

NFVIS serviceability, monitoring and observability

Objective

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

- Log files, on page 1
- Host vs. virtual machine monitoring, on page 3
- Cisco ThousandEyes, on page 59
- Troubleshooting NFVIS, on page 70

Log files

Objective

System logs are diagnostic files that

Log file types and configuration

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

NFVIS generates log files for troubleshooting issues. The configuration log and the operational log are the two main system log files. The configuration log has information related to configurations and actions performed on the system such as creation of networks. The operational log has information related to system operation such as statistics collection and monitoring.

Log entries can be one of these types:

Table 1: 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 2: System log APIs and commands

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

Host vs. virtual machine monitoring

Objective

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

Monitoring

System monitoring is a capability that

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

For larger durations average values are displayed.

Collection of system monitoring statistics

System monitoring statistics are displayed for the requested duration. The default duration is five minutes.

The supported duration values are 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, 30D with min as minutes, h and H as hours, d and D as days.

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

Example

This is a sample output of system monitoring statistics:

```
nfvis# show system-monitoring host cpu stats cpu-usage 1h state non-idle
system-monitoring host cpu stats cpu-usage 1h state non-idle
  collect-start-date-time  2019-12-20T11:27:20-00:00
  collect-interval-seconds 10
  cpu
     id              0
     usage-percentage "[7.67, 5.52, 4.89, 5.77, 5.03, 5.93, 10.07, 5.49,
     ...
```

The time at which the data collection started is displayed as collect-start-date-time. The sampling interval at which data is collected is shown as collect-interval-seconds. The data for the requested metric like host CPU statistics is displayed as an array. The first data point in the array was collected at the specified collect-start-date-time and each subsequent value at an interval specified by collect-interval-seconds. In the sample output, CPU id 0 has a utilization of 7.67% on 2019-12-20 at 11:27:20 as specified by collect-start-date-time. 10 seconds later, it had a utilization of 5.52% since the collect-interval-seconds is 10. The third value of CPU-utilization is 4.89% at 10 seconds after the second value of 5.52% and so on.

The sampling interval shown as collect-interval-seconds changes based on the specified duration. For higher durations, the collected statistics are averaged at a higher interval to keep the number of results reasonable.

Host system monitoring

NFVIS provides system monitoring commands and APIs to monitor the host's CPU utilization, memory, disk and ports.

Monitor host CPU usage

The percentage of time spent by the CPU in various states, such as executing user code, executing system code, waiting for IO operations, etc. is displayed for the specified duration.

CPU-state	Description
non-idle	100 – idle-CPU-percentage
interrupt	Indicates the percentage of the processor time spent in servicing interrupts
nice	The nice CPU state is a subset of the user state and shows the CPU time used by processes that have a lower priority than other tasks.
system	The system CPU state shows the amount of CPU time used by the kernel.
user	The user CPU state shows CPU time used by user space processes
wait	Idle time while waiting for an I/O operation to complete

The non-idle state is what the user usually needs to monitor. Use this CLI or API for monitoring CPU usage:

```
nfvis# show system-monitoring host cpu stats cpu-usage duration state cpu-state
```

```
/api/operational/system-monitoring/host/cpu/stats/cpu-usage/<duration>,<cpu-state>?deep
```

The data is also available in an aggregate form for the minimum, maximum, and average CPU utilization using this CLI and API:

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

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

Monitor the host port statistics

The statistics collection for non-switch ports is handled by the collectd daemon on all platforms. The input and output rate calculation per port is enabled and the rate calculations are done by the collected daemon.

Use the **show system-monitoring host port stats** command to display the outputs of the calculations done by collectd for packets/sec, errors/sec and now kilobits/sec. Use the system-monitoring host port table command to display the outputs of the collectd stats average for last 5 minutess for packets/sec and kilobits/sec values.

Monitor host memory

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

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

Use this CLI or API for monitoring host memory:

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

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

The data is also available in an aggregate form for the minimum, maximum, and average memory utilization using this CLI and API:

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

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

Monitor host disks

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

Monitor host disks operations

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

Field	Description
IO-time-ms	Average time spent doing I/O operations in milliseconds
IO-time-weighted-ms	Measure of both I/O completion time and the backlog that may be accumulating
merged-reads-per-sec	The number of read operations that could be merged into already queued operations, that is one physical disk access served two or more logical operations. The higher the merged reads, the better the performance.
merged-writes-per-sec	The number of write operations that could be merged into other already queued operations, that is one physical disk access served two or more logical operations. The higher the merged reads, the better the performance.
bytes-read-per-sec	Bytes read per second
bytes-written-per-sec	Bytes written per second
reads-per-sec	Number of read operations per second
writes-per-sec	Number of write operations per second
time-per-read-ms	The average time a read operation takes to complete
time-per-write-ms	The average time a write operation takes to complete
pending-ops	The queue size of pending I/O operations

Use this CLI or API for monitoring host disks:

```
nfvis# show system-monitoring host disk stats disk-operations duration

/api/operational/system-monitoring/host/disk/stats/disk-operations/<duration>?deep
```

