Procedure

Step 1 Reset NFVIS to factory default.

For the reset procedure, see Reset NFVIS to factory default.

Step 2 After the reset, disable FIPS mode and reboot the device.

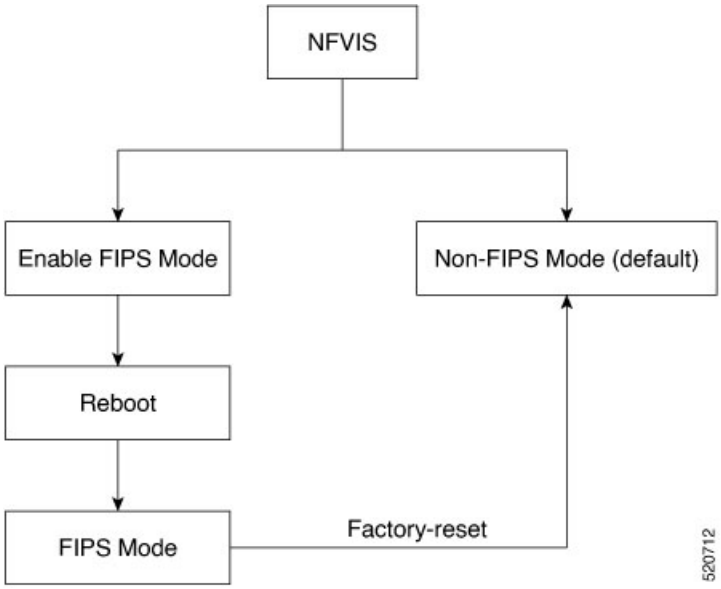
Configure FIPS mode

Before you begin

Before you configure FIPS mode, ensure that SNMP v1, v2, or v3 with the MD5 authentication protocol is not configured. FIPS mode configuration is terminated if any of these are configured.

NFVIS supports FIPS and non-FIPS modes. FIPS mode enables FIPS 140-3-compliant operation for SSH, TLS, and SNMP protocols.

The diagram shows the FIPS mode transition.



Procedure

Step 1 Enable FIPS mode.

Enter the following commands:

```
config terminal
security fips
commit
```

FIPS mode is enabled only after NFVIS reboots. Before reboot, verify that FIPS mode is configured and pending enablement:

```
nfvis# show security
security fips-status CONFIGURED_REBOOT_TO_ENABLE
```

Note FIPS mode can be disabled only with a factory reset. The **no** form of the command cannot disable FIPS mode:

```
config terminal
no security fips
commit
Aborted: This command can only be removed while factory reset
```

Step 2 Reboot NFVIS to enable FIPS mode.

Step 3 Verify the FIPS mode status after reboot.

Use the following command:

```
nfvis# show security
security fips-status ENABLED
```

FIPS mode configuration is terminated when SNMP v1 or v2 is configured, or when SNMP v3 is configured with MD5 authentication.

The following example shows the error returned 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
```

The following example shows the error returned when SNMP v3 is configured with MD5 authentication:

```
config terminal
security fips
commit
Aborted: SNMP version 3 MD5 auth-protocol configured other secure protocol and try again
```

Note After FIPS mode is enabled, you cannot configure SNMP v1 or v2, or SNMP v3 with MD5 authentication. The following examples show the errors returned for these configurations:

```
snmp group test_v1 snmp 1 noAuthNoPriv read test write test
commit
Aborted: Cannot configure SNMP group-version 1 because fips-status is ENABLED
config terminal
snmp user test_md5_v3
user-version 3
user-group test_v3
auth-protocol md5
auth-key 46:97:c3:b0:ba:45:fd:5e:be:99:44:c5:64:c9:bc:44
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
nfvis(config-user-test_md5_v3_passhd)#
```

FIPS mode is enabled, and NFVIS operates in FIPS 140-3-compliant mode for SSH, TLS, and SNMP protocols.

How backup and restore work in FIPS mode

Summary

This topic provides the details on how FIPS mode status is handled during NFVIS backup and restore operations.

Workflow

If you back up NFVIS configurations when FIPS mode is enabled, then upon restore, FIPS mode is configured but needs a manual reboot to enable it.

Process details

Backup configuration

```
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

After restore

```
nfvis#
nfvis# show running-config security
security fips
nfvis# show security fips-status
security fips-status CONFIGURED_REBOOT_TO_ENABLE
nfvis#
```

After reboot

```
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

```
Backup configurations
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#
```

Restore system configurations

```
Restore system configurations
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#
```

After restore

```
After restore
nfvis# show running-config security
% No entries found.
nfvis# show security
```

```
security fips-status UNCONFIGURED_REBOOT_TO_DISABLE
nfvis#
```

After reboot

```
After reboot
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#
```

FIPS operational status

Objective

This reference lists the operational states available when you enable FIPS mode and explains the possible state transitions.

FIPS operational states

These are the operational states when you try to ENABLE FIPS mode:

- DISABLED
- CONFIGURED-REBOOT-TO-ENABLE
- ENABLED
- UNCONFIGURED-REBOOT-TO-DISABLE
- FAILED

From	To	Description
DISABLED	CONFIGURED-REBOOT-TO-ENABLE	If the Oper data of FIPS-state leafs was previously set to DISABLED and if the security FIPS configuration is pushed
DISABLED	FAILED	If there is an error while pushing the security FIPS configuration
CONFIGURED-REBOOT-TO-ENABLE	ENABLED	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
CONFIGURED-REBOOT-TO-ENABLE	FAILED	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

From	To	Description
CONFIGURED-REBOOT-TO-ENABLE	DISABLED	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
ENABLED	DISABLED	After factory-reset (of any type)
ENABLED	FAILED	If there is an error while disabling the FIPS mode
ENABLED	UNCONFIGURED-REBOOT-TO-DISABLE	If the FIPS-mode is UNCONFIGURED while restoring from a backup package and the current FIPS-status is ENABLED
FAILED	CONFIGURED-REBOOT-TO-ENABLE	If the Oper date of FIPS-state leafs was previously set to FAILED and now configuring FIPS-mode
FAILED	DISABLED	If the Oper date of FIPS-state leafs was previously set to FAILED and now issued factory-reset
UNCONFIGURED-REBOOT-TO-DISABLE	DISABLED	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and then NFVIS is rebooted
UNCONFIGURED-REBOOT-TO-DISABLE	CONFIGURED-REBOOT-TO-ENABLE	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and FIPS-mode config is successful
UNCONFIGURED-REBOOT-TO-DISABLE	FAILED	If the Oper data of FIPS-state leafs was previously set to UNCONFIGURED-REBOOT-TO-DISABLE and if the FIPS-mode config was unsuccessful