API Reference for Cisco Enterprise Network Function Virtualization Infrastructure Software

First Published: 2017-03-31

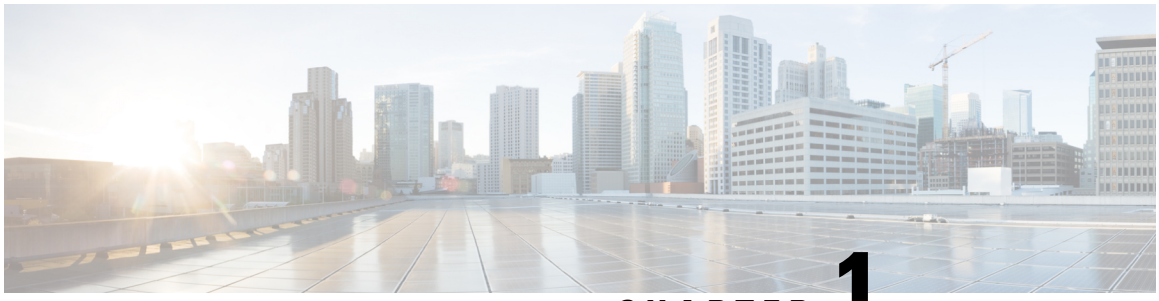
Last Modified: 2024-01-22

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

CONTENTS

PART I

NFVIS Related APIs



CHAPTER 1

Introduction to Cisco Enterprise NFVIS REST APIs

REST API Credentials

Ensure you include the following credential information in REST API requisition:

- User name: admin
- Password: password for admin

The payload in request can be in XML or JSON format. The headers (Content-Type and Accept) must be set accordingly.

The following two groups of headers are supported:

Table 1: Supported Headers

XML	Content-Type:application/vnd.yang.collection+xml Accept:application/vnd.yang.collection+xml
JSON	Content-Type:application/vnd.yang.collection+json Accept:application/vnd.yang.collection+json

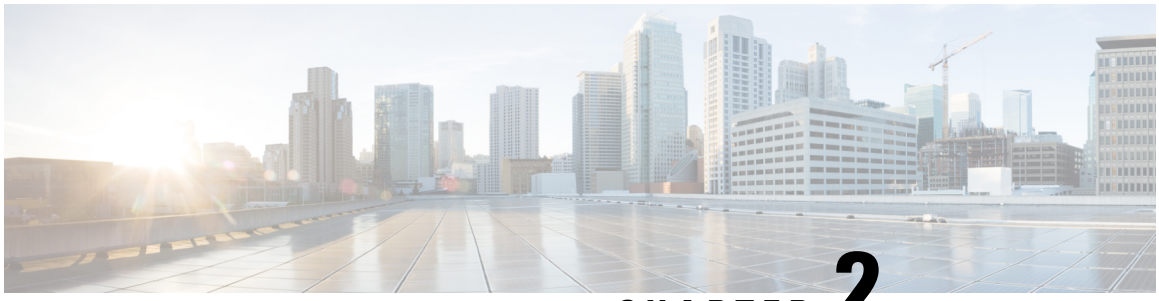
API Request Methods

The following are the supported REST API request methods:

HTTP Request Method	Description
GET	Retrieves the specified resource or representation. GET is a read-only operation that does not change the engine state or have any side effects. **Note** The GET method supports "*?deep*" query to get more detailed information.
POST	Submits data to be processed to the specified resource. The data to be processed is included in the request body. A POST operation can create a new resource.

HTTP Request Method	Description
PUT	Updates the specified resource with new information. The data that is included in the PUT operation replaces the previous data. • The PUT operation is used to replace or modify an existing resource. The PUT operation cannot be used to create a new resource. • The request body of a PUT operation must contain the complete representation of the mandatory attributes of the resource.
DELETE	Deletes a resource. If you delete a resource that has already been deleted, a 404 Not Found response is returned.

Note You can use any command line tool, such as curl, that supports transferring of data using the HTTPS protocol. All REST API commands must be preceded by *https://<host_server_ip>* . This is the Cisco Enterprise NFVIS host IP address.



CHAPTER 2

System and IP Configuration APIs

System Configuration APIs

Table 2: System Configuration APIs

Action	Method	Payload Required	API
To retrieve complete information on system configuration	GET	No	/api/operational/system/settings-native /api/config/system/settings
To configure the system by setting the default gateway, management IP address and/or WAN IP address	PUT	Yes	/api/config/system/settings

Example for System Configuration Payload

```
<system>
 <settings>
  <hostname>MyNFVIS123</hostname>
  <mgmt>
   <ip>
    <address>192.168.1.2</address>
    <netmask>255.255.255.0</netmask>
   </ip></mgmt>
  <wan>
   <dhcp/>
  </wan>
  </settings>
</system>
```

Note In the example, the management interface is configured with a static IP address and the WAN interface is set to DHCP. You can configure both the management and the WAN interface with static IP addresses; however, you can configure DHCP on only one of the interfaces.

Table 3: Description for System Details Payload

Property	Type	Description	Mandatory/Default Value
hostname	String	Hostname of the system. The hostname now follows RFC952 rules, allowing only alphabets, numbers and hyphen. The hostname can begin and end with either an alphabet or a digit. Host software must handle host names of up to 255 characters.	Yes
default-gw	String	IP address of the default gateway. **Note** The default gateway assigned through the DHCP configuration will take precedence over the default gateway for static configuration. Hence, to use the default gateway for static configuration, disable DHCP configuration for the WAN interface. When using default gateway, DHCP configuration is not allowed on any interface, include WAN and MGMT interfaces.	Yes

mgmt ip address	String	Management IP address **Note** When an interface is configured with a static IP address, DHCP is automatically disabled on that interface.	Yes
mgmt ip netmask	String	Netmask for the IP address.	Yes
wan dhcp	String	Set dhcp on the WAN interface. **Note** You can configure DHCP either on the WAN interface or the management interface; you cannot configure DHCP on both the interfaces simultaneously.	No

Example: PUT System Configuration API

```
curl -v -u admin:admin -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -k -X PUT
https://209.165.201.1/api/config/system -d
"<system>
 <settings>
  <hostname>Do3rdENCS75SettingsNoGW</hostname>
  <default-gw>172.19.183.1</default-gw>
  <mgmt>
   <ip>
    <address>172.19.183.75</address>
    <netmask>255.255.255.0</netmask>
   </ip>
  </mgmt>
  <wan>
   <ip>
    <address>4.3.2.5</address>
    <netmask>255.255.0.0</netmask>
   </ip><
  /wan>
 </settings>
</system>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (172.19.183.75) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
```

```
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Sep  2 17:03:09 2016 GMT
*  expire date: Aug 31 17:03:09 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system HTTP/1.1
> Host: 172.19.183.75
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 281
>
* upload completely sent off: 281 out of 281 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 07 Sep 2016 02:43:26 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 07 Sep 2016 02:43:25 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1473-216205-877863
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
sj22lab-as1:149>
```

Example: GET System Details API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X GET
https://209.165.201.1/api/operational/system/settings-native
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
```

```
*  start date: Sep  2 17:03:09 2016 GMT
*  expire date: Aug 31 17:03:09 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/settings-native HTTP/1.1
> Host: 172.19.183.75
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Tue, 06 Sep 2016 20:35:13 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<settings-native xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
  <mgmt>
   <ip-info>
      <interface>MGMT</interface>
      <ipv4_address>192.168.1.2</ipv4_address>
      <netmask>255.255.255.0</netmask>
      <ipv6_address>fe80::2f2:8bff:fec3:4a54</ipv6_address>
      <prefixlen>64</prefixlen>
      <mac_address>00:f2:8b:c3:4a:54</mac_address>
      <mtu>1500</mtu>
      <txqueuelen>1000</txqueuelen>
    </ip-info>
    <stats>
      <rx_packets>12481280</rx_packets>
      <rx_bytes>14392431432</rx_bytes>
      <rx_errors>0</rx_errors>
      <rx_dropped>210</rx_dropped>
      <rx_overruns>0</rx_overruns>
      <rx_frame>0</rx_frame>
      <tx_packets>3080505</tx_packets>
      <tx_bytes>238975886</tx_bytes>
      <tx_errors>0</tx_errors>
      <tx_dropped>0</tx_dropped>
      <tx_overruns>0</tx_overruns>
      <tx_carrier>0</tx_carrier>
      <tx_collisions>0</tx_collisions>
    </stats>
    <dhcp>
      <enabled>false</enabled>
      <offer>false</offer>
      <interface>NA</interface>
      <fixed_address>0.0.0.0</fixed_address>
      <subnet_mask>0.0.0.0</subnet_mask>
      <gateway>0.0.0.0</gateway>
      <lease_time>0</lease_time>
      <message_type>0</message_type>
      <name_servers>NA</name_servers>
      <server_identifier>0.0.0.0</server_identifier>
      <renewal_time>0</renewal_time>
      <rebinding_time>0</rebinding_time>
      <vendor_encapsulated_options>NA</vendor_encapsulated_options>
      <domain_name>NA</domain_name>
```

```
      <renew>0001-01-01T00:00:00-00:00</renew>
      <rebind>0001-01-01T00:00:00-00:00</rebind>
      <expire>0001-01-01T00:00:00-00:00</expire>
    </dhcp>
  </mgmt>
  <wan>
    <ip-info>
      <interface>wan-br</interface>
      <ipv4_address>209.165.201.22</ipv4_address>
      <netmask>255.255.255.0</netmask>
      <ipv6_address>fe80::2f2:8bff:fec3:49e0</ipv6_address>
      <prefixlen>64</prefixlen>
      <mac_address>00:f2:8b:c3:49:e0</mac_address>
      <mtu>1500</mtu>
      <txqueuelen>0</txqueuelen>
    </ip-info>
    <stats>
      <rx_packets>2971387</rx_packets>
      <rx_bytes>420208255</rx_bytes>
      <rx_errors>0</rx_errors>
      <rx_dropped>229</rx_dropped>
      <rx_overruns>0</rx_overruns>
      <rx_frame>0</rx_frame>
      <tx_packets>155</tx_packets>
      <tx_bytes>45522</tx_bytes>
      <tx_errors>0</tx_errors>
      <tx_dropped>0</tx_dropped>
      <tx_overruns>0</tx_overruns>
      <tx_carrier>0</tx_carrier>
      <tx_collisions>0</tx_collisions>
    </stats>
    <dhcp>
      <enabled>false</enabled>
      <offer>false</offer>
      <interface>NA</interface>
      <fixed_address>0.0.0.0</fixed_address>
      <subnet_mask>0.0.0.0</subnet_mask>
      <gateway>0.0.0.0</gateway>
      <lease_time>0</lease_time>
      <message_type>0</message_type>
      <name_servers>NA</name_servers>
      <server_identifier>0.0.0.0</server_identifier>
      <renewal_time>0</renewal_time>
      <rebinding_time>0</rebinding_time>
      <vendor_encapsulated_options>NA</vendor_encapsulated_options>
      <domain_name>NA</domain_name>
      <renew>0001-01-01T00:00:00-00:00</renew>
      <rebind>0001-01-01T00:00:00-00:00</rebind>
      <expire>0001-01-01T00:00:00-00:00</expire>
    </dhcp>
  </wan>
  <domain>NA</domain>
  <dns>
    <nameserver1>172.19.183.147</nameserver1>
    <nameserver2>0.0.0.0</nameserver2>
    <nameserver3>0.0.0.0</nameserver3>
  </dns>
 <hostname>Do3rdENCS75SettingsNoGW</hostname>
  <gateway>
    <ipv4_address>209.165.201.1</ipv4_address>
    <interface>MGMT</interface>
  </gateway>
</settings-native>
* Connection #0 to host 209.165.201.1 left intact
```

System Routes APIs

Table 4: System Routes APIs

Action	Method	Payload Required	API
To create a new route	POST	Yes	/api/config/system/routes
To modify an existing route	PUT	Yes	/api/config/system/routes/route/<host destination,netmask>
To retrieve the details of a route	GET	No	/api/operational/system/routes/route/<host destination,netmask> /api/config/system/routes
To delete a route	DELETE	No	/api/config/system/routes

Example for System Routes Payload

```
<route>
 <destination>209.165.201.1</destination>
 <prefixlen>16</prefixlen>
 <dev>lan-br</dev>
</route>
```

Table 5: System Routes Payload Description

Property	Type	Description	Mandatory/Default Value
destination	String	The route destination address.	Yes
prefixlen	Integer	The netmask for the destination address.	Yes
gateway	String	The gateway for the route.	No
dev	String	The device/interface that the route will use.	No

Note Though only the destination and prefixlen are mandatory parameters for creating a route, a valid route requires that you specify the gateway or the interface or both.

Example: POST System Route API

To create a new route:

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
```

```
"Content-Type:application/vnd.yang.data+xml" -X POST
https://209.165.201.1/api/config/system/routes -d
"<route><destination>209.165.201.5</destination><prefixlen>16</prefixlen></route>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug 27 06:20:53 2016 GMT
*  expire date: Aug 25 06:20:53 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/system/routes HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 75
>
* upload completely sent off: 75 out of 75 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 08:54:50 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://209.165.201.1/api/config/system/routes/route/21.1.0.0,16
< Connection: keep-alive
< Last-Modified: Sat, 27 Aug 2016 08:54:49 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1472-288089-901692
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Note The above example does not create a valid route because the gateway or device is not specified.

Example: PUT System Route API

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X PUT
```

```
https://209.165.201.1/api/config/system/routes/route/21.1.0.0,16 -d
"<route><destination>21.1.0.0</destination><prefixlen>16</prefixlen><dev>lan-br</dev></route>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug 27 06:20:53 2016 GMT
*  expire date: Aug 25 06:20:53 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/routes/route/21.1.0.0,16 HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 92
>
* upload completely sent off: 92 out of 92 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 09:00:45 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Sat, 27 Aug 2016 09:00:45 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1472-288445-682999
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET System Route API

To get route details and operational status for all routes:

```
curl -k -v -u "admin:admin" -X GET "https://209.165.201.1/api/operational/system/routes?deep"
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
```

```
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug 27 06:20:53 2016 GMT
*  expire date: Aug 25 06:20:53 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/routes?deep HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 09:07:19 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<routes xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
  <route>
    <destination>192.0.2.4</destination>
    <prefixlen>16</prefixlen>
    <gateway>192.0.2.1</gateway>
    <dev>lan-br</dev>
    <status>Success</status>
  </route>
  <route>
    <destination>192.0.2.5</destination>
    <prefixlen>16</prefixlen>
    <gateway>192.0.2.11</gateway>
    <dev>lan-br</dev>
    <status>Success</status>
  </route>
</routes>
* Connection #0 to host 209.165.201.1 left intact
```

Example: DELETE System Route API

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X DELETE
https://209.165.201.1/api/config/system/routes -d
"<route><destination>21.1.0.0</destination><prefixlen>16</prefixlen></route>"
*   Trying 209.165.201.1...
```

```
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug 27 06:20:53 2016 GMT
*  expire date: Aug 25 06:20:53 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/system/routes HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 75
>
* upload completely sent off: 75 out of 75 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 08:43:52 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Sat, 27 Aug 2016 08:43:52 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1472-287432-946952
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

VLAN APIs

The management VLAN is configured on the WAN interface.

Table 6: VLAN APIs

Action	Method	Payload Required	API
To configure a new VLAN or modify an existing VLAN	PUT	Yes	/api/config/bridges/bridge/wan-br/vlan

To get the configured VLAN info	GET	No	/api/config/bridges/bridge/wan2-br/vlan /api/config/bridges/bridge/user-br/vlan
To view the operational VLAN (the VLAN that is configured for the NFVIS management traffic on the wan-br).	GET	No	/api/operational/bridge-settings/bridge/wan-br/vlan
To delete a VLAN	DELETE	No	/api/config/bridges/bridge/wan-br/vlan

Example for VLAN Payload

```
<vlan> <vlan-id> </vlan>
```

The valid range for VLAN is from 1 to 4094.

Example: PUT VLAN API

Use the PUT VLAN API to create a new VLAN or modify an existing VLAN. When you modify a VLAN, the existing VLAN ID is replaced with the modified VLAN ID.

```
curl -k -v -u admin:Cisco#123 -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
PUT https://192.0.2.2/api/config/bridges/bridge/wan-br/vlan -d "<vlan>120</vlan>"
*   Trying 192.0.2.2...

* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)

* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

* successfully set certificate verify locations:

*   CAfile: /etc/pki/tls/certs/ca-bundle.crt

  CApath: none

* TLSv1.0 (OUT), TLS handshake, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Server hello (2):

* TLSv1.0 (IN), TLS handshake, Certificate (11):

* TLSv1.0 (IN), TLS handshake, Server key exchange (12):

* TLSv1.0 (IN), TLS handshake, Server finished (14):

* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):

* TLSv1.0 (OUT), TLS change cipher, Client hello (1):

* TLSv1.0 (OUT), TLS handshake, Finished (20):

* TLSv1.0 (IN), TLS change cipher, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Finished (20):
```

```
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Feb 15 23:33:39 2017 GMT
*  expire date: Feb 13 23:33:39 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/settings/wan/vlan HTTP/1.1
> Host: 192.0.2.2
> Authorization: Basic YWRtaW46Q2lzY28jMTIz
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 16
>
* upload completely sent off: 16 out of 16 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 16 Feb 2017 22:24:44 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 16 Feb 2017 22:24:36 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1487-283876-32584
< Pragma: no-cache
```

Example: GET VLAN API

Use this GET API to view the configured VLAN information.

```
curl -k -v -u admin:Cisco#123 -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/xml -k -X
```

```
GET https://192.0.2.2/api/config/bridges/bridge/wan-br/vlan
*   Trying 192.0.2.2...

* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)

* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

* successfully set certificate verify locations:

*   CAfile: /etc/pki/tls/certs/ca-bundle.crt

  CApath: none

* TLSv1.0 (OUT), TLS handshake, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Server hello (2):

* TLSv1.0 (IN), TLS handshake, Certificate (11):

* TLSv1.0 (IN), TLS handshake, Server key exchange (12):

* TLSv1.0 (IN), TLS handshake, Server finished (14):

* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):

* TLSv1.0 (OUT), TLS change cipher, Client hello (1):

* TLSv1.0 (OUT), TLS handshake, Finished (20):

* TLSv1.0 (IN), TLS change cipher, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Finished (20):

* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA

* Server certificate:

*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*  start date: Feb 15 23:33:39 2017 GMT

*  expire date: Feb 13 23:33:39 2027 GMT

*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*  SSL certificate verify result: self signed certificate (18), continuing anyway.

