Cisco Network Function Virtualization Infrastructure Software Configuration Guide, Release 26.2 and Later

First Published: 2026-06-29

Last Modified: 2026-06-29

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

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.

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.

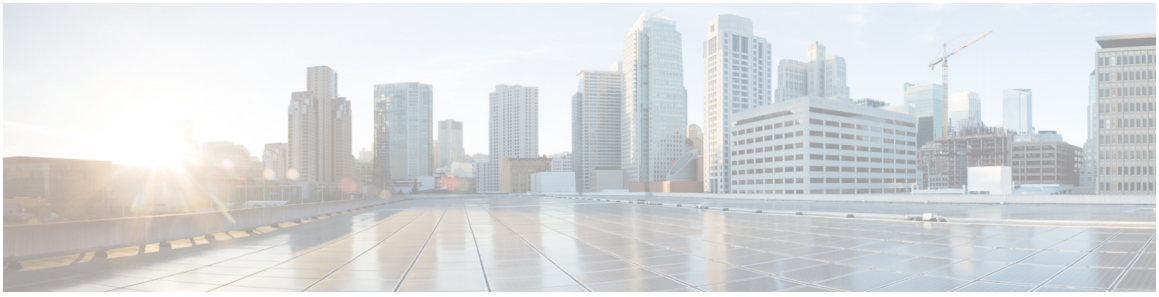
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY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

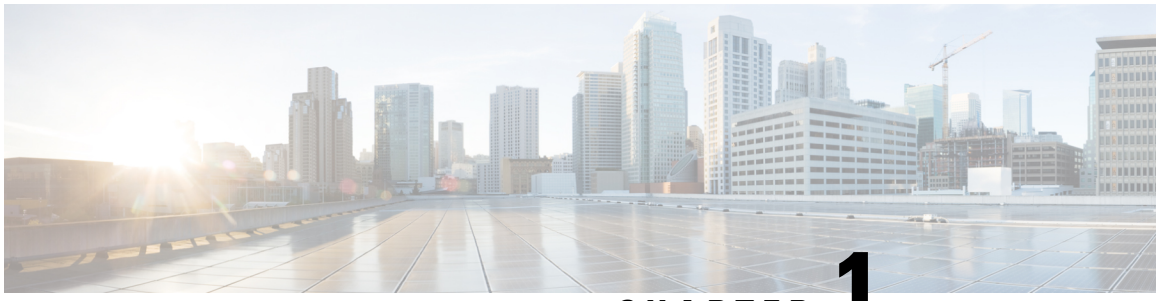
All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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



CHAPTER 1

About NFVIS

Cisco NFVIS

Cisco Network Function Virtualization Infrastructure Software (Cisco NFVIS) is a Linux-KVM based infrastructure software that

- helps service providers and enterprises design, deploy and manage network services, and
- dynamically deploys virtualized network functions on supported Cisco devices.

Benefits of Cisco NFVIS

The Cisco NFVIS platform provides several key benefits for network function virtualization and service deployment.

- Consolidates multiple physical network appliances into a single server running multiple VMs.
- Deploys services quickly and in a timely manner.
- VM life cycle management and provisioning.
- Life cycle management to deploy and chain VMs dynamically on the platform.
- Programmable APIs configured through Netconf interface, REST APIs and command-line interface as all the configurations are exposed through YANG models.

Supported hardware platforms

This reference provides information about the hardware platforms supported by Cisco NFVIS, specifically focusing on Cisco UCS Rack Servers across different generations.

Table 1: Supported PIDs

Server Model	Minimum Supported Release	Supported PIDs
Cisco UCS C M6 Rack Server	Cisco NFVIS Release 4.18.2a	• BE6K-M6-XU • BE6K-M6-K9 • BE7M-M6-K9 • BE7M-M6-XU • BE7H-M6-K9 • BE7H-M6-XU
Cisco UCS C M5 Rack Server	Cisco NFVIS Release 4.18.2a	• BE6M-M5-K9 • BE6M-M5-XU • BE6H-M5-K9 • BE6H-M5-XU • BE7M-M5-K9 • BE7M-M5-XU • BE7H-M5-K9 • BE7H-M5-XU
Cisco UCS C M7 Rack Server	Cisco NFVIS Release 4.18.2a	• BE6K-M7-K9 • BE6K-M7-XU • BE7M-M7-K9 • BE7M-M7-XU • BE7H-M7-K9 • BE7H-M7-XU • CE1400V-M7-K9
Cisco UCS C M8	Cisco NFVIS Release 26.2.1	• BE6K-M8-K9 • BE6K-M8-XU • BE7M-M8-K9 • BE7M-M8-XU • BE7H-M8-K9 • BE7H-M8-XU

Note For UCS C-Series M5 and M6 servers, if the rack servers have LOM (LAN on Motherboard) ports, the WAN bridge will be created on port GE0-0, which is a LOM port, and GE0-1 will be the LAN bridge on a LOM port. For UCS C-Series M7 Rack Servers, however, there are no LOM ports. Hence, the first slot in the chassis with a NIC inserted will have the WAN bridge and the LAN bridge.

Supported VMs

This reference lists the virtual machines that are supported by the system, including various Cisco Unified Communications applications and services.

Cisco Unified Communications Manager VMs

- Cisco Unified CM (**UCM**)
- Cisco Unified CM Session Management Edition (**SME**)
- Cisco IM and Presence (**IMP**)
- Cisco Unity Connection (**CUC**)
- Cisco Emergency Responder (**CER**)
- Cisco Expressway (X15.4)
- Webex Calling Dedicated Instance Enhanced Survivability Node (**DI ESN**)

Meetings VMs

- Video Mesh (VMN)
- Hybrid Data Security (HDS)
- Cisco Meeting Server v4.x (CMS)
- Cisco Meeting Management v4.x (CMM)

Contact Center VMs

- Unified Contact Center Express (UCCX)
- Customer Collaboration Platform (CCP)

Key tasks you can perform using Cisco NFVIS

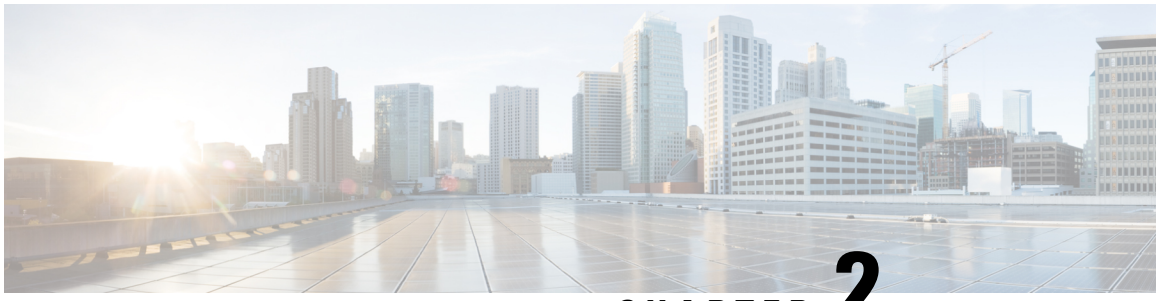
This reference describes the key tasks that you can perform using Cisco NFVIS for VM management, network operations, and system monitoring.

Cisco NFVIS supports these key operational tasks:

- Perform VM image registration and deployment

- Create new networks and bridges, and assign ports to bridges
- Perform service chaining of VMs
- Perform VM operations
- Verify system information including CPU, port, memory, and disk statistics

The APIs for performing these tasks are explained in the API Reference for Cisco Enterprise NFVIS.



CHAPTER 2

Install Cisco NFVIS

Install NFVIS on BEx7K/CE1400V appliances based out of UCS-C series servers

This task installs Cisco NFVIS on BEx7K/CE1400V appliances based out of UCS-C series servers by configuring boot order, mapping virtual media, and completing the installation process through CIMC.

UCS-C series devices has to configure RAID disk group before installing NFVIS. UCS-C supports only single RAID disk group for fresh installation.

Procedure

Step 1 Log in to CIMC.

The recommended CIMC version for UCS-C Series Servers and Cisco CSP platforms is 3.0(3c) or later version.

The recommended CIMC version for Cisco UCS-C Series Rack Servers is 4.3(2) or later versions.

Step 2 To launch KVM Console, Select **Launch KVM** from the CIMC homepage.

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

Step 3 To map virtual devices from the KVM Console:

a) 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. For more information see, Virtual Machine Security.

b) Select **Virtual Media** and then **Activate Virtual Devices**.

c) Select **Virtual Media** again and then **Map CD/DVD**. Browse and select the Cisco NFVIS ISO image. Click **Open** and Map Drive to mount the image.

d) Select **Virtual Media** again to ensure the NFVIS ISO image is now mapped to CD/DVD.

Step 4 To configure boot order:

a) From the **CIMC Compute**, select **BIOS**.

b) Select **Configure Boot Order** and the **Configure Boot Order** dialog box appears.
c) Select **Advanced**.
d) The **Add Boot Device** page appears. Select **Add Virtual Media**, and the **Add Virtual Media** dialog box appears.
e) Enter a name and select **KVM Mapped DVD**. Set state to **Enabled** and order as 1, and **Save Changes**.
f) The **Add Boot Device** page appears again, select **Add Local HDD**, and **Add Virtual Media** dialog box appears.
g) Enter a name, set state to **Enabled** and order as 2, and **Save Changes**.
h) Click **Close**.

Step 5 Power cycle server to start the installation:

From CIMC homepage, select **Host Power**. Reboot the server by selecting the **Power Off** option. 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 6 After the installation is complete, the system automatically reboots from the hard drive. Log into the system when the command prompt **NFVIS login** is displayed after the reboot.

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 7 Verify the installation using the System API, CLI, or by viewing the system information from the Cisco NFVIS portal.

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

Cisco NFVIS is successfully installed on the BEx7K/CE1400V appliances based out of UCS-C series servers and ready for configuration and virtual machine deployment.

Default system configuration

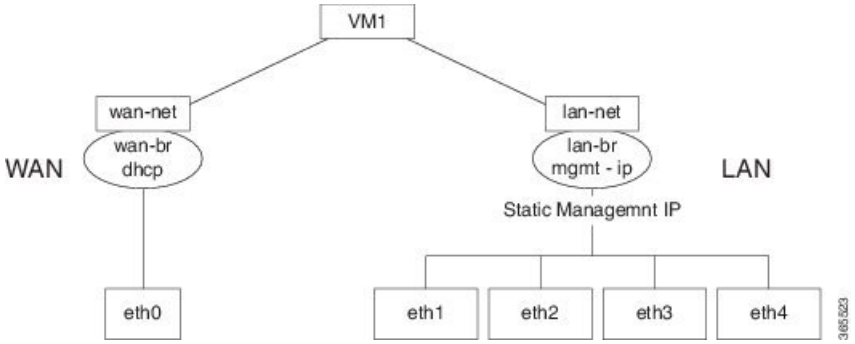
Default system configuration on the Cisco BEx7K/CE1400V appliances based out of UCS-C series servers is a network setup that

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

Network configuration details

This diagram illustrates the default network configuration:

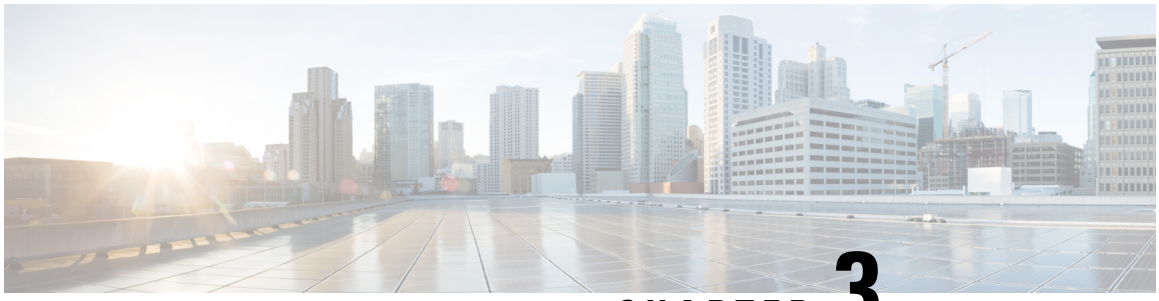
Figure 1: Default network configuration



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.



CHAPTER 3

Licensing Requirement for NFVIS Support

Prerequisites for Cisco BEx7K/CE1400V appliances based out of UCS-C series servers

This reference identifies the essential prerequisites that must be met before implementing Cisco NFVIS support for Cisco BEx7K/CE1400V appliances based out of UCS-C series servers.

- 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 BEx7K/CE1400V appliances based out of UCS-C series servers along with Cisco NFVIS Smart License.

Cisco smart licensing for Cisco BEx7K/CE1400V appliances based out of UCS-C series servers

Starting with Cisco NFVIS Release 26.2.1, Cisco Smart Licensing capability is added.

Cisco Smart Licensing is a flexible and unified licensing model that

- centralizes the management of all Cisco software licenses across organizations
- enables you to manage all product licenses from one central location, and
- allows licenses to be easily transferred between devices without hosting or issuing the license again or going through Return Material Authorization (RMA) or license re-issuance.

Smart licensing characteristics

The Cisco NFVIS smart licenses have these characteristics:

- Not node-locked to the device, so they can be used across all devices in your organization
- Can be ordered together with any Cisco BExK / CE1400V appliances through the Cisco Commerce Workspace (CCW) page
- Appear as **Licenses Available to Use** in your Virtual Account on the Cisco Smart Software Manager (CSSM) once purchased

For more information on Smart Licenses see, Cisco Software Licensing Guide.

Configure Cisco smart licensing for Cisco NFVIS

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 BEx7K/CE1400V appliances based out of UCS-C series servers 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

The Cisco BEx7K/CE1400V appliances based out of UCS-C series servers requires these inputs to operate in smart mode:

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

Note 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?

2. Provide the generated token as the input to the Cisco BEx7K/CE1400V appliances based out of UCS-C series servers. 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.
3. 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.

4. 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.

5. 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.

Use these methods to provide inputs to the Cisco BEx7K/CE1400V appliances based out of UCS-C series servers using Cisco NFVIS:

- Using the Cisco NFVIS CLI
- Using NETCONF/Rest APIs

When the device is 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

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

Verify smart mode using the CLI

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

```
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.

CSLU mode characteristics

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

CSLU mode has these characteristics:

- Ensure that you've 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 doesn't 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 aren't 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 backups the config to a file and sends the forced License Usage report to CSSM .

Monitor Cisco NFVIS support for Cisco BEx7K/CE1400V appliances based out of UCS-C series servers

This reference provides show commands used to monitor Cisco NFVIS Smart Licensing for Cisco BEx7K/CE1400V appliances based out of UCS-C series servers along with some configuration examples.

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
"$8$mbXijw3R0wmUU/A2JwDctH8aLwRF6ov+PScUXqpZJqINICFQCYnujpbJpYki8O3OWpjtW\nqS8FU21jisfd49mVF0qSS+F92V8/uD3iUowuGgjrWQ4POcZb770ojeAw+bD
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 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 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 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.

Action commands

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

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.

Enforcement behavior for smart licensing

Enforcement behavior for smart licensing is a Day-N behavior model that

- conditionally enforces Smart Licensing Using Policy in NFVIS
- transitions devices between states based on trust establishment and synchronization status, and
- maintains licensing compliance through operational recovery commands and state persistence.

Smart licensing states and transitions

When trust is established and account mapping is successful, the device transitions to AUTHORIZED state and reports the consumed license units to CSSM. In this state, usage and reporting continue at the configured intervals with no service-impacting enforcement action.

If trust is not established (for example after certificate loss, account/VA movement, or stale token/config mismatch), the device can show a PENDING state and should be recovered using **license smart trust** and/or **license smart sync**.

If synchronization is overdue beyond policy limits, the device can move to out-of-compliance related states and follow policy retry/overdue intervals.

Operationally, use these commands for lifecycle recovery and maintenance:

- **license smart trust** to re-establish trust certificates with CSSM.
- **license smart sync** to push policy/usage synchronization when state mismatch is observed.
- **license smart release** to release entitlement usage from CSSM before decommissioning or transfer workflows.

For steady state validation, check that:

- Licensing status is AUTHORIZED.
- Trust Code Installed has a valid timestamp.
- Last Attempt Status is success.
- Smart Account and Virtual Account are populated.

NFVIS persists Smart Licensing enforcement state in the **/data/cisco/licensing/state** file.

This file is JSON and is used to track enforcement-related runtime state across service restarts and upgrades.

The current state keys and meaning are:

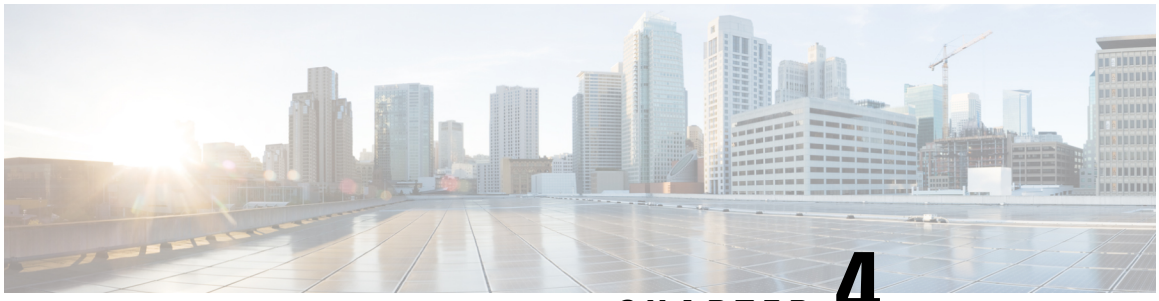
- **expired**: Boolean. True when grace has expired and out-of-compliance must be enforced.
- **grace**: Boolean. True when the device is currently in grace period.
- **overdue**: Boolean. True when synchronization is overdue (sync overdue event).
- **grace_expiry_ts**: UNIX timestamp (seconds). Grace period end time used to decide whether OOC is still within grace window.
- **grace_reason**: String. Reason text captured from usage-specific out-of-compliance information.

Default state when file is absent:

{"expired": false, "grace": false, "overdue": false, "grace_expiry_ts": 0, "grace_reason": ""}

State transitions used by NFVIS Smart Licensing workflow:

- **in_compliance event**: resets overdue/expired/grace and clears grace_expiry_ts and grace_reason.
- **grace_period event**: sets grace=true, expired=false, and updates grace_expiry_ts/grace_reason when available.
- **out_of_compliance event**: if grace window has expired, sets expired=true and clears grace metadata; if still in grace window, keeps grace=true.
- **sync_overdue event**: sets overdue=true.



CHAPTER 4

VM Life Cycle Management

VM life cycle management

VM life cycle management is a process that

- 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.

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).

Perform resource verification

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

```
nfvis# show resources cpu-info
resources cpu-info allocation total-sockets 2
resources cpu-info allocation cores-per-socket 20
```

```
resources cpu-info allocation total-logical-cpus 80
resources cpu-info allocation logical-cpus-used-by-system 4
resources cpu-info allocation logical-cpus-used-by-vnfs 4
resources cpu-info allocation logical-cpus-used-dedicated 0
resources cpu-info allocation logical-cpus-used-sharable 4
CPU  SOCKET  CORE  SYSTEM                LOW      VCPU
ID   ID      ID    USE     NAME     VCPUS  LATENCY  ID
-----------------------------------------------------------
0    0       0     true
20   1       20    true
40   0       0     true
60   1       20    true
1    0       1     false
41   0       1     false
2    0       2     false
42   0       2     false
3    0       3     false
43   0       3     false
4    0       4     false
44   0       4     false
5    0       5     false
45   0       5     false
6    0       6     false
46   0       6     false
7    0       7     false
47   0       7     false
8    0       8     false
48   0       8     false
9    0       9     false
49   0       9     false
10   0       10    false
50   0       10    false
11   0       11    false
51   0       11    false
12   0       12    false
52   0       12    false
13   0       13    false
53   0       13    false
14   0       14    false
54   0       14    false
15   0       15    false
55   0       15    false
16   0       16    false   OTHER76  1      false    0
56   0       16    false
17   0       17    false   OTHER99  1      false    0
57   0       17    false
18   0       18    false   OTHER57  1      false    0
58   0       18    false
19   0       19    false   OTHER95  1      false    0
```

```
nfvis# show resources cpu-info allocation
resources cpu-info allocation total-sockets 2
resources cpu-info allocation cores-per-socket 20
resources cpu-info allocation total-logical-cpus 80
resources cpu-info allocation logical-cpus-used-by-system 4
resources cpu-info allocation logical-cpus-used-by-vnfs 4
resources cpu-info allocation logical-cpus-used-dedicated 0
resources cpu-info allocation logical-cpus-used-sharable 4
```

```
nfvis# show resources cpu-info vnfs
                LOW      VCPU  SOCKET  CORE  CPU
NAME     VCPUS  LATENCY  ID    ID      ID    ID
-------------------------------------------------
OTHER95  1      false    0     0       19    19
OTHER57  1      false    0     0       18    18
```