Monitor host disk space

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

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

Use this CLI or API for monitoring host disk space:

```
nfvis# show system-monitoring host disk stats disk-space duration
```

```
/api/operational/system-monitoring/host/disk/stats/disk-space/<duration>?deep
```

Monitor host ports

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

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

Use this CLI or API for monitoring host ports:

```
nfvis# show system-monitoring host port stats port-usage duration
```

```
/api/operational/system-monitoring/host/port/stats/port-usage/<duration>?deep
```

The data is also available in an aggregate form for the minimum, maximum, and average port utilization using this CLI and API:

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

```
/api/operational/system-monitoring/host/port/table/port-usage/<duration>,<name>?deep
```

VNF system monitoring

NFVIS provides system monitoring commands and APIs to get statistics on the virtualized guests deployed on NFVIS. These statistics provide data on the VM's CPU utilization, memory, disk and network interfaces.

Monitoring the VNF CPU usage

The CPU utilization of a VM is displayed for the specified duration using these fields:

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

Use this CLI or API to monitor the CPU usage of the VNF:

```
nfvis# show system-monitoring vnf vcpu stats vcpu-usage duration

/api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/<duration>?deep

/api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/<duration>/vnf/<vnf-name>?deep
```

Monitoring VNF memory

These statistics are collected for VNF memory utilization:

Field	Description
total-MB	Total memory of the VNF in MB
RSS-MB	Resident Set Size (RSS) of the VNF in MB The Resident Set Size (RSS) is the portion of memory occupied by a process, that is held in the RAM. The rest of the occupied memory exists in the swap space or file system, because some parts of the occupied memory are paged out, or some parts of the executable are not loaded.

Use this CLI or API to monitor VNF memory:

```
nfvis# show system-monitoring vnf memory stats mem-usage duration

/api/operational/system-monitoring/vnf/memory/stats/mem-usage/<duration>?deep

/api/operational/system-monitoring/vnf/memory/stats/mem-usage/<duration>/vnf/<vnf-name>?deep
```

Monitoring VNF disks

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

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

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

Use this CLI or API to monitor VNF disks:

```
nfvis# show system-monitoring vnf disk stats duration
```

```
/api/operational/system-monitoring/vnf/disk/stats/disk-operations/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/disk/stats/disk-operations/<duration>/vnf/<vnf-name>?deep
```

Monitoring VNF ports

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

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

Use this CLI or API to monitor VNF ports:

```
nfvis# show system-monitoring vnf port stats port-usage duration
```

```
/api/operational/system-monitoring/vnf/port/stats/port-usage/<duration>?deep
```

```
/api/operational/system-monitoring/vnf/port/stats/port-usage/<duration>/vnf/<vnf-name>?deep
```

Syslog

Objective

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

Syslog message format and examples

The Syslog feature allows event notifications from NFVIS to be sent to remote syslog servers for centralized log and event collection. The syslog messages are based on the occurrence of specific events on the device and provide configuration and operational information such as creation of users, changes to the interface status, and failed login attempts. Syslog data is critical to recording day-to-day events as well as notifying operational staff of critical system alerts.

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

To configure remote syslog settings, see Configure syslog.

Syslog messages have the following format:

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

Sample Syslog messages:

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

Syslog event notifications include:

Table 3: Syslog events

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

Event	Trigger Condition	Syslog Messages
BRIDGE_DHCP_RENEW	bridge dhcp renew	nfvis %SYS-6-BRIDGE_DHCP_RENEW: Bridge DHCP IP address is being renewed
MGMT_DHCP_RENEW	dhcp renew from MGMT interface	nfvis %SYS-6-MGMT_DHCP_RENEW: wan-br DHCP IP address is being renewed
INTF_STATUS_CHANGE	interface status change	nfvis %SYS-6-INTF_STATUS_CHANGE: Interface eth0, changed state to up
UPGRADE_REGISTER	upgrade package registration	nfvis %SYS-6-UPGRADE_REGISTER: Upgrade package registration successful: Cisco_NFVIS_Upgrade-3.6.1-6.9.8-2017.04.02.0428.1.nfvispkg
UPGRADE_APPLY	upgrade process	nfvis %SYS-6-UPGRADE_APPLY: Upgrade Process: In Progress
RBAC_USER_CREATE	create a new user	nfvis %SYS-6-RBAC_USER_CREATE: Created user admin as administrators successfully
RBAC_USER_PASSWORD_UPDATE	change user's password	nfvis %SYS-6-RBAC_USER_PASSWORD_UPDATE: Set admin password successfully
RBAC_USER_ROLE_UPDATE	change user's role	nfvis %SYS-6-RBAC_USER_ROLE_UPDATE: Modified user: somebody successfully
RBAC_USER_DELETE	delete a user	nfvis %SYS-6-RBAC_USER_DELETE: Deleted rbac user successfully: somebody
RBAC_USERS_INACTIVATED	disable the user	nfvis %SYS-6-RBAC_USERS_INACTIVATED: Following users have been marked as INACTIVE, [user1, user2]. Please take necessary action.