* Server auth using Basic with user 'admin'

> GET /api/config/system/settings/wan/vlan HTTP/1.1

> Host: 192.0.2.2

> Authorization: Basic YWRtaW46Q2lzY28jMTIz

> User-Agent: curl/7.49.1

> Accept:application/vnd.yang.data+xml

> Content-Type:application/xml
>
< HTTP/1.1 200 OK

< Server: nginx/1.10.1
```

```
< Date: Thu, 16 Feb 2017 22:43:21 GMT

< Content-Type: application/vnd.yang.data+xml

< Transfer-Encoding: chunked

< Connection: keep-alive

< Last-Modified: Thu, 16 Feb 2017 22:24:36 GMT

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Etag: 1487-283876-32584

< Pragma: no-cache
```

Use this GET API to view the operational VLAN (the VLAN that is configured for the NFVIS management traffic on the wan-br).

```
curl -k -v -u admin:Cisco#123 -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/xml -k -X
GET https://192.0.2.2/api/operational/bridge-settings/wan-br/vlan
*   Trying 192.0.2.2...

* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)

* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

* successfully set certificate verify locations:

*   CAfile: /etc/pki/tls/certs/ca-bundle.crt

  CApath: none

* TLSv1.0 (OUT), TLS handshake, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Server hello (2):

* TLSv1.0 (IN), TLS handshake, Certificate (11):

* TLSv1.0 (IN), TLS handshake, Server key exchange (12):

* TLSv1.0 (IN), TLS handshake, Server finished (14):

* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):

* TLSv1.0 (OUT), TLS change cipher, Client hello (1):

* TLSv1.0 (OUT), TLS handshake, Finished (20):

* TLSv1.0 (IN), TLS change cipher, Client hello (1):

* TLSv1.0 (IN), TLS handshake, Finished (20):

* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA

* Server certificate:

*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*  start date: Feb 15 23:33:39 2017 GMT
```

```
*  expire date: Feb 13 23:33:39 2027 GMT

*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*  SSL certificate verify result: self signed certificate (18), continuing anyway.

* Server auth using Basic with user 'admin'

> GET /api/operational/system/settings-native/wan/vlan HTTP/1.1

> Host: 192.0.2.2

> Authorization: Basic YWRtaW46Q2lzY28jMTIz

> User-Agent: curl/7.49.1

> Accept:application/vnd.yang.data+xml

> Content-Type:application/xml

>

< HTTP/1.1 200 OK

< Server: nginx/1.10.1

< Date: Thu, 16 Feb 2017 22:44:37 GMT

< Content-Type: application/vnd.yang.data+xml

< Transfer-Encoding: chunked

< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Pragma: no-cache
<

<vlan xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">

 <tag>120</tag>
</vlan>
```

Example: DELETE VLAN API

```
curl -k -v -u admin:Cisco#123 -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
DELETE https://192.0.2.2/api/config/bridges/bridge/wan-br/vlan
*   Trying 192.0.2.2...

* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)

* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

* successfully set certificate verify locations:

*   CAfile: /etc/pki/tls/certs/ca-bundle.crt

  CApath: none
```

```
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Feb 15 23:33:39 2017 GMT
*  expire date: Feb 13 23:33:39 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/system/settings/wan/vlan HTTP/1.1
> Host: 192.0.2.2
> Authorization: Basic YWRtaW46Q2lzY28jMTIz
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 16 Feb 2017 22:48:59 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 16 Feb 2017 22:48:50 GMT
```

```
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Etag: 1487-285330-811423

< Pragma: no-cache
```

User Management APIs

Action	Method	Payload Required	API
Add a user	POST	Yes	/api/config/rbac/authentication/users/create-user
Modify a user (Changing the user password)	POST	Yes	/api/operations/rbac/authentication/users /user/<user-name>/change-password
Change the user role	POST	Yes	/api/operations/rbac/authentication/users/user /oper/change-role
Get all users	GET	No	/api/config/rbac/authentication/users/user?deep
Delete a user	Delete	Yes	/api/config/rbac/authentication/users/delete-user
Configure the minimum password length	POST	Yes	/api/config/rbac/authentication/
Configure the password lifetime	POST	Yes	/api/config/rbac/authentication/password-lifetime/
Configure the account inactivity period	POST	Yes	/api/config/rbac/authentication/account-inactivity/
Activate an inactive user account	POST	No	/api/operations/rbac/authentication/users/user/username/activate

Example for Add User Payload

```
<input>
 <name>testuser</name>
 <password>Test123#</password>
 <role>operators</role>
</input>
```

Example for Change Role Payload

```
<input>
 <old-role>auditors</old-role>
 <new-role>operators</new-role>
</input>
```

Example for Change Password Payload

```
<input>
 <old-password>Hello123#</old-password>
 <new-password>Hello123$</new-password>
 <confirm-password>Hello123$</confirm-password>
</input>
```

Example for Minimum Password Length Payload

```
<min-pwd-length>9</min-pwd-length>
```

Example for Password Lifetime Payload

```
<enforce>true</enforce>
<min-days>7</min-days>
<max-days>30</max-days>
```

Example for Account Inactivity Period Payload

```
<enforce>true</enforce>
<inactivity-days>50</inactivity-days>
```

Table 7: User Management API Payload Description

Property	Type	Description	Mandatory/Default Value
name	String	Name of the user	No
role	String	Role of the user	Yes
password	String	Password of the user	Yes
old-role	String	Existing role of the user	Yes
new-role	String	New role of the user	Yes
old-password	String	Existing password	Yes
new-password	String	New password for the user	Yes
confirm-password	String	Confirms the new password	Yes
min-pwd-length	Number	Minimum length required for passwords of all users. The minimum length must be between 7 to 128 characters.	Yes
enforce	String	Enforces or removes the rule. Valid values for this parameter are true and false.	Yes
min-days	Number	Number of days after which the users can change the password.	Yes
max-days	Number	Number of days before which the users must change the password.	Yes

inactivity-days	Number	Number of days after which an unused account is marked as inactive.	Yes

Example: POST Add User API

```
curl -X POST -v -k -u admin:Admin123$
https://209.165.201.1/api/operations/rbac/authentication/users/create-user -H
Content-Type:application/vnd.yang.data+xml
-d"<input><name>testname</name><password>Hello123#</password><role>operators</role></input>"
```

Example: POST Change Role API

```
curl -X POST -v -k -u admin:Cisco123#
https://209.165.201.1/api/operations/rbac/authentication/users/user/oper/change-role
 -H Content-Type:application/vnd.yang.data+xml -d
"<input><old-role>auditors</old-role><new-role>operators</new-role></input>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/rbac/authentication/users/user/oper/change-role HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 74
>
* upload completely sent off: 74 out of 74 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 16 Feb 2017 20:51:03 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
```

Example: POST Change Password API

```
curl -X POST -v -k -u admin:Admin123#
https://209.165.201.1/api/operations/rbac/authentication/users/user/testuser12/change-password
 -H
 Content-Type:application/vnd.yang.data+xml -d
"<input><old-password>Hello123#</old-password><new-password>Hello123$</new-password>
<confirm-password>Hello123$</confirm-password></input>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/rbac/authentication/users/user/testuser12/change-password HTTP/1.1
```

```
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 137
>
* upload completely sent off: 137 out of 137 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Thu, 22 Dec 2016 19:05:10 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
```

Example: GET Users API

```
curl -X GET -v -k -u "admin:Admin123#" -H "Content-Type: application/vnd.yang.collection+xml"
 -H "Accept: application/vnd.yang.collection+xml"
 "https://209.165.201.1/api/config/rbac/authentication/users/user?deep"
<collection xmlns:y="http://tail-f.com/ns/rest">
  <user xmlns="http://www.cisco.com/nfv/rbac">
    <name>admin</name>
    <role>administrators</role>
    <password>$7$K1dMMts4XHgdT//+YGlrrqh4YCZvYye4</password>
    <y:operations>

<change-password>/api/config/rbac/authentication/users/user/admin/_operations/change-password</change-password>


<change-role>/api/config/rbac/authentication/users/user/admin/_operations/change-role</change-role>

    </y:operations>
  </user>
  <user xmlns="http://www.cisco.com/nfv/rbac">
    <name>oper</name>
    <role>administrators</role>
    <password>$7$u76ZWuWU1Kn+gCPsImgEKpBkavgziDuO</password>
    <y:operations>

<change-password>/api/config/rbac/authentication/users/user/oper/_operations/change-password</change-password>


<change-role>/api/config/rbac/authentication/users/user/oper/_operations/change-role</change-role>

    </y:operations>
  </user>
  <user xmlns="http://www.cisco.com/nfv/rbac">
    <name>testuser12</name>
    <role>administrators</role>
    <password>$7$YhK1LGI2HTjzCTBVDZ8lxfWxTvqjjcvN</password>
    <y:operations>

<change-password>/api/config/rbac/authentication/users/user/testuser12/_operations/change-password</change-password>


<change-role>/api/config/rbac/authentication/users/user/testuser12/_operations/change-role</change-role>
```

```
    </y:operations>
  </user>
</collection>
```

Example: Delete User API

```
curl -X POST -v -k -u admin:Admin123#
https://209.165.201.1/api/operations/rbac/authentication/users/delete-user -H
Content-Type:application/vnd.yang.data+xml -d"<input><name>testname</name></input>"
```

Example: POST Configure Minimum Password Length

```
curl -X POST -v -k -u admin:Admin123# https://209.165.201.1/api/config/rbac/authentication/
 -H Content-Type:application/vnd.yang.data+xml -d "<min-pwd-length>9</min-pwd-length>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 34
>
* upload completely sent off: 34 out of 34 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 11:56:36 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Examples: POST Configure Password Lifetime

```
curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/config/rbac/authentication/password-lifetime/ -H
Content-Type:application/vnd.yang.data+xml -d "<enforce>true</enforce>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/password-lifetime/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 23
```

```
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 11:59:48 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

```
curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/config/rbac/authentication/password-lifetime/ -H
Content-Type:application/vnd.yang.data+xml -d "<min-days>1</min-days>"
```

```
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/password-lifetime/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 23
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 11:59:48 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

```
curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/config/rbac/authentication/password-lifetime/ -H
Content-Type:application/vnd.yang.data+xml -d "<max-days>30</max-days>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/password-lifetime/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 23
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 11:59:48 GMT
< Content-Type: text/html
< Content-Length: 0
```

```
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Examples: POST Configure Account Inactivity Period

```
curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/config/rbac/authentication/account-inactivity/ -H
Content-Type:application/vnd.yang.data+xml -d "<enforce>true</enforce>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/account-inactivity/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 23
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 12:00:52 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact

curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/config/rbac/authentication/account-inactivity/ -H
Content-Type:application/vnd.yang.data+xml -d "<inactivity-days>50</inactivity-days>"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/rbac/authentication/account-inactivity/ HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 23
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 12:00:52 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Activate an Inactive User Account

```
curl -X POST -v -k -u admin:Admin#123
https://209.165.201.1/api/operations/rbac/authentication/users/user/guest_user/activate -H
 Con-tent-Type:application/vnd.yang.data+xml
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: nfvis
* Server auth using Basic with user 'admin'
> POST /api/operations/rbac/authentication/users/user/guest_user/activate HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46QWRtaW4jMTIz
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Tue, 31 Oct 2017 12:11:31 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

TACACS+ Server APIs

Table 8: TACACS+ Server APIs

Action	Method	Payload Required	API
To configure a TACACS+ server	POST	Yes	/api/config/security_servers/tacacs-server
To modify a TACACS+ server configuration	PUT	Yes	/api/config/security_servers/tacacs-server
To get the TACACS+ server configuration details	GET	No	/api/config/security_servers/tacacs-server?deep
To delete a TACACS+ server configuration	DELETE	No	/api/config/security_servers/tacacs-server /host/<ip-address/domain-name>

Example for TACACS+ Server Payload

Table 9: TACACS+ Server Payload Description

Property	Type	Description	Mandatory/Default Value

Example: POST TACACS Server API

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+json -X
POST https://172.19.181.173/api/config/security_servers/tacacs-server -d
* Hostname was NOT found in DNS cache
*   Trying 172.19.181.173...
* Connected to 172.19.181.173 (172.19.181.173) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/security_servers/tacacs-server HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 172.19.181.173
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
> Content-Length: 122
>
* upload completely sent off: 122 out of 122 bytes
< HTTP/1.1 201 Created
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:14:46 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://172.19.181.173/api/config/security_servers/tacacs-server/host/5.5.5.5
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:14:46 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219286-189602
< Pragma: no-cache
```

Example: GET TACACS Server API

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+json -X
GET https://209.165.201.1/api/config/security_servers/tacacs-server?deep
* Hostname was NOT found in DNS cache
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
```

```
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/config/security_servers/tacacs-server?deep HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
>
< HTTP/1.1 200 OK
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:07:49 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Fri, 24 Feb 2017 01:13:51 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1487-898831-958028
< Pragma: no-cache
<

<tacacs-server xmlns="http://www.cisco.com/ns/test/security" xmlns:y="http://tail-

f.com/ns/rest"  xmlns:security="http://www.cisco.com/ns/test/security">
  <host>
    <server>10.2.2.2</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
  <host>
    <server>10.3.3.3</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
  <host>
    <server>10.1.1.1</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
```

```
</tacacs-server>
```

Example: PUT TACACS Server API

```
* Hostname was NOT found in DNS cache
*   Trying 172.19.181.173...
* Connected to 172.19.181.173 (172.19.181.173) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/security_servers/tacacs-server/host/5.5.5.5 HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 172.19.181.173
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
> Content-Length: 92
>
* upload completely sent off: 92 out of 92 bytes
< HTTP/1.1 204 No Content
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:20:13 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:20:13 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219613-571277
< Pragma: no-cache
```

Example: DELETE TACACS Server API

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+json -X
DELETE https://192.0.2.2/api/config/security_servers/tacacs-server/host/5.5.5.5
* Hostname was NOT found in DNS cache
*   Trying 192.0.2.2...
* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)
* successfully set certificate verify locations:
```

```
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/security_servers/tacacs-server/host/5.5.5.5 HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 192.0.2.2
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
>
< HTTP/1.1 204 No Content
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:21:30 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:21:30 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219690-404414
< Pragma: no-cache
```

Trusted IP Connection APIs

Table 10: Trusted IP Connection APIs

Action	Method	Payload Required	API
To add, modify, or remove the trusted source IP connection	PUT	Yes	/api/config/system/settings
To verify the configuration of the trusted source IP addresses	GET	No	/api/operational/system/settings-native/trusted-source
To verify the system settings	GET	No	/api/operational/system/settings-native?deep
To verify the trusted source or system settings	GET	No	/api/operational/system/settings?deep

Example for the Trusted IP Connection Payload

```
<settings>
    <hostname>nfvis</hostname>
    <trusted-source>192.0.2.0/24</trusted-source>
        <mgmt>
            <ip>
                <address>198.51.100.1</address>
                <netmask>255.255.255.0</netmask>
            </ip>
        </mgmt>
        <wan>
            <ip>
                 <address>198.51.100.2</address>
                 <netmask>255.255.255.0</netmask>
            </ip>
        </wan>
    <default-gw>198.51.100.3</default-gw>
</settings>
```

Table 11: Trusted IP Connection Payload Description

Property	Type	Description	Mandatory/Default Value
hostname	String	Hostname of the system	Yes
trusted-source	String	Source IP address You can specify a single IP address or a range of IP addresses.	No
mgmt ip address netmask	String	Specifies the management IP address and netmask.	Yes
wan ip address netmask	String	Specifies the WAN IP address and netmask.	Yes
default-gw	String	IP address of the default gateway	Yes

Example: PUT Trusted IP Connection API

Use this API to add, modify, or remove the trusted source IP address or addresses.