```
OTHER99  1      false    0     0       17    17
OTHER76  1      false    0     0       16    16
```

CPU Over-Subscription

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

Configure management IP subnet

Change the default subnet for all VMs with management interfaces from the default 10.20.0.1 subnet.

By default, all VMs with management interfaces will be provisioned with an IP address in the subnet of 10.20.0.1. You can change the default subnet by executing commands in a sequence to first delete the existing subnet and then add a new subnet in the network.

Before you begin

Ensure there are no managed VNFs in the system before you change the management network address.

Follow these steps to configure the management IP subnet:

Procedure

Step 1 To delete the existing subnet, use the **no vm_lifecycle networks network int-mgmt-net subnet int-mgmt-net-subnet** command.

Step 2 To create a new subnet, enter the following commands:

Example:

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

The management IP subnet is configured with the new subnet settings.

Workflow of vm life cycle management

VM life cycle management using REST APIs provides a structured workflow for creating, customizing, deploying, and managing virtual machines in the NFVIS environment.

Note Before performing the VM life cycle management tasks, you will have to upload the VM images to the NFVIS server or http/s, sftp, scp server.

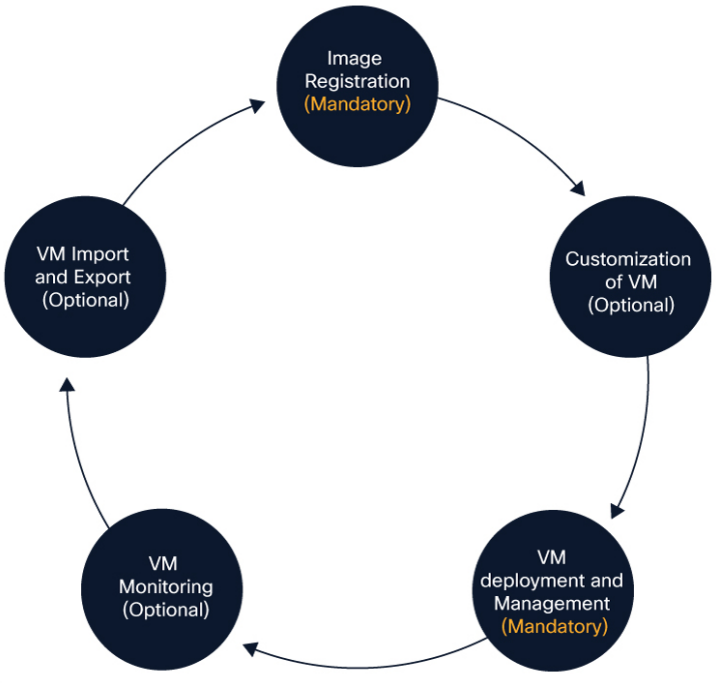
Summary

The key components involved in the VM life cycle management process are:

- VM Image: The source image that serves as the template for creating VM instances
- NFVIS Image Repository: The storage location where registered VM images are maintained
- Remote Servers: HTTP/HTTPS, FTP, or SCP servers that host the source images
- Custom Profiles/Flavors: Specific configurations for VM resources such as CPU and memory
- Deployment APIs: Interfaces that enable VM deployment and parameter configuration
- Management APIs: Tools for monitoring, controlling, and updating deployed VMs

Workflow

Figure 2: VM life cycle management



These stages describe how VM life cycle management works:

1. **Register a VM Image—**To create a VM instance, you must register the source image in the NFVIS image repository. Registering the image is a one-time activity. The source image is expected to be hosted on a remote server. In the Registration API you specify the URL on the remote server, where the file is located. When invoked, device pulls the API info, completes the file transfer and registration. Once registered, you can create one or more VM instances. Remote servers supported for hosting the source images (.tar.gz files) are HTTP/HTTPS servers. Starting with Cisco NFVIS release 4.12.1, remote servers can also be FTP servers or SCP servers.
2. **Customizing the VM—**After registering a VM image, you can optionally create a custom profile or flavor for the VM image if the profiles defined in the image file do not match your requirement. The flavor creation option lets you provide specific profiling details for a VM image, such as the virtual CPU on which the VM will run, and the amount of virtual memory the VM will consume.

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

3. **Deploy a VM—** A VM can be deployed using the deployment API. The deployment API allows you to provide values to the parameters that are passed to the system during deployment. Depending on the VM you are deploying, some parameters are mandatory and others optional.
4. **Manage and Monitor a VM—**You can monitor a VM using APIs and commands that enable you to get the VM status and debug logs. Using VM management APIs, you can start, stop, or reboot a VM, and view statistics for a VM such as CPU usage.

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

   The vNICs on a deployed VM can also be added or updated.

Image Registration

Image registration is a process that

- copies or downloads VM images to the NFVIS server or hosts them on http, https, FTP, or SCP servers
- uses the registration API to specify the file path to the location where the tar.gz file is hosted, and
- enables multiple VM deployments using the registered image once it is in active state.

Image registration characteristics

Image registration is a one-time activity. Once an image is registered on the http or https server and is in active state, you can perform multiple VM deployments using the registered image. All VM images are available in VM packaging and VM package content.

Register images for CUCM applications

Register both the ISO image (installation media) and the OVA file (metadata) with NFVIS for collaboration applications such as Cisco Unified Communications Manager (CUCM), Cisco Unity Connection, and Cisco IM and Presence Service (IM&P).

Collaboration applications require both ISO images and OVA metadata to be properly registered with NFVIS before deployment. The registration process automatically creates profiles and flavors from the OVA metadata.

Procedure

Step 1 Navigate to the Image Repository.

Access the image repository via **Configuration** > **Virtual Machine** > **Images** > **Image Repository**

Step 2 Upload the ISO image.

a. Click **Upload**.

b. Select the ISO file from your local file system.

Example: Bootable_UCSInstall_UCOS_15.0.1.ISO

Step 3 Configure image properties.

a. Update the pre-populated image name to a descriptive name.

Example: CUCM-15.0.1

b. Select the appropriate **VM Type** from the drop-down list. If your application type is not listed, select **Other**.

Step 4 Upload the OVA metadata.

a. Select the **Metadata** check box.

b. Click **Upload OVA** and select the OVA file.

Example: cucm_15.0_vmv17_v1.2.sha512.OVA

Step 5 Configure additional options.

Select the **Dedicated Cores** check box to allocate dedicated CPU cores for better performance.

Step 6 Submit the image registration request.

Click **Upload File** to submit the request.

NFVIS uploads the ISO, uploads and parse the OVA, register the image, and automatically create profiles and flavors from the OVA metadata.

Step 7 Verify successful registration.

a. Wait for the registration to complete.

Confirm the image appears in the Image Repository with an **Active** status.

b. Verify that flavors are populated from the OVA file.

The ISO image and OVA metadata are successfully registered with NFVIS, and profiles and flavors are automatically created from the OVA metadata. The image appears in the Image Repository with an Active status.

Remote server configuration

Here is a sample configuration to configure a remote server:

```
<remote_servers>
    <remote_server>
        <name>myserver</name>
        <base_url>ENTER_IP_ADDRESS</base_url>
        <username>ENTER_USERNAME</username>
        <password>ENTER_PASSWORD_HERE</password>
    </remote_server>
</remote_servers>
```

Register images using the SCP server

Here is a sample configuration to register images using the SCP server:

```
<images>
    <image>
```

```
        <name>C8Kv_IMAGE</name>
       <src>scp://myserver/nfvis/c8kv.tar.gz>  <============ Specify the remote server
 name created in device
        <remove_src_on_completion>
true</remove_src_on_completion>
    </image>
</images>
```

Register images using the HTTP/HTTPS server

Here is a sample configuration to register images using the HTTP/HTTPS server:

```
<images>
    <image>
        <name>C8Kv_IMAGE</name>
        <src>http://myserver/nfvis/c8kv.tar.gz>    <============ Specify the remote
server name created in device
        <remove_src_on_completion>
true</remove_src_on_completion>
    </image>
</images>
```

Register VM packages using REST API

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

Post image registration

```
curl -v -u -k admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml
-X POST https://209.165.201.1 /api/config/vm_lifecycle/images -d
'<image><name>WinServer2012R2.iso</name><src>file:///data/intdatastore/uploads/WinServer2012R2.iso/WinServer2012R2.iso</src></image>'

HTTP/1.1 201 Created
```

Get image status

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/operational/vm_lifecycle/opdata/images/image/isrv-03.16.02?deep

HTTP/1.1 200 OK
<image xmlns="http://www.cisco.com/nfvis/vm_lifecycle" xmlns:y="http://tail-f.com/ns/rest"
xmlns:esc="http://www.cisco.com/nfvis/vm_lifecycle">
<name>isrv.03.16.02</name>
<image_id>585a1792-145c-4946-9929-e040d3002a59</image_id>
<public>true</public>
<state>IMAGE_ACTIVE_STATE</state></image>
```

Get registered image status

```
Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/config/vm_lifecycle/images?deep

HTTP/1.1 200 OK
<images xmlns="[http://www.cisco.com/esc/esc|http://www.cisco.com/nfvis/vm_lifecycle]"
xmlns:y="[http://tail-f.com/ns/rest|http://tail-f.com/ns/rest]" 
xmlns:esc="[http://www.cisco.com/nfvis/vm_lifecycle|http://www.cisco.com/nfvis/vm_lifecycle]">
<image>
<name>isrv-9.16.03.01</name>
```

```
<src>http://data/nfvos-pkg/isr/isrv-universalk9.16.03.01.tar.gz</src>
</image>
</images>
```

Delete registered image

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
DELETE https://209.165.201.1/api/config/vm_lifecycle/images/image/isrv-3.16.0.1a

HTTP/1.1 204 No Content
```

For more information on REST APIs related to image registration, see API Reference for Cisco Enterprise NFVIS.

Register VM image with multiple root disks

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

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

```
<image_properties>
 …
 <root_file_disk_bus>virtio</root_file_disk_bus>
 <root_image_disk_format>qcow2</root_image_disk_format>
 <disk_1_file_disk_bus>virtio</ disk_1_file_disk_bus>
 <disk_1_image_format>qcow2</ disk_1_image_format>
 <disk_2_file_disk_bus>virtio</ disk_2_file_disk_bus>
 <disk_2_image_format>qcow2</ disk_2_image_format>
 …
</image_properties>
```

Note Image profiles apply to the first root disk only. The size of the remaining disks remain the same when deployed through different profiles with an exception to vWAAS VNF.

Register a VM image through a root disk

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

Default properties for root disk registration

Property Name	Property Tag	Default Value
Version	<version>	NA
VNF Type	<vnf_typ>	OTHER
VCPU Min	<vcpu_min>	1
VCPU Max	<vcpu_max>	64
Memory Min (MB)	<memory_mb_min>	256
Memory Max (MB)	<memory_mb_max>	1048576
Root Disk Min Size (GB)	<root_disk_gb_min>	1

Property Name	Property Tag	Default Value
Root Disk Max Size (GB)	<root_disk_gb_max>	10240
VNIC Max	<vnic_max>	8
Bootup Time	<bootup_time>	-1
Interface Hot Add	<interface_hot_add>	true
Interface Hot Delete	<interface_hot_delete>	false

Register a remote VM image

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

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

Example: POST remote image registration from webserver over HTTPS

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

HTTP/1.1 201 Created
```

Example: POST remote image registration from webserver over HTTPS

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

HTTP/1.1 201 Created
```

Specify storage location for a VM image

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

storage name	directory map
datastore1	/data
datastore2	/mnt/extdatastore1
datastore3	/mnt/extdatastore2
nfs:nfs_mount_name	/data/mount/nfs/nfs_mount_name
nfs or nfs:nfs	/data/mount/nfs/
nfs:nfs_storage or nfs_storage	/data/mount/nfs_storage

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

Example: VM image storage placement

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

HTTP/1.1 201 Created
```

Update VM image

Update specific VM image properties after a VM image has been registered to support interface hot add and delete functionality.

You can only update the following image properties after a VM image has been registered:

- interface_hot_add
- interface_hot_delete

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

Procedure

Step 1 Delete the previously set property value.

Example:

```
curl -k -v -u admin:Esc123# -H
Accept:application/vnd.yang.data+xml -H
Accept:application/vnd.yang.collection+xml -X DELETE
https://172.29.91.28/api/config/vm_lifecycle/images/image/ISR_IMAGE/properties/property/interface_hot_add/value

HTTP/1.1 204 No Content
```

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

Example:

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

HTTP/1.1 201 Created
```

Step 3 Get all properties and values to verify the update.

Example:

```
curl -k -v -u admin:Esc123# -H
Accept:application/vnd.yang.data+xml -H
Accept:application/vnd.yang.data+xml -H Content-
Type:application/vnd.yang.data+xml -X GET
https://172.29.91.28/api/config/vm_lifecycle/images/image/ISR_IMAGE/properties?deep

HTTP/1.1 200 OK
<properties xmlns="http://www.cisco.com/nfvis/vm_lifecycle" xmlns:y="http://tail-f.com/ns/rest"
xmlns:vmlc="http://www.cisco.com/nfvis/vm_lifecycle">
  <property>
    <name>interface_hot_add</name>
    <value>true</value>
  </property>
  <property>
    <name>interface_hot_delete</name>
    <value>false</value>
  </property>
</properties>
```

The VM image properties are successfully updated. The system returns the updated properties and values showing the new configuration.

Image Properties

The `image_properties.xml` file is a mandatory component of a VM image package. It contains the essential property configuration data required for the NFVIS image repository to successfully register and deploy a VM. If any mandatory properties are omitted during the registration process, the image registration will fail.

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

Table 2: Supported Image Properties

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
VNF Type	VM functionality provided. Router and firewall are predefined types.	<vnf_type>	Router, firewall, Windows, Linux, and custom_type	Mandatory
Name	Name associated with the VM packaging. This name is referenced for VM deployment.	<name>	Any	Mandatory
Version	Version of the package	<version>	Any	Mandatory
Boot-up time	Boot-up time (in seconds) of the VNF before it can be reachable via ping.	<bootup_time>	Any in seconds, (-1) to not monitor boot-up	Optional, default -1
Root Disk Image Bus	Root image disk bus	<root_file_disk_bus>	virtio, scsi, and ide	Mandatory
Boot Mode	Specifies the mode in which the VNF will boot. Used for Secure Boot feature	<boot_mode>	efi-secure-boot, bios	Optional, default bios
Shim Signature	If using efi-secure-boot boot mode, a shim signature must be provided	<shim_signature>	microsoft, N/A	Required if Boot Mode is specified
Disk-1 bus type	Additional disk1 image disk bus	<disk_1_file_disk_bus>	virtio, scsi, and ide	Optional

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
Disk-2 bus type	Disk2 image disk bus	<disk_2_file_disk_bus>	virtio, scsi, and ide	Optional
Disk-10 bus type	Disk10 image disk bus	<disk_10_file_disk_bus>	virtio, scsi, and ide	Optional
Root Disk Image format	Root image disk format	<root_image_disk_format>	qcow2 and raw	Mandatory
Disk-1 Image format	Additional disk 1 image format	<disk_1_image_format>	qcow2 and raw	Optional
Disk-2 Image format	Disk 2 image format	<disk_2_image_format>	qcow2 and raw	Optional
Disk-10 Image format	Disk 10 image format	<disk_10_image_format>	qcow2 and raw	Optional
Serial Console	Serial console supported	<console_type_serial>	true, false	Optional
Minimum vCPU	Minimum vCPUs required for a VM operation	<vcpu_min>		Mandatory
Maximum vCPU	Maximum vCPUs supported by a VM	<vcpu_max>		Mandatory
Minimum memory	Minimum memory in MB required for VM operation	<memory_mb_min>		Mandatory
Maximum memory	Maximum memory in MB supported by a VM	<memory_mb_max>		Mandatory
Minimum root disk size	Minimum disk size in GB required for VM operation	<root_disk_gb_min>		Optional
Maximum root disk size	Maximum disk size in GB supported by a VM	<root_disk_gb_max>		Optional

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
Maximum vNICs	Maximum number of vNICs supported by a VM	<vnic_max>		Mandatory
SRIOV support	SRIOV supported by VM interfaces. This should have a list of supported NIC device drivers.	<sriov_supported>	true, false	Optional
SRIOV driver list	List of drivers to enable SRIOV support	< sriov_driver_list>		Optional
PCI passthru support	PCI passthru support by VM interfaces	<pcie_supported>	true, false	Optional
PCIE driver list	List of VNICS to enable PCI passthru support	< pcie _driver_list>		Optional
bootstrap_cloud_init_drive_type	Mounts day0 config file as disk (default is CD-ROM)	<bootstrap_cloud_init_drive_type>	disk, cdrom	Optional
bootstrap_cloud_init_bus_type	Default is IDE	<bootstrap_cloud_init_bus_type>	virtio, ide	Optional

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
BOOTSTRAP	Bootstrap files for the VNF. Two parameters are required in the format of dst:src; dst filename including path has to match exactly to what the VM expects; up to 20 bootstrap files are accepted. For example: --bootstrap ovf-env.xml for ISRv and --bootstrap day0-config for ASAv	< bootstrap_file>	File name of the bootstrap file	Optional
Custom properties	List of properties can be defined within the custom_property tree. (Example: For ISRv, the technology packages are listed in this block.) If the Cisco NFV portal is used to deploy the VM, the portal prompts you for inputs for custom properties fields and can pass the values to the bootstrap configuration.	<custom_property>		Optional

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
Profiles for VM deployment	List of VM deployment profiles. Minimum one profile is required	<profiles>		Optional
Default profile	The default profile is used when no profile is specified during deployment.	<default_profile>		Optional
Monitoring Support	A VM supports monitoring to detect failures.	<monitoring_supported>	true, false	Mandatory
Monitoring Method	A method to monitor a VM. Currently, only ICMP ping is supported.	<monitoring_methods>	ICMPPing	Mandatory if monitoring is true
Low latency	If a VM's low latency (for example, router and firewall) gets dedicated resource (CPU) allocation. Otherwise, shared resources are used.	<low_latency>	true, false	Mandatory
Privileged-VM	Allows special features like promiscuous mode and snooping . By default, it is false.	<privileged_vm>	true, false	Optional
Disable Spoof Check	Used to disable spoof check for Privledged VMs	<disable_spoof_check>	true, false	Optional
Virtual interface model		<virtual_interface_model>		Optional

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
Interface Hot Add	If true, an active VNF's virtual interface can be added/updated without shutting down the VNF.	<interface_hot_add>	true, false	Optional, default true
Interface Hot Delete	If true, an active VNF's virtual interface can be deleted without shutting down the VNF.	<interface_hot_delete>	true, false	Optional, default false
Thick disk provisioning	During deployment, VM will be a fully allocated root disk with size specified by flavor.	<thick_disk_provisioning>	true, false	Optional, default false
Eager Zero	Used in conjunction with Thick disk provisioning. During deployment, root disk is zeroed out to improve I/O operations	<eager_zero>	true, false	Optional, only valid if Thick disk provisioning is enabled. Default false
Profile for VM deployment	A profile defines the resources required for VM deployment. This profile is referenced during VM deployment.	<profile>		Optional
Name	Profile name	<name>	Any	Mandatory
Description	Description of the profile	<description>	Any	Mandatory

Property Name	Description	Property Tag	Possible Values	Mandatory/ Optional
vCPU	vCPU number in a profile	<vcpus>		Mandatory
Memory	Memory - MB in profile	<memory_mb>		Mandatory
Root Disk Size	Disk size - MB in profile .	<root_disk_mb>		Mandatory
VNIC Offload	List of properties that can be set for vnic offload	<vnic_offload>		Optional
Generic Segmentation Offload	Turn generic segmentation offload on or off	<generic_segmentation_offload> (parent: <vnic_offload>)	on, off	Optional
Generic Receive Offload	Turn generic receive offload on or off	<generic_receive_offload> (parent: <vnic_offload>)	on, off	Optional
RX Checksumming	Turn RX checksumming on or off	<rx_checksumming> (parent: <vnic_offload>)	on, off	Optional
TX Checksumming	Turn TX checksumming on or off	<tx_checksumming> (parent: <vnic_offload>)	on, off	Optional
TCP Segmentation Offload	Turn TCP segmentation offload on or off	<tcp_segmentation_offload> (parent: <vnic_offload>)	on, off	Optional

Contents of an image_properties.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<image_properties>
    <vnf_type>ROUTER</vnf_type>
    <name>ISRV</name>
    <version>16.06.05</version>
    <bootup_time>600</bootup_time>
    <root_file_disk_bus>virtio</root_file_disk_bus>
    <root_image_disk_format>qcow2</root_image_disk_format>
    <vcpu_min>1</vcpu_min>
    <vcpu_max>8</vcpu_max>
    <memory_mb_min>4096</memory_mb_min>
    <memory_mb_max>8192</memory_mb_max>
    <vnic_max>8</vnic_max>
    <vnic_names>vnics:1:GigabitEthernet2</vnic_names>
    <vnic_names>vnics:2:GigabitEthernet3</vnic_names>
    <vnic_names>vnics:3:GigabitEthernet4</vnic_names>
    <vnic_names>vnics:4:GigabitEthernet5</vnic_names>
```

```
    <vnic_names>vnics:5:GigabitEthernet6</vnic_names>
    <vnic_names>vnics:6:GigabitEthernet7</vnic_names>
    <vnic_names>vnics:7:GigabitEthernet8</vnic_names>
    <root_disk_gb_min>8</root_disk_gb_min>
    <root_disk_gb_max>8</root_disk_gb_max>
    <console_type_serial>true</console_type_serial>
    <sriov_supported>true</sriov_supported>
    <sriov_driver_list>igb</sriov_driver_list>
    <sriov_driver_list>igbvf</sriov_driver_list>
    <sriov_driver_list>i40evf</sriov_driver_list>
    <pcie_supported>true</pcie_supported>
    <pcie_driver_list>igb</pcie_driver_list>
    <pcie_driver_list>igbvf</pcie_driver_list>
    <pcie_driver_list>i40evf</pcie_driver_list>
    <monitoring_supported>true</monitoring_supported>
    <monitoring_methods>ICMPPing</monitoring_methods>
    <low_latency>true</low_latency>
    <privileged_vm>true</privileged_vm>
    <cdrom>true</cdrom>
    <bootstrap_file_1>ovf-env.xml</bootstrap_file_1>
    <bootstrap_file_2>iosxe_config.txt</bootstrap_file_2>
    <custom_property>
        <tech_package>ax</tech_package>
        <tech_package>security</tech_package>
        <tech_package>ipbase</tech_package>
        <tech_package>appx</tech_package>
    </custom_property>
    <custom_property>
        <SSH_USERNAME> </SSH_USERNAME>
    </custom_property>
    <custom_property>
        <SSH_PASSWORD> </SSH_PASSWORD>
    </custom_property>
    <profiles>
        <profile>
    <name>ISRv-mini</name>
            <description>ISRv-mini</description>
    <vcpus>1</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8192</root_disk_mb>
</profile>
        <profile>
    <name>ISRv-small</name>
            <description>ISRv-small</description>
    <vcpus>2</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8192</root_disk_mb>
</profile>
        <profile>
    <name>ISRv-medium</name>
            <description>ISRv-medium</description>
    <vcpus>4</vcpus>
            <memory_mb>4096</memory_mb>
            <root_disk_mb>8192</root_disk_mb>
</profile>
    </profiles>
    <default_profile>ISRv-small</default_profile>
</image_properties>
```

Register a remote 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 3: Protocols

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

Before you begin

Follow these steps to register a remote virtual machine image:

Procedure

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

Step 2 Select the remote registration option:

a. Click **Register Image**

b. 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.

a. Select the Protocol from the drop-down list (HTTPS or SCP).

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

c. 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
Note: Authentication is not supported for HTTP or HTTPS protocols.

Step 6 Upload the OVA metadata.

a. Select the **Metadata** check box.

b. 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

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.

NFVIS downloads the ISO image and OVA file, parses the OVA file, registers the image, and automatically creates profiles and flavors from the metadata.

Step 9 Monitor the registration progress.

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

The VM image is registered, and the system automatically creates the necessary profiles and flavors, making the image available for deployment.

Customization of VM's

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

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

- 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.

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

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.

Procedure

Execute the following cURL command to create a VM profile using the REST API.

Example:

```
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.

Default configuration behavior

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

```
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
 !
!
```

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.

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.

Table 4: Example: Interface attachment to internal management network

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.

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

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

Restrictions for VNF volumes

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

Example: payload for creating volumes

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

Volume storage locations

The following are the accepted values for **storage_location** tags:

- <**storage_location**>*local*</**storage_location**>
- <**storage_location**>*nfs:NFS_MOUNT_NAME*</**storage_location**>
- <**storage_location**>*datastore1*</**storage_location**>
- <**storage_location**>*datastore2*</**storage_location**>
- <**storage_location**>*datastore3*</**storage_location**>

Volume deployment parameters

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

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

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>
  <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 containers 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

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

Volumes 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>
```

For more information, see VNF volumes, on page 42.

VM deployment and management

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

The VM name must meet these 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>'
```

Verify VM deployment

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

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

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

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

VNFs can be deployed using multiple mandatory and option parameters.

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.	

Parameter	Notes	Example
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>**

Parameter	Notes	Example
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>**
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)	

Parameter	Notes	Example
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.	**<interface><nicid>3</nicid><network>wan-net</network><model>virtio</mode**

Parameter	Notes	Example
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

VM bootstrap configuration options with VM deployment are methods that

- 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

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

- Bundle bootstrap configuration files into the VM package: In this method, the bootstrap configuration variables can be assigned tokens. Token names must be in bold text. For each variable with a token assigned, key-value pairs must be provided during deployment in the deployment payload.
- Bootstrap configuration as part of the deployment payload: The entire bootstrap configuration is copied to the payload without tokens.
- Bootstrap configuration file in the NFVIS server: In this method, the configuration file is copied or downloaded to the NFVIS server, and referenced from the deployment payload with the filename, which includes the full path.

For examples on how to use bootstrap configuration options in the deployment payload, see the API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software.

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

See VNF Deployment Parameters for more information on supported placement parameters and their accepted values.

Note If you are placing the VNF deployment on **nfs**, ensure that you have configured this storage option to be mounted on NFVIS using appropriate CLIs before deploying the VNF.

Example: VM deployment using placement

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/vm_lifecycle/tenants
/tenant/admin/deployments --data
'<deployment>
<name>WINIsodep</name>
<vm_group>
  <name>WINIsovmgrp</name>
  <image>WinServer2012R2.iso</image>
  <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>'
```

Deploy VNF using granular RBAC

Deploy Virtual Network Functions (VNFs) using granular role-based access control (RBAC) to ensure proper access management and security.