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

Event	Trigger Condition	Syslog Messages
CERT_SWITCH_CERT	switch to use different certificate	nfvis %SYS-6-CERT_SWITCH_CERT: switch certificate from ca-signed to self-signed.
CERT_CA_CERT_INSTALL	install CA signed certificate	nfvis %SYS-6-CERT_CA_CERT_INSTALL: ca-signed certificate file:// installed
REBOOT	system reboot	nfvis %SYS-6-REBOOT: System will be rebooted
SHUTDOWN	system shutdown	nfvis %SYS-6-SHUTDOWN: System will be shutdown
LOGGING_FAILURE	logging failure	nfvis %SYS-6-LOGGING_FAILURE: Unable to write to log file nfvis_config.log. Log message: log_config.CONFIG_LOGGER: File not found.
DISK_SPACE_ALMOST_FULL	disk space almost full	nfvis %SYS-6-DISK_SPACE_ALMOST_FULL: 'lv_data' currently occupies 95% of available disk space, which is more than or equal to the threshold of 90%.
ROTATED_LOGS_DELETE	delete rotated logfiles when accumulated rotated log files reach 2GB	nfvis %SYS-6-ROTATED_LOGS_DELETE: Deleted rotated logs from archive older than 30 days
TIME_UPDATE	Change system time manually	nfvis %SYS-6-TIME_UPDATE: Manual time updated successfully Manual time is now set to 2018-04-26 11:43:00
TIMEZONE_UPDATE	Change system timezone	nfvis %SYS-6-TIMEZONE_UPDATE: Timezone updated successfully. Timezone is now set to US/Eastern
FILE_COPY_STATUS	copy status of file	nfvis %SYS-6-FILE_COPY_STATUS: hostaction.py Copied Successfully.
CREATE_IMAGE	create image	nfvis %SYS-6-CREATE_IMAGE: Image creation successful: TinyLinux.tar.gz

Event	Trigger Condition	Syslog Messages
DELETE_IMAGE	delete image	nfvis %SYS-6-DELETE_IMAGE: Image deletion successful: TinyLinux.tar.gz
CREATE_FLAVOR	create flavor	nfvis %SYS-6-CREATE_FLAVOR: Profile creation successful: small
DELETE_FLAVOR	delete flavor	nfvis %SYS-6-DELETE_FLAVOR: Profile deletion successful: small
VM_DEPLOYED	vm deployment	nfvis %SYS-6-VM_DEPLOYED: VM deployment successful:
VM_ALIVE	vm alive	nfvis %SYS-6-VM_ALIVE: VM active successful:
SERVICE_ALIVE	service alive	nfvis %SYS-6-SERVICE_ALIVE: Service group deployment completed successfully!
VM_UNDEPLOYED	vm undeployed	nfvis %SYS-6-VM_UNDEPLOYED: VM undeployment successful: SERVICE_UNDEPLOYED service undeployed nfvis %SYS-6-SERVICE_UNDEPLOYED: Service group undeployment completed successfully
VM_UPDATED (update flavor)	vm updated	nfvis %SYS-6-VM_UPDATED: VM update successful: VM is resized with flavor [ISRv-medium].
VM_UPDATED (vnic add / delete / update)	vm updated	nfvis %SYS-6-VM_UPDATED: VM update successful: Added 1 interface: [managed, net=my-net-1, nicid=3] Updated 2 interface: [managed, net=lan-net, nicid=1],[managed, net=wan-net, nicid=2]
SERVICE_UPDATED	service updated	nfvis %SYS-6-SERVICE_UPDATED: Service group update completed successfully

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

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

Configure syslog

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

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

Procedure

Step 1 Configure a remote syslog server.

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

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

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

Step 2 Configure the syslog severity.

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

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

Table 4: Syslog severity levels

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

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

Step 3 Configure the syslog facility.

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

Example:

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

Note You can configure a facility from local0 through local7.

By default, NFVIS sends syslogs with the local7 facility.

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

NETConf

A NETCONF event notification is a monitoring mechanism that

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

Event notification types

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

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

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

You can view NFVIS and VM life cycle event notifications using the **show notification stream nfvisEvent** and **show notification stream vmlcEvent** commands respectively. For more information refer below