Note To delete all trusted source IP addresses, you need to remove the trusted source element (trusted-source) from the payload. You can modify a trusted source IP address by replacing it with a new IP address.

```
curl -k -v -u "admin:Cisco123#" -H "Content-Type:application/vnd.yang.data+xml" -X PUT
https://198.51.100.1/api/config/system/settings
-d "<settings><hostname>nfvis</hostname><trusted-source>192.0.2.0/24</trusted-source>
<mgmt><ip><address>198.51.100.1</address><netmask>255.255.255.0</netmask></ip></mgmt>
<wan><ip><address>198.51.100.2</address><netmask>255.255.255.0</netmask></ip></wan><default-gw>198.51.100.3</default-gw></settings>"

*   Trying 198.51.100.1...
* Connected to 198.51.100.1 (198.51.100.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
```

```
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 14 06:53:22 2017 GMT
*  expire date: Mar 12 06:53:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/settings HTTP/1.1
> Host: 198.51.100.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 343
>
* upload completely sent off: 343 out of 343 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Tue, 14 Mar 2017 21:19:21 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Tue, 14 Mar 2017 21:19:15 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-526355-690730
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.1 left intact
```

Example: GET Trusted IP Connection API

```
curl -v -k -u  admin:Cisco123# -X GET
'https://198.51.100.1/api/operational/system/settings-native/trusted-source'

Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 198.51.100.1...
* Connected to 198.51.100.1 (198.51.100.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
```

```
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 14 06:53:22 2017 GMT
*  expire date: Mar 12 06:53:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/settings-native/trusted-source HTTP/1.1
> Host: 198.51.100.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 14 Mar 2017 21:08:49 GMT
< Content-Type: application/vnd.yang.collection+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<collection xmlns:y="http://tail-f.com/ns/rest">
  <trusted-source xmlns="http://www.cisco.com/nfv">192.0.2.0/24</trusted-source>
  </collection>
* Connection #0 to host 198.51.100.1 left intact
```

Banner and Message APIs

Table 12: Banner and Message APIs

Action	Method	Payload Required	API
To configure or update a banner or message of the day or both	PUT	Yes	/api/config/banner-motd
To get system banner details and user-defined banner and message of the day	GET	No	/api/operational/banner-motd /api/operational/banner-motd/system-banner /api/operational/banner-motd/banner /api/operational/banner-motd/motd

To get user-defined banner and message of the day details	GET	No	/api/config/banner-motd /api/config/banner-motd/banner /api/config/banner-motd/motd
To delete the user-defined banner or message of the day	DELETE	No	/api/config/banner-motd /api/config/banner-motd/banner /api/config/banner-motd/motd

Example for Banner and Message Payload

```
<banner-motd>
    <banner> my banner </banner>
    <motd> my motd </motd>
</banner-motd>
```

Table 13: Banner and Message Payload Description

Property	Type	Description	Mandatory/Default Value
banner	String	Specifies the user-defined banner.	No
motd	String	Message of the day	No

Example: PUT Banner-MOTD API

```
curl -k -v -u "admin:Cisco123*" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT https://209.165.201.1/api/config/banner-motd -d '<banner-motd><banner>my
banner</banner><motd>my motd</motd></banner-motd>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> PUT /api/config/banner-motd HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 99
>
* upload completely sent off: 99 out of 99 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Tue, 27 Dec 2016 01:48:31 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Tue, 27 Dec 2016 01:48:31 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1482-803311-573328
< Pragma: no-cache
```

Example: GET Banner-MOTD API

Use this operational API to get information about the system-defined banner.

```
curl -k -v -u "admin:Cisco123*" -X GET
"https://209.165.201.1/api/operational/banner-motd/system-banner"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/banner-motd HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Tue, 27 Dec 2016 01:50:24 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<banner-motd xmlns="http://www.cisco.com/nfvis/banner" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:banner_motd="http://www.cisco.com/nfvis/banner">
  <banner>---my banner 111
2222
3333</banner>
  <motd>----my motd 1111</motd>
  <system-banner>
Cisco Enterprise Network Function Virtualization Infrastructure Software (NFVIS)

Copyright (c) 2015-2016 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco
Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

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


</system-banner>
</banner-motd>
```

Use this GET API to get information about the user-defined banner and message of the day.

```
curl -k -v -u "admin:Cisco123*" -X GET "https://209.165.201.1/api/config/banner-motd"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/config/banner-motd HTTP/1.1
```

```
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Tue, 27 Dec 2016 01:51:58 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Tue, 27 Dec 2016 01:48:31 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1482-803311-573328
< Pragma: no-cache
<

<banner-motd xmlns="http://www.cisco.com/nfvis/banner" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:banner_motd="http://www.cisco.com/nfvis/banner">
  <banner>my banner</banner>
  <motd>my motd</motd>
</banner-motd>
```

Example: DELETE Banner-MOTD API

Use this DELETE API to delete the user-defined banner.

```
curl -k -v -u "admin:Cisco123*" -X DELETE
"https://209.165.201.1/api/config/banner-motd/banner"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> DELETE /api/config/banner-motd/banner HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 08 Feb 2017 20:27:29 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 08 Feb 2017 20:27:29 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1486-585649-542089
< Pragma: no-cache
```

Use this DELETE API to delete the user-defined message of the day.

```
curl -k -v -u "admin:Cisco123*" -X DELETE "https://209.165.201.1/api/config/banner-motd/motd"
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
```

```
* Server auth using Basic with user 'admin'
> DELETE /api/config/banner-motd/motd HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 08 Feb 2017 20:33:52 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 08 Feb 2017 20:33:52 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1486-586032-109043
< Pragma: no-cache
<
```

Note After deleting the banner or message of the day, you can run the GET operational API to confirm the deletion. If you use the parameter "banner" or "motd" along with the GET API, you get a 404 error if the deletion is successful. If you run the GET API without the parameter (/api/operational/banner-motd), you get the output with empty "banner-motd" tag, if the deletion is successful.

Disk Space APIs

Table 14: Disk Space API

Action	Method	Payload Required	API
To get the information on disk space	GET	Yes	/api/operational/system/disk-space

Example: GET Disk Space API

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/system/disk-space?deep"
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
```

```
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=nfvis
*  start date: Oct 23 17:25:04 2018 GMT
*  expire date: Oct 22 17:25:04 2023 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/disk-space?deep HTTP/1.1
> Host: 172.25.221.106
> Authorization: Basic YWRtaW46MTIzI0FkbWlu
> User-Agent: curl/7.50.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 26 Oct 2018 01:10:37 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline';img-src 'self'
 data:; object-src 'none'; connect-src 'self' *
< X-Frame-Options: SAMEORIGIN
< Strict-Transport-Security: max-age=31536000; includeSubDomains
< Cache-Control: max-age=0, no-cache, no-store, must-revalidate
<

<disk-space xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
  <disk-info>
    <disk-name>lv_data</disk-name>
    <associated-physical-disk>sde2</associated-physical-disk>
    <total-size>41G</total-size>
    <size-used>8.6G</size-used>
    <size-available>32G</size-available>
    <use-percent>22%</use-percent>
  </disk-info>
  <disk-info>
    <disk-name>lv_var</disk-name>
    <associated-physical-disk>sde2</associated-physical-disk>
    <total-size>2.0G</total-size>
    <size-used>118M</size-used>
    <size-available>1.7G</size-available>
    <use-percent>7%</use-percent>
  </disk-info>
  <disk-info>
    <disk-name>lv_root</disk-name>
    <associated-physical-disk>sde2</associated-physical-disk>
    <total-size>7.8G</total-size>
    <size-used>1.8G</size-used>
    <size-available>5.7G</size-available>
    <use-percent>24%</use-percent>
```

```
  </disk-info>
  <disk-info>
    <disk-name>extdatastore2</disk-name>
    <associated-physical-disk>sdd</associated-physical-disk>
    <total-size>1.8T</total-size>
    <size-used>77M</size-used>
    <size-available>1.7T</size-available>
    <use-percent>1%</use-percent>
  </disk-info>
</disk-space>
* Connection #0 to host 209.165.201.1 left intact
```

System Time APIs

Table 15: System Time APIs

Action	Method	Payload Required	API
To set the manual time	PUT	Yes	• /api/config/system/time/set-manual-time
To configure the preferred and backup servers	PUT	Yes	• /api/config/system/time/ntp/preferred_server • /api/config/system/time/ntp/backup_server
To set the timezone	PUT	Yes	/api/config/system/time/timezone
To get the system time information	GET	No	/api/operational/system/time
To add NTP IPv6 server	POST	Yes	/api/config/system/time/
To delete NTP IPv6 server	DELETE	No	/api/config/system/time/ntp-ipv6/
To get time status	GET	NO	/api/operational/system/time

Example for System Time API Payload

```
<input><time>2017-01-01T00:00:00</time></input>
<preferred_server><ip-address></preferred_server>
<backup_server><ip-address></backup_server>
<timezone><zone/subzone></timezone>
<ntp-ipv6><ntp-server>2001:420:30d:201:ffff:ffff:fff4:35</ntp-server></ntp-ipv6>
```

Table 16: System Time API Payload Description

Property	Type	Description	Mandatory/Default Value

set-manual-time	String	Specifies manual time in YYYY-MM-DDTHH:MM:SS format.	Yes
preferred_server	String	Preferred server IP address or domain name.	Yes
backup_server	String	Backup server IP address or domain name.	No
timezone	String	Specifies the timezone.	No
ntp-server	String	Specifes the IPv6 address or domain name.	Yes

Example: PUT System Time Manual Time API

```
curl -v -k -u admin:Cisco123* -H "Content-Type: application/vnd.yang.data+xml" -X
PUT https://209.165.201.1/api/config/system/time/set-manual-time -d
'<input><time>2017-01-01T00:00:00</time></input>'

*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> PUT /api/config/system/time/set-manual-time HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMq

> User-Agent: curl/7.43.0

> Accept: */*

> Content-Type:application/vnd.yang.data+xml

> Content-Length: 46

>
* upload completely sent off: 46 out of 46 bytes

< HTTP/1.1 204 No Content

< Server: nginx/1.6.3

< Date: Wed, 01 Jan 2020 11:11:51 GMT

< Content-Type: text/html

< Content-Length: 0

< Connection: keep-alive

< Last-Modified: Wed, 30 Nov 2016 04:10:28 GMT

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
```

```
< Etag: 1480-479028-836845

< Pragma: no-cache

<
```

Example: PUT System Time Preferred Server API

```
curl -v -k -u admin:Cisco123* -H "Content-Type: application/vnd.yang.data+xml" -X
PUT https://209.165.201.1/api/config/system/time/ntp/preferred_server -d
'<preferred_server>209.165.201.2</preferred_server>'

*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> PUT /api/config/system/time/ntp/preferred_server HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMq

> User-Agent: curl/7.43.0

> Accept: */*

> Content-Type: application/vnd.yang.data+xml

> Content-Length: 49

>

* upload completely sent off: 49 out of 49 bytes

< HTTP/1.1 204 No Content

< Server: nginx/1.6.3

< Date: Wed, 01 Jan 2020 11:15:02 GMT

< Content-Type: text/html

< Content-Length: 0

< Connection: keep-alive

< Last-Modified: Wed, 01 Jan 2020 11:15:02 GMT

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Etag: 1480-479262-370866

< Pragma: no-cache
```

Example: PUT System Time Backup Server API

```
curl -v -k -u admin:Cisco123* -H "Content-Type: application/vnd.yang.data+xml" -X
PUT https:// 209.165.201.1/api/config/system/time/ntp/backup_server -d
'<backup_server>209.165.201.4</backup_server>'

  Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> PUT /api/config/system/time/ntp/backup_server HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMq

> User-Agent: curl/7.43.0

> Accept: */*

> Content-Type: application/vnd.yang.data+xml

> Content-Length: 43

>

* upload completely sent off: 43 out of 43 bytes

< HTTP/1.1 204 No Content

< Server: nginx/1.6.3

< Date: Wed, 01 Jan 2020 11:16:47 GMT

< Content-Type: text/html

< Content-Length: 0

< Connection: keep-alive

< Last-Modified: Wed, 01 Jan 2020 11:16:47 GMT

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Etag: 1480-479368-378871

< Pragma: no-cache
```

Example: PUT System Time Timezone API

```
curl -v -k -u admin:Cisco123* -H "Content-Type: application/vnd.yang.data+xml" -X
PUT https://209.165.201.1/api/config/system/time/timezone -d
'<timezone>America/New_York</timezone>'
```

```
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> PUT /api/config/system/time/timezone HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Type: application/vnd.yang.data+xml
> Content-Length: 37
>
* upload completely sent off: 37 out of 37 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 01 Jan 2020 11:19:44 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 01 Jan 2020 16:19:44 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1480-479547-383761
< Pragma: no-cache
```

Example: GET System Time API

```
curl -v -k -u admin:Cisco123* -H "Content-Type: application/vnd.yang.data+xml" -X
GET https://209.165.201.1/api/operational/system/time?deep

*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/system/host_time HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMq
> User-Agent: curl/7.43.0
```

```
> Accept: */*
> Content-Type: application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Wed, 01 Jan 2020 11:21:13 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<time xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
<ntp>
<status>
<remote>209.165.201.4</remote>
<refid>.GPS.</refid>
<st>1</st>
<t>u</t>
<when>2</when>
<poll>512</poll>
<reach>377</reach>
<delay>71.547</delay>
<offset>-1.862</offset>
<jitter>0.764</jitter>
</status>
</ntp>
<current-time>2017-01-01T12:12:12</current-time>
<current-timezone>UTC (UTC, +0000)</current-timezone>
</time>
```

Platform Details API

Table 17: Platform Details APIs

Action	Method	Payload Required	API
To get information about the hardware	GET	No	/api/operational/platform-detail

Sample Output for the Platform Details API

```
curl -k -v -u admin:Cisco123# -X GET 'https://172.19.162.209/api/operational/platform-detail'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
```

```
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
* subject: CN=nfv
* start date: Aug 17 11:21:43 2017 GMT
* expire date: Aug 15 11:21:43 2027 GMT
* issuer: CN=nfv
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/platform-detail HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
>< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 18 Aug 2017 13:21:47 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<


<platform-detail
 xmlns="http://www.cisco.com/nfvos/platform-info"
 xmlns:y="http://tail-f.com/ns/rest"
 xmlns:platform_info="http://www.cisco.com/nfvos/platform-info">
 <hardware_info>
  <Manufacturer>Cisco Systems Inc</Manufacturer>
  <PID>UCSC-C220-M4S</PID>
  <SN>FCH1924V2AH</SN>
  <hardware-version>74-12419-01</hardware-version>
  <UUID>663F3347-5499-0D49-A76E-533A4AA9C755</UUID>
  <Version>3.6.0-916</Version>
  <Compile_Time>Monday, August 07, 2017 [01:30:11 PDT]</Compile_Time>
  <CPU_Information>Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz 8 cores</CPU_Information>
  <Memory_Information>65701956 kB</Memory_Information>
  <Disk_Size>1000.2 GB</Disk_Size>
  <CIMC_IP>NA</CIMC_IP>
 </hardware_info>
 <software_packages>
  <Kernel_Version>3.10.0-514.10.2.el7.x86_64</Kernel_Version>
  <QEMU_Version>1.5.3</QEMU_Version>
  <LibVirt_Version>2.0.0</LibVirt_Version>
  <OVS_Version>2.3.2</OVS_Version>
 </software_packages>
 <port_detail>
  <Name>eth0</Name>
 </port_detail>
 <port_detail>
  <Name>eth1</Name>
 </port_detail>
 <port_detail>
  <Name>eth2</Name>
```

```
 </port_detail>
 <port_detail>
  <Name>eth3</Name>
 </port_detail>
 <port_detail>
  <Name>eth4</Name>
 </port_detail>
 <port_detail>
  <Name>eth5</Name>
 </port_detail>
 <switch_detail>
  <UUID>NA</UUID>
  <Type>NA</Type>
  <Name>NA</Name>
  <Ports>8</Ports>
 </switch_detail>
</platform-detail>
```

Port Details APIs

Table 18: Port Details APIs

Action	Method	Payload Required	API
To get information about the physical port	GET	No	/api/operational/platform-detail/port_detail

Sample Output for the Port Details API

```
curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/platform-detail/port_detail'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
* subject: CN=nfv
* start date: Aug 17 11:21:43 2017 GMT
```

```
* expire date: Aug 15 11:21:43 2027 GMT
* issuer: CN=nfv
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/platform-detail/port_detail HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
>< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 18 Aug 2017 13:24:32 GMT
< Content-Type: application/vnd.yang.collection+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<collection
 xmlns:y="http://tail-f.com/ns/rest">
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth0</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>up</Link>
  <Speed>1000</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:4a:8c:56</MAC>
  <PCI_detail>01:00.0</PCI_detail>
 </port_detail>
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth1</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>up</Link>
  <Speed>1000</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:4a:8c:57</MAC>
  <PCI_detail>01:00.1</PCI_detail>
 </port_detail>
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth2</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>down</Link>
  <Speed>0</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:37:0f:28</MAC>
  <PCI_detail>04:00.0</PCI_detail>
 </port_detail>
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth3</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>down</Link>
  <Speed>0</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:37:0f:29</MAC>
  <PCI_detail>04:00.1</PCI_detail>
```

```
 </port_detail>
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth4</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>down</Link>
  <Speed>0</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:37:0f:2a</MAC>
  <PCI_detail>04:00.2</PCI_detail>
 </port_detail>
 <port_detail
  xmlns="http://www.cisco.com/nfvos/platform-info">
  <Name>eth5</Name>
  <Type>physical</Type>
  <Media>Twisted Pair</Media>
  <Link>down</Link>
  <Speed>0</Speed>
  <MTU>1500</MTU>
  <MAC>80:e0:1d:37:0f:2b</MAC>
  <PCI_detail>04:00.3</PCI_detail>
 </port_detail>
</collection>
```

Portal Access APIs

Table 19: Portal Access APIs

Action	Method	Payload Required	API
To enable or disable the portal access	PUT	Yes	/api/config/system/portal
To get the portal access status	GET	No	/api/operational/system/portal/status

Example for a Portal Access Payload

```
<portal>
    <access>enabled</access>
</portal>
```

Table 20: Portal Access Payload Description

Property	Type	Description	Mandatory/Default Value
access	String	Specify the portal access as "enabled" or "disabled".	Yes

Example: PUT Portal Access (Enable/Disable)

```
curl -v -k -u "admin:Cisco123#" -H "Content-Type:application/vnd.yang.data+xml" -X
PUT https://209.165.201.1/api/config/system/portal -d
"<portal><access>enabled</access></portal>"

*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 14 06:53:22 2017 GMT
*  expire date: Mar 12 06:53:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/portal HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 41
>
* upload completely sent off: 41 out of 41 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Tue, 14 Mar 2017 19:34:42 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Tue, 14 Mar 2017 19:34:42 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-520082-470197
< Pragma: no-cache
```

Example: GET Portal Access API

```
curl -v -k -u  admin:Cisco123# -X GET
'https://209.165.201.1/api/operational/system/portal/status'

Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
```

```
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 14 06:53:22 2017 GMT
*  expire date: Mar 12 06:53:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/portal/status HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 14 Mar 2017 19:35:05 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
```

System Log APIs

Action	Method	Payload Required	API
To set system logs	POST	Yes	/api/operations/system/set-log
To get the system log configuration details	GET	No	/api/operational/system/logging-level

Example for System Log Payload

```
<input>
    <logtype>all</logtype>
    <level>warning</level>
</input>
```

Table 21: Payload Description for Setting Log Level

Property	Type	Description	Mandatory/Default Value
logtype	String	Type of the log. There are two types: configuration and operational. You can specify one of the following: • configuration • operational • all (includes both configuartion and opeartional logs)	Yes
level	String	Indicates the log level. The supported log levels are: debug, info, warning, error, and critcal. **Note** The info and warning log levels are set by default respectively for the configuration and operational log types. You can change them as required. However, the change to the log level is not persisted across a reboot. After a reboot, the default log levels are used.	Yes

Example: POST System Log API

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/operations/system/set-log -d
'<input><logtype>all</logtype><level>warning</level></input>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
```

```
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Dec  8 07:50:20 2016 GMT
*  expire date: Dec  6 07:50:20 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/operations/system/set-log HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 59
>
* upload completely sent off: 59 out of 59 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Thu, 05 Jan 2017 03:49:32 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
```

Example: GET System Log API

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/operational/system/logging-level

*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Dec  8 07:50:20 2016 GMT
*  expire date: Dec  6 07:50:20 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/logging-level HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
```