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. For more information on granular RBAC and resource groups, see Granular role-based access control, on page 103.

Procedure

Step 1 Create a VM with a group name.

Example:

```
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 Update a VM from one resource group to another.

Example:

```
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 list of VMs with their group information.

Example:

```
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 overall group, 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.

Deploy CUCM application

Deploy a CUCM virtual machine on Cisco NFVIS using a Bootstrap ISO for automated Day0 configuration.

This procedure is performed after successfully registering the application image in NFVIS. The deployment creates a functional CUCM virtual machine with automated initial configuration.

Before you begin

- Ensure you have a registered CUCM application image (ISO and OVA) in NFVIS.
- Ensure you have a Bootstrap ISO with Day0 configuration for the CUCM application.

Refer to the CUCM documentation for instructions on creating the Bootstrap ISO with Day0 configuration: *Touchless Installation Task Flow*

Follow these steps to deploy a CUCM virtual machine on Cisco NFVIS with a Bootstrap ISO for Day0 configuration:

Procedure

Step 1 From the NFVIS portal, click **Configuration** > **Deploy**.

Step 2 Select a VM type from the drop-down list.

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

Step 3 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 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 network connections.

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

b. Add additional networks as required.

c. Drag a line from the VM icon to the desired network to connect to the appropriate network(s) for your deployment.

Step 6 Select the **Bootstrap ISO** check box.

Step 7 Click **Upload** and select your Day0 configuration ISO.

Step 8 Click **Deploy**.

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

Step 9 Track the deployment progress and wait until the VM state changes to **ACTIVE** status.

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

Table 5: Virtual machine deployment statuses

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

a. Image Registration – Register the ISO and OVA metadata (same process as CUCM).

b. Virtual Machine (VM) Deployment – Deploy the VM with a Bootstrap ISO (same process as CUCM).

c. Virtual Machine (VM) Management – Perform power operations, console access, and network updates (same as CUCM).

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

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

VM states

VM states represent the lifecycle phases of virtual machines or VNFs deployed on NFVIS. These states track the progression from initial deployment through operational status to termination.

VM States	Description
VM_UNDEF_STATE	The initial STATE of a VM or VNF before deployment of this VM.
VM_DEPLOYING_STATE	The VM or VNF is being deployed on to the NFVIS.
VM_MONITOR_UNSET_STATE	The VM or VNF is deployed in NFVIS but the monitoring rules are not applied.
VM_MONITOR_DISABLED_STATE	Due to a VM action request or recovery workflow, the monitoring or KPI rules applied to the VM were not enabled.
VM_STOPPING_STATE	VM or VNF is being stopped.
VM_SHUTOFF_STATE	VM or VNF is in stopped or SHUTOFF STATE.
VM_STARTING_STATE	VM or VNF is being started.
VM_REBOOTING_STAT	VM or VNF is being rebooted.
VM_INERT_STATE	VM or VNF is deployed but not ALIVE. The KPI MONITOR is applied and waiting for the VM to become ALIVE.
VM_ALIVE_STATE	VM or VNF is deployed and successfully booted up or ALIVE as shown in the KPI metric.
VM_UNDEPLOYING_STATE	The deployment of a VM or VNF is being terminated.
VM_ERROR_STATE	The VM or VNF is in an ERROR STATE because the deployment or some other operation has failed.

NFVIS container deployment

NFVIS container deployment is a network function virtualization process that

- follows the same workflow as VM deployment
- supports SRIOV/Host Interface, Day 0 Configuration, Configuration Options, vCPU Pinning, and Volumes, and
- excludes import and export operations for containers.

Container deployment restrictions and capabilities

NFVIS container deployment has the following restriction:

- Import and export of containers is not supported.

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

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

SRIOV/Host interface configuration

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

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

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

Day 0 configuration

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

Configuration options setup

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

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

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

Table 6: Supported configuration options

Option Name	Type	Description
ENV_VARIABLE	String	Environment VARIABLE to set within container: name/value pair (example: varName=varValue)
LOW_LATENCY	Boolean	vCPU pinning option: When the value is True, the container is allocated a dedicated core.

VNF deployment update

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

Update VNF flavor

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

VNF flavor updates allow you to modify CPU and memory resources for your virtual network functions to meet changing requirements.

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

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

Before you begin

Follow these steps to update a VNF flavor:

Procedure

Step 1 Get a list of available flavors.

Example:

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

Step 2 Check the CPU usage of the system.

Example:

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

Step 3 Update the flavor of the VNF.

Example:

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

The VNF flavor is updated with the new CPU and memory configuration.

Changing the flavor of a VNF from flavor from ASAv5 to ASAv10

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

Update CPU topology

This task allows you to update a VM to use a custom-defined CPU topology that you created when creating a VM flavor.

Updating the CPU topology of a VNF involves updating a VM to a custom-defined topology. The process is similar to updating a VNF flavor—by replacing the name of the CPU topology. For more details, see VM profiles or flavors, on page 39.

Procedure

Step 1 Update the CPU topology using the curl command.

Example:

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

Example of updating ISRv to include a CPU topology

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

--data '<flavor>Isrv_CPUToplogy</flavor>'
```

Step 2 Verify the changed configuration using the virsh command.

Example:

```
support virsh dumpxml <uid>
```

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

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

The VNF is successfully updated with the new CPU topology configuration.

VNF interface updates

VNF interface updates are VM deployment operations that

- support adding an interface, deleting an interface, and moving a VNIC of a VM from one interface to another
- can be performed as hot updates when a VM is in ACTIVE STATE or cold updates when a VM is in VM_SHUTOFF_STATE, and
- depend on custom image properties set during image registration.

Hot and cold update methods

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

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

Cold Update: Cold update refers to an update operation that runs after a VM is put in a VM_SHUTOFF_STATE. In such cases, the VM reboots during the update.

Note The need to do a hot or cold update depends on the custom image properties set during image registration. Refer to the table below for various custom image properties related to hot and cold updates.

Custom Image Property	Value	Interface Update Description	Default Value
interface_hot_add	true	Hot add an interface to a VM	true
interface_hot_add	false	Cold add an interface to VM	
interface_hot_delete	true	Hot delete a VM interface	false
interface_hot_delete	false	Cold delete a VM interface	

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

The VNF interface update prerequisites are:

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

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

Interface Type	Hot Add	Hot Delete	Hot Update for Moving VNIC
VIRTIO	Yes	Yes	Yes
SRIOV	Yes	Yes	Yes
DPDK	Yes	Yes	Yes

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

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

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

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

Update interfaces

This task enables you to perform various interface management operations on VM deployments, including adding new interfaces, removing existing interfaces, and moving VNICs between networks.

Interface updates are necessary when modifying VM deployment configurations to change network connectivity, add additional network connections, or optimize network topology.

Procedure

Step 1 Add a single interface to a VM deployment.

Example:

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

Step 2 Add multiple interfaces to a VM deployment.

Example:

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

HTTP/1.1 204 No Content
```

Step 3 Delete an interface from a VM deployment.

Remove the required nicID along with content between <interface> and </interface> tags.

Example:

```
curl -k -v -u admin:<password> -H
Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X PUT
https://<NfvisIpAddress>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ASAdep/vm_group/ASAvmgrp/interfaces
 --data '
<interfaces>
<interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
    </interface>
**** Note: Remove the required nicID along with content between <interface> and </interface> ******

    </interfaces>'
```

Example:

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

HTTP/1.1 204 No Content
```

In this example, NIC ID 2 has been excluded from the REST API for it to be deleted from the deployment.

Step 4 Move VNICs from one network to another.

Example:

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

Example:

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

```
            </interface>
            <interface>
                <nicid>1</nicid>
                <network>wan2-net</network>
          </interface>
</interfaces>'

HTTP/1.1 204 No Content
```

This example shows how to move nicid 1 from wan-net to wan2-net.

The interface configuration is updated on the VM deployment. You should receive an HTTP 204 No Content response indicating successful completion of the operation.

Access VNFs

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

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

Access VNFs using VNC console

Access VNFs through a VNC client to manipulate the VNF through the NFVIS portal.

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

Before you begin

Follow these steps to access the VNF using the VNC console:

Procedure

Step 1 Using the NFVIS portal, from the **Manage Deployments**, click **Configuration** > **Virtual Machine** > **Manage**

Step 2 Select the desired VM from the list.

The VM must be in active status.

Step 3 Click **Terminal** in the VM row.

A console window opens in your browser.

Step 4 For added security, enable VNC passphrase by including the following contents in your configuration.

Example:

```
<vm_lifecycle>
<tenants>
<name>admin</name>
<deployments>
<deployment>
…
<vnc>
```

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

Access to the VNC console can be restricted through a passphrase.

You can now access and manipulate VNFs through the VNC console in your browser.

Access VMs using serial console

The serial console allows you to access the VM using the serial interface provided by the VM itself.

This method is applicable only if the VM supports serial interfaces in both, its image and its image properties.

Before you begin

Follow these steps to access VMs using serial console:

Procedure

To access the VNF through the serial console, use the command `vmConsole` followed by the name of the VNF.

Example:

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

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

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

nfvis#
```

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

You have successfully accessed the VM through the serial console interface and can now interact with the VM directly.

Access a VM using port forwarding

This task enables you to access a virtual machine through port forwarding configuration, which allows secure remote access to VMs by mapping external ports to internal VM ports.

Port forwarding is used when you need to access a VM that is deployed behind a network address translation (NAT) or firewall. This method creates a mapping between external and internal ports to enable connectivity.

Procedure

Step 1 Deploy a VNF using the deployment payload with **port_forwarding** configuration:

Example:

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

Step 2 Log into VNF using SSH and port number given in the example payload (20122):

Example:

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

isrv-encs#
```

You have successfully accessed the VM using port forwarding. The SSH connection is established and you can now manage the virtual machine remotely through the forwarded port.

VM monitoring

VM monitoring is a system capability that

- monitors deployed VMs periodically based on metrics defined in the KPI section of deployment data model
- can be enabled or disabled by modifying the <actionType> tag, and

- uses bootup_time settings to control monitoring behavior.

VM monitoring configuration details

VM monitoring behavior is controlled through specific settings and parameters:

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

Disabling VM monitoring

This example shows how to disable monitoring for a VM.

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

VM import and export

VM export with selective disk

You can exclude certain disks or volumes from a VM export.

```
nfvis# show running-config vm_lifecycle tenants tenant admin deployments
vm_lifecycle tenants tenant admin
 deployments deployment linuxdep
  vm_group gp_01
   image              Linux_IMAGE
   flavor             centos-disk-large
   vim_vm_name        linux_vm
   bootup_time        -1
   recovery_wait_time 0
   recovery_policy action_on_recovery REBOOT_ONLY
   interfaces interface 0
    model   virtio
    network wan-net
   !
   scaling min_active 1
   scaling max_active 1
   placement zone_host
    host datastore1
   !
   vmexport_policy disk_exclusion Linux_IMAGE:512G-file.qcow2 =====> Disk to be excluded
on Export
   !
  !
 !
!
```

NFVIS VM import and export

NFVIS VM import and export is a backup and recovery mechanism that

- allows you to backup or export (vmExportAction) VMs for disaster recovery
- enables you to restore or import (vmImportAction) VMs from backup files, and
- provides VM-level backup and recovery separate from full NFVIS system backup.

VM export and import limitations

VM export and import operations have specific requirements and limitations:

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

To export a VM ensure that:

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

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

```
nfvis# show system file-list disk local | display xpath | include backup

/system/file-list/disk/local[si-no='84']/name tiny_backup.vmbkp
nfvis# show system file-list disk local 84
SI NO  NAME              PATH               SIZE  TYPE               DATE MODIFIED
-------------------------------------------------------------------------------------
84     tiny_backup.vmbkp  /mnt/extdatastore1  17M   VM Backup Package  2019-01-31 19:31:32
```

To import a VM ensure that:

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

```
nfvis# vmImportAction importPath /mnt/extdatastore1/tiny_backup.vmbkp
System message at 2019-01-31 19:53:32...
```

```
Commit performed by admin via ssh using maapi.
```

An optional unique MAC UID support is added to VM import.

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

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

VM backup using vmExportAction:

Export and import failures

These examples show export failures:

- Original deployment is not deleted

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

- OVS network used by original deployment is deleted.

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

Table 7: Feature comparison table for VM backup using vmExportAction

Features	NFVIS 4.18.2a and Later Releases
Default file location for backup vmExportAction vmName sample exportName vmbackup exportPath <datastore>:	/data/intdatastore/vmbackup.vmbkp /mnt/extdatastore1/vmbackup.vmbkp /mnt/extdatastore2/vmbackup.vmbkp
Default file location for backup on USB vmExportAction vmName sample exportName backup02 exportPath usb:usb1	/mnt-USB/usb1/vmbackup.bkup
Check disk space before backup	Supported
VM backup format	Diff disk backup
Backup image and flavor	Supported
VM live export snapshot	Supported
VM Export with Selective Disk	Supported

VM restore using vmImportAction:

Table 8: Feature comparison table for VM restore using vmImportAction

Features	NFVIS 4.18.2a and Later Releases
Default file location for backup vmImportAction importPath <datastore>:vmbackup.vmbkp	/data/intdatastore/vmbackup.vmbkp /mnt/extdatastore1/vmbackup.vmbkp /mnt/extdatastore2/vmbackup.vmbkp
Default file location for restore on USB vmImportAction importPath usb:usb1/vmbackup.vmbkp	/mnt-USB/usb1/vmbackup.vmbkp
Check disk space before backup	Supported
Restore backing images and flavors	Supported
VM Export with Selective Disk	Supported

Additional capabilities

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.

Before you begin

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.

If the Recovery ISO registration is not allowed, upload the respective VM OVA as well and register it.

Name the image descriptively. 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.

   Wait for the VM state to move to a **SHUTDOWN** state.

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 now 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.

   The VM boots from the Recovery ISO.

e. Follow the application-specific recovery procedure.

   This step involves interacting with the application's recovery prompts via the console.

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.

Before you begin

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.

What to do next

Refer to the official CUCM documentation for any final application-specific configuration steps required after the password reset.

CDROM attachment and detachment

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.

CDROM attachment and detachment process

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

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.

   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.

Before you begin

- 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:

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 corresponding to the recovery ISO you wish 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.

VM graceful stop

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.

Perform a graceful stop

Perform a graceful stop to safely shut down a VM while allowing running processes to complete and prevent data loss.

Use a graceful stop when you need to shut down a VM while ensuring that all running processes complete properly and the system shuts down cleanly. This method is the recommended approach for routine VM shutdowns.

Procedure

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

The **Graceful stop** checkbox is selected by default for VMs deployed with type as **OTHER**.

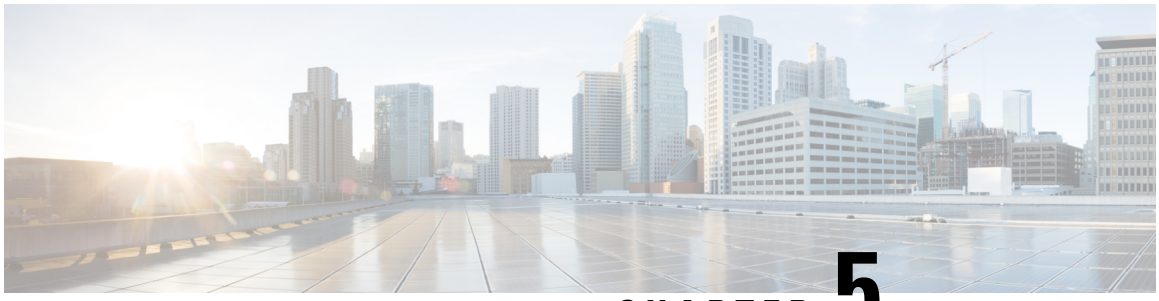
Step 2 Click **OK** to confirm VM shutdown.

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 will automatically move to an error state. In this case, you may need to perform a forceful shutdown by repeating this procedure and unchecking the **Graceful stop** option.

The VM status transitions from **active** to **shutdown**, indicating the graceful stop was successful.



CHAPTER 5

Cisco NFVIS ThousandEyes

Cisco NFVIS ThousandEyes support

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

Cisco NFVIS ThousandEyes support is a pre-integrated solution that

- allows deploying ThousandEyes Enterprise Agents as a container 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.

ThousandEyes container upgrade and performance benefits

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

Prerequisites for Cisco NFVIS ThousandEyes support

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

- 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.

ThousandEyes deployment on Cisco NFVIS

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.

Deploy ThousandEyes container from NFVIS web interface

This task enables you to deploy ThousandEyes containers on Cisco NFVIS infrastructure and perform upgrades when newer container images become available.

Before deploying ThousandEyes containers, you must download the latest ThousandEyes container image file from the **Add New Enterprise Agent** dialog. You also need to upload the ThousandEyes container image file to the Cisco NFVIS and register the image. For more information on uploading and registering the image files see, Uploading VM images to an NFVIS server, on page 19 and Register a remote virtual machine image, on page 38.

Before you begin

- Download the latest **ThousandEyes** container image file from **Add New Enterprise Agent** dialog.
- Upload the **ThousandEyes** container image file to the Cisco NFVIS and register the image.

Follow these steps to deploy ThousandEyes VM:

Procedure

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

Step 2 In the **Select VM or Container** menu, click **TE** and the ThousandEyes node is displayed inside the topology.

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

Field	Description
VM Name	Enter the container's name.
Image	Choose the registered **thousandeyes.docker** image from the drop-down or click + button to add a new image.
Profile	Choose the profile for the ThousandEyes agent from the drop-down or click the + button to add a new profile. For more information, see Enterprise Agent System Requirements for agent without 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. **Note** Do not choose NFS Store as a deployment disk from the drop-down.
Add Bootstrap Config	Add bootstrap config data in the explanded **Add Bootstrap Config** pane. Check the **Is External File Path** check box if your bootstrap file path is from an external source.
Add Config Options	Add the additional configuration in the expanded **Add Config Options** pane. Choose the **Config Option Name**, Enter the **Config Option Value** after = in the pre-populated config option value. For example, TEAGENT_INET=4
Add Volume	Volume required to enable ThousandEyes on Cisco NFVIS are pre-populated with the minimum required volume. You can add or remove volumes based on your requirement.
Remove Volume	You can remove volumes based on your requirement.
Exclude Disks from Export	Not supported by ThousandEyes.

Step 4 Click **Deploy**.

Step 5 To upgrade ThousandEyes Agent Container from NFVIS Web Interface, use the **Upgrade** icon from the **Manage Deployments** page.

- To upgrade a registered image:
  a. From Cisco NFVIS portal, click **Configuration** > **Virtual Machine** > **Manage**.
  b. Click the **Upgrade** icon.
  c. Choose the image from a drop-down list, displays all previously registered images.
  d. Click **Submit** to upgrade your existing configuration to the selected image.

- To upgrade using a new image:
  a. From Cisco NFVIS portal, click **Configuration** > **Virtual Machine** > **Manage**.
  b. Click the **Upgrade** icon.

c. Click **New Image**.

d. Choose a protocol.

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

e. Click **Submit** to register your 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 container on NFVIS using CLI configs

This task allows you to deploy and manage ThousandEyes Enterprise Agent containers on NFVIS infrastructure using various configuration methods including CLI, NETCONF, and RESTCONF.

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

Before you begin

Follow these steps to deploy ThousandEyes Container on NFVIS using CLI configurations:

Procedure

Step 1 Register the ThousandEyes Docker Image

Note
The URL in **`src`** is provided by **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 using CLI configurations, choose one of two 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>
    '
```

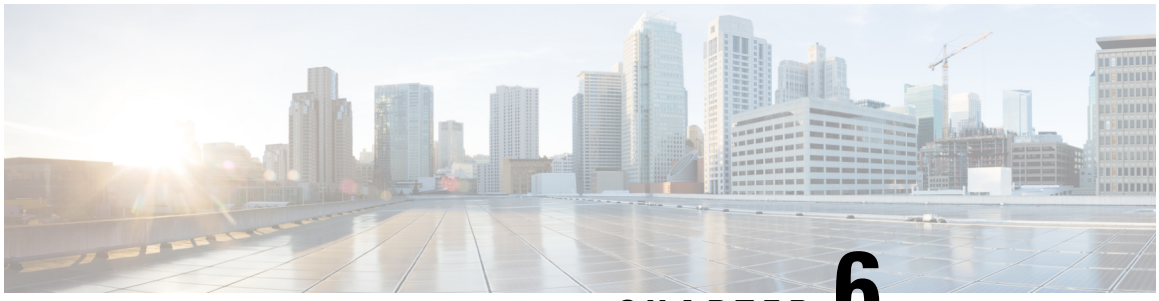
Step 9 Upgrade ThousandEyes Container Deployment on NFVIS Using RESTCONF

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's 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 container is successfully deployed on NFVIS and ready to monitor network performance. The container will be registered with ThousandEyes and begin collecting network metrics according to the configured parameters.



CHAPTER 6

System Access Configuration

Host system requirements

These resources are required for a standalone Cisco NFVIS.

Table 10: CPU allocation

Total Cores	NFVIS 4.10.x 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)

Total Cores	NFVIS 4.10.x and later Releases
* 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.	

Note

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

Table 11: Memory 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	For UCSC-M6 : 7 GB/ 8 GB*	For UCSC-M6 : 11 GB/ 12 GB*	For UCSC-M6 : 11 GB/ 12 GB*	For UCSC-M6 : 11 GB/ 12 GB*	For UCSC-M6 : 11 GB/ 13 GB*	For UCSC-M6 : 16 GB/ 20 GB*
* 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.						

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

Requirements: system setting hostname

You must adhere to the following rules for hostname on NFVIS:

- Must contain minimum length of 2 and maximum length of 255.
- Must begin with a letter or digit and can contain alphabets, numbers and hyphen.
- Must not be deleted.
- The hostname range is from 1 to 58. The hostname range must contain a letter or a digit, it may contain alphabets, numbers, and hyphens.

Access NFVIS

This task enables you to gain initial access to the NFVIS system and configure network connectivity for ongoing management operations.

NFVIS provides multiple access methods including portal, CLI, console, and PNP. The system requires immediate password change after first login for security purposes. Network connectivity can be established through WAN, WAN2, and management interfaces with support for both IPv4 and IPv6 configurations.

Before you begin

Ensure physical connectivity to the NFVIS system through one of the available interfaces.

Follow these steps to access NFVIS and configure initial connectivity:

Procedure

Step 1 Log in using the default credentials.

For initial login, use **admin** as the default user name, and **Admin123#** as the default password. Immediately after the initial login, the system prompts you to change the default password. You must set a strong password as per the on-screen instructions to proceed with the application. All other operations are blocked until default password is changed. API returns 401 unauthorized error if the default password is not reset.

If WAN-br or wan2-br have not obtained IP addresses through DHCP, the zero touch deployment is terminated. To manually apply the IP configurations answer 'y' and the system proceeds with DHCP assignment on WAN-br until the configurations are changed. For DHCP assignment to continue to request IP address for PNP flow on both WAN interfaces answer 'n'.

Step 2 Create a strong password that meets the security requirements.

You must adhere to these rules to create a strong password:

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

You can change the default password in three ways:

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

Example:

```
NFVIS Version: 3.10.0-9

Copyright (c) 2015-2018 by Cisco Systems, Inc.
```

```
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco
Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

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

nfvis login: console (automatic login)

login:
login:
login:
login:
login: admin

Cisco Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS Version: 3.10.0-9

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

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

admin@localhost's password:

admin connected from ::1 using ssh on nfvis
nfvis# show version


NFVIS Version: 3.12.3

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

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

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


Cisco Network Function Virtualization Infrastructure Software (NFVIS)

NFVIS Version: 3.12.3-RC8

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

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

```
admin connected from ::1 using ssh on nfvis
admin logged with default credentials
Setting admin password will disable zero touch deployment behaviors.
Do you wish to proceed? [y or n]y
Please provide a password which satisfies the following criteria:
        1.At least one lowercase character
        2.At least one uppercase character
        3.At least one number
        4.At least one special character from # _ - * ?
        5.Length should be between 7 and 128 characters
Please reset the password :
Please reenter the password :


 Resetting admin password


New admin password is set

nfvis#
System message at 2020-01-08 03:10:10...
Commit performed by system via system using system.
nfvis#
```

Step 3 Connect to the system using IPv4.