nfvisEvent

Table 5: NFVIS event notification output examples

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

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

Event Type	Notification Trigger	Notification Output Example
NETWORK_DELETE	A network is deleted.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-09-22T12:42:03.391986+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_message>Network deleted succesfully</status_message> <event_type>NETWORK_DELETE</event_type> <network_name>testn1</network_name> <network_bridge/> <network_sriov/> <network_vlan/> <network_trunk/> </nfvisEvent> </notification>
UPGRADE_REGISTER	System upgrade is registered.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T15:57:50.434636+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Upgrade package registration successful: Cisco_NFVIS_Upgrade-3.6.1-698-20170402_042811.nfvispkg</status_message> <event_type>UPGRADE_REGISTER</event_type> </nfvisEvent> </notification>
UPGRADE_APPLY	System upgrade is applied.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T16:02:43.885516+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Upgrade Process: In Progress</status_message> <event_type>UPGRADE_APPLY</event_type> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
ROTATED_LOGS_DELETE	Rotated logs older than 30 days are deleted by the system.	<?xml version="1.0" encoding="UTF-8"?> <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1"> <ok/> </rpc-reply> <?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T17:38:10.321152+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Deleted rotated logs from archive older than 30 days</status_message> <details>NA</details> <event_type>ROTATED_LOGS_DELETE</event_type> </nfvisEvent> </notification>
ROTATED_LOGS_DELETE	Older logs deleted by the system when the total file size of rotated logs exceeds 2GB.	<?xml version="1.0" encoding="UTF-8"?> <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1"> <ok/> </rpc-reply> <?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-04-26T17:42:10.321152+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>NA</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Rotated logs had exceeded 2G, older logs have been deleted to make space</status_message> <details>NA</details> <event_type>ROTATED_LOGS_DELETE</event_type> </nfvisEvent> </notification>

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

Event Type	Notification Trigger	Notification Output Example
PNIC_SRIOV_ENABLE	Enable SR-IOV to 2 vfs	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Physical Interface: eth0-1 Number of VFs 2</status_message> <details>NA</details> <event_type>PNIC_SRIOV_ENABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>
PNIC_SRIOV_DISABLE	Disable SR-IOV	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Physical Interface: eth0-1 Number of VFs 0</status_message> <details>NA</details> <event_type>PNIC_SRIOV_DISABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
PNIC_SRIOV_ENABLE	Enable of disable SR-IOV in error	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Physical Interface: eth0-1 Number of VFs 2</status_message> <details>NA</details> <event_type>PNIC_SRIOV_ENABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>
PNIC_SRIOV_UPDATE	Set switchmode to veb/vepa	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Physical Interface: eth0-1 Switchmode vepa</status_message> <details>NA</details> <event_type>PNIC_SRIOV_UPDATE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
PNIC_SRIOV_UPDATE	Set switchmode to veb/vepa in error	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2019-03-15T21:46:28.034133+00:00</eventTime> <nfvisEvent xmlns="http://www.cisco.com/nfvis/notifier"> <user_id>admin</user_id> <config_change>true</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Physical Interface: eth0-1 Switchmode vepa</status_message> <details>NA</details> <event_type>PNIC_SRIOV_UPDATE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>
PROMISC_ENABLED	Enable promiscuous mode	<?xml version="1.0" encoding="UTF-8"?> <notification <eventTime>2020-03-27T19:58:52.333682+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>GE0-0: promiscuous mode enabled</status_message> <details>GE0-0: promiscuous mode enabled</details> <event_type>PROMISC_ENABLED</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
[illegible]	When FIPS mode is configured but a reboot is required to enable it.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2020-06-07T21:26:06.231166+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Fips-mode configured and a manual reboot is required to enable it</status_message> <details>NA</details> <event_type>FIPS_CONFIGURED_REBOOT_TO_ENABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>
FIPS_ENABLED	FIPS mode is enabled.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2020-06-07T21:26:06.231166+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Fips-mode successfully enabled</status_message> <details>NA</details> <event_type>FIPS_ENABLED</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
FIPS_FAILED		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2020-06-07T21:26:06.231166+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Fips-mode configuration is unsuccessful</status_message> <details>NA</details> <event_type>FIPS_FAILED</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>
FIPS_DISABLED	FIPS mode is disabled.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2020-06-07T21:26:06.231166+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Fips-mode successfully disabled</status_message> <details>NA</details> <event_type>FIPS_DISABLED</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
[illegible]		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2020-06-07T21:26:06.231166+00:00</eventTime> <nfvisEvent <user_id>admin</user_id> <config_change>false</config_change> <transaction_id>0</transaction_id> <status>SUCCESS</status> <status_code>0</status_code> <status_message>Fips-mode unconfigured and a manual reboot is required to disable it</status_message> <details>NA</details> <event_type>FIPS_UNCONFIGURED_REBOOT_TO_DISABLE</event_type> <severity>INFO</severity> <hostname>NFVIS</hostname> </nfvisEvent> </notification>