```
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Thu, 05 Jan 2017 03:45:53 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<logging-level xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
  <configuration>info</configuration>
  <operational>warning</operational>
</logging-level>
```

DPDK Support APIs

Action	Method	Payload Required	API
To enable DPDK and VM migration	POST	Yes	/api/config/system/settings/
To Disable DPDK (in error state)	DELETE	No	/api/config/system/settings/dpdk
To get the status of DPDK	GET	No	/api/operational/system/settings-native/dpdk-status

Table 22: Payload Description for DPDK Support

Property	Type	Description	Mandatory/Default Value
dpdk	String	Specify enabling DPDK	Yes

Example : POST to enable DPDK

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/config/system/settings/
--data '{"dpdk": "enable"}'
```

Example: DELETE to disable DPDK

```
curl -k -v -u admin:admin -X DELETE https://localhost/api/config/system/settings/dpdk
```

Example: GET to get the status of DPDK:

```
curl -k -v -u admin:admin -X GET
https://localhost/api/operational/system/settings-native/dpdk-status
```

Backup and Restore APIs

Backup APIs

Action	Method	Payload Required	API
To start configuration-only backup	POST	Yes	/api/operations/hostaction/backup/configuration-only/
To start configuration-and-vms backup	POST	Yes	/api/operations/hostaction/backup/configuration-and-vms/

Table 23: Payload Description for Setting Log Level

Property	Type	Description	Mandatory/Default Value
file-path	String	Path representing location to the file	Yes

Example: POST to start a configuration-only backup

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/operations/hostaction/backup/configuration-only/
--data '{"input": {"file-path": "intdatastore:sample.bkup"}}'
```

Example: POST to start configuration-and-vms backup:

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/operations/hostaction/backup/configuration-and-vms/
--data '{"input": {"file-path": "intdatastore:sample.bkup"}}'
```

Restore APIs

Action	Method	Payload Required	API
To start restore from a backup package	POST	Yes	/api/operations/hostaction/restore/

Table 24: Payload Description for Setting Log Level

Property	Type	Description	Mandatory/Default Value
restore-option	String	Option to restore without connectivity settings. Accepted values: except-connectivity	No

Example: To start a restore

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/operations/hostaction/restore/
--data '{"input": {"file-path": "intdatastore:sample.bkup"}}'
```

Example: To start a restore while preserving connectivity settings:

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/operations/hostaction/restore/
--data '{"input": {"restore-option": "except-connectivity", "file-path":
"intdatastore:sample.bkup"}}'
```

Route Distribution APIs

Action	Method	Payload Required	API
To configure route distribution	POST	Yes	/api/config/route-distributions
To update route distribution configuration	GET	No	/api/config/route-distributions?deep
To delete route distribution configuration	DELETE	No	/api/config/route-distributions
To get route distribution state data	GET	No	/api/operational/route-distributions

Example for route distribution payload

```
<route-distribute>
 <neighbor-address>172.25.221.106</neighbor-address>
 <local-bridge>wan-br</local-bridge>
 <local-as>65000</local-as>
 <remote-as>65000</remote-as>
 <network-subnet>
  <subnet>10.20.0.0/24</subnet>
 </network-subnet>
</route-distribute>
```

Table 25: Payload Description for Route Distribution

Property	Type	Description	Mandatory/Default Value
neighbor-address	String	Neighbor IPv4 address secure overlay connection.	Yes
local-address	String	Local IPv4 address	No
local-bridge	String	Local bridge name for overlay (default wan-br)	No

local-as	String	Local autonomous system number	Yes
remote-as	String	Remote autonomous system number	Yes
router-id	String	Local router id IP address	No
network-subnet	String	List of subnets to be announced. H.H.H.H/N (atleast one subnet needs to be announced)	Yes
next-hop	String	IPv4 address of any local interface	No

Example: POST create route distribution

```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/route-distributions -d
'<route-distribute><ip-address>172.25.221.106</ip-address><local-bridge-name>br</local-bridge-name><local-as>500</local-as><remote-as>500</remote-as><network-subnet>10.0.0.0/24</network-subnet></route-distribute>'
```

Example: GET update route distribution

```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X PUT
https://209.165.201.1/api/config/route-distributions/route-distribute/172.25.221.106 -d
'<route-distribute><ip-address>172.25.221.106</ip-address><local-bridge-name>br</local-bridge-name><local-as>500</local-as><remote-as>500</remote-as><network-subnet>10.0.0.0/24</network-subnet></route-distribute>'
```

Example: GET route distributions state information

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/route-distributions?deep"
```

Example: DELETE all route distributions

```
curl -k -v -u "admin:admin" -X DELETE "https://209.165.201.1/api/config/route-distributions"
```

Dynamic SR-IOV APIs

Action	Method	Payload Required	API
To enable SR-IOV	PUT	Yes	/api/config/pnics/pnic/eth0-1/sriov/numvfs --
To set switchmode	PUT	No	/api/config/pnics/pnic/eth0-1/sriov/switchmode
To disable SR-IOV	DELETE	No	/api/config/pnics/pnic/eth0-1/sriov
To get SR-IOV operational data	GET	No	/api/operational/pnics/pnic/eth0-1/sriov
To create SR-IOV network with trunk mode	POST	Yes	/api/config/networks

Action	Method	Payload Required	API
To create SR-IOV network with access mode	POST	Yes	/api/config/networks
To delete SR-IOV network	DELETE	No	/api/config/networks/network/eth0-1-SRIOV-1

Example: PUT enable SR-IOV

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X PUT
https://209.165.201.1/api/config/pnics/pnic/eth0-1/sriov/numvfs --data '<numvfs>1</numvfs>'
```

Example: DELETE disable SR-IOV

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X DELETE
https://209.165.201.1/api/config/pnics/pnic/eth0-1/sriov
```

Example: GET SR-IOV operational data

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X GET
https://209.165.201.1/api/operational/pnics/pnic/eth0-1/sriov
```

Example: POST create SR-IOV network with trunk mode

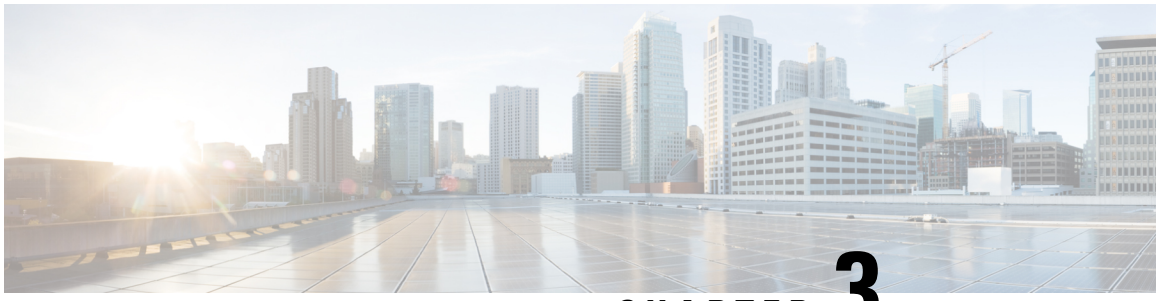
```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/networks
 --data '<network><name>eth0-1-SRIOV-1</name><sriov>true</sriov></network>'
```

Example: POST create SR-IOV network with access mode

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/networks
 --data
'<network><name>eth0-1-SRIOV-1</name><sriov>true</sriov><trunk>false</trunk><vlan>30</vlan></network>'
```

Example: DELETE SR-IOV network

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X DELETE
https://209.165.201.1/api/config/networks/network/eth0-1-SRIOV-1
```



CHAPTER 3

PnP APIs

Certificate Creation APIs

Table 26: Certificate Creation APIs

Action	Method	Payload Required	API
To create a certificate signing request	POST	Yes	/api/operations/system/certificate/signing-request
To install a certificate, which will be used by the local portal and REST API	POST	Yes	/api/operations/system/certificate/install-cert
To switch between self-signed and CA signed certificates	POST	Yes	/api/operations/system/certificate/use-cert

Example for Signing Request Payload

```
<signing-request>
 <country-code>US</country-code>
```

```
 <state>California</state>
 <locality>San Jose</locality>
 <organization>Cisco</organization>
 <organization-unit-name>Cisco</organization-unit-name>
 <common-name>nfvis.cisco.com</common-name>
</signing-request>
```

Table 27: Description for Signing Request Payload

Property	Type	Description	Mandatory/Default Value
<country-code>	String	Two-letter ISO abbreviation for your country.	No
<state>	String	Name of the state where your organization's head office is located.	No
<locality>	Boolean	Name of the city where your organization's head office is located.	No
<organization>	Boolean	Name of the organization	No
<organization-unit-name>	String	Name of the department or group that will use the certificate.	No
<common-name>	URL	Fully qualified domain name that you want to secure.	Yes

Example for Install Certificate Payload

```
<install-cert>
  <path>file:///data/upload1/servercert.pem</path>
</install-cert>
```

Table 28: Description for Install Certificate Payload

Property	Type	Description	Mandatory/Default Value
<install-cert> <path>	URL	Full path of the certificate.	Yes

Example for Use Certificate Payload

```
<use-cert>
  <cert-type>ca-signed</cert-type>
</use-cert>
```

The <cert-type> parameter is mandatory in the use certificate payload. You can .

Table 29: Description for Use Certificate Payload

Property	Type	Description	Mandatory/Default Value

<use-cert> <cert-type>	string	The <self-signed> or <ca-signed> certificate type.	Yes

Example: POST Signing Request API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
POST -d <signing-request><country-code>US</country-code><state>California</state><locality>San
 Jose</locality><organization>Cisco</organization>
<organization-unit-name>Cisco</organization-unit-name><common-name>nfvis.cisco.com</common-name></signing-request>

https://209.165.201.1/api/operations/system/certificate/signing-request
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Apr 04 23:26:13 2016 GMT
* expire date: Apr 02 23:26:13 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/system/certificate/signing-request HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 250
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Wed, 06 Apr 2016 23:29:39 GMT
< Content-Type: application/vnd.yang.operation+xml
< Content-Length: 85
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Vary: Accept-Encoding
< Pragma: no-cache
<
<output xmlns='http://www.cisco.com/nfv'>
 <url>/download/nfvis.csr</url>
</output>
* Connection #0 to host 209.165.201.1 left intact
* Closing connection #0
```

Example: POST Install Certificate API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
POST -d <install-cert><path>file:///data/upload1/servercert.pem</path></install-cert>
 https://209.165.201.1/api/operations/system/certificate/install-cert
* About to connect() to 209.165.201.1 port 443 (#0)
```

```
* Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Apr 04 23:26:13 2016 GMT
* expire date: Apr 02 23:26:13 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/system/certificate/install-cert HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 81
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 06 Apr 2016 23:19:33 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
* Closing connection #0
```

Example: POST Use Certificate API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
POST -d <use-cert><cert-type>ca-signed</cert-type></use-cert>
https://209.165.201.1/api/operations/system/certificate/use-cert
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Apr 04 23:26:13 2016 GMT
* expire date: Apr 02 23:26:13 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/system/certificate/use-cert HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
```

```
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 57
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Wed, 06 Apr 2016 23:23:19 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
* Closing connection #0
```

PnP Action APIs

Table 30: PnP Action API

Action	Method	Payload Required	API
To start, stop, and restart a PnP action	POST	Yes	/api/operations/pnp/action

Example for PnP action Payload

```
<input>
<command><start><stop><restart>
```

Example: POST PnP Action API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST  https://209.165.201.1/api/operations/pnp/action -d
'<input><command>start</command></input>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/pnp/action HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 39
>
* upload completely sent off: 39 out of 39 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 12 Aug 2016 14:38:13 GMT
< Content-Type: text/html
```

```
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
```

PnP APIs

PnP Server APIs

Table 31: PnP Server APIs

Action	Method	Payload Required	API
To get the PnP IP address and port number	GET	No	/api/config/pnp?deep
To get the PnP operational status	GET	No	/api/operational/pnp/status
To modify the PnP IP address and port number	PUT	Yes	/api/config/pnp
To delete the PnP IP address and port number	DELETE	No	/api/config/pnp
To add PnP static IPv6 address	PUT	Yes	/api/config/pnp

Example for PnP Server Payload (Static Mode)

```
<pnp>
<static>
<ip-address>192.0.2.1</ip-address>
<port>80</port>
</static>
<automatic>
<dhcp>disable</dhcp>
<dns>disable</dns>
<cco>disable</cco>
</automatic>
</pnp>
```

Example for PnP Server Payload (Automatic Mode)

```
<pnp>
<automatic>
<dhcp>enable</dhcp>
<dns>enable</dns>
<cco>enable</cco>
<timeout>100</timeout>
```

```
</automatic>
</pnp>
```

Table 32: PnP Server Payload Description

Property	Type	Description	Mandatory/Default Value
<static> <ip-address>	number	Static IP address	Yes (if you disable the automatic option)
<port>	number	Port number	Yes (in static mode)
<dhcp>disable</dhcp> <dhcp>enable</dhcp>	text	Enable or disable DHCP	Yes (one of the options is mandatory)
<dns>disable</dns> <dns>enable</dns>	text	Enable or disable DNS	Yes (one of the options is mandatory)
<cco>disable</cco> <cco>enable</cco>	text	Enable or disable CCO	Yes (one of the options is mandatory)
<timeout>	number	Timeout in seconds. Default is 60 seconds.	No

Example: PUT PnP Server API

Use this API to enable static mode for PnP discovery.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT https://209.165.201.1/api/config/pnp -d
'<pnp><static><ip-address>209.165.201.2</ip-address><port>50</port></static>
<automatic><dhcp>disable</dhcp><dns>disable</dns><cco>disable</cco></automatic></pnp>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> PUT /api/config/pnp HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 154
>
* upload completely sent off: 154 out of 154 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 12 Aug 2016 14:32:04 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Fri, 12 Aug 2016 14:32:04 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1471-12324-598715
< Pragma: no-cache
```

```
<
* Connection #0 to host 209.165.201.1 left intact
```

Use this API to enable automatic mode for PnP discovery.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT https://209.165.201.1/api/config/pnp -d
'<pnp><automatic><timeout>100</timeout><dhcp>enable</dhcp>
<dns>enable</dns><cco>enable</cco></automatic></pnp>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> PUT /api/config/pnp HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 110
>
* upload completely sent off: 110 out of 110 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 12 Aug 2016 14:34:38 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Fri, 12 Aug 2016 14:34:37 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1471-12477-787708
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET PnP Server API

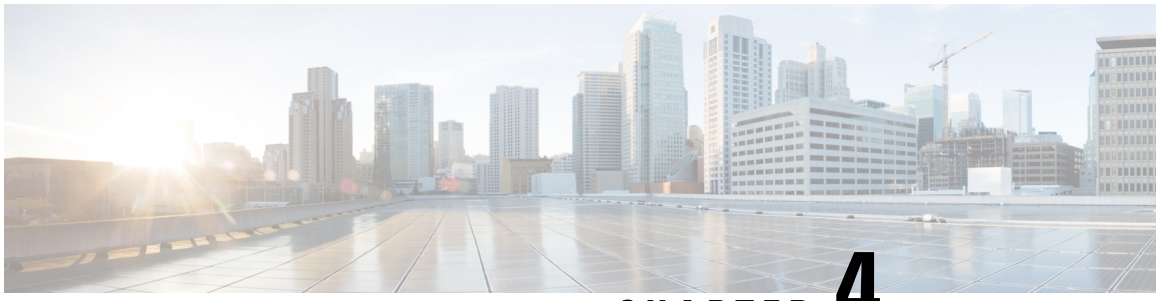
Use this API to get the PnP IP address and port number.

```
curl -X GET -v -k -u admin:admin https://192.0.2.2/api/config/pnp -H
Content-type:application/vnd.yang.data+xml
*   Trying 192.0.2.1...
* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/config/pnp HTTP/1.1
> Host: 192.0.2.2
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.43.0
> Accept: */*
> Content-type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Fri, 19 Aug 2016 09:04:21 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
```

```
< Connection: keep-alive
< Last-Modified: Fri, 19 Aug 2016 08:39:52 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1471-595992-889103
< Pragma: no-cache
<
<pnp xmlns="http://www.cisco.com/nfv/pnp" xmlns:y="http://tail-f.com/ns/rest"
xmlns:pnp="http://www.cisco.com/nfv/pnp">
  <static>
    <ip-address>192.0.2.1</ip-address>
    <port>32</port>
  </static>
  <automatic>
    <dhcp>disable</dhcp>
    <dns>disable</dns>
    <cco>disable</cco>
  </automatic>
  <y:operations>
    <action>/api/config/pnp/_operations/action</action>
  </y:operations>
</pnp>
```

Example: DELETE PnP Server API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
DELETE  https://209.165.201.1/api/config/pnp
*Trying 209.165.201.1...
*Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> DELETE /api/config/pnp HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 12 Aug 2016 14:36:30 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Fri, 12 Aug 2016 14:36:30 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1471-12590-573502
< Pragma: no-cache
<
```



CHAPTER 4

Resource APIs

CPU Allocation Summary API

This API provides the total number of CPUs available for use, and the total number of CPUs that are already used by VMs.

Table 33: CPU Allocation Summary API

Action	Method	Payload Required	API
To get information on the number of CPUs allocated to VMs, and the CPUs that are already used by VMs.	GET	No	api/operational/resources/cpu-info/allocation

Example: GET CPU Allocation Summary API

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/resources/cpu-info/allocation?deep"
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Aug 26 07:41:22 2016 GMT
* expire date: Aug 24 07:41:22 2026 GMT
```

```
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/resources/cpu-info/allocation?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 06:35:48 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<allocation xmlns="http://www.cisco.com/nfvis/resources" xmlns:y="http://tail-f.com/ns/rest"
 xmlns:resource-info="http://www.cisco.com/nfvis/resources">
 <total-sockets>1</total-sockets>
 <cores-per-socket>8</cores-per-socket>
 <total-logical-cpus>16</total-logical-cpus>
 <logical-cpus-used-by-system>2</logical-cpus-used-by-system>
 <logical-cpus-used-by-vnfs>14</logical-cpus-used-by-vnfs>
 <logical-cpus-used-dedicated>12</logical-cpus-used-dedicated>
 <logical-cpus-used-sharable>2</logical-cpus-used-sharable>
</allocation>
* Connection #0 to host 209.165.201.1 left intact
```

Resources CPU APIs

These APIs return CPU information for each CPU or the user specified CPU (cpu-id). These APIs also display a list of VMs (VNF name, VCPU number, VCPU ID) pinned to the CPU or CPUs.

Table 34: Resources CPU APIs

Action	Method	Payload Required	API
To get the VMs running in each physical CPU in the system.	GET	No	• api/operational/resources/cpu-info/cpus • /api/operational/resources/cpu-info/cpus/cpu
To get the VMs running in a specific physical CPU in the system.	GET	No	/api/operational/resources/cpu-info/cpus/cpu/<cpu-id>

Example: GET Resources CPU API

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/resources/cpu-info/cpus/cpu/7?deep"
```

```
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Aug 26 07:41:22 2016 GMT
* expire date: Aug 24 07:41:22 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/resources/cpu-info/cpus/cpu/7?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 06:32:52 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<cpu xmlns="http://www.cisco.com/nfvis/resources" xmlns:y="http://tail-f.com/ns/rest"
xmlns:resource-info="http://www.cisco.com/nfvis/resources">
 <cpu-id>7</cpu-id>
 <socket-id>0</socket-id>
 <core-id>7</core-id>
 <system-use>false</system-use>
 <vnf>
 <name>1472148428.ROUTER</name>
 <vcpus>4</vcpus>
 <low-latency>true</low-latency>
 <vcpu-id>0</vcpu-id>
 </vnf>
</cpu>
* Connection #0 to host 209.165.201.1 left intact
```

Resource Precheck APIs

Use the resource precheck APIs in the following scenarios to check if sufficient resources are available:

- Right before deploying a new VM. Do not proceed to deploy the VM if no sufficient resources are available.
- Right before updating a flavor of a deployed VM. Do not modify the VM if no sufficient resources are available.