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

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

You can connect to the system in these ways:

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

Step 4 Perform static configuration without DHCP if needed.

Step 5 Verify the initial configuration.

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

Example:

```
system settings-native mgmt ip-info interface lan-br
system settings-native mgmt ip-info ipv4_address 192.168.1.2
system settings-native mgmt ip-info netmask 255.255.255.0
!
!
!
system settings-native mgmt dhcp disabled
system settings-native wan ip-info interface wan-br
```

```
system settings-native wan ip-info ipv4_address 209.165.201.22
system settings-native wan ip-info netmask 255.255.255.0
!
!
!
system settings-native wan dhcp disabled
!
!
system settings-native gateway ipv4_address 209.165.201.1
system settings-native gateway interface wan-br
```

Here is an extract from the output of the **show system settings-native** command when the management interface has a DHCP configuration and the WAN interface has a static configuration:

```
system settings-native mgmt ip-info interface MGMT
system settings-native mgmt ip-info ipv4_address 192.168.1.2
system settings-native mgmt ip-info netmask 255.255.255.0
!
!
!
system settings-native mgmt dhcp enabled
system settings-native wan ip-info interface wan-br
system settings-native wan ip-info ipv4_address 209.165.201.22
system settings-native wan ip-info netmask 255.255.255.0
!
!
!
system settings-native wan dhcp disabled
```

You have successfully accessed NFVIS, changed the default password, and configured network connectivity. The system is now ready for management operations through your chosen access method.

VLAN configuration for NFVIS management traffic

A VLAN is a logical network segmentation technology that

- creates independent logical networks within a physical network
- uses VLAN tagging to insert a VLAN ID into a packet header to identify which VLAN the packet belongs to, and
- enables isolation of Cisco NFVIS management traffic from VM traffic when configured on bridge interfaces.

VLAN configuration details

You can configure a VLAN tag on these bridge interfaces:

- WAN bridge (WAN-br) interface to isolate Cisco NFVIS management traffic from VM traffic
- wan2-br for ENCS5400 or ENCS 5100

- user-br for all systems

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

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

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

Configure the IP receive ACL

This task configures the IP receive ACL to filter out unwanted traffic by allowing or blocking traffic based on IP addresses and service ports.

Use the IP receive ACL feature to control access to the management interface by specifying which source networks are permitted to connect.

Before you begin

Follow these steps to configure the IP receive ACL:

Procedure

Step 1 Configure the source network for Access Control List (ACL) access.

Example:

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

Step 2 Verify the trusted IP connection using the **show running-config system settings IP-receive-ACL** command.

This command displays the configured source network for ACL access to the management interface.

Example:

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

The IP receive ACL is configured and the trusted IP connection settings are verified. Traffic from the specified source network is now allowed based on the configured ACL rules.

Configure port 22222 and management interface ACL

Use this task to configure an IP receive Access Control List (ACL) to filter traffic. This allows you to block or allow specific traffic based on IP addresses and service ports to enhance network security.

The IP receive ACL enables you to control access to the management interface. Note that port 22222, used for the SCP server, is closed by default. You must explicitly open this port if you need to SCP files into NFVIS from an external server.

Note The SCP command cannot be used to copy files between two NFVIS devices.

Procedure

Step 1 Configure the source network for ACL access.

Follow these steps to define the source network allowed to access the management interface:

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

Step 2 Open port 22222 for SCP access.

If you need to SCP files from an external server, follow these steps to open the port:

Example:

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

Note
The ACL is identified by the address. If you remove this ACL, all other ACLs sharing the same address are also removed. You must reconfigure those ACLs if necessary.

Step 3 Verify the interface configuration using the **show running-config system settings ip-receive-ACL** command.

Example:

```
nfvis# show running-config system settings ip-receive-acl

system settings ip-receive-acl 10.156.0.0/16

 service  [ ssh https scpd ]

 action   accept

 priority 100

!
```

Port 22222 is now open and configured for SCP file transfer from external servers to the NFVIS system.

Configure secondary IP address and source interface

Configure secondary IP addresses and source interfaces to enable multiple IP addresses per interface and control packet source addressing on NFVIS.

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

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

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

Before you begin

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

Procedure

Step 1 Configure Secondary IP Address:

Example:

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

Step 2 Configure source Interface:

Example:

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

The secondary IP address and source interface are configured. The secondary IP address and source interface related errors are logged in **`show log nfvis_config.log`** file.

Users, roles, and authentication

Configure local user account management

Configure role-based access control to enable administrators to manage different levels of access to the system's compute, storage, database, and application services using access control concepts such as users, groups, and rules.

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

Table 12: Supported user roles and privileges

User Role	Privilege
Administrators	Owns everything, can perform all tasks including changing user roles, but cannot delete basic infrastructure. An admin's role can't be changed
Operators	Start, stop, and delete a VM. Clear logs and view all information
Auditors	Read-only permission and can't perform any tasks

Before you begin

The user passwords must meet these requirements:

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

Procedure

Step 1 Create a user and assign a role.

The administrator can create users and define user roles as required. You can assign a user to a particular user group. For example, the user "test1" can be added to the user group "administrators".

Example:

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

Step 2 Delete a user if needed.

Example:

```
rbac authentication users delete-user name test1
```

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

Step 3 Configure the minimum length for passwords.

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

Example:

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

Step 4 Configure password lifetime values.

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

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

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

Example:

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

Step 5 Configure automatic deactivation of inactive user accounts.

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

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

Example:

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

Step 6 Activate an inactive user account when needed.

The admin user can activate the account of an inactive user.

Example:

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

Local user account management is configured with role-based access control, password requirements, and account lifecycle management policies.

User and role management in the NFVIS portal

User and role management in the NFVIS portal is a graphical interface that

- provides an intuitive alternative to command-line operations for managing user accounts and roles
- enables administrators to create, modify, and delete user accounts, and
- allows assignment of specific roles to define user access levels within the system.

Create new users

Create new user accounts in NFVIS portal to provide controlled access to system resources based on assigned roles and permissions.

Use this procedure when you need to grant access to the NFVIS system for new team members or when setting up role-based access control for different user types.

Before you begin

Log in to the NFVIS portal as an administrator.

Follow these steps to create a new user account through the NFVIS portal:

Procedure

Step 1 Navigate to **Configuration** > **Host** > **Security** > **Users and Roles**.

Step 2 From the **Users and Roles** page, click the **+** icon to create a new user.

Step 3 Enter the following details:

Field	Description
Name	Enter the unique login name for the new user account..

Field	Description
Role	Select the predefined role that defines the user's permissions and access level within the NFVIS system. Options typically include: • Administrator: Full access to configure, update, or delete resources, and manage user roles. • Operator: Can view, start, stop, and delete Virtual Machines (VMs). • Auditor: Read-only access to system information.
Password	Enter the password for the new user account. This must comply with the configured strong password policies (e.g., minimum length, complexity requirements including uppercase, lowercase, numbers, and special characters).
Confirm Password	Re-enter the password exactly as entered in the **Password** field to confirm accuracy and prevent typing errors.
Group	Choose a user group to assign this user to. User groups are used for granular role-based access control (RBAC) and can define specific access policies for sets of users or resources.
Preferred Language	Choose the default language for a user's NFVIS portal interface: • English • Japanese When this user logs in, the NFVIS portal will be displayed in the language chosen here.

Step 4 Click **Submit** to create the new user account with the specified details and save the configuration.

- Click **Cancel** to discard any entered information and return to the previous screen without creating the user.
- Click **Reset** to clear all fields on the form, allowing you to re-enter information.

The new user account is created with the specified role, permissions, and configuration settings. The user can now log in to the NFVIS portal using the assigned credentials.

Modify users

Modify user account details such as role, password, group, and preferred language to maintain proper access control and user management.

User accounts may need to be updated periodically to reflect changes in user roles, security requirements, or personal preferences. This task allows administrators to modify existing user accounts through the NFVIS portal interface.

Before you begin

Log in to the NFVIS portal as an administrator.

Follow these steps to modify a user account through the NFVIS portal:

Procedure

Step 1 Navigate to **Configuration** > **Host** > **Security** > **Users and Roles**.

Step 2 From the **Users and Roles** page, click the edit icon to modify an existing user account.

When modifying a user account, you can update details such as their Role, password, Group, and Preferred Language.

Step 3 Click **Submit** to apply the changes and save the updated configuration.

Click **Cancel** to discard any changes and return to the previous screen without modifying the user.

Click **Reset** to clear all fields on the form, allowing you to re-enter information.

The user account is successfully updated with the new configuration details, and the changes are saved in the system.

Delete users

Delete user accounts to remove access and manage security within the NFVIS portal.

Use this procedure when you need to remove user accounts from the NFVIS system through the portal interface.

Before you begin

Log in to the NFVIS portal as an administrator.

Follow these steps to delete users through the NFVIS portal:

Procedure

Step 1 Navigate to **Configuration** > **Host** > **Security** > **Users and Roles**.

Step 2 From the **Users and Roles** page, identify the user account you want to delete and click the Delete icon associated with it.

A confirmation message is displayed to confirm the deletion of the user account.

Step 3 In the confirmation dialog, use the following options:

- Delete: Click this button to proceed with the permanent deletion of the selected user account(s).
- Cancel: Click this button to abort the deletion process and return to the "Users and Roles" page without removing the user(s).

The selected user account is permanently deleted from the NFVIS system.

Change language preferences in the NFVIS portal

Configure the NFVIS portal interface language to support user interaction in their preferred language (English or Japanese).

Cisco NFVIS portal now supports both English and Japanese languages, providing users with the flexibility to interact with the interface in their preferred language.

Users can adjust their language preferences through two ways methods within the NFVIS portal:

- From the current user session, using the **Settings** icon.
- From the Users and Roles page. For more information on setting the default language for user accounts, see *Create New Users*.

Before you begin

Follow these steps to change the language for your current portal session:

Procedure

Step 1 Log in to the NFVIS portal

Step 2 Click the user profile icon.

Step 3 From the drop-down menu, choose **Language Preferences**

Step 4 Choose your desired language (English or Japanese) from the available options.

Once a language is chosen, the portal interface will immediately update to the chosen language, and a confirmation notification will appear. This change applies only to your current session.

The portal interface displays in your selected language for the current session.

User groups

Granular Role-Based access control

Restrictions for granular Role-Based access control

When configuring granular role-based access control, observe these restrictions to ensure proper system operation:

- A group can only be associated with one policy, either the `resource-access-control` policy or the `local-authentication-only` policy.
- One user can be assigned to one group only.
- A VM can only belong to one group.

Granular role-based access control

Granular Role-Based Access Control (Granular RBAC) is a security feature that

- restricts VM management to a particular set of users
- enables system administrators to define resource groups and assign VNFs and system resources to these groups, and
- allows user assignment to resource groups for access to associated VNFs.

Resource management components

The system administrator can define a set of resource groups, and assign VNFs and system resources such as VMs, disk files, and system level configurations to these defined groups. When you create a user, you can assign that user to one of the resource groups, and this enables the user to access the associated VNFs.

Roles

The three roles defined by the system are:

- 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

The admin user is a special user account with unique 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

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.

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.

For more details on Granular RBAC feature capabilities, see Appendix.

Create groups and assign local users to the groups

This task allows you to establish role-based access control by creating groups and assigning local users to them, enabling organized user management and policy enforcement.

Groups provide a way to organize users and apply policies for role-based access control (RBAC). You can create groups with or without specific policies and then assign local users to these groups to manage their access permissions.

Procedure

Step 1 Create a group with or without a policy.

Example:

```
nfvis(config)# aaa groups group rac_group policy resource-access-control
nfvis(config-policy-resource-access-control)# commit
Commit complete.
```

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

Example:

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

Step 3 View a list of RBAC users with role and group information.

Example:

```
nfvis# 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 groups are created and local users are assigned to them. You can verify the configuration by viewing the RBAC users with their associated roles and groups.

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 the group using the following commands:

Example:

```
nfvis(config)# aaa groups group tac_group user remote_admin1
nfvis(config-user-remote_admin1)# user remote_admin2
nfvis(config-user-remote_admin2)# user remote_operator3
nfvis(config-user-remote_operator3)# policy resource-access-control
nfvis(config-policy-resource-access-control)# commit
Commit complete.
nfvis(config-policy-resource-access-control)# end
nfvis# 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
!
!
nfvis#
nfvis# show rbac authentication users
NAME
-------
admin
```

Note

In the above example, `remote_admin1`, `remote_admin2`, and `remote_operator3` are TACACS+/RADIUS users.

Remote users are successfully 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 will now bypass external TACACS authentication and only go through local authentication.

RADIUS support

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

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

Note You can configure 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

The key components involved in RADIUS authentication are:

- 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

This task configures RADIUS authentication to enable secure client-server communication with encrypted secret key support.

RADIUS secret encryption is supported on NFVIS. You can configure either secret key or encrypted secret key at a given time. Use encrypted secret if special characters are used in secret. NFVIS encrypts both shared-secret and encrypted-shared-secret configurations and supports both TLS and non-TLS client-server communication for RADIUS.

- 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 RADIUS support using one of the following methods:

- For non-TLS RADIUS configuration, use these commands:

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

- For TLS RADIUS configuration, use these commands:

```
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 using the **show running-config RADIUS-server** command.

Example:

```
nfvis# show running-config radius-server radius-server host 1.2.3.4
shared-secret $8$lZwgccBvs9x1oQpx6f1s8/JkZ4rLdQ95VMUjRrhD9Z8=
```

```
admin-priv    15
oper-priv     11
```

The shared secret is displayed in encrypted form.

RADIUS authentication is successfully configured with encrypted secret key support for secure client-server communication.

TACACS+ support

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

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.

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 operates

Summary

The key components involved in TACACS operation are:

- 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 encryped-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 TACACS+ without TLS.

Example:

```
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 TACACS+ with TLS.

Example:

```
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.

Example:

```
nfvis# show running-config tacacs-server
tacacs-server host 1.2.3.4
shared-secret $8$JkMZFgA3DkbjAHOrmdBr3U2cLg2qY1FuHAIJiIp7nSw=
admin-priv    14
oper-priv     9
```

Use the **show running-config TACACS-server** command to verify the configuration. The shared secret is displayed in encrypted form.

The TACACS+ server is configured with the specified host, shared secret, and privilege levels. Users can now authenticate through the TACACS+ server with appropriate 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

After you have identified the TACACS+ and RADIUS server and defined an associated TACACS+ and RADIUS authentication key, you must define method lists for TACACS+ and RADIUS authentication. Because TACACS+ and RADIUS authentication is operated through AAA, you need to issue the AAA authentication command, specifying TACACS+ or RADIUS as the authentication method.

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

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

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

- 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.

Networking

Bridges

A bridge is a network connectivity component that

- enables NFVIS connectivity through IPv4 or IPv6 configurations such as Static IP, DHCP, SLAAC, or VLAN
- can have a port or port channel associated with it, and
- provides default LAN and WAN connectivity on NFVIS installations.

Bridge configuration information

The IP configuration on bridges and the **show bridge-settings** command were added in NFVIS 3.10.1 release.

NFVIS is installed with LAN and WAN bridges by default. A service bridge can also be created. On all NFVIS systems, LAN-br and WAN-br are generated by default and populated with the appropriate ports for that system. On ENCS 5000 series platforms, wan2-br is also generated by default for the dual WAN initialization.

The default LAN bridge is configured with a static IP address 192.168.1.1 and the WAN bridges uses DHCP for initial NFVIS connectivity.

IPv4 bridge configuration:

- If the system has a DHCP server connected to a bridge with DHCP configured, the bridge receives the IP address from the server. You can use this IP address to connect to the system.
- You can also connect to the server locally with an ethernet cable using a static IP address. To connect to the device remotely using a static IP address, you must configure the default gateway or setup an appropriate static route.
- DHCP and a default gateway cannot be configured on NFVIS simultaneously. NFVIS only supports one system level default gateway. If DHCP is configured, the default gateway is assigned to the system through the DHCP server. Also, only one bridge can be configured with DHCP at any time.

IPv6 bridge configuration:

- IPv6 can be configured in static, DHCP stateful, and Stateless Auto configuration (SLAAC) modes. By default, DHCP IPv6 stateful is configured on the WAN interface.
- If DHCP stateful is not enabled on the network, the router advertisement (RA) flag decides which state the network stays in. If the RA shows the Managed (M) flag, then the network stays in DHCP mode, even if there is no DHCP server in the network. If the RA shows the Other (O) flag, then the network switches from DHCP server to SLAAC mode.
- SLAAC provides IPv6 address and a default gateway. Stateless DHCP is enabled in the SLAAC mode. If the server has DNS and domain configured, then SLAAC also provides those values through stateless DHCP.
- Similar to IPv4, IPv6 DHCP and IPv6 default gateway cannot be configured on the system simultaneously, nor can stateful and stateless IPv6 DHCP. Also, only one bridge can be configured with either stateful or stateless IPv6 DHCP at any time.

Create bridges

This task allows you to create and configure a new bridge in the system.

Use this procedure when you need to establish a new bridge configuration for network connectivity.

Procedure

Step 1 Configure a new bridge.

Example:

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

Step 2 Verify the bridge generation using the **show bridge-settings** command.

Example:

```
nfvis# show bridge-settings my-br ip-info interface
ip-info interface my-br
```

The bridge is successfully created and configured. The verification command displays the bridge settings and interface information.

Configure bridge port

Configure a bridge port to establish the connection between a bridge and a physical interface or port channel.

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

Procedure

Step 1 Configure a port on a bridge.

Example:

```
configure terminal
bridges bridge my-br port eth3
commit
```

Step 2 Configure a port channel on a bridge.

Example:

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

Step 3 Verify the port settings applied to a bridge using the **support ovs vsctl** command.

Example:

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

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

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

The bridge port is configured and can be verified using the support ovs vsctl command to display the configured ports.

Configure bridge IP connectivity

This task enables you to establish network connectivity for bridges by configuring IP addressing methods, VLAN isolation, and MAC learning parameters on Cisco NFVIS systems.

Bridge IP connectivity configuration is essential for proper network operation in NFVIS environments. You can configure different connectivity options including DHCP for automatic IP assignment, static IP for fixed addressing, VLAN tagging for traffic isolation, and MAC aging time for optimal MAC address table management.

Before you begin

Follow these steps to configure bridge IP connectivity:

Procedure

Step 1 Configure DHCP on the bridge if automatic IP assignment is required.

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

Example:

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

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

```
nfvis# show bridge-settings my-br dhcp

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

Step 2 Configure a static IP address on the bridge if fixed IP assignment is required.

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

Example:

```
configure terminal
```

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

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

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

Step 3 Configure VLAN tagging on the bridge to isolate traffic.

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

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

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

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

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

Example:

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

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

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

Step 4 Configure MAC aging time on the bridge to optimize MAC address table management.

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

Example:

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

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

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

The bridge is configured with the appropriate IP connectivity settings including DHCP or static IP addressing, VLAN tagging for traffic isolation, and optimized MAC aging time for efficient network operation.

Physical network interface cards

Configure LLDP

LLDP enables network devices to advertise their identity, capabilities, and neighbors, allowing connected devices to see each other as neighbors.

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

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

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

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

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

Procedure

Step 1 To enable LLDP on a PNIC:

Example:

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

Step 2 To disable LLDP on a PNIC:

Example:

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

Step 3 Use the **show LLDP neighbors** command to display the peer information:

Example:

```
nfvis# show lldp neighbors eth0
------------------------------------------------------------
DEVICE
NAME ID      HOLDTIME  CAPS  PLATFORM  PORTID  DESCRIPTION

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

Step 4 Use the **show LLDP stats** command to display the tx and rx information:

Example:

```
nfvis# show lldp stats eth0
-------------------------------------------------------------------
TX      DISCARD  ERROR  RX      DISCARDED  UNREC
NAME    FRAMES  RX       RX     FRAMES    TLVS       TLVS   AGEOUTS
-------------------------------------------------------------------
eth0    23      0        0       19667     0         0         0
```

LLDP is configured on the specified PNIC, enabling the device to exchange identity and capability information with connected neighbors.

Configure the administrative status of a port

Configure the administrative status of a port to control its operational state.

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

Note Administrative status cannot be enabled on port channel.

Procedure

Step 1 Configure the admin status on a pnic for a VM.

Example:

```
configure terminal
pnic GE0-1 admin status down
commit
```

Step 2 Verify the admin status configuration.

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

Example:

```
nfvis# show pnic GE0-1 link_state
link_state down
```

Note
Speed and duplex values in **show pnic** and **ethtool** outputs may differ depending on the peer device's interface speed and duplex settings.

The administrative status of the port has been configured and verified. The port's operational state is now controlled according to the specified administrative setting.

Configure speed, duplex and autonegotiation

This task enables you to configure speed and duplex settings on Physical Network Interface Cards (PNICs) to control autonegotiation behavior and ensure optimal network connectivity.

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

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

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

When autonegotiation is enabled on a port, it does not automatically determine the configuration of the port on the other side of the ethernet cable to match it. Autonegotiation only works if it is enabled on both sides of the link. If one side of a link has auto-negotiation enabled, and the other side of the link does not, then autonegotiation cannot determine the speed and duplex configurations of the other side. If autonegotiation is enabled on the other side of the link, the two devices decide together on the best speed and duplex mode. Each interface advertises the speed and duplex mode at which it can operate, and the best match is selected. Higher speed and full duplex is the preferred mode.

If one side of a link does not have autonegotiation enabled, then the speed and duplex on both sides must match so that the data can transmit without collisions. Autonegotiation fails on 10/100 links, if one side of the link has been set to 100/full, and the other side has been set to autonegotiation which is 100/half.

Procedure

Step 1 To disable autonegotiation on a PNIC, configure speed and duplex:

Example:

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

Step 2 To enable autonegotiation on a PNIC:

Example:

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

Step 3 To configure speed and duplex with non auto values:

Example:

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

Step 4 Use the **show PNIC GE0-0 operational-speed**, **show PNIC GE0-0 operational-duplex** and **show PNIC GE0-0 autoneg** to verify the configurations.

Example:

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

nfvis# show pnic GE0-0 operational-duplex
operational-duplex full

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

Step 5 To verify the PNIC speed and duplex configurations, use the **show notification stream NFVIS Event** command.