vmlcEvent

Table 6: VM life cycle event notification output examples

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

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

Event Type	Notification Trigger	Notification Output Example
VM_DEPLOYED	The VM is deployed.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:19:16.927+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VIM Driver: VM successfully created, VM Name: [SystemAdmin_Tena_ROUTER_0_df6733c1-0768-4ae6-8dbe-b223ecdb036c]</status_message> <depname>1479341445</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> <interfaces> <interface> <nicid>0</nicid> <port_id>vnet0</port_id> <network>int-mgmt-net</network> <subnet>N/A</subnet> <ip_address>10.20.0.2</ip_address> <mac_address>52:54:00:31:c5:7f</mac_address> <netmask>255.255.255.0</netmask> <gateway>10.20.0.1</gateway> </interface> <interface> <nicid>1</nicid> <port_id>vnet1</port_id> <network>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:59:52:41</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> </interfaces> </vm_source> <vm_target/> <event> <type>VM_DEPLOYED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_ALIVE	The state of a monitored VM becomes ACTIVE.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:22:47.306+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VM_Alive event received, VM ID: [SystemAdmin_Tena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c]</status_message> <depname>1479341445</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> <interfaces> <interface> <nicid>0</nicid> <port_id>vnet0</port_id> <network>int-mgmt-net</network> <subnet>N/A</subnet> <ip_address>10.20.0.2</ip_address> <mac_address>52:54:00:31:c5:7f</mac_address> <netmask>255.255.255.0</netmask> <gateway>10.20.0.1</gateway> </interface> <interface> <nicid>1</nicid> <port_id>vnet1</port_id> <network>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:59:52:41</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> </interfaces> </vm_source> <vm_target/> <event> <type>VM_ALIVE</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_UNDEPLOYED	The VM is undeployed.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:31:40.6+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>204</status_code> <status_message>VIM Driver: VM successfully deleted</status_message> <depname>1479341445</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> <interfaces> <interface> <nicid>0</nicid> <port_id>vnet0</port_id> <network>int-mgmt-net</network> <subnet>N/A</subnet> <ip_address>10.20.0.2</ip_address> <mac_address>52:54:00:31:c5:7f</mac_address> <netmask>255.255.255.0</netmask> <gateway>10.20.0.1</gateway> </interface> <interface> <nicid>1</nicid> <port_id>vnet1</port_id> <network>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:59:52:41</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> </interfaces> </vm_source> <vm_target/> <event> <type>VM_UNDEPLOYED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
SERVICE_UPDATED	The VM is updated.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:51:45.5+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Service group update completed successfully</status_message> <depname>1479342258</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <depid>827e871a-30d5-4f5f-a05a-263b7ee3a734</depid> <vm_source/> <vm_target/> <event> <type>SERVICE_UPDATED</type> </event> </vmlcEvent> </notification>
VM_STOPPED	The VM is stopped per VM action requrest.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:26:05.762+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Successfully stopped VM [SystemAdminTena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223ecdb036c].</status_message> <depname>1479341445</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_STOPPED</type> </event> </vmlcEvent> </notification>

Event Type	Notification Trigger	Notification Output Example
VM_STARTED	The VM is started per VM action request.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:26:40.398+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Started VM [SystemAdmin_Tena_ROUTER_0_df6733c1-0768-4ae6-8dce-b223eedb036c].</status_message> <depname>1479341445</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>c64d79db-3a29-41a8-8114-c80d42731a5b</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d18dd252-80c8-44f2-ab66-d4481790bb79</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_STARTED</type> </event> </vmlcEvent> </notification>
VM_REBOOTED	The VM is rebooted per VM action request.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T17:36:56.5+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Rebooted VM [SystemAdmin_Tena_ROUTER_0_f17fc494-8535-4b05-b88d-f0fd2effdc7d]</status_message> <depname>1479342258</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>827e871a-30d5-4f5f-a05a-263b7ee3a734</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>d918a3b1-f2a9-4065-9d8e-2135b0a37d87</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target/> <event> <type>VM_REBOOTED</type> </event> </vmlcEvent> </notification>

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

Event Type	Notification Trigger	Notification Output Example
VM_RECOVERY_REBOOT	Recovery reboot starts for the monitored VM, which is not reachable.	<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2016-11-17T16:27:53.979+00:00</eventTime> <vmlcEvent xmlns="http://www.cisco.com/nfvis/vm_lifecycle"> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VM [SystemAdminTena_ROUTER_0_40ae18be-5930-4d94-95ff-dbb0b56ef12b] is being rebooted. </statu s_message> <depname>1479328919</depname> <tenant>admin</tenant> <tenant_id>AdminTenantId</tenant_id> <svcid>NULL</svcid> <depid>9e7fe4f8-a5f4-4a6d-aad7-121405be4ba4</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>000883fc-77f3-4b9e-aaf6-0f31d88a8f67</vmid> <hostid>NFVIS</hostid> <hostname>NFVIS</hostname> </vm_source> <vm_target></vm_target> <event> <type>VM_RECOVERY_REBOOT</type> </event> </vmlcEvent> </notification>