Table 35: Resource Precheck APIs

Action	Method	Payload Required	API

Check if there are sufficient resources for the deployment of a VM.	GET	No	/api/operational/resources/precheck/vnf/<vnf_name>,<flavor_name>,<true or false for low-latency>
Check if there are sufficient resources for updating a deployed VM.	GET	No	/api/operational/resources/precheck/vnf/<deployment_name>.<vm_group_name>

Note When the low-latency property of a VM is true, the VM will require one or more dedicated CPUs.

For a new VM, the <vnf_nam> can be any string (for example, "new-vnf"). For updating a deployed VM, the <vnf_name> must be the <deployment_name>.<vm_group_name>.

Example: GET Resource Precheck API

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/resources/precheck/vnf/newvnf,csr1kv-large,true
?deep"
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Aug 26 07:41:22 2016 GMT
* expire date: Aug 24 07:41:22 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/resources/precheck/vnf/newvnf,csr1kv-large,true?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 06:28:59 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<vnf xmlns="http://www.cisco.com/nfvis/resources" xmlns:y="http://tail-f.com/ns/rest"
xmlns:resource-info="http://www.cisco.com/nfvis/res
ources">
 <vnf-name>newvnf</vnf-name>
```

```
 <flavor-name>csr1kv-large</flavor-name>
 <low-latency>true</low-latency>
 <sufficient-resources>false</sufficient-resources>
 <cause>No enough CPU resources</cause>
</vnf>
* Connection #0 to host 209.165.201.1 left intact
```

Resources VM APIs

These APIs return CPU information for each VM or the user specified VM. These APIs also display a list CPUs pinned by the VM.

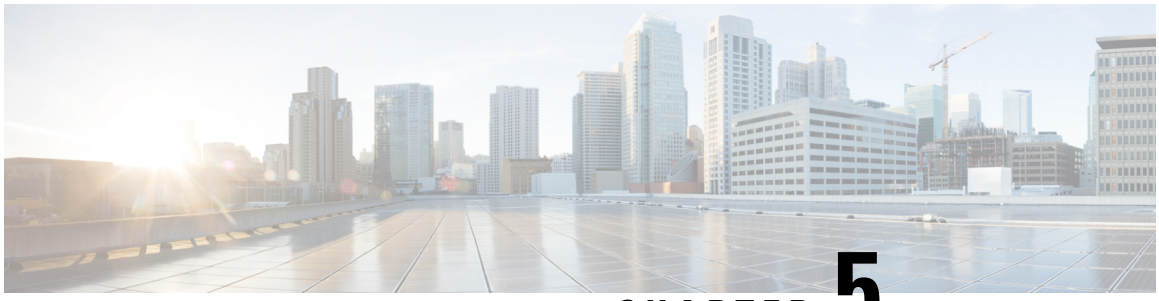
Table 36: Resources VM APIs

Action	Method	Payload Required	API
To get the CPUs and VCPUs allocated to each of the VMs in the system.	GET	No	• /api/operational/resources/cpu-info/vnfs • /api/operational/resources/cpu-info/vnfs/vnf
To get the CPUs and VCPUs allocated to a specific VM in the system.	GET	No	/api/operational/resources/cpu-info/vnfs/vnf/ <deployment_name>.<vm_group_name>

Example: GET Resources VNF API

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.1/api/operational/resources/cpu-info/vnfs/vnf/1472148662.ROUTER2?deep"
* About to connect() to 209.165.201.1 port 443 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Aug 26 07:41:22 2016 GMT
* expire date: Aug 24 07:41:22 2026 GMT
* common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/resources/cpu-info/vnfs/vnf/1472148662.ROUTER2?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
```

```
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 27 Aug 2016 06:35:15 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<vnf xmlns="http://www.cisco.com/nfvis/resources" xmlns:y="http://tail-f.com/ns/rest"
xmlns:resource-info="http://www.cisco.com/nfvis/resources">
 <name>1472148662.ROUTER2</name>
 <vcpus>2</vcpus>
 <low-latency>true</low-latency>
 <cpu>
 <vcpu-id>0</vcpu-id>
 <socket-id>0</socket-id>
 <core-id>3</core-id>
 <cpu-id>3</cpu-id>
 </cpu>
 <cpu>
 <vcpu-id></vcpu-id>
 <socket-id>0</socket-id>
 <core-id>3</core-id>
 <cpu-id>11</cpu-id>
 </cpu>
 <cpu>
 <vcpu-id>1</vcpu-id>
 <socket-id>0</socket-id>
 <core-id>2</core-id>
 <cpu-id>2</cpu-id>
 </cpu>
 <cpu>
 <vcpu-id></vcpu-id>
 <socket-id>0</socket-id>
 <core-id>2</core-id>
 <cpu-id>10</cpu-id>
 </cpu>
</vnf>
* Connection #0 to host 209.165.201.1 left intact
```



CHAPTER 5

Networks and Bridges APIs

Bridge APIs

By default, a LAN bridge (lan-br), a WAN bridge (wan-br) and wan2-br for ENCS 5000 series are created in the system.

Table 37: Bridge APIs

Action	Method	Payload Required	API
To create a bridge	POST	Yes	/api/config/bridges
To verify a bridge configuration	GET	No	/api/config/bridges?deep
To get specific IP/DHCP info for all bridges	GET	No	/api/operational/bridge-settings/<ip/dhcp_configuration>
To get specific IP/DHCP info for specific bridge	GET	No	/api/operational/bridge-settings/<br_name>/<ip/dhcp_configuration>
To modify a bridge, and attach a port to the bridge	PUT	Yes	/api/config/bridges/bridge/<bridge name>
To delete a bridge	DELETE	No	/api/config/bridges/bridge/<bridge name>

Example for Bridge Payload

```
<bridge>
 <name>sc-br</name>
 <port>
  <name>eth3</name>
```

```
 </port>
</bridge>
```

Table 38: Bridge Payload Description

Property	Type	Description	Mandatory/Default Value
bridge name	String	Name of the bridge.	Yes
port name	String	Name of the port the bridge is attached to.	Yes
dhcp		Flag to specify DHCP configuration	No
ip address	String	IP address	No
ip netmask	String	Netmask	No
dhcp-ipv6		Flag to specify DHCP IPv6 configuration	No
slaac-ipv6		Flag to specify SLAAC IPv6 configuration	No
ipv6 address	String	IPv6 address and prefix length	No
vlan	Integer	VLAN tag	No

Example: POST Bridge Creation API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/config/bridges -d
"<bridge><name>sc-br</name><port><name>eth3</name></port><dhcp/><dhcp-ipv6/></bridge>”. "
* About to connect() to 209.165.201.1 port 443 (#0)
*   Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 21 20:02:15 2016 GMT
*  expire date: Mar 19 20:02:15 2026 GMT
*  common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/config/bridges HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 66
>
< HTTP/1.1 201 Created
< Server: nginx/1.6.3
< Date: Sat, 02 Apr 2016 00:21:25 GMT
< Content-Type: text/html
```

```
< Content-Length: 0
< Location: https://209.165.201.1/api/config/bridges/bridge/sc-br
< Connection: keep-alive
< Last-Modified: Sat, 02 Apr 2016 00:21:24 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1459-556484-952070
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Bridge Configuration API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
GET "https://209.165.201.1/api/config/bridges?deep"
* About to connect() to 209.165.201.1 port 443 (#0)
*   Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 21 20:02:15 2016 GMT
*  expire date: Mar 19 20:02:15 2026 GMT
*  common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/config/bridges?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.6.3
< Date: Sat, 02 Apr 2016 00:18:44 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Sat, 02 Apr 2016 00:16:51 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1459-556211-275675
< Pragma: no-cache
<

<bridges xmlns="http://www.cisco.com/nfv/network" xmlns:y="http://tail-f.com/ns/rest"
xmlns:network="http://www.cisco.com/nfv/network">
  <bridge>
    <name>lan-br</name>
    <port>
      <name>eth0</name>
    </port>
  </bridge>
  <bridge>
    <name>wan-br</name>
    <port>
      <name>eth1</name>
```

```
    </port>
    <ip>
      <address>209.165.201.1</address>
      <netmask>255.255.255.0</netmask>
    </ip>
    <ipv6>
      <address>2001:DB8:1:1::72/64</address>
    </ipv6>
  </bridge>
  <bridge>
    <name>sc-br</name>
    <port>
      <name>eth3</name>
    </port>
  </bridge>
</bridges>

* Connection #0 to host 209.165.201.1 left intact
```

Example: GET IPv4 address for all bridges

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X GET
https://localhost/api/operational/bridge-settings/ip-info/ipv4_address
```

Example: GET dhcp enabled under wan-br

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X GET
https://localhost/api/operational/bridge-settings/wan-br/dhcp/enabled
```

Example: DELETE Bridge API

```
curl -k -v -u admin:admin -X
DELETE https://209.165.201.1/api/config/bridges/bridge/sc-br
* About to connect() to 209.165.201.1 port 443 (#0)
*   Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 21 20:02:15 2016 GMT
*  expire date: Mar 19 20:02:15 2026 GMT
*  common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> DELETE /api/config/bridges/bridge/sc-br HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
```

```
< Date: Sat, 02 Apr 2016 00:19:30 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Sat, 02 Apr 2016 00:19:30 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1459-556370-37827
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Network Creation APIs

By default a LAN network (lan-net), a WAN network (wan-net) and wan2-net for ENCS 5000 series are created in the system.

Table 39: Network Creation APIs

Action	Method	Payload Required	API
To create a network	POST	Yes	/api/config/networks
To verify network configuration details	GET	No	/api/config/networks?deep
To modify a network	PUT	Yes	/api/config/networks/network/<network name>
To delete a network	DELETE	No	/api/config/networks/network/<network name>

Example for Network Creation Payload

```
<network>
 <name>sc-net</name>
 <bridge>sc-bridge</bridge>
</network>
```

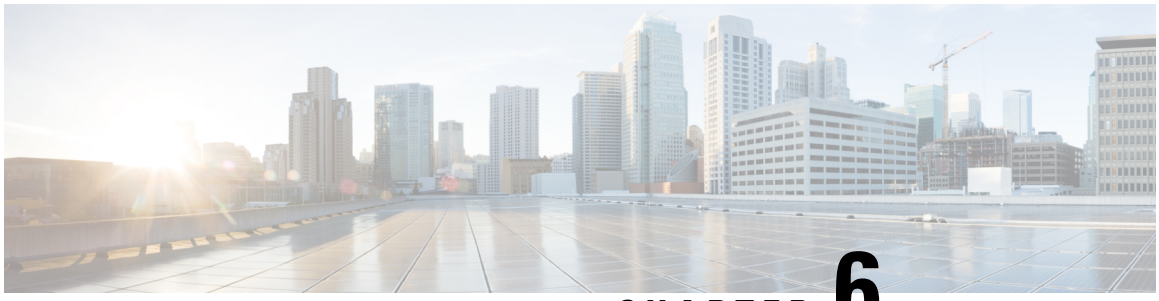
Table 40: Network Creation Payload Description

Property	Type	Description	Mandatory/Default Value
network name	String	Name of the network.	Yes
bridge	String	Name of the bridge the network is attached to.	Yes
trunk	Boolean	Network set to trunk mode.	No/true
sriov	Boolean	SR-IOV supported on the network.	No/false
native-tagged	Boolean	Specifies if the netowrk is tagged or not.	No

native-vlan	Integer	Specifies a native VLAN. It sets the native characteristics when the interface is in **trunk** mode. If you do not configure a native VLAN, the default VLAN 1 is used as the native VLAN.	No
vlan	Integer	Specifies the VLAN number. If the **trunk** parameter is configured as true, this parameter specifies a set of VLAN numbers and ranges If **trunk** parameter is false, access mode is true, then this parameter can have only one VLAN number.	No

Example: POST Network API

```
curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X
POST  https://209.165.201.1/api/config/networks -d
"<network><name>sc-net</name><bridge>sc-bridge</bridge></network>"

* About to connect() to 209.165.201.1 port 443 (#0)
*   Trying 209.165.201.1... connected
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 21 20:02:15 2016 GMT
*  expire date: Mar 19 20:02:15 2026 GMT
*  common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/config/networks HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: 209.165.201.1
> Accept: */*
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 62
>
< HTTP/1.1 201 Created
< Server: nginx/1.6.3
< Date: Sat, 02 Apr 2016 00:14:37 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://209.165.201.1/api/config/networks/network/sc-net
< Connection: keep-alive
< Last-Modified: Sat, 02 Apr 2016 00:14:37 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1459-556077-695828
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```



CHAPTER 6

VM Lifecycle Management APIs

VM Image Registration APIs

Table 41: VM Registration APIs

Action	Method	Payload Required	API
Image registration	POST	Yes	/api/config/vm_lifecycle/images
Get image configuration	GET	No	/api/config/vm_lifecycle/images?deep
Get image status	GET	No	/api/operational/vm_lifecycle/opdata/images/image /<image_name>?deep
Image Unregistration	DELETE	No	/api/config/vm_lifecycle/images/image/<image_name>

Example for Image Registration Payload

```
<image>
 <name>isrv9.16.03.01</name>
 <src>http://<filename_with_full-path-of-the-file>/isrv-universalk9.16.03.01.tar.gz</src>
</image>
```

```
<image>
 <name> mytiny2</name>
 <src>file:///data/mount/nfs_storage/repository/TinyCore-current.iso</src>
 <properties>
  <property>
   <name>placement</name>
   <value>nfs_storage</value>
  </property>
 </properties>
</image>
```

Added in NFVIS 3.12.x release:

```
<image>
        <name>isrv</name>
        <src>https://esc-soltest-124/nfvis/isrv-universalk9.16.03.01.tar.gz</src>
        <certificate_validation>true</certificate_validation>
        <certificate_string>"-----BEGIN CERTIFICATE-----
MIID2TCCAsGgAwIBAgIJAOySjdTedBEyMA0GCSqGSIb3DQEBCwUAMIGCMQswCQYD
VQQGEwJVUzELMAkGA1UECAwCQ0ExCzAJBgNVBAcMAlNKMQ4wDAYDVQQKDAVDaXNj
bzEMMAoGA1UECwwDRVNDMRgwFgYDVQQDDA9lc2Mtc29sdGVzdC0xMjQxITAfBgkq
hkiG9w0BCQEWEmFydmluZGtzQGNpc2NvLmNvbTAeFw0xOTA0MDUxMDQ3NDRaFw0y
MDA0MDQxMDQ3NDRaMIGCMQswCQYDVQQGEwJVUzELMAkGA1UECAwCQ0ExCzAJBgNV
BAcMAlNKMQ4wDAYDVQQKDAVDaXNjbzEMMAoGA1UECwwDRVNDMRgwFgYDVQQDDA9l
c2Mtc29sdGVzdC0xMjQxITAfBgkqhkiG9w0BCQEWEmFydmluZGtzQGNpc2NvLmNv
bTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAN+f6mSR3C5f7bm0622r
hy8IVmF64cjPEjJL7uVa0wid4ohqH7PYjZLlxecWjzwqboMBMX8f5dpqoCCfpIwV
aMDMNQPAWDkLB8D04GgHfZUGmbrKnC/9vzopfIr6zhIsuHU1UGfMu9V+gSK8/1Yd
DXsco4s+JOO+ke+sO+cxghKKzh36R+O6aYNlqNCE3vCIQ91abfx/8pOVGy7+T01g
t9y4v3nTIU0cGAvj6ag6QnQFacU75mYrdHq1SoDF6sJoQhUq3YmiAKVnEKp836sB
jHIDveWqgsj+0aiqHg8z4a6t0WTF1ssVES5mORVY7R2MLcYPtpGedWWW56emuGIe
sbUCAwEAAaNQME4wHQYDVR0OBBYEFBKP8tNQEJCEsof5DqaoDuv4VMWjMB8GA1Ud
IwQYMBaAFBKP8tNQEJCEsof5DqaoDuv4VMWjMAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQELBQADggEBADg04m/U1H12lIacF9ZeItoxp62YDvjszrblj9iKQWxPzPr/
5kbafATak0rAZQ4tHwAGHD6uvmW5zeo5RUMFHDx/FHU+tzjP3dmwSnBAkhicZBu4
uG6ri3PWEPXUlgx/v7liiwYmgT8gCZ7ToD1XzR8x1fPAGwAL48xRmXqiW57cuHWN
RireQ+aIBr7IT61TjdiXnldnjfXcIHGRJStOqoE1QKD44Awq8oguhzOnIyOlZ/AQ
YTv1IYXBvKfDa91EdMS5k6hjeLWjMLYjHWrrB94elQonP6nGfKwD/Zfhsz+1KG6U
JmPyR3GTWwbpB8TmiD80hSXDJNxuHpTRdSO5BUc=
-----END CERTIFICATE-----"</certificate_string>
    </image>
```

```
<image>
        <name>isrv</name>
        <src>https://esc-soltest-124/nfvis/isrv-universalk9.16.03.01.tar.gz</src>
        <certificate_validation>true</certificate_validation>

<certificate_file>file:///data/intdatastore/uploads/esc-soltest-124.cert</certificate_file>