Example:

```
notification
event Time 2019-12-16T22:52:49.238604+00:00
nfvisEvent
    user_id admin
    config_change true
    transaction_id 0
    status FAILURE
    status_code 0
    status_message Pnic GE0-1 speed did not update successfully
    details NA
    event_type PNIC_SPEED_UPDATE
    severity INFO
    host_name nfvis
    !
!
notification
event Time 2019-12-16T22:53:05.01598+00:00
nfvisEvent
    user_id admin
    config_change true
    transaction_id 0
    status SUCCESS
    status_code 0
    status_message Pnic GE0-1 duplex updated successfully:full
    details NA
    event_type PNIC_DUPLEX_UPDATE
    severity INFO
    host_name nfvis
    !
!
```

The PNIC speed and duplex settings are configured according to your requirements, and autonegotiation is either enabled or disabled as specified.

Port Channels

Port channels

A port channel is a logical link that

- combines individual links into a group to provide the aggregate bandwidth of up to eight physical links
- increases bandwidth and redundancy and load balances traffic between the member ports, and
- switches traffic from a failed port to the remaining member ports when a member port fails.

Port channel configuration requirements

Port channel configuration has these requirements:

- Port channels must have at least two ports and can be configured using static mode or Link Access Control Protocol (LACP).
- Configuration changes that are applied to the port channel are applied to each member port of the port channel.
- A port channel can also be added to a bridge. When a port channel has two or more than two members and the port channel is added to a bridge, a bond is created.
- A port can be a member of only one port channel and all the ports in a port channel must be compatible.
- Each port must use the same speed and operate in full-duplex mode.

Note

- The Physical Network Interface Controllers (PNICs) added to the port channel should be uniform. For example, all the PNICs associated with the port channel must have SRIOV VFs or they should not have SRIOV VFs.
- The Data Plane Development Kit (DPDK) can be associated only with port channels that have no SRIOV VFs attached to them. When a port channel is attached to a bridge and if the port channel has SRIOV VFs attached, the bridge gets automatically downgraded to a non-DPDK bridge.

Port channel bond modes include:

- **`active-backup`**: In this mode, one of the ports in the aggregated link is active and all others ports are in the standby mode.
- **`balance-slb`**: In this mode, load balancing of traffic is done based on the source MAC address and VLAN.
- **`balance-tcp`**: In this mode, 5-tuple (source and destination IP, source and destination port, protocol) is used to balance traffic across the ports in an aggregated link.

Port channel LACP modes include:

- **`off`**: Indicates that no mode is applicable.
- **`active`**: Indicates that the port initiates transmission of LACP packets.

- **passive**: Indicates that the port only responds to the LACP packets that it receives but does not initiate the LACP negotiation.

Configure a port channel

Port channels provide link aggregation to increase bandwidth and provide redundancy between network devices. This task allows you to create and manage port channels for optimal network performance.

Port channels combine multiple physical interfaces into a single logical interface. You can add ports to existing port channels and integrate them into bridge configurations for enhanced network connectivity.

Procedure

Step 1 Create a port channel.

Example:

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

Note
Ensure to commit the changes.

Step 2 Add ports to the port channel.

You can add a port to a new port channel or a port channel that already contains ports. Adding GE0-0 and GE0-1 to egroup:

Example:

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

Note
Ensure to commit the changes.

Example:

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

Note
Ensure to commit the changes.

Step 3 Add the port channel to a bridge.

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

Example:

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

Note

Ensure to commit the changes.

Step 4 Verify port channel configurations using the **show port-channel** command.

Example:

```
nfvis# show port-channel

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

slave GE0-0: enabled
may_enable: true

slave GE0-1: enabled
active slave >>>active slaveport should show active
may_enable: true
```

The port channel is successfully configured with the specified ports and added to the bridge. The verification output displays the port channel status, including LACP negotiation status and active slave information.

What to do next

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

Enable promiscuous mode

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

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

Procedure

Step 1 Enable promiscuous mode on the interface.

Example:

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

Step 2 Verify that promiscuous mode has been enabled.

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

Promiscuous mode is enabled on the interface, allowing monitoring of all incoming packets.

Configure dynamic SR-IOV

Configure dynamic SR-IOV to control SR-IOV functionality on Physical Network Interface Controllers, allowing you to enable or disable SR-IOV and manage SR-IOV networks based on virtual function requirements.

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

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

Procedure

Step 1 Disable SR-IOV on a PNIC by ensuring all SR-IOV networks on the PNIC are deleted and the PNIC is not attached to a bridge.

Example:

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

Step 2 Enable SR-IOV on a PNIC by ensuring the PNIC supports SR-IOV, the numvfs field is populated with a value less than the maximum number of virtual functions supported, and the PNIC is not attached to a bridge.

Example:

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

To display the SR-IOV status of all PNICs, use the **show PNIC SRIOV** command. To display the SR-IOV state of an individual PNIC use the **show PNIC eth0-1 SRIOV** command.

Step 3 Create SR-IOV networks when the PNIC has SR-IOV enabled and configured with numvfs, using the format <pnic_name>-SRIOV-<num> where <num> is greater than 0 and less than the number of VFs.

To create an SR-IOV network in trunk mode:

Example:

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

To create an SR-IOV network in access mode:

Example:

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

Step 4 Delete SR-IOV networks by ensuring no VMs are attached to the network.

Example:

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

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

SR-IOV is successfully configured on the PNIC with the appropriate virtual functions and networks created or deleted as specified.

System routes

A system route is a static routing configuration that

- directs traffic that should not go through the default gateway
- provides connectivity when certain destinations are not reachable through the default routes, and
- updates the system routing table when configured.

System route configuration requirements

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

Configure system routes

Configure additional static routes to enable network connectivity to specific destinations through designated gateways or devices.

System routes define how network traffic is directed to different destinations. Static routes provide explicit control over routing decisions for specific network segments.

Procedure

Step 1 Configure the system routes using the following commands:

Example:

```
configure terminal
system routes route 172.25.222.024 gateway 172.25.221.1
system routes route 172.25.223.024 dev wan-br
commit
```

Step 2 Verify the system routes configuration using the **show system routes** command.

Example:

```
nfvis# show system routes

DESTINATION    PREFIXLEN       STATUS
---------------------------------------
172.25.222.0      24           Success
172.25.223.0      24           Success
```

The system routes are configured and verified. The output displays the configured destinations with their prefix lengths and successful status.

Troubleshoot system route configuration errors

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

```
nfvis# show system routes

DESTINATION      PREFIXLEN       STATUS
-----------------------------------------
172.25.222.0        24          Failure(1)
172.25.223.1        24          Failure(2)
```

You can find the cause for each error in the *nfvos-confd* log.

Network unreachable error:

```
Failure 1) result=RTNETLINK answers: Network is unreachable
```

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

Invalid argument error:

```
Failure 2)result=RTNETLINK answers: Invalid argument
```

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

Cisco network Plug-n-Play support

Cisco Network Plug-n-Play support is a provisioning solution that

- provides a simple, secure, unified, and integrated offering for enterprise network customers to ease new branch or campus device rollouts
- enables provisioning updates to an existing network, and
- delivers a unified approach to provision enterprise networks comprising Cisco routers, switches, and wireless devices with a near zero touch deployment experience.

Cisco network plug and play client capabilities

You can use the Cisco Network Plug and Play client to:

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

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

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

Configuration file formats - Sample 1

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <rbac xmlns="http://www.cisco.com/nfv/rbac">
    <authentication>
      <users>
        <user>
          <name>admin</name>
          <password>Cisco123#</password>
          <role>administrators</role>
        </user>
        <user>
          <name>test1</name>
          <password>Test1239#</password>
          <role>administrators</role>
        </user>
        <user>
          <name>test2</name>
          <password>Test2985#</password>
          <role>operators</role>
        </user>
      </users>
    </authentication>
  </rbac>
</config>
```

Configuration file formats - Sample 2

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

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

PnP discovery methods

A PnP discovery method is a network discovery mechanism that

- enables Cisco Network PnP agents to locate and connect to the PnP server in Cisco APIC-EM
- activates automatically when a device powers on for the first time without a startup configuration file, and
- provides multiple fallback options to ensure successful server discovery.

Discovery method types

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

- Static IP address—The IP address of the Cisco Network PnP server is specified using the **set pnp static IP-address** command.
- DHCP with option 43—The Cisco PnP agent automatically discovers the IP address of the Cisco Network PnP server specified in the DHCP option 43 string. For more details on how to configure DHCP for PnP server auto-discovery, see the Solution Guide for Cisco Network Plug and Play
- Domain Name System (DNS) lookup—If DHCP discovery fails to get the IP address of the PnP server, for example, because option 43 is not configured, the Cisco Plug and Play Agent falls back on a DNS lookup method. Based on the network domain name returned by the DHCP server, it constructs a fully qualified domain name (FQDN) for the PnP server, using the preset hostname "pnpserver". For more details on how to configure DNS for PnP server auto-discovery, see the Solution Guide for Cisco Network Plug and Play.

Note DNS FQDN Only lookup method is supported since 3.10.1 release.

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

Configure PnP discovery methods

Configure PnP discovery methods to enable device provisioning and management through static configuration with specific IP addresses or FQDN, or automatic discovery through DHCP, DNS, and CCO methods.

PnP discovery enables automatic device provisioning in network environments. You can configure discovery methods using static mode with specific IP addresses, IPv6 addresses, or FQDN, or use automatic mode that leverages DHCP, DNS, and CCO services for device discovery.

Procedure

Step 1 Enable static mode for PnP discovery using IPv4.

Example:

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

Step 2 Enable static mode for PnP discovery using IPv6.

Example:

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

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

Step 3 Enable static mode for PnP discovery using FQDN.

Example:

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

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

Step 4 Enable automatic mode for PnP discovery using IPv4.

Example:

```
configure terminal
pnp automatic dhcp enable
pnp automatic dns enable
pnp automatic cco enable
```

```
pnp automatic timeout 100
commit
```

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

Step 5 Enable automatic mode for PnP discovery using IPv6.

Example:

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

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

Step 6 Verify the PnP status using the **show pnp** command in privileged EXEC mode.

Example:

```
nfvis# show pnp
pnp status response "PnP Agent is running\n"
pnp status ip-address 192.0.2.8
pnp status ipv6-address ""
pnp status port 80
pnp status transport http
pnp status cafile ""
pnp status created_by user
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status dhcp-ipv6 0
pnp status dns-ipv6 0
pnp status cco-ipv6 0
pnp status timeout 100
nfvis#
```

FQDN

```
nfvis# show pnp
pnp status response "PnP Agent is running\nserver-connection\n    status: Success\n    time: 06:23:11
 Jun 17\ndevice-info\n    status: Success\n    time: 06:23:06 Jun 17\nbackoff\n    status: Success\n
    time: 06:23:11 Jun 17\ncertificate-install\n    status: Success\n    time: 06:21:38 Jun
17\ncli-exec\n    status: Success\n    time: 06:22:50 Jun 17\ntopology\n    status: Success\n
time: 06:23:00 Jun 17\n"
pnp status ip-address apic-em-fqdn.cisco.com
pnp status ipv6-address ""
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by user
pnp status dhcp_opt43 0
pnp status dns_discovery 0
pnp status cco_discovery 0
pnp status dhcp-ipv6 0
pnp status dns-ipv6 0
```

```
pnp status cco-ipv6 0
pnp status timeout 0
nfvis#
```

The following sample output shows that the static discovery mode is disabled, and the automatic discovery mode is enabled for DHCP, DNS, and CCO:

DHCP:

```
nfvis# show pnp
pnp status response "PnP Agent is running\nserver-connection\n    status: Success\n    time: 05:05:59
 Jun 17\ninterface-info\n    status: Success\n    time: 05:05:56 Jun 17\ndevice-info\n    status:
Success\n    time: 05:05:38 Jun 17\nbackoff\n    status: Success\n    time: 05:05:59 Jun
17\ncapability\n    status: Success\n    time: 05:05:44 Jun 17\ncertificate-install\n    status:
Success\n    time: 05:01:19 Jun 17\ncli-exec\n    status: Success\n    time: 04:58:29 Jun 17\ntopology\n
    status: Success\n    time: 05:05:49 Jun 17\n"
pnp status ip-address 192.0.2.8
pnp status ipv6-address ""
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by dhcp_discovery
pnp status dhcp_opt43 1
pnp status dns_discovery 1
pnp status cco_discovery 1
pnp status dhcp-ipv6 1
pnp status dns-ipv6 1
pnp status cco-ipv6 1
pnp status timeout 60
```

DNS:

```
nfvis# show pnp
pnp status response "PnP Agent is running\nserver-connection\n    status: Success\n    time: 05:13:55
 Jun 17\ndevice-info\n    status: Success\n    time: 05:13:49 Jun 17\nbackoff\n    status: Success\n
    time: 05:13:55 Jun 17\ncertificate-install\n    status: Success\n    time: 05:12:26 Jun
17\ncli-exec\n    status: Success\n    time: 05:13:34 Jun 17\ntopology\n    status: Success\n
time: 05:13:45 Jun 17\n"
pnp status ip-address pnpserver.apic-em-fqdn.cisco.com
pnp status ipv6-address ""
pnp status port 443
pnp status transport https
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by dns_discovery
pnp status dhcp_opt43 1
pnp status dns_discovery 1
pnp status cco_discovery 1
pnp status dhcp-ipv6 1
pnp status dns-ipv6 1
pnp status cco-ipv6 1
pnp status timeout 60
```

CCO:

```
nfvis# show pnp
pnp status response "PnP Agent is running\nserver-connection\n    status: Success\n    time: 05:24:25
 Jun 17\ninterface-info\n    status: Success\n    time: 05:23:13 Jun 17\ndevice-info\n    status:
Success\n    time: 05:23:01 Jun 17\nbackoff\n    status: Success\n    time: 05:24:25 Jun
17\ncapability\n    status: Success\n    time: 05:23:06 Jun 17\nredirection\n    status: Success\n
   time: 05:09:43 Jun 17\ncli-exec\n    status: Success\n    time: 05:09:53 Jun
17\ncertificate-install\n    status: Success\n    time: 05:18:43 Jun 17\ntopology\n    status:
Success\n    time: 05:23:10 Jun 17\n"
pnp status ip-address 192.0.2.8
pnp status ipv6-address ""
pnp status port 443
pnp status transport https
```

```
pnp status cafile /etc/pnp/certs/trustpoint/pnplabel
pnp status created_by cco_discovery
pnp status dhcp_opt43 1
pnp status dns_discovery 1
pnp status cco_discovery 1
pnp status dhcp-ipv6 1
pnp status dns-ipv6 1
pnp status cco-ipv6 1
pnp status timeout 60
```

The output displays the current PnP configuration status, showing whether static or automatic discovery modes are enabled and the specific parameters configured for each method.

PnP discovery methods are configured according to your network requirements. The device can now discover and connect to the PnP server using the specified method (static IP/IPv6/FQDN or automatic DHCP/DNS/CCO discovery).

PnP Root Certificate and Static Configuration

This reference provides the technical specifications, file requirements, and command syntax for uploading PnP root certificates and configuring static PnP settings on the NFVIS host.

A certificate can be used as a PnP root certificate through Command Line Interface (CLI). The following command is used to upload a certificate:

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

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

The following are examples to show how to upload certificates:

```
nfvis# system certificate input filepath intdatastore:uploads/apic_em_oneline_02.pem pem-data
 "-----BEGIN
[redacted]
 CERTIFICATE-----\n-----BEGIN
[redacted]
 CERTIFICATE-----\n"
nfvis#
```

```
nfvis# system certificate input filepath
Value for 'filepath' (<string>): intdatastore:uploads/test_cert.pem
Value for 'pem-data' (<string>):
[Multiline mode, exit with ctrl-D.]
> -----BEGIN CERTIFICATE-----
> MIIDvzCCAqegAwIBAgIUKBZWJ4U2c1gBYJFBFWNlNQSVd2MwDQYJKoZIhvcNAQEL
> BQAwgYgxLTArBgNVBAMTJDEyYjQ0ZDhmLWRhZWItNGIxNi1iYWZhLWRiYTQxZGRk
> MWY3MzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExEDAOBgNVBAcT
> B1Nhbkpvc2UxEzARBgNVBAsTCkFQSUNFTS1TRE4xDjAMBgNVBAoTBUNpc2NvMB4X
> DTE4MDcyMTIwNDMyN1oXDTIzMDcyMDIwNDMyN1owXjEVMBMGA1UEAxMMMTcyLjI1
> LjIxNy44MRMwEQYDVQQIEwpDYWxpZm9ybmlhMQswCQYDVQQGEwJVUzEOMAwGA1UE
> ChMFQ2lzY28xEzARBgNVBAsTCkFQSUNFTS1TRE4wggEiMA0GCSqGSIb3DQEBAQUA
> A4IBDwAwggEKAoIBAQDD616jSBX+DfxI5kGT3JCmcIgOHYDDhJJd1L8qhxDAOM7P
> qLjLEdUCOTap8Lu/dKpVClN+hE0LNr9HyQSZ7Mn8+UrM7wQqTKEA6p3dw323wWbp
> ia/XSgByznj4JEY3xW8/trKGkoxfZ7D2JA/cLTf7hK3v44Nev4ONdKtnwNXV32ms
> E7Mpx/Wb/110hYn37DJ9sN5+o34F1KkF1quaXPOgmQA8PMH2Y1LMOh/Z3g3afali
> uOsbPpeGxawEWgfc5pK5EJbziWy1JUSh/c2DaqrfvnKnlprM/HfjZLgzit9FnLyL
> 0lxW54U77AMDRcNALtSq0G3YDX12pdb+ateFWznFAgMBAAGjSjBIMAkGA1UdEwQC
> MAAwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAP
> BgNVHREECDAGhwSsGdkIMA0GCSqGSIb3DQEBCwUAA4IBAQAiD3XiYgUTFK0MuxuB
> DYanVIuAPBImqULdidh3uDlnoTUlzDFU+feYug+XHjAPk7rczZ1Yoc5Hvo90/gQU
> WQUd9TftJygftRn4vG4MMb4fjDPqoh7rxj0P51NRJIA64ro6/gvbQn+T70a62/Wt
> sDGIdXhp8MUTzA/R99vxYUS9YfTMFVAEv5KAW9ZRwC+CTPT+YeN0B+iNO3cXmZ3A
> C0YDXgN27yfF/+1Lna1aOMCTXdTFzNCplOIhhWKulM74Re9QN0Lwvkt9PDim3rX+
> GaDvd4nINMcIXRXy40ieQOOv0W9xnYnvgFD3xjhKQhzDEXX1BStXVJwaqAwWlDlY
> 8EdE
> -----END CERTIFICATE-----
> -----BEGIN CERTIFICATE-----
> MIIDrzCCApegAwIBAgIGCMgojJNOMA0GCSqGSIb3DQEBCwUAMIGIMS0wKwYDVQQD
> EyQxMmI0NGQ4Zi1kYWViLTRiMTYtYmFmYS1kYmE0MWRkZDFmNzMxCzAJBgNVBAYT
> AlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRAwDgYDVQQHEwdTYW5Kb3NlMRMwEQYD
> VQQLEwpBUElDRU0tU0ROMQ4wDAYDVQQKEwVDaXNjbzAeFw0xODA3MjEyMDQzMjVa
> Fw00ODA3MTMyMDQzMjVaMIGIMS0wKwYDVQQDEyQxMmI0NGQ4Zi1kYWViLTRiMTYt
> YmFmYS1kYmE0MWRkZDFmNzMxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9y
> bmlhMRAwDgYDVQQHEwdTYW5Kb3NlMRMwEQYDVQQLEwpBUElDRU0tU0ROMQ4wDAYD
> VQQKEwVDaXNjbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALd/HK9i
> 6H45KJ42G3awGiehp5ZJWeUNqbmnkD0eiR6PNJIVNGBGMV3k+SW0kIfTjxIDkqSc
> K+K5tfEtTRc0R+WmgtdT+asDhPOmfmL73POqIXLr2RE98J0FLkcdhpRU2K1/tDu3
> 4+7WBA1OT/uIb9LVkVrk1pwolVKAWfgwBgCfmLmebtNfvexI3hvk2awcl4fmVb+d
> CMGVXHBw+yQW2AE3RH66VebS/E6bc3ifAvXHlWYWMt/tPRQRftH+0ZrgXt/dar5o
> 24Qq8taPwY1UIse41Nnnp1Q+FuoE9elLVmLidVaJ7qraDw5Yi6ZSQNGud813lU8s
> 6RtbNS48SSB3U0MCAwEAAaMdMBswDAYDVR0TBAUwAwEB/zALBgNVHQ8EBAMCAQYw
> DQYJKoZIhvcNAQELBQADggEBAHUddWTSeJR7QLxud9aaoIYzbL8lwJx7Ot49K5So
> RrymEKbrUSGtZaBp+hdam3l2ByDS44OmwDolelMsdIN61hV5o8DgpaJLwMhoY8gS
> HXyUTdpiLQD6RoxNWim6wGJGDTiMZFYaiaz5oFeazzOG7D8qu9agQ+/7Ky0RVkUK
> 6IoZZPSQ4BgBnsKg3OdBSPUWSIB7+rMz6ww5ELNOBIDo+yo0wCNgvmy19QgabM0/
> 5SJc2pMyidQy2yj3h1+DpMlZ6nWPqF+9IZlzoErKYABySH/Stho4l2kMXY5libCi
> AEqsyhYpxeO9S466uzcujQwJFARog2GGO0vJEDICqKm+lDI=
> -----END CERTIFICATE-----
>
nfvis#


nfvis#
nfvis# system certificate input filepath intdatastore:uploads/pnp_cert7.pem
Value for 'pem-data' (<string>):
[Multiline mode, exit with ctrl-D.]
> -----BEGIN CERTIFICATE-----
> MIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGEwJCRTEPMA0G
> A1UEChMGR251VExTMSUwIwYDVQQLExxHbnVUTFMgY2VydGlmaWNhdGUgYXV0aG9y
> aXR5MQ8wDQYDVQQIEwZMZXV2ZW4xJTAjBgNVBAMTHEdudVRMUyBjZXJ0aWZpY2F0
> ZSBhdXRob3JpdHkwHhcNMTEwNTIzMjAzODIxWhcNMTIxMjIyMDc0MTUxWjB9MQsw
> CQYDVQQGEwJCRTEPMA0GA1UEChMGR251VExTMSUwIwYDVQQLExxHbnVUTFMgY2Vy
> dGlmaWNhdGUgYXV0aG9yaXR5MQ8wDQYDVQQIEwZMZXV2ZW4xJTAjBgNVBAMTHEdu
```

```
> dVRMUyBjZXJ0aWZpY2F0ZSBhdXRob3JpdHkwWTATBgcqhkjOPQIBBggqhkjOPQMB
> BwNCAARS2I0jiuNn14Y2sSALCX3IybqiIJUvxUpj+oNfzngvj/Niyv2394BWnW4X
> uQ4RTEiywK87WRcWMGgJB5kX/t2no0MwQTAPBgNVHRMBAf8EBTADAQH/MA8GA1Ud
> DwEB/wQFAwMHBgAwHQYDVR0OBBYEFPC0gf6YEr+1KLlkQAPLzB9mTigDMAoGCCqG
> SM49BAMCA0gAMEUCIDGuwD1KPyG+hRf88MeyMQcqOFZD0TbVleF+UsAGQ4enAiEA
> l4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
> -----END CERTIFICATE-----
>
nfvis# config
Entering configuration mode terminal
nfvis(config)# pnp automatic cco disable
nfvis(config)# pnp automatic cco-ipv6 disable
nfvis(config)# pnp automatic dns disable
nfvis(config)# pnp automatic dns-ipv6 disable
nfvis(config)# pnp automatic dhcp disable
nfvis(config)# pnp automatic dhcp-ipv6 disable
nfvis(config)# commit
Commit complete.
nfvis(config)# pnp static ip-address 10.0.0.7 port 443 transport https cafile
/data/intdatastore/uploads/pnp_cert7.pem
nfvis(config)# commit
Commit complete.
nfvis(config)# end
nfvis# exit
```

DPDK support on NFVIS

DPDK support on NFVIS is a network performance feature that

- increases network throughput by allowing applications to pull data directly from the Network Interface Card (NIC) without involving the kernel
- delivers high-performance user-space network I/O by allowing network traffic to bypass NFVIS kernel and directly reach deployed VNFs and service chains, and
- reserves additional cores and memory to enhance system performance.

DPDK support features

DPDK support on NFVIS includes:

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

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

DPDK restrictions include:

- You must enable DPDK using the **system settings DPDK enable** command before you commit any other configurations.
- DPDK is not supported on wan-br or wan2-br on any NFVIS platform.
- SR-IOV interfaces and DPDK support: To enable DPDK, every device driver must be supported by DPDK. NFVIS does not support SR-IOV interface upgrade to enable DPDK because SR-IOV device

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

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

DPDK support system requirements include:

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

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

DPDK operational status values are:

Table 13: DPDK status values

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

Configuring DPDK support takes up to a minute and network changes can be observed during the process. NFVIS provides an operational status for DPDK support which indicates if DPDK support is enabled or not.

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

After enabling DPDK, physical NICs configured with SR-IOV will not be able to interact with DPDK bridges. To add a physical NIC to a DPDK bridge, all SR-IOV networks created on the interface should be removed first. NFVIS will not allow adding an SR-IOV configured interface to a DPDK bridge. For more information, see #unique_139.

DPDK configuration examples

To enable DPDK support:

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

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

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

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

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

config terminal
no system settings dpdk
commit

nfvis# show system setting-native dpdk-status
system settings-native dpdk-status disabled
```

Storage access

Configure network file system support

Configure Network File System (NFS) to enable file access on remote devices using Remote Procedure Calls (RPC) to route requests between users and servers.