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

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

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

Event Type	Notification Trigger	Notification Output Example
		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-12-08T00:50:39.034+00:00</eventTime> <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'> <status>SUCCESS</status> <status_code>200</status_code> <status_message>VM is resized with flavor [ISRv-medium].</status_message> <user_name>admin</user_name> <depname>1512766000</depname> <tenant>admin</tenant> <tenant_id>adminUUID</tenant_id> <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid> <vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname> <hostid>NFVIS</hostid> <hostname>nfvis</hostname> <interfaces> <interface> <nicid>0</nicid> <type>virtual</type> <port_id>vnic0</port_id> <network>int-mgmt-net</network> <subnet>N/A</subnet> <ip_address>10.20.0.3</ip_address> <mac_address>52:54:00:3c:ee:5b</mac_address> <netmask>255.255.255.0</netmask> <gateway>10.20.0.1</gateway> </interface> <interface> <nicid>1</nicid> <type>virtual</type> <port_id>vnic1</port_id> <network>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:70:06:4a</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> <interface> <nicid>2</nicid> <type>virtual</type> <port_id>vnic2</port_id> <network>lan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:c7:30:1c</mac_address> <netmask>255.255.255.0</netmask> <gateway>192.168.1.1</gateway> </interface> </interfaces> </vm_source> <event> <type>VM_UPDATED</type> </event> </vmlcEvent></notification>]]>]]>

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

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

Event Type	Notification Trigger	Notification Output Example
		<?xml version="1.0" encoding="UTF-8"?> <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"> <eventTime>2017-12-08T02:10:56.184+00:00</eventTime> <vmlcEvent xmlns='http://www.cisco.com/nfvis/vm_lifecycle'> <status>SUCCESS</status> <status_code>200</status_code> <status_message>Added 1 interface: [managed, net=my-net-1, nicid=3] Updated 2 interface: [managed, net=lan-net, nicid=1],[managed, net=wan-net, nicid=2]</status_message> <user_name>admin</user_name> <depname>1512766000</depname> <tenant>admin</tenant> <tenant_id>adminUUID</tenant_id> <depid>92c11aa1-f6dd-47d1-948f-c8c65b9ef70f</depid> <vm_group>ROUTER</vm_group> <vm_source> <vmid>1a6f587e-2779-4087-b84d-c0a2c8a481b1</vmid> <vmname>1512766000_ROUTER_0_60d15064-0c6d-49b9-aa4a-80587f626004</vmname> <hostid>NFVIS</hostid> <hostname>nfvis</hostname> <interfaces> <interface> <nicid>0</nicid> <type>virtual</type> <port_id>vnic0</port_id> <network>int-mgmt-net</network> <subnet>N/A</subnet> <ip_address>10.20.0.3</ip_address> <mac_address>52:54:00:3c:ee:5b</mac_address> <netmask>255.255.255.0</netmask> <gateway>10.20.0.1</gateway> </interface> <interface> <nicid>1</nicid> <type>virtual</type> <port_id>vnic1</port_id> <network>lan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:70:06:4a</mac_address> <netmask>255.255.255.0</netmask> <gateway>192.168.1.1</gateway> </interface> <interface> <nicid>2</nicid> <type>virtual</type> <port_id>vnic2</port_id> <network>wan-net</network> <subnet>N/A</subnet> <mac_address>52:54:00:c7:30:1c</mac_address> <netmask>255.255.255.0</netmask> <gateway>172.19.181.152</gateway> </interface> <interface> <nicid>3</nicid> <type>virtual</type>

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

SNMP

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

- provides a message format for communication between SNMP managers and agents, and
- provides a standardized framework and a common language used for the monitoring and management of devices in a network.

The SNMP framework has three parts:

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

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

SNMP operations

SNMP operations are network management procedures that

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

SNMP operation types

SNMP applications perform three types of operations to retrieve data, modify SNMP object variables, and send notifications:

- SNMP GET - The SNMP GET operation is performed by a Network Management Server (NMS) to retrieve SNMP object variables.

- SNMP SET - The SNMP SET operation is performed by a Network Management Server (NMS) to modify the value of an object variable.
- SNMP Notifications - A key feature of SNMP is its capability to generate unsolicited notifications from an SNMP agent.

SNMP GET

Objective

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

SNMP GET command syntax

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

The command for SNMP GET is:

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

SNMP walk

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

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

The command for SNMP walk with SNMP v2 is:

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

Show more

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

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

This is a sample configuration of SNMP walk with SNMP v3:

```
snmpwalk -v 3 -u user3 -a sha -A changePassphrase -x aes -X changePassphrase
 -l authPriv -n snmp 172.16.1.101 system
SNMPv2-MIB::sysDescr.0 = STRING: Cisco ENCS 5412, 12-core Intel, 8 GB, 8-port PoE LAN, 2
HDD, Network Compute System
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.2377
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (16944068) 1 day, 23:04:00.68
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 70
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

SNMP notifications

Objective

SNMP notifications are asynchronous messages that

SNMP notification behavior

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

Notification types

Objective

SNMP notifications include these types:

SNMP notification types

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

SNMP versions

Objective

SNMP versions are network management protocol variants that

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

Supported SNMP versions

Cisco NFVIS supports these versions of SNMP:

- SNMP v1—The Simple Network Management Protocol: A Full Internet Standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.
- SNMP v2c—The community-string based Administrative Framework for SNMPv2. SNMPv2c (the "c" stands for "community") is an Experimental Internet Protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic), and uses the community-based security model of SNMPv1.
- SNMPv3—Version 3 of SNMP. SNMPv3 is an interoperable standards-based protocol defined in RFCs 3413 to 3415. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network.

The security features provided in SNMPv3 are as follows:

- Message integrity—Ensuring that a packet has not been tampered with in transit.
- Authentication—Determining that the message is from a valid source.
- Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.

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

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

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

SNMP MIB support

Objective

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

Supported SNMP MIBs

CISCO-MIB

CISCO-MIB OID 1.3.6.1.4.1.9.2.1.3. hostname

IF-MIB (1.3.6.1.2.1.31)

- ifDescr
- ifType
- ifPhysAddress
- ifSpeed
- ifOperStatus
- ifAdminStatus
- ifMtu
- ifName
- ifHighSpeed
- ifPromiscuousMode
- ifConnectorPresent
- ifInErrors
- ifInDiscards
- ifInOctets
- ifOutErrors
- ifOutDiscards
- ifOutOctets
- ifOutUcastPkts

- ifHCInOctets
- ifHCInUcastPkts
- ifHCOutOctets
- ifHCOutUcastPkts
- ifInBroadcastPkts
- ifOutBroadcastPkts
- ifInMulticastPkts
- ifOutMulticastPkts
- ifHCInBroadcastPkts
- ifHCOutBroadcastPkts
- ifHCInMulticastPkts
- ifHCOutMulticastPkts

Entity MIB (1.3.6.1.2.1.47)

- entPhysicalIndex
- entPhysicalDescr
- entPhysicalVendorType
- entPhysicalContainedIn
- entPhysicalClass
- entPhysicalParentRelPos
- entPhysicalName
- entPhysicalHardwareRev
- entPhysicalFirmwareRev
- entPhysicalSoftwareRev
- entPhysicalSerialNum
- entPhysicalMfgName
- entPhysicalModelName
- entPhysicalAlias
- entPhysicalAssetID
- entPhysicalIsFRU

CISCO process MIB (1.3.6.1.4.1.9.9.109)

- cpmCPUTotalPhysicalIndex (.2)

- cpmCPUTotal5secRev (.6.x)*
- cpmCPUTotal1minRev (.7.x)*
- cpmCPUTotal5minRev (.8.x)*
- cpmCPUMonInterval (.9)
- cpmCPUMemoryUsed (.12)
- cpmCPUMemoryFree (.13)
- cpmCPUMemoryKernelReserved (.14)
- cpmCPUMemoryHCUsed (.17)
- cpmCPUMemoryHCFree (.19)
- cpmCPUMemoryHCKernelReserved (.21)
- cpmCPULoadAvg1min (.24)
- cpmCPULoadAvg5min (.25)
- cpmCPULoadAvg15min (.26)

CISCO environmental monitor MIB notification

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

VM-MIB (1.3.6.1.2.1.236)

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

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

Configure SNMP support

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

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

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

Procedure

Step 1 Create an SNMP community.

Example:

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

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

Step 2 Create an SNMP group.

Example:

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

Table 7: SNMP group command variables

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

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

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

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

- For authPriv without a separate encryption passphrase:

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

- For authPriv with a separate encryption passphrase:

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

- For authNoPriv:

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

- For noAuthNoPriv:

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

Table 8: SNMP user command variables

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

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

Step 4 Enable SNMP traps.

Example:

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

trap_event can be linkup or linkdown.

Step 5 Create an SNMP trap host.

Example:

```
configure terminal
snmp host host_name host-ip-address ip_address host-port port host-user-name user_name
host-version version host-security-level security_level
```

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

Table 9: SNMP trap host command variables

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

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

SNMP configuration reference

Objective

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

SNMP v3 configuration

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

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

```
configure terminal
snmp group testgroup3 snmp 3 authPriv notify test write test read test
!
snmp user user3 user-version 3 user-group testgroup3 auth-protocol sha
priv-protocol aes passphrase changePassphrase encryption-passphrase
encryptPassphrase
! Configure the SNMP host to enable SNMP v3 traps
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
 host-security-level authPriv host-port 162
!
```

SNMP v1 and v2 configuration syntax

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

```
configure terminal
snmp community public community-access readOnly
!
snmp group testgroup snmp 2 noAuthNoPriv read read-access write write-access
 notify notify-access
!
snmp user public user-group testgroup user-version 2
!
snmp host host2 host-ip-address 2.2.2.2 host-port 162 host-user-name public
 host-version 2 host-security-level noAuthNoPriv
!
snmp enable traps linkup
snmp enable traps linkDown
```

SNMP v3 configuration with MD5 authentication

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

```
configure terminal
snmp group testgroup3 snmp 3 authPriv notify test write test read test
!
snmp user user3 user-version 3 user-group testgroup3 auth-protocol md5
priv-protocol aes passphrase changePassphrase
! Configure the SNMP host to enable SNMP v3 traps
snmp host host3 host-ip-address 3.3.3.3 host-version 3 host-user-name user3
 host-security-level authPriv host-port 162
!
```

SNMP security-level modification

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

```
configure terminal
!
snmp group testgroup4 snmp 3 authNoPriv notify test write test read test
```

```
!
snmp user user4 user-version 3 user-group testgroup4 auth-protocol md5
passphrase changePassphrase
! Configure the SNMP host to enable SNMP v3 traps
snmp host host4 host-ip-address 4.4.4.4 host-version 3 host-user-name user4
 host-security-level authNoPriv host-port 162
!
snmp enable traps linkUp
snmp enable traps linkDown
```

SNMP context configuration

Use these examples to configure specific SNMP contexts.