    </image>
```

Table 42: Image Registration Payload Description

Property	Type	Description	Mandatory/Default Value
name	String	Name of the VM image	Yes

src	URL	Full path of the VM image	Yes
Added in NFVIS 3.12.x release			
certificate_validation	True/false	Enable certificate validation by setting this tag to "true"	Yes
certificate_string	String	Validate Web-server with the raw contents of a certificate file	Yes
certificate_file	URL	Validate Web-server with a certificate file	Yes

Example: POST Image Registration API

```
curl -v -u -k admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml
 -X POST https://<NFVIS_IP>/api/config/vm_lifecycle/images -d
'<image><name>WinServer2012R2.iso</name><src><filename_with_full-path-of
the-file>/WinServer2012R2.iso</src></image>'
\* About to connect() to 209.165.201.1 port 80 (#0)
\* Trying 209.165.201.1...
\* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
\* Server auth using Basic with user 'admin'
> POST /api/config/vm_lifecycle/images HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 87
>
\* upload completely sent off: 87 out of 87 bytes
< HTTP/1.1 201 Created
< Server:
< Location: [http://209.165.201.1/api/config/vm_lifecycle/images/image/WinServer2012R2.iso]
< Date: Thu, 10 Dec 2015 11:15:50 GMT
< Last-Modified: Thu, 10 Dec 2015 11:15:50 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-746150-710421
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
\* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Image Registration to External Disk API

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/vm_lifecycle/images -d
'<image><name>linux</name><src>file:///mnt/xdatastore2/ploads/Tinylinux.tar.gz</src><properties><property><name>datastore</name><value>datastore3</value></property></properties></image>'
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 209.165.201.1...
* TCP_NODELAY set
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
```

```
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/ssl/cert.pem
CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* ALPN, server did not agree to a protocol
* Server certificate:
* subject: CN=nfvis
* start date: Jun 12 19:40:33 2018 GMT
* expire date: Jun 11 19:40:33 2023 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/vm_lifecycle/images HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.54.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 190
>
* upload completely sent off: 190 out of 190 bytes
< HTTP/1.1 201 Created
< Server: nginx
< Date: Tue, 12 Jun 2018 22:59:05 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://172.25.221.106/api/config/vm_lifecycle/images/image/Linuxnew
< Connection: keep-alive
< Last-Modified: Tue, 12 Jun 2018 22:59:04 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1528-844344-814906
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Image Configuration API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/config/vm_lifecycle/images?deep
\* About to connect() to 209.165.201.1 port 80 (#0)
\* Trying 209.165.201.1...
\* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
\* Server auth using Basic with user 'admin'
> GET /api/config/vm_lifecycle/images?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
```

```
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server:
< Date: Thu, 10 Dec 2015 11:16:10 GMT
< Last-Modified: Thu, 10 Dec 2015 11:15:50 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-746150-710421
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Pragma: no-cache
<
<images xmlns="[http://www.cisco.com/esc/esc|http://www.cisco.com/nfvis/vm_lifecycle]"
xmlns:y="[http://tail-f.com/ns/rest|http://tail-f.com/ns/rest]" 
xmlns:esc="[http://www.cisco.com/nfvis/vm_lifecycle|http://www.cisco.com/nfvis/vm_lifecycle]">

 <image>
 <name>isrv-9.16.03.01</name>
 <src>http://data/nfvos-pkg/isr/isrv-universalk9.16.03.01.tar.gz</src>
 </image>
</images>
\* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Image Status API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/operational/vm_lifecycle/opdata/images/image/isrv-03.16.02?deep
/* About to connect() to 209.165.201.1 port 80 (#0)
/*   Trying 209.165.201.1...
/* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
/* Server auth using Basic with user 'admin'
> GET /api/operational/vm_lifecycle/opdata/images/image/isr-image?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> < HTTP/1.1 200 OK
< Server:
< Date: Thu, 10 Dec 2015 11:16:22 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Content-Type: application/vnd.yang.data+xml< Transfer-Encoding: chunked
< Pragma: no-cache<
<image xmlns="http://www.cisco.com/nfvis/vm_lifecycle" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:esc="http://www.cisco.com/nfvis/vm_lifecycle">
<name>isrv.03.16.02</name>
<image_id>585a1792-145c-4946-9929-e040d3002a59</image_id>
<public>true</public>
<state>IMAGE_ACTIVE_STATE</state></image>
/* Connection #0 to host 209.165.201.1 left intact
```

Note The supported image states are:

- IMAGE_UNDEF_STATE
- IMAGE_CREATING_STATE
- IMAGE_ACTIVE_STATE
- IMAGE_DELETING_STATE
- IMAGE_DELETED_STATE
- IMAGE_ERROR_STATE

Example: DELETE Image Registration API

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


/*About to connect() to 209.165.201.1 port 80 (#0)
/*   Trying 209.165.201.1...
/* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
/* Server auth using Basic with user 'admin'
> DELETE /api/config/vm_lifecycle/images/image/isr-image HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml>
< HTTP/1.1 204 No Content
< Server:
< Date: Thu, 10 Dec 2015 12:44:28 GMT
< Last-Modified: Thu, 10 Dec 2015 12:44:28 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-751468-864441
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
</* Connection #0 to host 209.165.201.1 left intact
```

Custom Flavor Creation APIs

After registering a VM, you can define custom flavors of the VM based on your requirements. These flavors are also known as profiles.

Table 43: Flavor Creation APIs

Action	Method	Payload Required	APIs

To create a flavor	POST	Yes	/api/config/vm_lifecycle/flavors
To get configuration details of a flavor	GET	No	• /api/config/vm_lifecycle/flavors • /api/config/vm_lifecycle/flavors?deep • /api/config/vm_lifecycle /flavors/flavor/<flavor_name>?deep
To view the operational status of a flavor	GET	No	/api/operational/vm_lifecycle /opdata/flavors/flavor/<flavor_name>?deep
To delete a flavor	DELETE	No	/api/config/vm_lifecycle/flavors/flavor/<flavor-name>

Example for Flavor Creation Payload

```
<flavor>
 <name>ISR_FLAVOR</name>
 <vcpus>2</vcpus>
 <memory_mb>4096</memory_mb>
 <root_disk_mb>0</root_disk_mb>
 <ephemeral_disk_mb>0</ephemeral_disk_mb>
 <swap_disk_mb>0</swap_disk_mb>
</flavor>
```

Table 44: Description for Flavor Creation Payload

Property	Type	Description	Mandatory/Default Value
name	String	Name of the flavor.	Yes
vcpus	Number	Number of virtual CPUs.	Yes
memory_mb	Number	Amount of memory in Mega Bytes.	Yes
root_disk_mb	Number	Virtual root disk size in gigabytes.	Yes **Note** Added support in 3.7.1
ephemeral_disk_mb	Number	A temporary storage that is added to your instance.	No
swap_disk_mb	Number	The space used on a hard disk as RAM	No

Example: POST Flavor API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/config/vm_lifecycle/flavors -d
'<flavor><name>windows</name><ephemeral_disk_mb>0</ephemeral_disk_mb><memory_mb>4096</memory_mb><root_disk_mb>1228</root_disk_mb><swap_disk_mb>0</swap_disk_mb><vcpus>2</vcpus></flavor>'
* About to connect() to 209.165.201.1 port 80 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> POST /api/config/vm_lifecycle/flavors HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 188
>
* upload completely sent off: 188 out of 188 bytes
< HTTP/1.1 201 Created< Server:
< Location: http://209.165.201.1/api/config/vm_lifecycle/flavors/flavor/ISR_FLAVOR_demo
< Date: Fri, 11 Dec 2015 11:15:23 GMT
< Last-Modified: Fri, 11 Dec 2015 11:15:23 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-832523-873124
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Flavor Configuration API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
GET https://209.165.201.1/api/config/vm_lifecycle/flavors?deep
* About to connect() to 209.165.201.1 port 80 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /api/config/vm_lifecycle/flavors?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server:
< Date: Fri, 11 Dec 2015 11:11:31 GMT
< Last-Modified: Fri, 11 Dec 2015 01:32:26 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-797546-701321
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Pragma: no-cache
<
<flavors xmlns="http://www.cisco.com/nfvis/vm_lifecycle" xmlns:y="http://tail-f.com/ns/rest"
```

```
  xmlns:esc="http://www.cisco.com/nfvis/vm_lifecycle">
 <flavor>
  <name>ASAv10</name>
  <description>ASAv10 profile</description>
  <vcpus>1</vcpus>
  <memory_mb>2048</memory_mb>
  <root_disk_mb>8192</root_disk_mb>
  <ephemeral_disk_mb>0</ephemeral_disk_mb>
  <swap_disk_mb>0</swap_disk_mb>
  <properties>
    <property>
    <name>source_image</name>
    <value>ASAv_IMAGE</value>
    </property>
   </properties>
  </flavor>
  <flavor>
   <name>ASAv30</name>
   <description>ASAv30 profile</description>
   <vcpus>4</vcpus>
   <memory_mb>8192</memory_mb>
   <root_disk_mb>16384</root_disk_mb>
   <ephemeral_disk_mb>0</ephemeral_disk_mb>
   <swap_disk_mb>0</swap_disk_mb>
   <properties>
     <property>
     <name>source_image</name>
     <value>ASAv_IMAGE</value>
     </property>
   </properties>
  </flavor>
</flavors>
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Flavor Status API

```
curl -k -v -u admin:admin -X
GET https://209.165.201.1/api/operational/vm_lifecycle/flavors?deep
* About to connect() to 209.165.201.1 port 80 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /api/operational/vm_lifecycle/flavors?deep HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept: */*>
< HTTP/1.1 200 OK< Server:
< Date: Fri, 11 Dec 2015 10:58:48 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Pragma: no-cache
<flavors xmlns="http://www.cisco.com/nfvis/vm_lifecycle" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:esc="http://www.cisco.com/nfvis/vm_lifecycle">
 <flavor>
   <name>ASAv10</name>
   <description>ASAv10 profile</description>
   <vcpus>1</vcpus>
   <memory_mb>2048</memory_mb>
```

```
   <root_disk_mb>8192</root_disk_mb>
   <ephemeral_disk_mb>0</ephemeral_disk_mb>
   <swap_disk_mb>0</swap_disk_mb>
    <properties>
     <property>
     <name>source_image</name>
     <value>ASAv_IMAGE</value>
     </property>
    </properties>
 </flavor>
 <flavor>
   <name>ASAv30</name>
   <description>ASAv30 profile</description>
   <vcpus>4</vcpus>
   <memory_mb>8192</memory_mb>
   <root_disk_mb>16384</root_disk_mb>
   <ephemeral_disk_mb>0</ephemeral_disk_mb>
   <swap_disk_mb>0</swap_disk_mb>
    <properties>
     <property>
     <name>source_image</name>
     <value>ASAv_IMAGE</value>
     </property>
    </properties>
 </flavor>
</flavors>
* Connection #0 to host 209.165.201.1 left intact
```

VM Deployment APIs

Table 45: VM Deployment APIs

Action	Method	Payload Required	API
Deploy a VM	POST	Yes	/api/config/vm_lifecycle/tenants/tenant/admin/deployments
Get deployment configuration	GET	No	/api/config/vm_lifecycle/tenants/tenant/admin/deployments?deep
Get deployment status and details	GET	No	• /api/operational/vm_lifecycle/tenants/tenant /admin/deployments?deep • /api/operational/vm_lifecycle/opdata/tenants/tenant/admin /deployments/(deployment_name),-,-?deep
Undeploy a VM	DELETE	No	/api/config/vm_lifecycle/tenants/tenant/admin/deployments /deployment/<deployment_name>

Example for VM Export with selective disk Payload

```
<vmexport_policy>
            <disk_exclusion>
              <disk_name>Linux_IMAGE:512G-file.qcow2</disk_name>
          </disk_exclusion>
        </vmexport_policy>
```

Example: POST VM Deployment API for Cisco ISRv

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


/* About to connect() to 209.165.201.1 port 80 (#0)
/* Trying 209.165.201.1...
/* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
/* Server auth using Basic with user 'admin'
> POST /api/config/vm_lifecycle/tenants/tenant/admin/deployments HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 1313
> Expect: 100-continue
> * Done waiting for 100-continue
< HTTP/1.1 201 Created
< Server:
< Location:
http://209.165.201.1/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/WinServer2012R2
```

```
< Date: Thu, 10 Dec 2015 11:17:53 GMT
< Last-Modified: Thu, 10 Dec 2015 11:17:53 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-746273-842306
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
/* Connection #0 to host 209.165.201.1 left intact
```

Note To enable NIM support on a Cisco ISRv running on Cisco ENCS, you must use the following variable in the ISRv deployment payload.

```
<variable>
     <name>ngio</name>
     <val>enable</val>
</variable>
```

Table 46: Description for VM Deployment Payload

Property	Type	Description	Mandatory/Default Value
deployment name	string	Name of the deployment	Yes
vm_group name vim_vm_name	string	Name of the VM group.	Yes
vm_group image	string	Image name that was used to register.	Yes
bootup_time	integer	Bootup time could vary depending on the VM image that you have chosen. For example, bootup time is 600 seconds for a Cisco ISRv image. If no monitoring is required for the VM, set the bootup time as -1. **Note** A monitored VM must have a valid bootup time. The corresponding KPI fields are mandatory for the monitored VM. In the case of an unmonitored VM, KPI fields are optional.	Yes
placement type	string	Set VM deployment placement. For example deploying the VM on external datastore if the system has external datastore. Must set value to "zone_host" if deploying the VM on external data store or NFS.	No

placement host	string	Specify placement datastore. For example ENCS system has external datastore. Specify placement host. Allowed values are: datastore2, datastore3, nfs_storage	No
recovery_wait_time	integer	Time in seconds that this VM takes to perform a normal warm reboot. This will be used to avoid premature VM recovery in case VM becomes unresponsive due to operator reboot. This is important as VM recovery will results in loss of data that is stored on root disk. If speedy recovery is more important than the data on the root disk, this value can be optionally set to 0.	
recovery_policy action_on_recovery	string	The action performed during recovery. Possible values: REBOOT_ONLY; REDEPLOY_ONLY; REBOOT_THEN_REDEPLOY	Yes (for monitored VMs)
interface nicid	integer	The network interface card ID. **Note** At least one NIC ID is mandatory for monitored VMs. It is optional for unmonitored VMs.	Yes (for monitored VMs)
network	string	Name of the network attached to the NIC ID. All networks (such as LAN and WAN) except the internal management network require an IP address. The vNIC attachment to the internal management network is only required for VMs, which require monitoring. If this interface is for monitoring, network must be set to "int-mgmt-net"	Yes (for monitored VMs)
ip_address	string	IPv4 address	Yes
port_forwarding	-	**Note** If port forwarding is included, all elements under it are mandatory.	No
port type	enum	SSH, HTTPS, TCP, and Telnet	No
protocol	string	TCP	No
vnf_port	integer	Port number corresponding to the protocol used.	No
external_port_range start end	integer	Unique port number to specify the start and end range.	No

scaling	container	Specifies how many instances of a particular type of VM need to be instantiated, and whether elastic scale-in and scale-out are required.	Yes
min_active	integer	Describes the minimum number of VMs to be activated. Value currently supported: 1	Yes
max_active	integer	Describes the maximum number of VMs to be activated. Value currently supported: 1	Yes
kpi_data	-	Key performance indicators data.	Yes (for monitored VMs)
event_name	string	Name of the event.	Yes (for monitored VMs)
metric_value	string	The metric threshold value of the KPI.	Yes (for monitored VMs)
metric_cond	enum	Specifies the direction of the metric value change for this KPI. There are four valid values: • GE & GT—An alarm is sent when the metric value increases from a lower position to equal or exceed the specified value. • LE & LT—An alarm is sent when the metric value decreases from a higher position to equal or go down the specified value.	Yes (for monitored VMs)
metric_type	integer	Supported metric types are INT8, UINT8, INT16, UINT16, INT32, UINT32, FLOAT, DOUBLE, and STRING .	Yes (for monitored VMs)
metric_collector type	String	If the image boot-up time is provided, monitoring must be set to ICMPPing. This field type can be empty if boot-up time is -1.	Yes (for monitored VMs)
nicid>	Integer	The card ID of the interface through which this VM is monitored. It should be the ID specified in one of interfaces section in the payload.	Yes (for monitored VMs)
poll_frequency	Integer		Yes (for monitored VMs)
polling_unit	string		Yes (for monitored VMs)
continuous alarm	boolean	Continuous events needs to be generated. Value supported: false, true	Yes (for monitored VMs)

rule event_name	string	Name of the event.	No
action	string	• Always log—Whether the event is pingable or not, the details are always logged. • TRUE servicebooted.sh—The action identified by this keyword in the dynamic mapping file is triggered when the VM moves from a non-pingable to a pingable state. • FALSE recover autohealing—The action identified by this keyword is triggered, and the VM is recovered without the administrator's intervention.	No
configuration dst	string	If the VM supports the bootstrap configuration file in the VM package, and a token is included in the configuration file, this token must be filled in the bootstrap template during the VM deployment.	No
variable name	string	TECH_PACKAGE is the token for a Cisco ISRv image. This needs to be specified in the variable name. This varies with each VM.	Yes

Example: DELETE VM Deployment API

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
DELETE
https://209.165.201.1/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISRdep1
/* About to connect() to 209.165.201.1 port 80 (#0)
/* Trying 209.165.201.1...
/* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
/* Server auth using Basic with user 'admin'
> DELETE /api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISRdep1
HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server:
< Date: Thu, 10 Dec 2015 12:43:31 GMT
< Last-Modified: Thu, 10 Dec 2015 12:43:31 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1449-751411-880440
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
```

```
/* Connection #0 to host 209.165.201.1 left intact
```

Examples for VM Deployment Payload with Bootstrap Configuration Options

Note You need to specify the exact name of the VM bootstrap configuration file under the <dst> element in the deployment payload. This name can vary with each VM. For example, the Cisco ASAv bootstrap configuration file is "day0-config".

Option 1 Example: Deployment Payload for Bundling Bootstrap Configuration Files into the VM Package

In this method, the bootstrap configuration variables can be tokenized. You have to provide token values at the time of deployment using the deployment payload.