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

Procedure

Step 1 Mount NFS storage on the system.

Example:

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

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

Step 2 Register images on NFS.

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

To register tar.gz images on NFS:

Example:

```
configure terminal
```

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

Similar configuration can be used for the various images formats.

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

Step 3 Deploy a VM on NFS.

To deploy a VM on NFS, under deployment VM group, use the **placement type zone_host host nfs_storage** command.

NFS is configured and ready for file operations, image registration, and VM deployment on the mounted storage.

Host System Operations

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

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.

Table 14: Power Management

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

System File Management

Use these commands to list, copy, and delete files within the NFVIS environment.

List System Files

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

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

Table 15: System Files

Disk Type	Files
local	Files present in the internal datastore and external datastores

Disk Type	Files
nfs	Files on NFS
usb	Files on the mounted USB drive

Copy System Files

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

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

The `system file-copy` command can also be used to copy a file from the given source path to the given destination path.

The allowed directories for source path and destination path are:

- /data/intdatastore
- /mnt/extdatastore1
- /mnt/extdatastore2
- /mnt/extdatastore3
- /data/mount

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

Delete System Files

The `system file-delete` command is used to delete a file from one of these directories: /data/intdatastore, /mnt/extdatastore1, /mnt/extdatastore2, /mnt/extdatastore3, /mnt-usb/ or /data/mount.

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

Secure Copy (SCP)

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.

Syntax: **scp**<*source*> <*destination*>

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

SCP between two NFVIS devices is not supported.

Examples

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

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

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

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

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

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

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

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

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

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

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

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

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
```

Backup and Restore NFVIS and VM Configurations

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

Note

- To backup and restore a single VM, use vmExportAction (for VM backup) and vmImportAction (for VM restore) APIs.
- Perform the following hostaction backup that avoids loss of VMs during hostaction restore due to insufficient disk space:
  1. Stop the functioning of the VMs that are associated with Cisco NFVIS.
  2. Perform individual image backups of the VMs using the **vmExportAction** command.
  3. Once the backup is successful, delete the VMs and the images from Cisco NFVIS.
  4. When you delete the VMs and the images, perform a host level backup with configurations-only option using the command **hostaction backup configuration-only file-path** *extdatastore2:sample-dir/sample* .
  5. Copy the backup files to a file server.
  6. Perform a factory reset using the **factory-default-reset** command.
  7. Paste the backup copied to a file server and restore the host level backup file using the **hostaction restore** *file-path extdatastore2:sample-dir/sample.bkup* command.
  8. When the restore fails due to disk storage issues, restore the configurations-only backup. When the restore is successful, restore the VMs and their images using the **vmImportAction** *importPath /mnt/extdatastore1/tiny_backup.vmbkp* command.

Restrictions for Backup and Restore on NFVIS

- The backup includes all deployed VMs and the registered images except uploaded files.
- VM restore using hostaction restore and vmImportAction requires original registered image to be on the system, on the same datastore. Missing registered image or image registered in a different datastore results in VM restore failure.
- The time taken to backup a VM depends on the option you choose:
  - *configuration-only* - within 1 min.
  - *configuration-and-vms* - depends on the number of VM deployments on your system, system disk write speed, and compress the VM disks into one bundle.
- You can either backup all the VMs or none.
- The final backup is a compressed file which requires temporary disk space to create the VM backup file. If the system has only one datastore, the maximum deployment backups in a single file is around one-third to half of the datastore disk space. If the deployments occupies more disk space, use *vmExportAction* to backup an individual VM instead of relying on host backup for all VM deployments.
- NFVIS only supports backup or restore on the same release. For example, backup created in Cisco NFVIS release 4.1.1 cannot be used to restore on Cisco NFVIS release 4.2.1.

Feature Comparison Table for Backup and Restore

Backup using hostaction backup:

Feature	NFVIS 26.2.1 Release
Default file location for backup	/data/intdatastore/backup.bkup /mnt/extdatastore1/backup.bkup /mnt/extdatastore2/backup.bkup /mnt/extdatastore3/backup.bkup
VM backup format	Diff disk backup
Registered Image and Flavors	Yes
Status monitoring	Yes
Check disk space before backup	Yes

Restore using hostaction restore:

Feature	NFVIS 26.2.1 Release
Default file location for backup	/data/intdatastore/backup.bkup /mnt/extdatastore1/backup.bkup /mnt/extdatastore2/backup.bkup /mnt/extdatastore3/backup.bkup
Restore images and flavors	Yes
Unique Mac Uid for VM	Yes
Status monitoring	Yes
SNMP v3 user/passphrase restore (with uniqMacUid)	If system engine ID is the same as backup, restore all v3 users. If system engine ID is different from backup, ignore v3 users restoration.
SNMP engine ID restore on different system	Engine ID changed to same as backup bundle

VM backup using vmExportAction:

Feature	NFVIS 26.2.1 Release
VM backup format	Diff disk backup

Backup and Restore

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

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

Action	Backup configuration-only	Backup configuration-and-vms
Save system configurations	Yes	Yes
Save system upgrade configurations	Yes	Yes
Save system upgrade file	No	No
Save images and flavors configurations	Yes	Yes
Save image disks	No	Yes
Save deployments configurations	Yes	Yes
Save deployments disks	No	Yes

The following examples shows the backup options:

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

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

The following example shows the backup stored on a USB:

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

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

Use the **show hostaction backup status** command to view the status of the overall backup 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 backup process is complete:

```
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     ""
nfvis#
```

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

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

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. Restoree 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.
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)

Condition	dpdk-disabled while backup	dpdk-enable while backup
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     ""
nfvis#
```

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.

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 fails to restore if:

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

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

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

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

Reset to 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:

Procedure

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

Table 16: 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.

Configure banner, message of the day and system time

Configure your banner and message of the day

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

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.

Procedure

Enter global configuration mode and configure the banner and message:

Example:

```
configure terminal
banner-motd banner "This is a banner" motd "This is the message of the day"
commit
```

Note
Currently, you can create banners and messages in English only. You can view the system-defined banner using the **show banner-motd** command. This command does not display the user-defined banner or message.

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 with NTP

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.

Procedure

Step 1 Set the system time manually.

Example:

```
configure terminal
```

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

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

Step 2 Set the system time using NTP IPv4.

Example:

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

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

Example:

```
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.

Configure DNS name servers

DNS name servers are used for domain name resolution. Configuring them allows the system to resolve domain names to IP addresses.

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

Procedure

Step 1 Configure name servers.

Example:

```
config terminal
system settings name-server 209.165.201.24 209.165.201.23 2001:420:30d:201:ffff:ffff:fff4:36
commit
```

Step 2 To update name servers, first unconfigure existing name servers and then configure new ones.

Example:

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

Step 3 To unconfigure name servers, use the no form of the command.

Example:

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

The DNS name servers are configured and will be used for domain name resolution. The configured servers are prepended to any DHCP-provided DNS servers.

Configure the IP host

Use this task to specify static mapping between hostnames and IP addresses on the NFVIS host. This configuration allows you to resolve hostnames locally without relying on external DNS services.

Procedure

Step 1 Enter configuration mode

Example:

```
configure terminal
```

Step 2 Configure the IP host mapping.

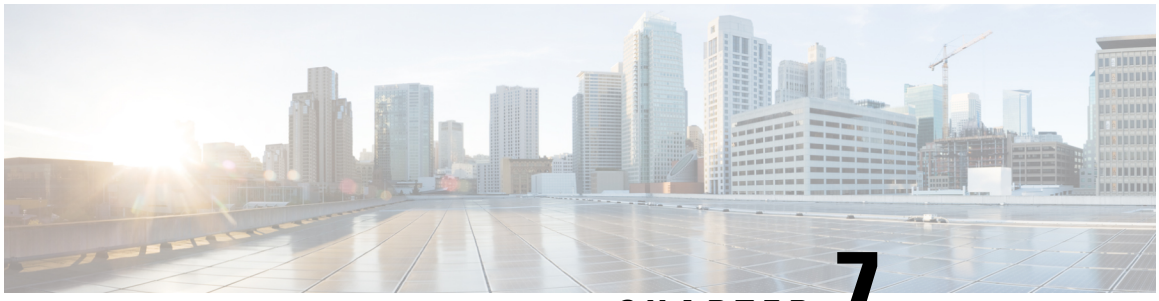
Define the hostname and its associated IP addresses.

```
ip host test2.com 2.2.2.3 2.2.2.1
```

Step 3 Commit the configuration

```
commit
```

The static mapping between the hostname and the specified IP addresses is saved to the system configuration.



CHAPTER 7

Access Cisco NFVIS Portal

Access NFVIS portal

Access the NFVIS portal to view the dashboard which provides a summary of activities on the device.

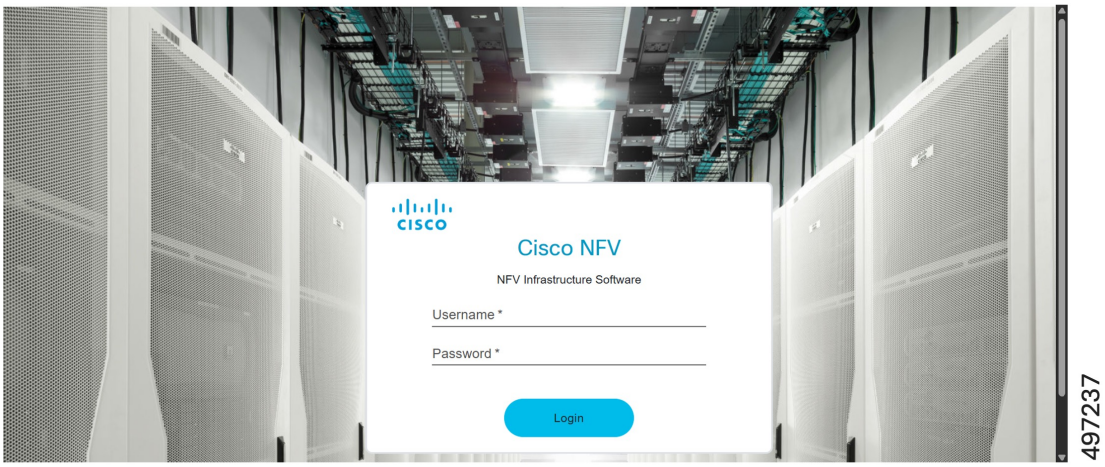
The NFVIS portal is accessed through a web browser using the local ethernet management network connection.

Procedure

Step 1 Connect your laptop to the local ethernet management network and enter https://10.29.43.84 in your web browser's address bar.

We recommend that you use Google Chrome.

Step 2 Login to NFVIS portal using username **admin** and the new generated password.



You will see the NFVIS dashboard which provides a summary of activities on the device.

You have successfully accessed the NFVIS portal and can view the dashboard with device activity summaries.

Create and deploy a generic VM

This task allows you to deploy a virtual machine through the NFVIS portal with network connectivity options for your infrastructure needs.

Use this procedure when you need to create and deploy a generic virtual machine with network connections in an NFVIS environment. The deployment process involves selecting images, configuring network connections, and monitoring the deployment progress.

Before you begin

Ensure that VM images are available in the Image Repository and that you have access to the NFVIS portal.

Follow these steps to create and deploy a generic VM:

Procedure

Step 1 From the NFVIS portal, choose **Configuration** > **Virtual Machine** > **Images** > **Image Repository** from the navigation tree on the top of the interface.

Here you will see all the previously uploaded images in the device. In **Images** you can see information about the available images and make a note of the version for an upgrade if required. The **ACTIVE** state of the image indicates that the image is registered and ready for deployment.

Step 2 Choose **Configuration** > **Deploy**.

You can catalog various VM or Containers at the top of the page. The default configuration of the device at the center of the page has LAN, WAN, and SR-IOV networks.

Step 3 To create an **OTHER (generic VM)** instance with a LAN and WAN connection click and drag **OTHER (generic VM)** (generic VM) to the center of the page.

To configure a connection to the WAN, click **OTHER (generic VM)** (generic VM) on the page and drag it to the `WAN-net` line. Select the connected line to view the details. In the vNIC details panel you will see that the vNIC id and Model is associated with the WAN (`WAN-net`). Record this interface name to use the same name to configure the WAN subnet later.

To configure a LAN connection, click **OTHER (generic VM)** again and this time drag it to the `LAN-net` line. Select the connected line to view the details. In the vNIC details pane you will see vNIC id and Model is associated with the LAN (`LAN-net`). Record this interface name to use this same name to configure the local subnet later.

Step 4 Click **OTHER (generic VM)** and enter the VM Details.

The virtual machine was deployed through the NFVIS **Deployment Details** section. The deployment was configured with the VM name **OTHER99** and the image **TinyIsoTest.ISO**. The selected profile was **NFVIS,** and the deployment disk was set to **datastore1 (internal).** Dedicated cores were not enabled for this deployment. **No GPU devices** were assigned, and no Bootstrap ISO was selected. The **Group Name** and **VNC Password** fields were left unconfigured.

Step 5 Click **Deploy** to deploy the **OTHER99** VM and see the progress of the deployment on the right side of the page.

A successful deployment is indicated through a pop-up message on the corner of the page.

Step 6 To monitor the progress of the OTHER VNF booting, choose **Configuration** > **Virtual Machine** > **Manage**.

The status of the deployment is displayed in **Status** column. Click on the **Refresh** button to get the latest status.

When the VNF is ready you can see all the data related to it along with some useful action items like edit, terminal, CD-Rom, etc.

Configure Day-N CD-ROM attach-detach workflow

This task allows you to attach and detach CD-ROM images to virtual machines, enabling you to boot from recovery ISOs or remove existing disks as needed.

CD-ROM attach and detach operations require the VM to be in Shut Off State. The attachment process involves registering the Recovery ISO and then attaching it to the desired VM through the CD-ROM action.

Before you begin

The VM must be in Shut Off State to perform CD-ROM attach and detach functions. The attachment process consists of two steps: first, register the Recovery ISO using the existing image registration workflow; then, navigate to the CD-ROM action for the desired VM to attach the registered ISO.

Follow these steps to attach and detach CD-ROM images:

Procedure

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

Step 2 Click the **CD-ROM** icon.

The **Attach** and **Detach** options are displayed.

Step 3 Select the **Recovery ISO** from the image dropdown.

Step 4 Click **Submit**.

Note
To use this Recovery ISO, you need to start the VM and then enter the Boot menu through the **terminal** Action button.

A pop-up notification will appear in the top-right corner confirming that the CD-ROM is attached.

Step 5 In the boot menu, select the **CD-ROM** option.

The system will then boot from the Recovery ISO.

Step 6 To detach a CD-ROM, select the disk you wish to detach from the **Disks** dropdown menu and click **Submit**.

The **Detach** option allows you to remove existing disks.

A confirmation alert will appear in the top-right corner upon successful detachment. You can confirm the detachment by checking the **Disks** drop-down menu.

You have successfully configured the CD-ROM attach-detach workflow. The VM can now boot from the attached Recovery ISO, or you have successfully detached the CD-ROM as needed.

Shut down a Virtual Machine

Use this procedure to shut down a Virtual Machine (VM) gracefully to ensure data integrity and proper system state management.

Procedure

Step 1 To perform a graceful shutdown when needed, navigate to **Configuration** > **Virtual Machine** > **Manage**.

Step 2 In the **Action** column, select **Switch Power**.

Step 3 Select the **Graceful Shutdown** checkbox to initiate the process.

The Graceful Shutdown process has a 20-minute timeout. If the VM does not shut down gracefully within this period, it will enter an **Error** state. If this occurs, you have the option to perform a **Forceful Shutdown** to power off the VM.

The VM initiates the graceful shutdown process. If successful, the VM powers off and its status updates to reflect that it is no longer active. If the shutdown does not complete within the 20-minute timeout period, the VM enters an *Error* state, at which point you must perform a Forceful Shutdown to power off the VM.

Accessibility features in the NFVIS GUI

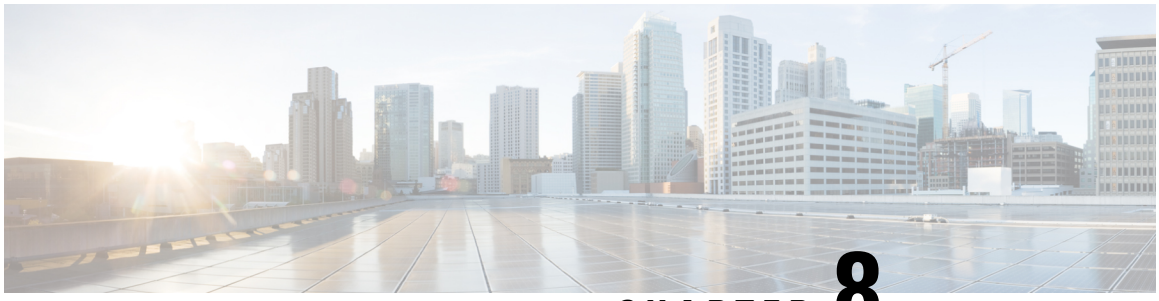
The NFVIS GUI includes accessibility enhancements designed to support users with disabilities and improve overall interface usability through keyboard navigation, screen reader compatibility, visual clarity improvements, and assistive technology integration.

Accessibility enhancements

1. Improved Keyboard Navigation: Enhanced operability and focus order across all workflows to support keyboard-only users.
2. Enhanced Screen Reader Support: Updated ARIA roles and descriptive labels to ensure compatibility with screen readers.
3. Optimized UI Color Contrast: Adjusted color schemes to meet WCAG 2.1 AA guidelines for improved readability.
4. Meaningful Alternative Text: Added descriptive text to icons and visuals, ensuring that users who rely on non-visual access receive the same information as sighted users.
5. Refined Form Structure: Corrected label associations for form controls to improve compatibility with assistive technologies.

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.
- 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.



CHAPTER 8

Cisco NFVIS Upgrade

Cisco NFVIS upgrade

A Cisco NFVIS upgrade is a software update process that

- requires copying the upgrade image to the server before starting the process, and
- ensures cryptographic integrity and authenticity through signed RPM packages.
- shuts down the VM and retains VM data during the upgrade, and after the upgrade automatically restores the VM to its prior operational state upon completion.

Upgrade image formats and security

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

Note Currently, downgrade is not supported.

Image copy requirements

Ensure that you copy the image to the Cisco NFVIS server before starting the upgrade process. Always specify the exact path of the image when registering the image. Use the **scp** command to copy the upgrade image from a remote server to your Cisco NFVIS server. When using the **scp** command, you must copy the image to the "`/data/intdatastore/uploads`" folder on the Cisco NFVIS server.

Upgrade matrix for upgrading Cisco NFVIS

This matrix shows the supported upgrade paths from current Cisco NFVIS versions to target versions, including the required image types for each upgrade scenario.

Table 17: Upgrade matrix for upgrading Cisco NFVIS

Running Version	Supported Upgrade Version	Supported Upgrad
4.18.x	26.2	iso

Restrictions for Cisco NFVIS ISO file upgrade

Cisco NFVIS supports .ISO upgrade only from version N to versions N+1, N+2 and N+3.

- Image downgrade using .ISO file is not supported.
- Upgrade from 4.18 to 26.1 is not supported.

In case of an error while upgrading Cisco NFVIS rolls back to the image version N.

Upgrade NFVIS software using Cisco NFVIS portal

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

The NFVIS upgrade is a two-step process: first, registering the NFVIS upgrade image, and then initiating the upgrade.

Note

- During the upgrade process, the NFVIS application is not available for use.
- All VMs in a running state will be powered off automatically.
- After the upgrade completes, the NFVIS application will be accessible, and all VM will be powered on.
- The upgrade process typically takes approximately 20-30 minutes to complete.

Procedure

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

Step 2 Register the upgrade image using one of these methods:

- If the upgrade image is on your LOCAL computer:
  - **a.** Select the **LOCAL** tab.
  - **b.** Click **Select File (.iso)** and choose the NFVIS upgrade image.
- If the upgrade image is on a REMOTE server:
  - **a.** Select the **REMOTE** tab.
  - **b.** Fill in the following fields:

Table 18: Upgrade details

Field	Description
Image Name	Enter a descriptive name for the upgrade image.
Protocol	Select **HTTP** from the drop-down list. **Note** REMOTE registration currently supports only HTTP protocol.
IP Address	Enter the server IP address.
Port	Enter the port number (optional, defaults to 80).
Image File Path	Enter the path to the upgrade image.
Image Checksum	Provide the SHA512 checksum value.

c. Click **Submit**.

The system will upload or download the image, validate the supported upgrade path, and register the image. Once complete, the image appears in the **Upgrade Image List** with **Status**: **Valid**.

Note
A progress bar for image registration is not available for REMOTE uploads. The image registration and download process runs in the background and the image will appear in the Image Repository table once the process completes.

Step 3 Verify the image is registered successfully by checking the **Upgrade Image List** section.

Verify the image shows:

- **Status**: Valid
- **Version**: Correct upgrade version
- **Upload Date**: Registration timestamp

Step 4 In the **Upgrade Image** section, click the + icon to schedule an upgrade.

Step 5 Select the registered upgrade image from the list.

Step 6 Choose the scheduling option:

- **Upgrade Now**: Set the schedule time to 0 hours.
- **Upgrade Later**: Specify hours from now for the upgrade to begin.

Step 7 Click **Submit**.

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

Step 8 After the upgrade completes, from the NFVIS portal, click **Platform** > **About Platform** to verify the new version.

Step 9 Verify all VMs are powered on and operational.

The NFVIS platform is successfully upgraded to the newer version with all VMs powered on and operational.

Upgrade using ISO file

This task enables you to upgrade Cisco NFVIS using an ISO file through command line operations or portal interface.

You can upgrade the system by copying the upgrade image using SCP commands or by uploading through the Cisco NFVIS portal. The upgrade process involves registering the image and then applying the upgrade.

Procedure

Step 1 Copy the upgrade image using one of the following methods:

- To copy the upgrade image, use the **scp** command from Cisco NFVIS CLI:

```
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
```

- To copy the upgrade image, use the **scp** command from remote linux:

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

scp -P22222 Cisco_NFVIS-4.8.0-13-20220123_020232.iso
admin@172.27.250.128:/data/intdatastore/uploads/Cisco_NFVIS-4.8.0-13-20220123_020232.iso
```

- Upload the image to the Cisco NFVIS server using the **System Upgrade** option from the Cisco NFVIS portal.