```
configure terminal
!
snmp group testgroup5 devop 3 authPriv notify test write test read test
!
snmp user user5 user-version 3 user-group testgroup5 auth-protocol md5
priv-protocol des passphrase changePassphrase
!
```

Empty context and noAuthNoPriv configuration

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

```
configure terminal
!
snmp group testgroup6 "" 3 noAuthNoPriv read test write test notify test
!
snmp user user6 user-version 3 user-group testgroup6
!
```



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

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

Do not use auth-key and priv-key to configure SNMP v3 passphrase. These keys are generated differently between different NFVIS systems for the same passphrase.

SNMP agent status

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

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

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

SNMP trap state

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

```
nfvis# show snmp traps

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

SNMP statistics

Use the **show snmp stats** command to verify the SNMP stats:

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

SNMP running configuration

Use the **show running-config SNMP** command to verify the interface configuration for SNMP:

```
nfvis# show running-config snmp
snmp agent enabled true
snmp agent engineID 00:00:00:09:11:22:33:44:55:66:77:88
snmp enable traps linkUp
snmp community pub_comm
community-access readOnly
!
snmp community tachen
community-access readOnly
!
snmp group tachen snmp 2 noAuthNoPriv
read   test
write  test
notify test
!
snmp group testgroup snmp 2 noAuthNoPriv
read   read-access
write  write-access
notify notify-access
!
snmp user public
user-version  2
user-group    2
auth-protocol md5
priv-protocol des
!
snmp user tachen
user-version 2
user-group   tachen
!
snmp host host2
host-port           162
host-ip-address     2.2.2.2
host-version        2
```

```
host-security-level noAuthNoPriv
host-user-name      public
!
```

SNMP configuration limits

Upper limit for SNMP configurations:

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

SNMP APIs and commands

Table 10: SNMP support APIs and commands

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

SNMP security mode behavior

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

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

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

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

Table 11: SNMP security settings by mode

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

Cisco ThousandEyes

Objective

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

ThousandEyes support in NFVIS

Cisco NFVIS ThousandEyes support is a pre-integrated solution that

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

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

This support provides these benefits:

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

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

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

Prerequisites

Objective

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

ThousandEyes deployment requirements

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

Deploy ThousandEyes container from NFVIS web interface

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

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

Procedure

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

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

The ThousandEyes node appears in the topology.

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

Table 12: VM Details fields

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

Step 4 Click **Deploy**.

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

To upgrade a registered image:

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

b. Click the **Upgrade** icon.

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

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

To upgrade using a new image:

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

b. Click the **Upgrade** icon.

c. Click **New Image**.

d. Choose a protocol.

Table 13: Image source fields

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

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

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

Deploy ThousandEyes on NFVIS using CLI configs

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

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

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

Procedure

Step 1 Register the ThousandEyes Docker Image

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

Example:

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

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

Step 2 Configure Resource Details as a Flavor

Example:

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

Step 3 Deploy the ThousandEyes Agent Container

Example:

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

Step 4 Commit the configuration

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

Option 1: Register and Update the Image

Register the New Docker Image:

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

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

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

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

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

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

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

 !
```

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

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

Image Registration with Proxy:

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

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

Deployment Configuration with Proxy:

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

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

Step 7 Deploy ThousandEyes Container on NFVIS Using NETCONF

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



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

Image Registration

From Docker Hub:

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

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

From ThousandEyes Webserver:

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

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

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

Flavor Registration

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

Deployment

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

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

ThousandEyes Container Deployment Upgrade on NFVIS Using NETCONF:

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

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

Step 8 Deploy ThousandEyes Container on NFVIS Using RESTCONF

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

Image Registration

From Docker Hub:

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

From ThousandEyes Webserver:

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

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

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

Flavor Creation

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

ThousandEyes Agent Container Deployment

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

Step 9 Upgrade ThousandEyes Container Deployment on NFVIS Using RESTCONF

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

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

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

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

Troubleshooting NFVIS

Objective

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

Log and show commands

Objective

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

Support and show commands

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

Support commands and show commands

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

Table 14: NFVIS support and show commands

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

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

Log files

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

Table 15: NFVIS log commands

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

Configure packet capture

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

The Packet Capture feature helps you capture all packets being transmitted and received over physical and virtual network interface controllers (physical port and vNIC) for analysis. These packets are inspected to diagnose and solve network problems.

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

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

Procedure

Step 1 Configure packet capture on a physical port.

Example:

configure terminal

tcpdump port `eth0`

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

Step 2 Configure packet capture on a vNIC.

Example:

configure terminal

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

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

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

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