The following is the extract from the Cisco ASAv bootstrap configuration with tokenized variables. Tokenized variables are highlighted in this example.

```
ASA Version 9.4.1
firewall transparent
ssh version 2
!
interface management0/0
description vnf-mgmt-net
nameif vnf-mgmt
security-level 100
ip address ${VNF_MGMT_IP} ${VNF_MGMT_NETMASK}
no shutdown
!
interface GigabitEthernet0/0
description service-net
nameif outside
security-level 0
bridge-group 10
no shutdown
!
interface GigabitEthernet0/1
description lan-net
nameif inside
bridge-group 10
security-level 100
no shutdown
!
interface BVI10
ip address ${BRIDGE_IP} ${BRIDGE_MASK}
!
snmp-server enable
snmp-server community public
http server enable
http 0.0.0.0 0.0.0.0 management
crypto key generate rsa modulus 2048
username test password test123
ssh 0.0.0.0 0.0.0.0 management
aaa authentication ssh console LOCAL
route vnf-mgmt 0.0.0.0 0.0.0.0 ${VNF_MGMT_GW} 1
route outside 0.0.0.0 0.0.0.0 ${BRIDGE_GW} 1
```

The following is an example for the Cisco ASAv deployment payload with the tokenized variables.

```
<deployment>
 <name>ASAv</name>
 <vm_group>
  <name>FirwallGroup</name>
  <image>asavImage</image>
  <flavor>IASAv5l</flavor>
  <bootup_time>600</bootup_time>
  <recovery_wait_time>0</recovery_wait_time>
  <interfaces>
   <interface>
    <nicid>0</nicid>
    <network>int-mgmt-net</network>
    <port_forwarding>
     <port>
      <type>ssh</type>
      <protocol>tcp</protocol>
      <vnf_port>22</vnf_port>
      <external_port_range>
       <start>20024</start>
       <end>20024</end>
      </external_port_range>
     </port>
    </port_forwarding>
   </interface>
   <interface>
    <nicid>1</nicid>
    <network>sc-net</network>
   </interface>
   <interface>
    <nicid>2</nicid>
    <network>lan-net</network>
   </interface>
  </interfaces>
  <kpi_data>
   <enabled>true</enabled>
   <kpi>
    <event_name>VM_ALIVE</event_name>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_collector>
     <type>ICMPPing</type>
     <nicid>0</nicid>
     <poll_frequency>3</poll_frequency>
     <polling_unit>seconds</polling_unit>
     <continuous_alarm>false</continuous_alarm>
    </metric_collector>
   </kpi>
  </kpi_data>
  <rules>
   <admin_rules>
    <rule>
     <event_name>VM_ALIVE</event_name>
     <action>ALWAYS log</action>
     <action>FALSE recover autohealing</action>
     <action>TRUE servicebooted.sh</action>
    </rule>
   </admin_rules>
   <user_rules/>
  </rules>
  <scaling>
```

```
   <min_active>1</min_active>
   <max_active>1</max_active>
  </scaling>
  <config_data>
   <configuration>
    <dst>day0-config</dst>
    <variable>
     <name>VNF_MGMT_IP</name>
     <val>192.0.2.6</val>
    </variable>
    <variable>
     <name>VNF_MGMT_NETMASK</name>
     <val>255.255.255.0</val>
    </variable>
    <variable>
     <name>BRIDGE_IP</name>
     <val>192.0.2.10</val>
    </variable>
    <variable>
     <name>BRIDGE_MASK</name>
     <val>255.255.255.0</val>
    </variable>
    <variable>
     <name>VNF_MGMT_GW</name>
     <val>192.0.2.7</val>
    </variable>
    <variable>
     <name>BRIDGE_GW</name>
     <val>192.0.2.12</val>
    </variable>
   </configuration>
  </config_data>
 </vm_group>
</deployment>
```

Option 2 Example: Bootstrap Configuration without Tokens in the Deployment Payload

In this example, the entire Cisco ASAv bootstrap configuration is copied under the **<data>** element.

```
<deployment>
 <name>ASAv</name>
 <vm_group>
  <name>ASAvGroup</name>
  <bootup_time>-1</bootup_time>
  <config_data>
   <configuration>
    <dst>day0-config</dst>
    <data>
    ASA Version 9.4.1
    firewall transparent
    ssh version 2
    interface management0/0
    description vnf-mgmt-net
    nameif vnf-mgmt
    security-level 100
    ip address 11.20.0.3 255.255.255.0
    no shutdown
    interface GigabitEthernet0/0
    description service-net
    nameif outside
    security-level 0
    bridge-group 10
```

```
    no shutdown
!
    interface GigabitEthernet0/1
    description lan-net
    nameif inside
    bridge-group 10
    security-level 100
    no shutdown
    interface BVI10
    ip address 12.20.0.3 255.255.255.0
!
    snmp-server enable
    snmp-server community public
    http server enable
    http 0.0.0.0 0.0.0.0 management
    crypto key generate rsa modulus 2048
    username test password test123
    ssh 0.0.0.0 0.0.0.0 management
    aaa authentication ssh console LOCAL
    route vnf-mgmt 0.0.0.0 0.0.0.0 11.20.0.1 1
    route outside 0.0.0.0 0.0.0.0 12.20.0.1 1
    </data>
 <image>ASAvImage</image>
  <interfaces>
   <interface>
   <nicid>0</nicid>
   <network>vnf-mgmt-net</network>
   </interface>
   <interface>
   <nicid>1</nicid>
   <ip_address>12.20.0.68</ip_address>
   <network>sc-net</network>
   </interface>
  </interfaces>
  <kpi_data>
   <kpi>
   <event_name>VM_ALIVE</event_name>
   <metric_collector>
   <continuous_alarm>false</continuous_alarm>
   <nicid>0</nicid>
   <poll_frequency>3</poll_frequency>
   <polling_unit>seconds</polling_unit>
   <type>ICMPPing</type>
   </metric_collector>
   <metric_cond>GT</metric_cond>
   <metric_type>UINT32</metric_type>
   <metric_value>1</metric_value>
   </kpi>
  </kpi_data>
  <recovery_wait_time>0</recovery_wait_time>
  <rules>
   <admin_rules>
    <rule>
    <event_name>VM_ALIVE</event_name>
    <action>ALWAYS log</action>
    <action>TRUE servicebooted.sh</action>
    <action>FALSE recover autohealing</action>
    </rule>
   </admin_rules>
  </rules>
  <scaling>
   <max_active>1</max_active>
   <min_active>1</min_active>
  </scaling>
```

```
 </vm_group>
</deployment>
```

Option 3 Example: Deployment Payload with Local Bootstrap Configuration File

In this example, a reference to the Cisco ASAv local bootstrap configuration file is provided from the payload under the **<configuration>** element. If the bootstrap configuration file has tokens, you have to provide token values in the deployment payload under the configuration section.

```
<deployment>
    <name>asaV</name>
    <vm_group>
        <name>firewall_Group</name>
        <image>ASAvImage</image>
        <bootup_time>600</bootup_time>
        <recovery_wait_time>0</recovery_wait_time>
        <recovery_policy>
            <action_on_recovery>REBOOT_ONLY</action_on_recovery>
        </recovery_policy>
        <interfaces>
            <interface>
                <nicid>0</nicid>
                <network>int-mgmt-net</network>
                <port_forwarding>
                    <port>
                        <type>ssh</type>
                        <protocol>tcp</protocol>
                        <vnf_port>22</vnf_port>
                        <external_port_range>
                            <start>20022</start>
                            <end>20022</end>
                        </external_port_range>
                    </port>
                </port_forwarding>
            </interface>
           <interface>
                <nicid>1</nicid>
                <network>wan-net</network>
                <ip_address>172.19.181.42</ip_address>
            </interface>
            <interface>
                <nicid>2</nicid>
                <network>lan-net</network>
                <ip_address>192.168.0.20</ip_address>
            </interface>
        </interfaces>
        <scaling>
            <min_active>1</min_active>
            <max_active>1</max_active>
        </scaling>
        <kpi_data>
            <kpi>
                <event_name>VM_ALIVE</event_name>
                <metric_value>1</metric_value>
                <metric_cond>GT</metric_cond>
                <metric_type>UINT32</metric_type>
                <metric_collector>
                    <type>ICMPPing</type>
                    <nicid>0</nicid>
                    <poll_frequency>3</poll_frequency>
                    <polling_unit>seconds</polling_unit>
                    <continuous_alarm>false</continuous_alarm>
```

```
                </metric_collector>
            </kpi>
        </kpi_data>
        <rules>
      <admin_rules>
                <rule>
                    <event_name>VM_ALIVE</event_name>
                    <action>ALWAYS log</action>
                    <action>TRUE servicebooted.sh</action>
                    <action>FALSE recover autohealing</action>
                </rule>
            </admin_rules>
        </rules>
        <config_data>
           <configuration>
                <dst>day0-config</dst>
               <file>file://data/upload1/day0-config</file>
           </configuration>
        </config_data>
    </vm_group>
</deployment>
```

Adding or Editing a vNIC Using the VM Deployment API

Using the VM deployment API, you can add, edit, or delete as many vNICs as you want. For these actions, you will have to use the PUT method of the VM deployment API. VM's vNIC can be updated when VM is active or stopped.

Note Editing vNIC (add / delete / changing network) will reboot the VM is the VM does not support vNIC hot-add / hot-delete / hot-modify.

Example: Adding more than one vNIC

You should know the deployment name and the VM group name to use the PUT form of the VM deployment API. To get them, use the following commands before running the PUT form of the VM deployment API :

- GET https://<server_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments—Provides the names of all VMs that are deployed.
- GET https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISR1—Provides the VM group name for a particular deployment.

Additional interfaces are passed into the same deployment URL as shown in this example. A new vNIC (NIC ID 2) is added to the deployed VM, ISR1.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT
https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISR1/vm_group/ISR-VM/interfaces
 --data
'<interfaces>
    <interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
    </interface>
```

```
    <interface>
        <nicid>1</nicid>
        <network>sc-net</network>
    </interface>
    <interface>
        <nicid>2</nicid>
        <network>lan-net</network>
    </interface>
</interfaces>'
```

Example: Editing a vNIC

You can edit the attributes of an existing vNIC. In this example, the network is changed from **sc-net** to **wan-net** for NIC ID 1.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT
https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISR1/vm_group/ISR-VM/interfaces
 --data
'<interfaces>
    <interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
    </interface>
    <interface>
        <nicid>1</nicid>
        <network>wan-net</network>
    </interface>
    <interface>
        <nicid>2</nicid>
        <network>lan-net</network>
    </interface>
</interfaces>'
```

Example: Deleting a vNIC

To delete a vNIC that is part of the VM deployed, remove the vNIC ID from the payload, and then run the PUT form of the VM deployment API. For example, assume that you want to remove vNIC 2 from the above configuration (ISR1 deployment), use the PUT form of the VM deployment API as shown in the example:

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
PUT
https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISR1/vm_group/ISR-VM/interfaces
 --data
'<interfaces>
    <interface>
        <nicid>0</nicid>
        <network>int-mgmt-net</network>
    </interface>
    <interface>
        <nicid>1</nicid>
        <network>sc-net</network>
    </interface>
  </interfaces>'
```

See the Example: POST VM Deployment API for Cisco ISRv, on page 93 for details on the API command.

Changing the Flavor Using the VM Deployment API

Using this deployment API, you can change or update the flavor. Before changing an existing flavor to a new one, ensure that you have the new flavor created using the flavor creation API. VM's flavor change be updated when VM is active or stopped.

Example: Changing the Flavor

In this example, the existing flavor ID is changed to **isr-flavor** for the VM deployed as ISR1.

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml
-X PUT
https://<nfvis_ip>/api/config/vm_lifecycle/tenants/tenant/admin/deployments/deployment/ISR1/vm_group/ISR-VM/flavor
 --data
'<flavor>isr-flavor</flavor>'
```

Note A VM is automatically power cycled when a flavor of the VM is changed.

See the Example: POST VM Deployment API for Cisco ISRv, on page 93 for details on the API command.

VM Action APIs

Note You may want to get the VM name before running the VM operations API. To get the VM name, use the following operational status API:

/api/operational/vm_lifecycle/opdata/tenants/tenant/admin/deployments/<deploy name>,-,-?deep

Table 47: VM Operations APIs

Action	Method	Payload Required	APIs
To start a VM	POST	Yes	/api/operations/vmAction
To stop a VM	POST	Yes	/api/operations/vmAction
To reboot a VM	POST	Yes	/api/operations/vmAction
To enable VM monitoring	POST	Yes	/api/operations/vmAction
To disable VM monitoring	POST	Yes	/api/operations/vmAction
To backup a VM	POST	Yes	/api/operations/vmExportAction
To restore a VM	POST	Yes	/api/operations/vmImportAction

VM Live Export Support

Only VMs deployed with QCOW2 disks on non-nfs datastores can be live exported. All VMs do not support live export. To verify if a VM supports live export, use the following api:

```
nfvis# show vm_lifecycle opdata tenants tenant admin deployments OTHER vm_group
supported_export_type
       SUPPORTED
       EXPORT
NAME   TYPE
------------------
OTHER  live
```

Example of VM import API:

```
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+json -H
content-type:application/vnd.yang.data+json -X POST
https://209.29.91.165/api/operations/vmImportAction -d
'{"vmImportAction":{"importPath":"intdatastore:/test_export_isrv.vmbkp"}}'
```

Example for VM Operations Payload

This section provides an example of operations payload for starting a VM. You can change the action type value to STOP, REBOOT, ENABLE_MONITOR or DISABLE_MONITOR as required.

```
<vmAction>

    <actionType>START</actionType>
    <vmName>ISR</vmName>

 </vmAction>
```

Table 48: Description for VM Operations Payload

Property	Type	Description	Mandatory/Default Value
vmAction actionType	String	Type of VM action. Value supported: STOP, START, REBOOT, ENABLE_MONITOR, DISABLE_MONITOR	Yes
vmName	String	Name of the VM instance.	Yes

Example: POST Start VM API

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/vmAction --data
'<vmAction><actionType>START</actionType><vmName>
 <vm-instance name></vmName></vmAction>'
* About to connect() to 209.165.201.1 port 80 (#0)
```

```
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> POST /api/operations/vmAction HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 130
>
* upload completely sent off: 130 out of 130 bytes
< HTTP/1.1 204 No Content
< Server:
< Date: Fri, 11 Dec 2015 11:36:33 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Content-Length: 0
< Content-Type: text/html< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Stop VM API

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X
POST
[https://209.165.201.1/api/operations/vmAction|http://209.165.201.1/api/operations/vmAction]
 \--data
'<vmAction><actionType>STOP</actionType><vmName><vm-instance name></vmName></vmAction>'
\* About to connect() to 209.165.201.1 port 80 (#0)
\*Trying 209.165.201.1...
\* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
\* Server auth using Basic with user 'admin'
> POST /api/operations/vmAction HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 129
>
\* upload completely sent off: 129 out of 129 bytes
< HTTP/1.1 204 No Content
< Server:
< Date: Fri, 11 Dec 2015 11:34:36 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
\* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Restart VM API

```
curl -k -v -u "admin:admin" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/vmAction --data
'<vmAction><actionType>REBOOT</actionType><vmName>
 <vm-instance name></vmName></vmAction>'
* About to connect() to 209.165.201.1 port 80 (#0)
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> POST /api/operations/vmAction HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 131
>
* upload completely sent off: 131 out of 131 bytes
< HTTP/1.1 204 No Content
< Server:
< Date: Fri, 11 Dec 2015 11:30:28 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Content-Length: 0
< Content-Type: text/html
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Enable VM Monitoring API

```
curl -k -v -u "admin:password" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://209.165.201.1/api/operations/vmAction --data
'<vmAction><actionType>ENABLE_MONITOR</actionType><vmName><vm-instance
name></vmName></vmAction>'

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/ssl/cert.pem
CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* ALPN, server did not agree to a protocol
```

```
* Server certificate:
* subject: CN=nfvis
* start date: Apr 18 18:54:43 2018 GMT
* expire date: Apr 15 18:54:43 2028 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/operations/vmAction HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.54.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 133
>
* upload completely sent off: 133 out of 133 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 25 Apr 2018 21:57:32 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: POST Disable VM Monitoring API

```
curl -k -v -u "admin:password" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://209.165.201.1/api/operations/vmAction --data
'<vmAction><actionType>DISABLE_MONITOR</actionType><vmName><vm-instance
name></vmName></vmAction>'

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/ssl/cert.pem
CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* ALPN, server did not agree to a protocol
* Server certificate:
* subject: CN=nfvis
* start date: Apr 18 18:54:43 2018 GMT
* expire date: Apr 15 18:54:43 2028 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
```

```
* Server auth using Basic with user 'admin'
> POST /api/operations/vmAction HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.54.0
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 133
>
* upload completely sent off: 133 out of 133 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 25 Apr 2018 21:57:32 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

VM Network APIs

Table 49: VM Network APIs

Action	Method	Payload Required	APIs
To delete an existing subnet	DELETE	No	/api/config/vm_lifecycle/networks/network/int-mgmt-net/subnet/int-mgmt-net-subnet
To create a new subnet	POST	Yes	/api/config/vm_lifecycle/networks/network/int-mgmt-net

Example for VM Networks Payload

This section provides an example of networks payload.

```
<subnet>
 <name>int-mgmt-net-subnet</name>
 <dhcp>false</dhcp>
 <address>105.20.0.0</address>
 <netmask>255.255.255.0</netmask>
 <gateway>105.20.0.1</gateway>
</subnet>
```

Table 50: Description for VM Networks Payload

Property	Type	Description	Mandatory/Default Value
name	String	Management subnet name	Yes - Should be set to int-mgmt-net-subnet
address	String	Subnet address for this network	Yes

netmask	String	Netmask for the network	No
gateway	String	The gateway IP	No

Network File System APIs

Table 51: Network File System APIs

Action	Method	Payload Required	APIs
To mount NFS	POST	Yes	/api/config/system
To unmount NFS	DELETE	No	/api/config/system/storage
To register images on NFS	POST	Yes	/api/config/vm_lifecycle/images
To unregister images on NFS	DELETE	No	/api/config/vm_lifecycle/images/image
To deploy VM on NFS using an image	POST	Yes	/api/config/vm_lifecycle/tenants/tenant/admin/deployments

Example for Network File System Payload

This section provides an example of NFS payload.