Note
When 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.

Step 2 Register the image using the **system upgrade image-name** command.

Step 3 Upgrade the image using the **system upgrade apply-image** command.

The system upgrade process completes successfully with the new ISO image installed and operational.

Register an image

Register an image in the system to prepare it for upgrade operations.

Image registration is a prerequisite step before applying system upgrades. This process registers the image file in the system's upgrade directory and validates its integrity.

Procedure

Step 1 Register the image using the system upgrade command.

Example:

```
config terminal
system upgrade image-name Cisco_NFVIS-4.8.0-13-20220123_020232.iso location
/data/intdatastore/uploads/Cisco_NFVIS-4.8.0-13-20220123_020232.iso
commit
```

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

Step 2 Verify the image registration status.

Example:

```
nfvis# show system upgrade

        PACKAGE
NAME                                    LOCATION
        VERSION   STATUS  UPLOAD DATE
-------------------------------------------------------------------------------
Cisco_NFVIS-4.8.0-13-20220123_020232.iso
/data/upgrade/register/Cisco_NFVIS-4.8.0-13-20220123_020232.iso  4.8.0-13  Valid
2022-01-24T02:40:29.236057-00:00


nfvis# show system upgrade reg-info

        PACKAGE
NAME                                    LOCATION
        VERSION   STATUS  UPLOAD DATE
-------------------------------------------------------------------------------
Cisco_NFVIS-4.8.0-13-20220123_020232.iso
/data/upgrade/register/Cisco_NFVIS-4.8.0-13-20220123_020232.iso  4.8.0-13  Valid
2022-01-24T02:40:29.236057-00:00
```

The image is successfully registered in the system and shows a "Valid" status, indicating it is ready for upgrade operations.

Upgrade the registered image

This task applies a registered image upgrade to the system at a scheduled time.

Use this procedure when you need to upgrade the system software to a newer version using a previously registered image.

Procedure

Step 1 To upgrade the registered image, use the following command:

Example:

```
config terminal
system upgrade apply-image Cisco_NFVIS-4.8.0-13-20220123_020232.iso scheduled-time 5
commit
```

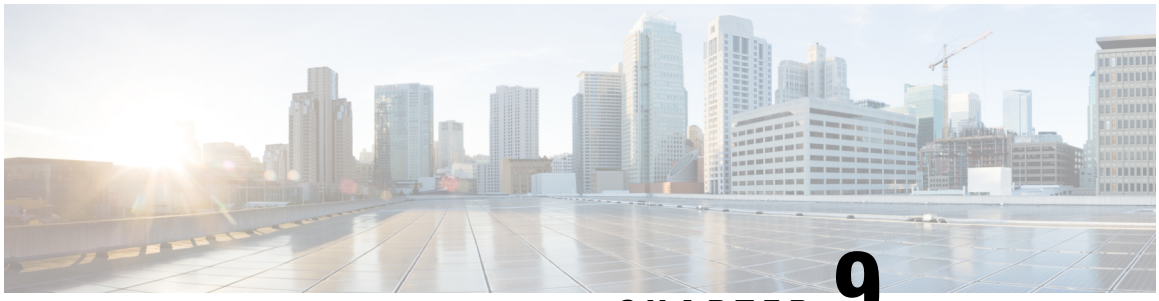
Step 2 To verify the upgrade status, use the **show system upgrade apply-image** command in the privileged EXEC mode.

Example:

```
nfvis# show system upgrade
                                                    UPGRADE  UPGRADE
NAME                                        STATUS  FROM     TO
-------------------------------------------------------------------------
Cisco_NFVIS-4.8.0-13-20220123_020232.iso  SCHEDULED  -        -


        PACKAGE
NAME                                        LOCATION
        VERSION   STATUS  UPLOAD DATE
-------------------------------------------------------------------------------
Cisco_NFVIS-4.8.0-13-20220123_020232.iso
/data/upgrade/register/Cisco_NFVIS-4.8.0-13-20220123_020232.iso  4.8.0-13  Valid
2022-01-24T02:40:29.236057-00:00
```

The upgrade is scheduled and the system displays the upgrade status showing the image as SCHEDULED.



CHAPTER 9

NFVIS Security Considerations

Security considerations

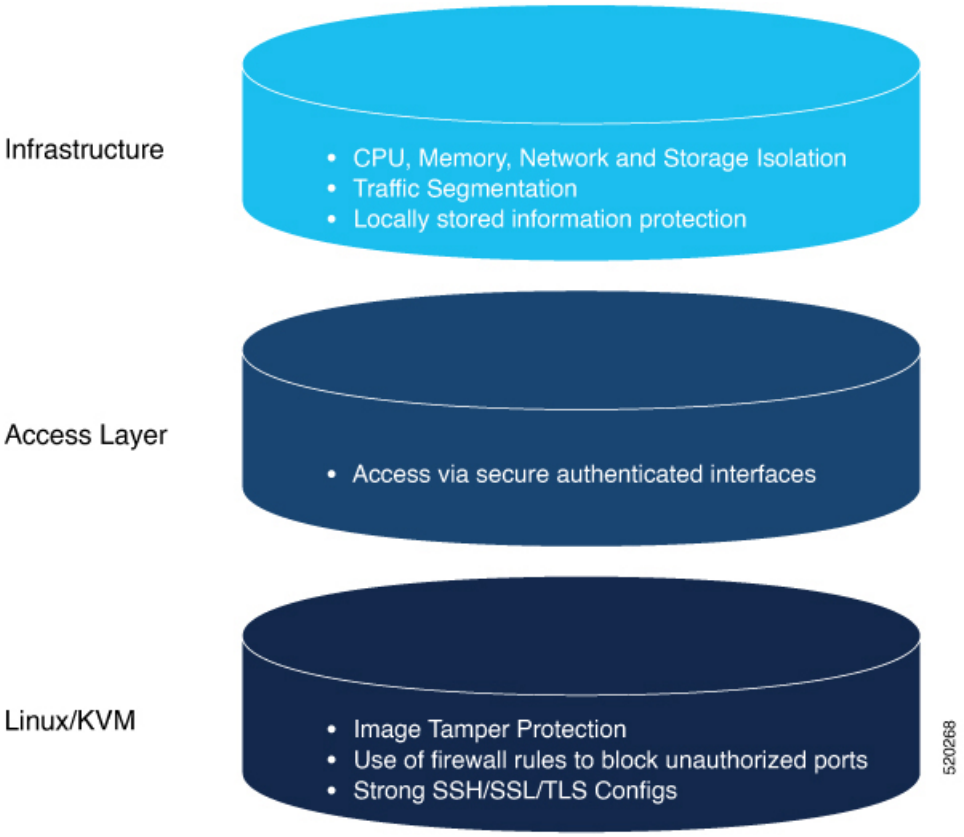
Security considerations are security features and components in NFVIS that

- 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 3: Software layers



Installation

To ensure that the NFVIS software has not been tampered with , the software image is verified before installation using the following mechanisms:

Image tamper protection

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

RPM signing

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 software verifies the signature of all the RPM packages before an installation or upgrade. If signature verification fails during an installation or upgrade, the upgrade is aborted.

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.

Hash verification process

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

```
% /usr/bin/sha512sum <ImageFile>
c2122783efc18b039246ae1bcd4eec4e5e027526967b5b809da5632d462dfa6724a9b20ec318c74548c6bd7e9b8217ce96b5ece93dcddd74fda5e01bb382ad607
  <ImageFile>
```

Secure unique device identification

A Secure Unique Device Identification (SUDI) is a security mechanism that

- provides NFVIS with an immutable identity used to verify that the device is a genuine Cisco product
- ensures that the device is well-known to the customer's inventory system, and
- enables secure, remote on-boarding of devices through authenticated and automated configuration using Zero Touch Provisioning (ZTP).

SUDI certificate and key-pair characteristics

The SUDI consists of an X.509v3 certificate and an associated key-pair which are protected in hardware. The SUDI certificate contains the product identifier and serial number and is rooted in Cisco Public Key Infrastructure. The key pair and the SUDI certificate are inserted into the hardware module during manufacturing, and the private key can never be exported.

The SUDI-based identity enables authenticated and automated configuration using Zero Touch Provisioning (ZTP). This ensures that the orchestration server is talking to a genuine NFVIS device. A backend system can

issue a challenge to the NFVIS device to validate its identity and the device will respond to the challenge using its SUDI based identity. This allows the backend system to not only verify against its inventory that the right device is in the right location but also provide encrypted configuration that can only be opened by the specific device, thereby ensuring confidentiality in transit.

SUDI workflow diagrams

These workflow diagrams illustrate how NFVIS uses SUDI:

Figure 4: Plug and play (PnP) server authentication

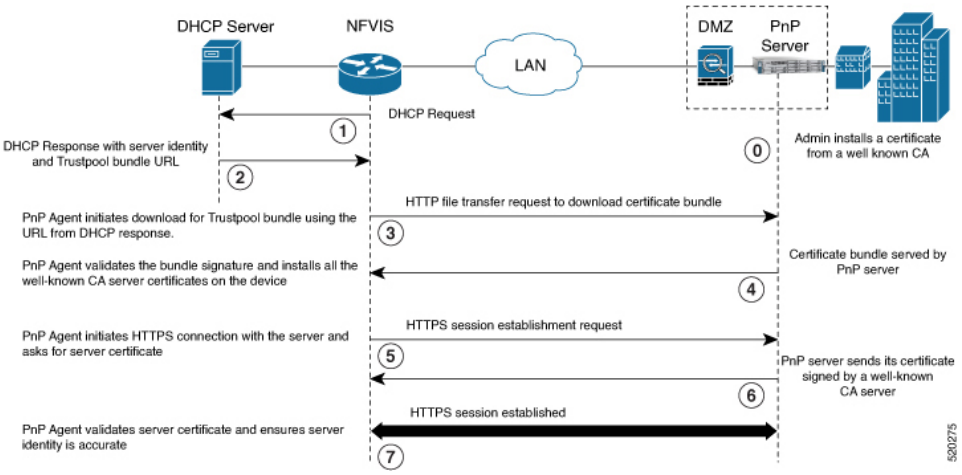
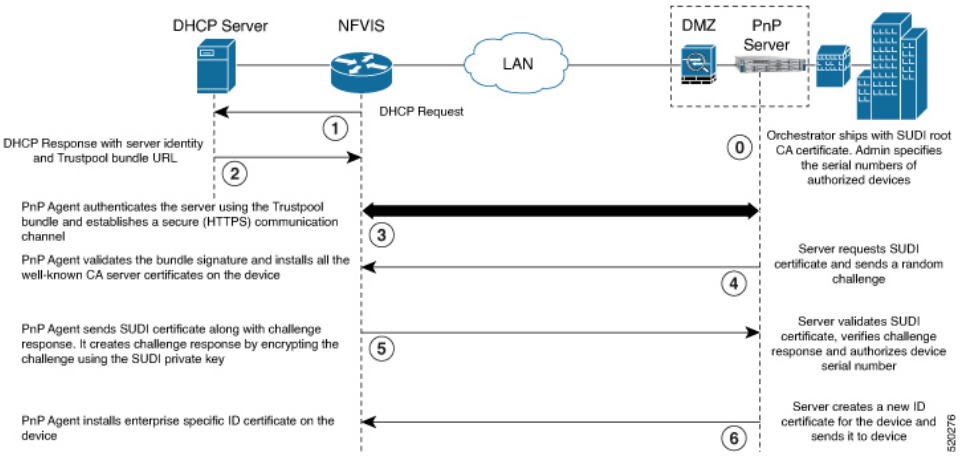


Figure 5: Plug and play device authentication and authorization



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

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.

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.

Additional information

For more information, see Access NFVIS, on page 91.

Restricting login vulnerabilities

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

Enforcement of strong password

An authentication mechanism is only as strong as its credentials. For this reason, it is important to ensure users have strong passwords. NFVIS checks that a strong password is configured as per these rules.

Password must contain:

- At least one uppercase character
- At least one lowercase character
- At least one number
- At least one of these special characters: hash (#), underscore (_), hyphen (-), asterisk (*), or question mark (?)
- Seven characters or more
- The password length should be between 7 and 128 characters.

Configure minimum length for passwords

Configure the minimum password length requirement to strengthen 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 configure the minimum length required for passwords of all users. The minimum length must be between 7 and 128 characters. By default, the minimum length required for passwords is set to 7 characters.

Procedure

Configure the minimum password length using CLI or API.

- CLI:

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

- API:

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

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

Configure password lifetime

Configure password lifetime settings to determine how long a password can be used before the user is required to change it.

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

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.

Recommendation: restrict frequency of login attempts

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 the enforcement of account inactivity using CLI or API.

- Using CLI:

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

- Using API:

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

The default value for inactivity-days is 35.

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

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.

Procedure

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

- Use CLI commands:

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

- Use API:

```
/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

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.

For more information on external authentication, see Configure RADIUS, on page 109 and Configure a TACACS+ server, on page 111.

Authentication cache for external authentication server

An authentication cache for external authentication server is a local storage mechanism that

- stores hash entries using the username and OTP after successful TACACS server authentication
- maintains expiration timestamps that match the SSH session idle timeout value of 15 minutes, and
- authenticates subsequent requests with the same username against the local hash value first before contacting the TACACS server.

Authentication cache behavior

The NFVIS portal uses the same One-Time Password (OTP) for all API calls after the initial authentication. The API calls fail as soon as the OTP expires. This feature supports TACACS OTP authentication with the NFVIS portal.

After you have successfully authenticated through the TACACS server using an OTP, NFVIS creates a hash entry using the username and the OTP and stores this hash value locally. This locally stored hash value has an expiration time stamp associated with it. The time stamp has the same value as the SSH session idle timeout value which is 15 minutes. All the subsequent authencation requests with the same username are authenticated against this local hash value first. If the authentication fails with the local hash, NFVIS authenticates this request with TACACS server and creates a new hash entry when the authentication is successful. If a hash entry already exists, its time stamp is reset to 15 minutes.

If you are removed from the TACACS server after successfully logging into the portal, you can continue to use the portal until the hash entry in NFVIS expires.

When you explicitly log out from the NFVIS portal or are logged out due to idle time, the portal calls a new API to notify NFVIS backend to flush the hash entry. The authentication cache and all of its entries are cleared out after NFVIS reboot, factory reset, or upgrade.

Role based access control

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 doesn't pertain to them, and
- uses an employee's role in the enterprise to determine the permissions granted.

NFVIS user roles and privileges

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.

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

The 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, can't 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 is 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's permitted for each role. If a user is found to have unnecessary access to a certain system, change the user's role.

Granular Role-Based Access Control

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 Granular role-based access control, on page 103.

Recommendation: restrict device accessibility

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

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.

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.

Security benefits

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.

Enable only essential ports by default

Only those services which are absolutely necessary to setup and manage NFVIS are available by default. This removes the user effort needed to configure firewalls and deny access to unnecessary services.

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, *Introduction about 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

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.

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, on page 95

Access privileged debug shell

This task provides secure shell access to NFVIS for interactive debugging when standard troubleshooting methods are insufficient and Cisco Technical Assistance Center or development team requires system-level access.

The super-user account on NFVIS is disabled by default, to prevent all unrestricted, potentially adverse, system-wide changes and NFVIS does not expose the system shell to the user. However, for some hard to debug issues on the NFVIS system, the Cisco Technical Assistance Center team (TAC) or development team might require shell access to the customer's NFVIS. NFVIS has a secure unlock infrastructure to ensure that privileged debug access to a device in the field is restricted to authorized Cisco employees. To securely access the Linux shell for this kind of interactive debugging, a challenge-response authentication mechanism is used between NFVIS and the Interactive debugging server maintained by Cisco. The admin user's password is also required in addition to the challenge-response entry to ensure that the device is accessed with the customer's consent.

Before you begin

Follow these steps to access the shell for interactive debugging:

Procedure

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

Example:

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

Step 2 Copy the challenge string displayed on the screen.

Example:

```
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.

Development or TAC team uses the shell to proceed with the debugging.

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

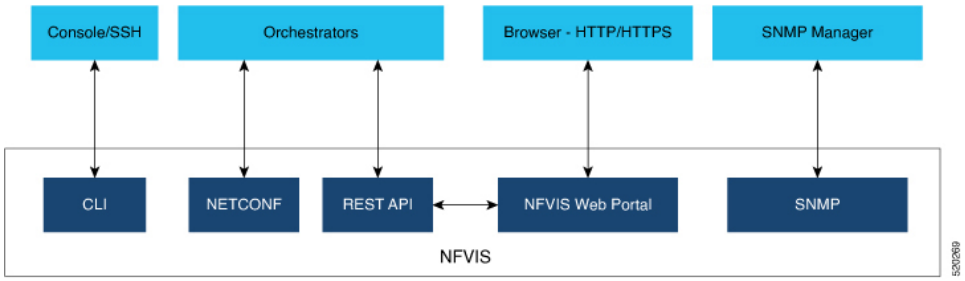
You have successfully accessed the privileged debug shell and can perform interactive debugging with Cisco support team assistance.

Secure interfaces

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 access is allowed using the interfaces shown in the diagram. The following sections describe security best practices for these interfaces to NFVIS.

Figure 6: Secure interfaces



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.

SSH implementation details

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

By logging in via SSH, a user establishes a session with NFVIS. While the user is logged in, if the user leaves the logged-in session unattended, this can expose the network to a security risk. Session security limits the risk of internal attacks, such as one user trying to use another user's session.

To mitigate this risk, NFVIS times out CLI sessions after 15 minutes of inactivity. When the session timeout is reached, the user is automatically logged out.

NETCONF

The Network Configuration Protocol (NETCONF) is a network management protocol 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 capabilities

NETCONF allows NFVIS to expose an XML-based API that the network operator can use to set and get configuration data and event notifications securely over SSH.

For more information see, NETCONF event notifications, on page 207.

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.

REST API session management

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.

Details on all the REST APIs can be found in the NFVIS API Reference guide.

NFVIS web portal

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

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

NFVIS encrypts the user's session. The AES-256-CBC cipher is used to encrypt the session contents with an HMAC-SHA-256 authentication tag. A random 128-bit Initialization Vector is generated for each encryption operation.

An Audit record is started when a portal session is created. Session information is deleted when the user logs out or when the session times out.

The default idle timeout for portal sessions is 15 minutes. However, this can be configured for the current session to a value between 5 and 60 minutes on the Settings page. Auto-logout will be initiated after this period. Multiple sessions are not permitted in a single browser. The Maximum number of concurrent sessions are set to 30.

The NFVIS portal utilizes cookies to associate data with the user. It uses 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're currently authenticated. To prevent this, NFVIS uses **CSRF** tokens to validate every REST API that is invoked during each session.

URL Redirection

In typical web servers, when a page is not found on the web server, the user gets a 404 message; for pages that exist, they get a login page. The security impact of this is that an attacker can perform a brute force scan and easily detect which pages and folders exist.

To prevent this on NFVIS, all non-existent URLs prefixed with the device IP are redirected to the portal login page with a 301 status response code. This means that irrespective of the URL requested by an attacker, they will always get the login page to authenticate themselves.

All HTTP server requests are redirected to HTTPS and have these headers configured:

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

Portal Disablement

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:

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

HTTPS

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.

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).

Manage certificates

Replace the default self-signed certificate with a CA-signed certificate to enhance security and follow best practices for certificate management in NFVIS.

Procedure

Step 1 Generate a Certificate Signing Request (CSR) on NFVIS.

A Certificate Signing request (CSR) is a file with a block of encoded text that is given to a Certificate Authority when applying for an SSL Certificate. This file contains information that should be included in the certificate such as the organization name, common name (domain name), locality, and country. The file also contains the public key that should be included in the certificate. NFVIS uses a 2048-bit public key since encryption strength is higher with a higher key size.

To generate a CSR on NFVIS, run the following command:

```
nfvis# system certificate signing-request [common-name  country-code  locality  organization
organization-unit-name  state]
```

The CSR file is saved as */data/intdatastore/download/NFVIS.CSR*.

Step 2 Get an SSL certificate from a CA using the CSR.

From an external host, use the scp command to download the Certificate Signing Request.

```
[myhost:/tmp] > scp -P 22222 admin@<NFVIS-IP>:/data/intdatastore/download/nfvis.csr
<destination-file-name>
```

Contact a Certificate authority to issue a new SSL server certificate using this CSR.

Step 3 Install the CA Signed Certificate.

From an external server, use the scp command to upload the certificate file into NFVIS to the *data/intdatastore/uploads/* directory.

```
[myhost:/tmp] > scp -P 22222 <certificate file> admin@<NFVIS-IP>:/data/intdatastore/uploads
```

Install the certificate in NFVIS using the following command.

```
nfvis# system certificate install-cert path file:///data/intdatastore/uploads/<certificate file>
```

Step 4 Switch to using the CA Signed Certificate.

Use the following command to start using the CA signed certificate instead of the default self-signed certificate.

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

NFVIS now uses the CA-signed certificate instead of the default self-signed certificate for secure HTTPS communications.

SNMP access

SNMP access is a network management mechanism that

- collects and organizes information about managed devices on IP networks
- modifies that information to change device behavior, and
- provides different security levels through three significant versions.

SNMP version security features

Three significant versions of SNMP have been developed. NFVIS supports SNMP version 1, version 2c and version 3. SNMP versions 1 and 2 use community strings for authentication, and these are sent in plain-text. So, it is a security best practice to use SNMP v3 instead.

SNMPv3 provides secure access to devices by using three aspects: - users, authentication, and encryption. SNMPv3 uses the USM (User-based Security Module) for controlling access to information available via SNMP. The SNMP v3 user is configured with an authentication type, a privacy type as well as a passphrase. All users sharing a group utilize the same SNMP version, however, the specific security level settings (password, encryption type, etc.) are specified per-user.

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

```
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

```
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.
```

For more details on SNMP, see SNMP, on page 207.

Recommendation: legal notification banners

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.

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

Discuss this issue with your own legal counsel to ensure that the notification banner meets company, local, and international legal 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

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.

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, on page 148.

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 reset scenarios

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.
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.

For more information, see Reset to factory default, on page 147.

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

Network devices can connect to the OOB network in different ways:

- NFVIS supports a built-in management interface that can be used to connect to the OOB network. NFVIS allows the configuration of a predefined physical interface, the MGMT port on the ENCS, as a dedicated management interface.
- 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

For further information, see Configure the IP receive ACL, on page 95 and Configure port 22222 and management interface ACL, on page 96.

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.

Additional information

For further information see VLAN configuration for NFVIS management traffic, on page 94.

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.

Locally stored information protection

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

NFVIS requires locally-stored passwords for specific cases even when centralized AAA servers are deployed:

- 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

A secure copy from NFVIS is initiated through the SCP command. The secure copy (SCP) command allows only the admin user to securely copy files from NFVIS to an external system, or from an external system to NFVIS.

The syntax for the SCP command is:

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

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 (address)/(mask lenth)  service scpd priority (number) action
 accept
commit
```

To prevent users from accessing system directories, secure copy can be performed only to or from intdatastore:, extdatastore1:, extdatastore2:, usb: and nfs:, if available. Secure copy can also be performed from logs and techsupport.

Logging

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.

Logged information

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. For more information on syslog messages and event notifications, see Appendix.

Virtual machine security

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.

VNC console access process

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.

VNC console command example

```
nfvis# vncconsole start deployment-name 1510614035 vm-name ROUTER
vncconsole-url :6005/vnc_auto.html
```

Pointing your browser to https://<NFVIS ip>:6005/vnc_auto.html will connect to the ROUTER VM's VNC console.

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.

Additional information

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.

For more information see, VM deployment parameters, on page 47

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.

Checksum verification process

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.

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

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.

More details can be found at Register a remote VM image, on page 27

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

The Network Function Virtualization (NFV) architecture consists of:

- Virtualized network functions (VNFs), which are Virtual Machines running software applications that deliver network functionality such as a router, firewall, load balancer, and so on.

- Network functions virtualization infrastructure, which consists of the infrastructure components—compute, memory, storage, and networking, on a platform that supports the required software and hypervisor.

With NFV, network functions are virtualized so that multiple functions can be run on a single server. As a result, less physical hardware is needed, allowing for resource consolidation. In this environment, it is essential to simulate dedicated resources for multiple VNFs from a single, physical hardware system. Using NFVIS, VMs can be deployed in a controlled manner such that each VM receives the resources it needs. Resources are partitioned as needed from the physical environment to the many virtual environments.

As a result, CPU, memory, network and storage are protected.

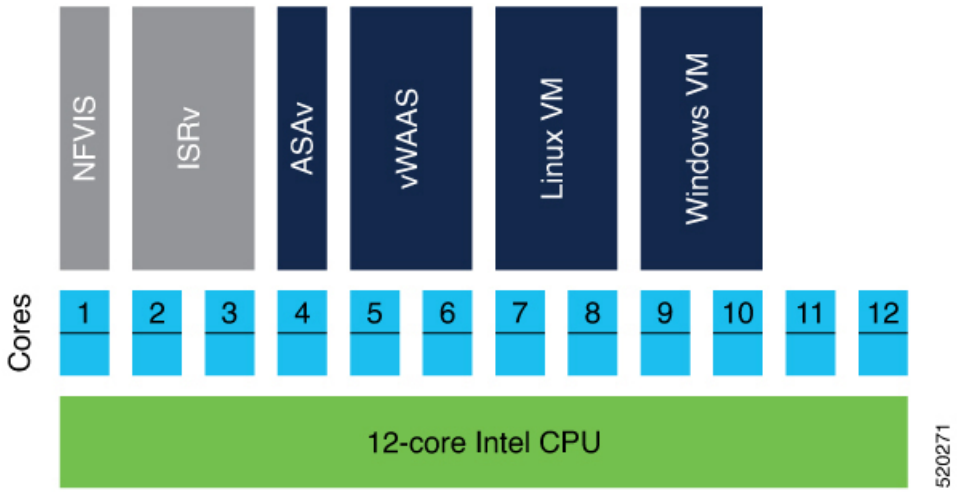
CPU isolation

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

NFVIS handles CPU allocation differently based on VM latency requirements.



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

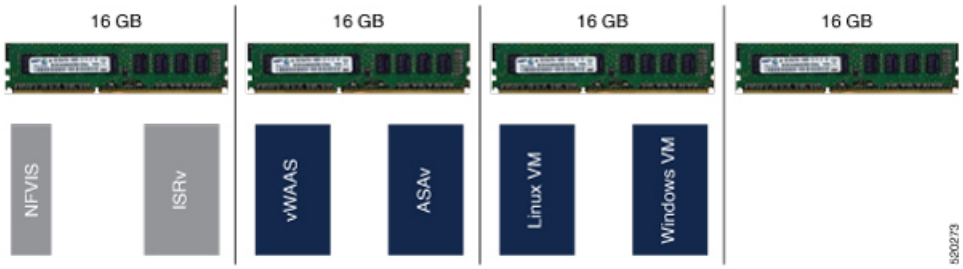
Memory allocation is a resource management mechanism that

- 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.

Memory allocation process

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.

VMs are not allowed to directly access the host file system and storage.



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

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 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

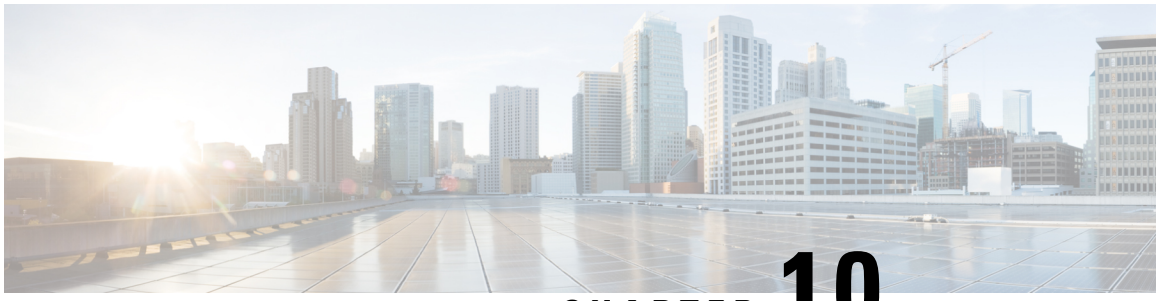
A secure development lifecycle is a software development process that

- 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.

NFVIS SDL processes

Every NFVIS release goes through these processes:

- Following Cisco-internal and market-based Product Security Requirements
- Registering 3rd party software with a central repository at Cisco for vulnerability tracking
- Periodically patching software with known fixes for CVEs
- Designing software with Security in mind
- Following secure coding practices such as using vetted common security modules like CiscoSSL, running Static Analysis and implementing input validation for Preventing command injection
- Using Application Security tools such as IBM AppScan, Nessus, and other Cisco internal tools.



CHAPTER 10

FIPS Mode on Cisco NFVIS

FIPS mode on NFVIS

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.

FIPS mode behavior

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.

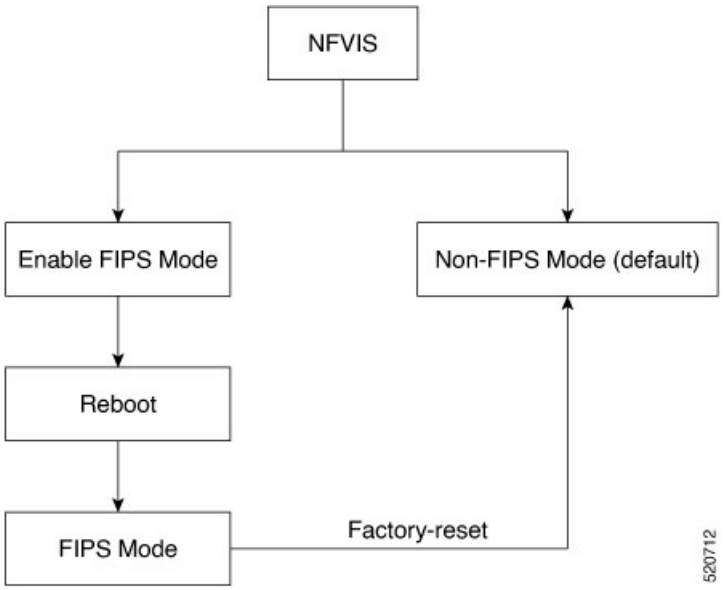
Disable FIPS mode

To disable FIPS mode, you must first perform a factory reset. After the reset, disable FIPS mode and reboot the device. The same steps apply when re-enabling FIPS mode.

Configure FIPS mode

FIPS mode enables FIPS 140-3 compliant operation for SSH, TLS and SNMP protocols on NFVIS.

NFVIS supports both FIPS and non-FIPS mode of operation.



Before you begin

Ensure that SNMP v1, v2, or v3 with MD5 auth protocol is not configured, as FIPS mode configuration will be terminated if these are present.

Procedure

Step 1 Enable FIPS mode.

Example:

```
config terminal
security fips
commit
```

Only after NFVIS reboot after the configuration FIPS mode is enabled.

FIPS mode can be disabled only with factory reset. If you try to disable FIPS mode with the no form of the command:

Example:

```
config terminal
no security fips
commit
Aborted: This command can only be removed while factory reset
```

Example:

The following is an example, where FIPS mode is successfully configured, but not enabled:

```
nfvis# show security
security fips-status CONFIGURED_REBOOT_TO_ENABLE
```

Step 2 Verify the status of the FIPS mode after reboot:

Example:

```
nfvis# show security
security fips-status ENABLED
```

FIPS mode configuration is terminated when:

- SNMP v1 or v2 is configured.

  The following is an example of FIPS mode configuration failure when SNMP v1 or v2 is configured:

  ```
  config terminal
  security fips
  commit
  Aborted: SNMP version 1 and/or SNMP version 2 is configure. Please unconfigure SNMPv1 and SNMPv2
   and then try again
  ```

- SNMP v3 is configured with auth protocol MD5.

  The following is an example of FIPS mode configuration failure when SNMP v3 is configured with auth protocol md5:

  ```
  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, SNMP v1or v2 and SNMP v3 with auth protocol MD5 cannot be configured.

```
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.

Backup and restore behavior for FIPS mode

This topic provides the details on how FIPS mode status is handled during NFVIS backup and restore operations.

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.

```
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
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#

Restore system configurations
nfvis#
nfvis# show running-config security
security fips
nfvis# show security
security fips-status ENABLED
nfvis#

After restore
nfvis# show running-config security
% No entries found.
nfvis# show security
security fips-status UNCONFIGURED_REBOOT_TO_DISABLE
nfvis#

After reboot
nfvis# show running-config security fips
% No entries found.
nfvis# show security fips-status
security fips-status DISABLED
nfvis#
```

FIPS operational status

This topic provides reference information about the operational states available when you try to ENABLE FIPS mode and the possible operational state transitions for FIPS mode.

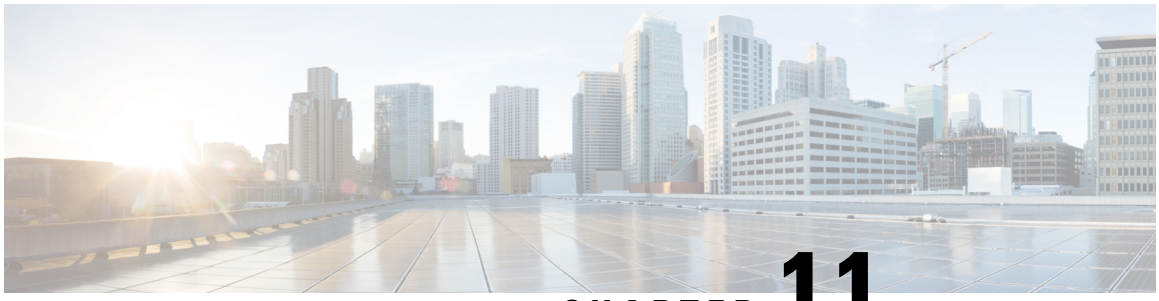
These are the operational states when you try to ENABLE FIPS mode:

- DISABLED
- CONFIGURED-REBOOT-TO-ENABLE
- ENABLED
- UNCONFIGURED-REBOOT-TO-DISABLE
- FAILED

Table 19: FIPS mode operational state transitions

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
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

From	To	Description
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



CHAPTER 11

System Logging

System logs

System logs are diagnostic files that

- 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.

System log 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 20: 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

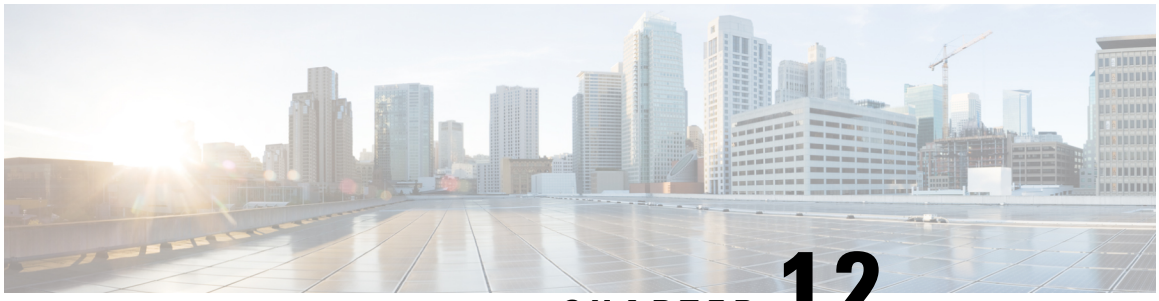
To verify the system log configuration, use the **show system logging-level** command as shown below:

```
nfvis# show system logging-level
system logging-level configuration error
system logging-level operational warning
```

System log APIs and commands:

Table 21: 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



CHAPTER 12

Cisco NFVIS Monitoring

NFVIS monitoring

Configure syslog

The Syslog feature allows event notifications from NFVIS to be sent to remote syslog servers for centralized log and event collection. The syslog messages are based on the occurrence of specific events on the device and provide configuration and operational information such as creation of users, changes to the interface status, and failed login attempts. Syslog data is critical to recording day-to-day events as well as notifying operational staff of critical system alerts.

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

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
```

Procedure

Step 1 Configure a remote syslog server by specifying its IP address or DNS name along with the protocol to send syslogs and the port number on the syslog server.

Example:

```
configure terminal
system settings logging host 172.24.22.186
```

```
port 3500
transport tcp
commit
```

Note
A maximum of 4 remote syslog servers can be configured. The remote syslog server can be specified using its IP address or DNS name. The default protocol for sending syslogs is UDP with a default port of 514. For TCP, the default port is 601.

Step 2 Configure syslog severity to describe the importance of the syslog message.

Example:

```
configure terminal
system settings logging severity <debug | informational | notice | warning| error| critical | alert
 | emergency>
```

Table 22: 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 of syslogs is informational which means all syslogs at informational severity and higher will be logged. Configuring a value for severity will result in syslogs at the configured severity and syslogs which are more severe than the configured severity.

Step 3 Configure syslog facility to logically separate and store syslog messages on the remote syslog server.

Example:

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

The syslog facility can be used to logically separate and store syslog messages on the remote syslog server. For example, syslogs from a particular NFVIS can be assigned a facility of local0 and can be stored and processed in a different directory location on the syslog server. This is useful to separate it from syslogs with a facility of local1 from another device.

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

By default, NFVIS sends syslogs with the facility of local7

You have successfully configured syslog settings including remote server, severity level, and facility. NFVIS will now send syslog messages to the configured remote syslog servers for centralized log and event collection.

NETCONF event notifications

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 to Event Notifications.

SNMP support on NFVIS

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.

SNMP framework components

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

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

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

The command for SNMP GET is:

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

SNMP walk

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

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

The command for SNMP walk with SNMP v2 is:

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

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

SNMP notifications are asynchronous messages that

- 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

SNMP notifications include these 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.

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

SNMP versions

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

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

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)

Note * indicates the support data required for a single CPU core starting from NFVIS 3.12.3 release.

CISCO environmental MIB (1.3.6.1.4.1.9.9.13)

- Voltage Sensor:
  - ciscoEnvMonVoltageStatusDescr
  - ciscoEnvMonVoltageStatusValue
- Temperature Sensor:
  - ciscoEnvMonTemperatureStatusDescr
  - ciscoEnvMonTemperatureStatusValue
- Fan Sensor
  - ciscoEnvMonFanStatusDescr
  - ciscoEnvMonFanState

Note Sensor support for hardware platforms:

- ENCS 5400 series: all
- ENCS 5100 series: none
- UCS-E: voltage, temperature
- UCS-C: all
- CSP: CSP-2100, CSP-5228, CSP-5436 and CSP5444 (Beta)

CISCO environmental monitor MIB notification

- 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

Configure SNMP support to enable network monitoring and management through SNMP protocols. This configuration allows administrators to monitor device status, collect statistics, and receive trap notifications.

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

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

Follow these steps to configure SNMP support:

Procedure

Step 1 Create SNMP community.

Example:

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

SNMP community name string supports [A-Za-z0-9_-] and maximum length of 32. NFVIS supports only **readOnly** access.

Step 2 Create SNMP Group.

Example:

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

Variables	Description
group_name	Group name string. Supporting string is [A-Za-z0-9_-] and maximum length is 32.
context	Context string, default is SNMP. Maximum length is 32. Minimum length is 0 (empty context).
version	1, 2 or 3 for SNMP v1, v2c and v3.
security_level	authPriv, authNoPriv, noAuthNoPriv **Note** SNMP v1 and v2c uses noAuthNoPriv only.

Variables	Description
notify_list/read_list/write_list	It can be any string. read_list and notify_list is required to support data retrieval by SNMP tools. write_list can be skipped because NFVIS SNMP does not support SNMP write access.

Step 3 Create SNMP v3 user based on the security level.

When security level is authPriv

Example:

```
configure terminal
snmp user <user_name> user-version 3 user-group <group_name> auth-protocol <auth> priv-protocol <priv>
 passphrase <passphrase_string>
```

Example:

```
configure terminal
snmp user <user_name> user-version 3 user-group <group_name> auth-protocol <auth> priv-protocol <priv>
 passphrase <passphrase_string> encryption-passphrase <encryption_passphrase>
```

When security level is authNoPriv:

Example:

```
configure terminal
snmp user <user_name> user-version 3 user-group <group_name> auth-protocol <auth> passphrase
<passphrase_string>
```

When security level is noAuthNopriv

Example:

```
configure terminal
snmp user <user_name> user-version 3 user-group <group_name>
```

Variables	Description
user_name	User name string. Supporting string is [A-Za-z0-9_-] and maximum length is 32. This name has to be the same as community_name.
version	1 and 2 for SNMP v1 and v2c.
group_name	Group name string. This name has to be same as the group name configured in the NFVIS.
auth	md5 or sha
priv	aes or des
passphrase_string	Passphrase string. Supporting string is [A-Za-z0-9\-_#@%$*&!].
encryption_passphrase	Passphrase string. Supporting string is [A-Za-z0-9\-_#@%$*&!]. The user must configure passphrase first to 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 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 noAuthNoPriv
```

Variables	Description
host_name	User name string. Supporting string is [A-Za-z0-9_-] and maximum length is 32. This is not FQDN host name, but an alias to IP address of traps.
ip_address	IP address of traps server.
port	Default is 162. Change to other port number based on your own setup.
user_name	User name string. Must be the same as user_name configured in NFVIS.
version	1, 2 or 3 for SNMP v1, v2c or v3.
security_level	authPriv, authNoPriv, noAuthNoPriv **Note** SNMP v1 and v2c uses noAuthNoPriv only.

SNMP community, 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 Examples

This reference provides standard configuration syntax for SNMP versions 1, 2, and 3 on NFVIS systems. These examples serve as a ready reckoner for administrators to implement SNMP monitoring and trap configurations.

SNMP v3 configuration

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 privprotocol aes
passphrase changePassphrase encryption-passphrase encryptPassphrase
! configure snmp host to enable snmp v3 trap
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
host-security-level authPriv host-port 162
!!
```

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

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 sha priv-protocol aes
passphrase changePassphrase
! configure snmp host to enable snmp v3 trap
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
host-security-level authPriv host-port 162
!!
```

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

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 configuration verification

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

SNMP agent status verification

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 verification

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 verification

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 verification

Use the **show running-config SNMP** command to verify the interface configuration for snmp:

```
nfvis# show running-config snmp

snmp agent enabled true
snmp agent engineID 00:00:00:09:11:22:33:44:55:66:77:88
snmp enable traps linkUp
snmp community pub_comm
community-access readOnly
!
snmp community tachen
community-access readOnly
!
snmp group tachen snmp 2 noAuthNoPriv
read   test
write  test
notify test
!
snmp group testgroup snmp 2 noAuthNoPriv
read   read-access
write  write-access
notify notify-access
!
snmp user public
user-version  2
user-group    2
auth-protocol md5
priv-protocol des
!
snmp user tachen
user-version 2
user-group   tachen
!
snmp host host2
host-port           162
host-ip-address     2.2.2.2
host-version        2
host-security-level noAuthNoPriv
host-user-name      public
!
```

Upper limit for SNMP configurations

Upper limit for SNMP configurations:

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

SNMP Support APIs and Commands

APIs	Commands
• /api/config/snmp/agent • /api/config/snmp/communities • /api/config/snmp/enable/traps • /api/config/snmp/hosts • /api/config/snmp/user • /api/config/snmp/groups	• agent • community • trap-type • host • user • group

In Cisco NFVIS Release 26.1.1, 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 will generate warning messages for SNMP1, SNMP2 and non-auth type for SNMPv3.

Table 23: SNMP

Sl. No	SNMP Type	SNMP Version	Security Protocol	System Mode
1	SNMP HOST SNMP GROUP SNMP USER	V3	noAuthNoPriv noAuthNoPriv auth-protocol: md5,sha priv-protocol: des	insecure
2	SNMP HOST SNMP GROUP SNMP USER	V3	AuthNoPriv AuthNoPriv auth-protocol: md5,sha priv-protocol: des	insecure

Sl. No	SNMP Type	SNMP Version	Security Protocol	System Mode
3	SNMP HOST SNMP GROUP SNMP USER	V3	authPriv authPriv auth-protocol: sha256 priv-protocol: aes	secure
4	All type SNMP	V1	All type	insecure
5	All type SNMP	V2	All type	insecure

System 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 the host CPU usage

The percentage of time spent by the CPU in various states, such as executing user code, executing system code, waiting for IO operations, etc. is displayed for the specified duration.

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 collectd daemon.

Use the **show system-monitoring host port stats** command to display the outputs of the calculations done by collectd for packets/sec, errors/sec and now kilobits/sec. Use the **system-monitoring host port table** command to display the outputs of the collectd stats average for last 5 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 can't 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

Field	Description
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
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:

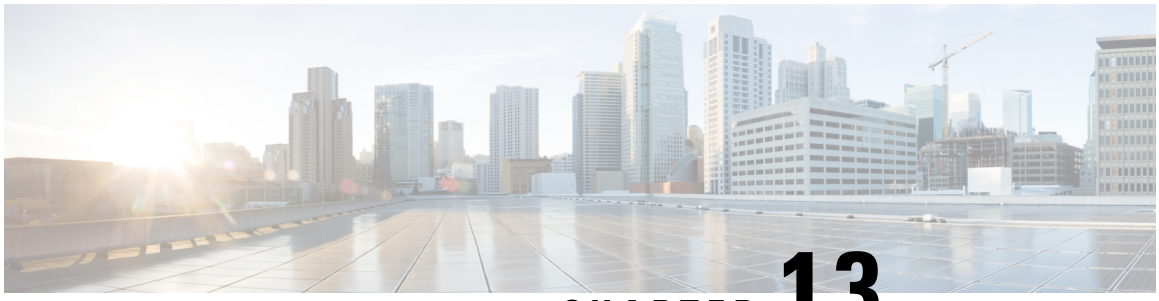
Field	Description
total-packets-per-sec	Total packets received and transmitted per second
rx-packets-per-sec	Packets received per second
tx-packets-per-sec	Packets transmitted per second
total-errors-per-sec	Total error rate for packet reception and transmission
rx-errors-per-sec	Error rate for receiving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



CHAPTER 13

Troubleshooting

Log 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:

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.
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 about one VM index by <num>
`support virsh domiflist <num>`	To display the list of interfaces on VM index by <num> and MAC address of the VNICs.

Command	Description
support flush cache memory	To clear cache memory and free up some system memory for seamless performance of Cisco NFVIS. Clearing caches using 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	

Log files

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

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

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
```

Table 24: Types of errors

Error	Scenario
Port/vnic not found	When non-existing interface is given as input.
File/directory not created	When the system is running out of disk space.
The **tcpdump** command fails	When the system is running out of disk space.

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.