```
<image>
<name> myas10</name>
<src>file:///data/mount/nfs_storage/repository/asav961.tar.gz</src>
<properties>
<property>
<name>placement</name>
<value>nfs_storage</value>
</property>
</properties>
</image>
```

```
{"deployment":
    {"name":"15065483181",
     "vm_group":
        {"name":"myasav1",
         "image":"asav961",
         "flavor":"ASAv5",
         "bootup_time":"-1",
         "recovery_wait_time":"0",
         "placement":{"type":"zone_host",
         "host":"nfs_storage"},
         "recovery_policy":{"action_on_recovery":"REBOOT_ONLY"},
         "interfaces":{"interface":[{"nicid":0,"network":"lan-net","model":"virtio"}]},
         "scaling":{"min_active":"1","max_active":"1"}}}}
```

Added in NFVIS 3.12.x release:

```
<image>
        <name>ubuntu</name>
        <src>file:///data/intdatastore/uploads/ubuntu_raw.tar.gz</src>
        <properties>
           <property>
                <name>placement</name>
                <value>iscsi:test</value>
           </property>
        </properties>
    </image>



<tenant>
                        <name>admin</name>
                        <deployments>
                                <deployment>
                                        <name>ubuntu</name>
                                        <vm_group>
                                                <name>ubgrp</name>
                                                <image>ubiscsi</image>
                                                <flavor>ubuntu-small-flav</flavor>
                                                <bootup_time>-1</bootup_time>
                                                <placement>
                                                    <type>zone_host</type>
                                                    <host>iscsi:test</host>
                                                </placement>
                                                <recovery_wait_time>0</recovery_wait_time>
                                                <interfaces>
                                                        <interface>
                                                                <nicid>0</nicid>

<network>int-mgmt-net</network>
                                                        </interface>
                                                </interfaces>
                                        </vm_group>
                                </deployment>
                        </deployments>
                </tenant>
```

VNC Console Start API

Table 52: VNC Console Start API

Action	Method	Payload Required	API
To start a VNC console	POST	No	/api/operations/vncconsole/start

VM Multi Serial Port APIs

Table 53: VM Multi Serial Port API

Action	Method	Payload Required	API
To deploy attaching serial port to VNF	POST	Yes	/api/config/vm_lifecycle/tenants/tenant/admin/deployments

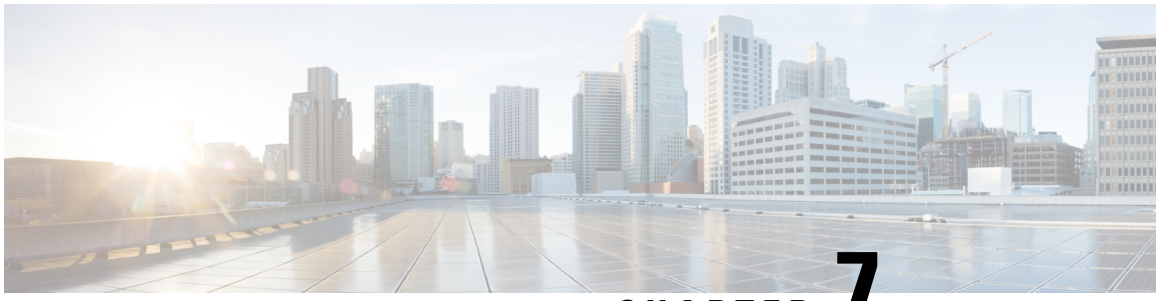
Example for VM Multi Serial Port API

```
{
    "deployment": {
        "name": "15065483181",
        "vm_group": {
            "name": "myasav1",
            "image": "asav961",
            "flavor": "ASAv5",
            "bootup_time": "-1",
            "recovery_wait_time": "0",
            "interfaces": {
                "interface": [{
                    "nicid": 0,
                    "network": "lan-net",
                    "model": "virtio"
                }]
            },
           "serial_ports": {
                "serial_port": [{
                    "serial": 0,
                    "serial_type": "telnet",
                    "service_port": 7000
                }]
            }
```

```
}
```

Table 54: Description for VM Multi Serial Port Payload

Property	Type	Description	Mandatory/Default Value
serial	String	Serial port number	Yes
serial_type	String	Serial type, telnet or console	Yes
service_port	String	Service port number	Yes



CHAPTER 7

System Monitoring APIs

The system monitoring APIs are used to get statistics on the host and VNFs running on the host. These statistics are used by the portal for pictorial representation. These statistics are collected over a specified duration. For large durations, average values are returned. The default duration for all host and VNF queries is set to five minutes. If data is not available for a particular interval during the specified duration, the API returns "na" (not available) for that interval.

Host CPU Stats APIs

Table 55: Host CPU Stats APIs

Action	Method	Payload Required	API
To get the host CPU utilization of a CPU state	GET	No	• /api/operational/system-monitoring/host/cpu/stats • /api/operational/system-monitoring/host/cpu/stats/cpu-usage?deep • /api/operational/system-monitoring/host/cpu/stats /cpu-usage/<duration>,<cpu-state>?deep

Valid duration: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host CPU Stats API

```
curl -k -v -u "admin:admin" -X GET
https://192.0.2.2/api/operational/system-monitoring/host/cpu/stats/cpu-usage/5min,non-idle?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 192.0.2.2...
* Connected to 192.0.2.2 (192.0.2.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 3 05:02:29 2017 GMT
* expire date: Feb 1 05:02:29 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/cpu/stats/cpu-usage/5min,non-idle?deep HTTP/1.1
> Host: 192.0.2.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 07 Feb 2017 03:44:43 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<cpu-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <state>non-idle</state>
 <collect-start-date-time>2017-02-07T03:39:40-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <cpu>
 <id>0</id>
 <usage-percentage>[1.62, 1.16, 1.22, 1.44, 1.41, 1.46, 1.63, 1.82, 3.77, 2.61, 0.94, 1.32,
 1.36, 1.14, 1.34, 1.38, 2.75, 2.33, 1.4, 1.28, 1.2, 1.26, 1.42, 1.44, 1.76, 1.22, 1.0,
1.32, 1.16]</usage-percentage>
 </cpu>
 <cpu>
 <id>1</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
```

```
 <cpu>
 <id>2</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>3</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>4</id>
 <usage-percentage>[29.91, 20.67, 5.82, 0.38, 0.25, 0.1, 0.25, 0.88, 5.72, 7.48, 6.58, 7.37,
 12.95, 17.53, 19.24, 20.78]</usage-percentage>
 </cpu>
 <cpu>
 <id>5</id>
 <usage-percentage>[0.0, 0.06, 0.04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>6</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.07, 0.04, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.13, 0.09, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>7</id>
 <usage-percentage>[2.14, 4.37, 8.71, 5.46, 2.14, 2.22, 2.16, 2.05, 6.19, 4.8, 2.01, 2.2,
2.01, 1.99, 2.37, 2.47, 2.23, 2.23, 2.33, 2.39, 2.49, 2.29, 2.24, 2.14, 2.01, 2.01, 2.33,
2.47, 3.5]</usage-percentage>
 </cpu>
 <cpu>
 <id>8</id>
 <usage-percentage>[1.44, 1.26, 1.54, 1.88, 1.58, 1.36, 3.81, 5.12, 2.87, 1.51, 1.56, 1.72,
 1.68, 1.6, 1.55, 1.38]</usage-percentage>
 </cpu>
 <cpu>
 <id>9</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>10</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>11</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</usage-percentage>

 </cpu>
 <cpu>
 <id>12</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
 <id>13</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]</usage-percentage>
 </cpu>
 <cpu>
```

```
 <id>14</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</usage-percentage>

 </cpu>
 <cpu>
 <id>15</id>
 <usage-percentage>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</usage-percentage>

 </cpu>
</cpu-usage>
```

Table 56: Field Description for Host CPU Stats API Response

duration	Required duration
usage-percentage	It is the percentage of CPU used for the requested state.
state	CPU state The allowed CPU states are: non-idle, interrupt, nice, system, user, and wait.

Host CPU Table API

Table 57: Host CPU Table APIs

Action	Method	Payload Required	API
To get the host CPU utilization statistics table (minimum, maximum, and average) of all CPU states on each of the CPUs	GET	No	• /api/operational/system-monitoring/host/cpu/table • /api/operational/system-monitoring/host/cpu/table?deep • /api/operational/system-monitoring/host/cpu/table/cpu-usage • /api/operational/system-monitoring/host/cpu/table/cpu-usage?deep • /api/operational/system-monitoring/host /cpu/table/cpu-usage/<duration>? deep

Valid duration: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host CPU Table API

```
 curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/system-monitoring/host/cpu/table/cpu-usage/1h?deep'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
```

```
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 3 05:02:29 2017 GMT
* expire date: Feb 1 05:02:29 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/cpu/table/cpu-usage/1h?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 07 Feb 2017 04:10:56 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<


<cpu-usage
 xmlns="http://www.cisco.com/nfvos/system-monitoring"
 xmlns:y="http://tail-f.com/ns/rest"
 xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>1h</duration>
 <cpu>
  <id>0</id>
  <states>
   <state>non-idle</state>
   <min-percentage>0.9</min-percentage>
   <max-percentage>13.56</max-percentage>
   <average-percentage>1.72</average-percentage>
  </states>
  <states>
   <state>interrupt</state>
   <min-percentage>0.0</min-percentage>
   <max-percentage>0.0</max-percentage>
   <average-percentage>0.0</average-percentage>
  </states>
  <states>
   <state>nice</state>
   <min-percentage>0.0</min-percentage>
   <max-percentage>0.06</max-percentage>
   <average-percentage>0.0</average-percentage>
  </states>
  <states>
   <state>softirq</state>
   <min-percentage>0.0</min-percentage>
```

```
  <max-percentage>0.18</max-percentage>
  <average-percentage>0.01</average-percentage>
 </states>
 <states>
  <state>steal</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
 <states>
  <state>system</state>
  <min-percentage>0.36</min-percentage>
  <max-percentage>5.69</max-percentage>
  <average-percentage>0.72</average-percentage>
 </states>
 <states>
  <state>user</state>
  <min-percentage>0.34</min-percentage>
  <max-percentage>5.68</max-percentage>
  <average-percentage>0.65</average-percentage>
 </states>
 <states>
  <state>wait</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>2.64</max-percentage>
  <average-percentage>0.35</average-percentage>
 </states>
</cpu>

...


<cpu>
 <id>15</id>
 <states>
  <state>non-idle</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
 <states>
  <state>interrupt</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
 <states>
  <state>nice</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
 <states>
  <state>softirq</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
 <states>
  <state>steal</state>
  <min-percentage>0.0</min-percentage>
  <max-percentage>0.0</max-percentage>
  <average-percentage>0.0</average-percentage>
 </states>
```

```
  <states>
   <state>system</state>
   <min-percentage>0.0</min-percentage>
   <max-percentage>0.0</max-percentage>
   <average-percentage>0.0</average-percentage>
  </states>
  <states>
   <state>user</state>
   <min-percentage>0.0</min-percentage>
   <max-percentage>0.0</max-percentage>
   <average-percentage>0.0</average-percentage>
  </states>
  <states>
   <state>wait</state>
   <min-percentage>0.0</min-percentage>
   <max-percentage>0.0</max-percentage>
   <average-percentage>0.0</average-percentage>
  </states>
 </cpu>
</cpu-usage>
```

Table 58: Field Description for Host CPU Table API Response

Field	Description
duration	Duration of this collection
cpu states	Indicates the CPU state. This can be non-idle, interrupt, nice, soft interrupt request line (IRQ), steal, system, user and wait.
cpu states min-percentage	Minimum percentage of CPU usage
cpu states max-percentage	Maximum percentage of CPU usage
cpu states average-percentage	Average percentage of CPU usage

Host Disk Stats APIs

Table 59: Host Disk Stats APIs

Action	Method	Payload Required	API

To get arrays of utilizations (per type) for the list of disks or disk partitions on the host server	GET	No	• /api/operational/system-monitoring/host/disk • /api/operational/system-monitoring/host/disk/stats • /api/operational/system-monitoring/host/disk/stats?deep • /api/operational/system-monitoring/host /disk/stats/disk-operations?deep • /api/operational/system-monitoring/host /disk/stats/disk-operations/<duration>?deep • /api/operational/system-monitoring/host /disk/stats/disk-space?deep /api/operational/system-monitoring/host /disk/stats/disk-space/<duration>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host Disk Stats API

Example 1: disk-operations

```
curl -k -v -u "admin:admin" -X GET
"https://209.165.201.2/api/operational/system-monitoring/host/disk/stats/disk-operations/5min?deep
<
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/disk/stats/disk-operations/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
```

```
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 05:56:14 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<disk-operations xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <collect-start-date-time>2017-02-22T05:51:10-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <disk>
 <name>disk-sda</name>
 <io-time-ms>[45.3, 37.5, 117.98, 137.86, 27.6, 30.58, 13.14, 13.22, 25.46, 26.48, 15.62,
27.14, 30.62, 15.68,
 18.68, 78.0, 147.2, 102.5, 44.86, 27.0, 23.66, 15.22, 30.22, 30.16, 15.24,
15.62]</io-time-ms>
 <io-time-weighted-ms>[1986.62, 2263.14, 12979.04, 15663.62, 477.62, 660.3, 17.74, 33.28,
355.26, 484.1,
 27.18, 415.68, 596.6, 27.32, 501.26, 7602.3, 16598.74, 9846.72, 708.42, 359.36, 360.22,
21.82, 407.72, 587.76, 21.3, 21.84]</io-time-weighted-ms>
 <merged-reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</merged-reads-per-sec>
 <merged-writes-per-sec>[5.26, 2.46, 2.72, 2.4, 1.68, 2.28, 1.94, 2.14, 2.28, 1.54, 2.12,
3.5, 4.88, 4.88,
3.36, 3.58, 3.7, 2.58, 3.4, 2.82, 1.96, 1.94, 2.42, 2.1, 3.02]</merged-writes-per-sec>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[92897.28, 112394.24, 356515.84, 375029.76, 29163.52, 33669.12,
14008.32, 18186.24,
31703.04, 31293.44, 15319.04, 36864.0, 49971.2, 28344.32, 49397.76, 224952.32, 414023.68,
268615.68,
58081.28, 33341.44, 27279.36, 14336.0, 35225.6, 40550.4, 22364.16, 23101.44,
40878.08]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[14.66, 19.14, 67.56, 72.26, 5.1, 5.52, 1.12, 1.72, 4.96, 5.4, 1.52, 4.92,
 6.18, 1.5, 7.14, 41.76,
78.92, 49.64, 7.56, 4.54, 4.16, 1.28, 5.24, 6.5]</writes-per-sec>
 <time-per-read-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</time-per-read-ms>
 <time-per-write-ms>[128.4, 109.58, 164.04, 141.78, 66.52, 84.36, 15.76, 17.36, 52.06,
63.96, 17.04, 54.6, 70.24, 18.18, 42.06,
131.5, 210.24, 164.54, 68.04, 57.46, 75.28, 16.88, 48.7, 64.12, 16.4, 16.74,
48.94]</time-per-write-ms>
 <pending-ops>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 20.8, 31.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4]</pending-ops>
 </disk>
 <disk>
 <name>disk-sda1</name>
 <io-time-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</io-time-ms>
 <io-time-weighted-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</io-time-weighted-ms>
```

```
 <merged-reads-per-sec></merged-reads-per-sec>
 <merged-writes-per-sec></merged-writes-per-sec>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</writes-per-sec>
 <time-per-read-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</time-per-read-ms>
 <time-per-write-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</time-per-write-ms>
 <pending-ops></pending-ops>
 </disk>
 <disk>
 <name>disk-sda2</name>
 <io-time-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</io-time-ms>
 <io-time-weighted-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</io-time-weighted-ms>
 <merged-reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</merged-reads-per-sec>
 <merged-writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</merged-writes-per-sec>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</writes-per-sec>
 <time-per-read-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</time-per-read-ms>
 <time-per-write-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</time-per-write-ms>
 <pending-ops></pending-ops>
 </disk>
 <disk>
 <name>disk-sda3</name>
 <io-time-ms>[45.3, 37.5, 117.98, 137.86, 27.6, 30.58, 13.14, 13.22, 25.46, 26.48, 15.62,
27.14, 30.62, 15.68, 18.68,
78.0, 147.2, 102.5, 44.86, 27.0, 23.66, 15.22, 30.22, 30.16, 15.24, 15.62]</io-time-ms>
 <io-time-weighted-ms>[1986.62, 2263.14, 12979.04, 15663.62, 477.62, 660.3, 17.74, 33.28,
355.26, 484.1, 27.18,
415.68, 596.6, 27.32, 501.26, 7602.3, 16598.74, 9846.72, 708.42, 359.36, 360.22, 21.82,
407.72, 587.76, 21.3, 21.84]</io-time-weighted-ms>
 <merged-reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</merged-reads-per-sec>
 <merged-writes-per-sec>[5.26, 2.46, 2.72, 2.4, 1.68, 2.28, 1.94, 2.14, 2.28, 1.54, 2.12,
3.5, 4.88, 4.88, 3.36, 3.58,
3.7, 2.58, 3.4, 2.82, 1.96, 1.94, 2.42, 2.1, 3.02, 3.74, 4.58]</merged-writes-per-sec>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[92897.28, 112394.24, 356515.84, 375029.76, 29163.52, 33669.12,
14008.32, 18186.24, 31703.04,
31293.44, 15319.04, 36864.0, 49971.2, 28344.32, 49397.76, 224952.32, 414023.68, 268615.68,
 58081.28, 33341.44, 27279.36,
14336.0, 35225.6, 40550.4, 22364.16, 23101.44]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[14.66, 19.14, 67.56, 72.26, 5.1, 5.52, 1.12, 1.72, 4.96, 5.4, 1.52, 4.92,
 6.18, 1.5, 7.14, 41.76,
78.92, 49.64, 7.56, 4.54, 4.16, 1.28, 5.24, 6.5, 1.3, 1.32]</writes-per-sec>
 <time-per-read-ms>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</time-per-read-ms>
 <time-per-write-ms>[128.4, 109.58, 164.04, 141.78, 66.52, 84.36, 15.76, 17.36, 52.06,
63.96, 17.04, 54.6, 70.24, 18.18,
42.06, 131.5, 210.24, 164.54, 68.04, 57.46, 75.28, 16.88, 48.7, 64.12, 16.4,
16.74]</time-per-write-ms>
 <pending-ops>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 20.8,
31.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4]</pending-ops>
 </disk>
</disk-operations>
```

Table 60: Field Description for Host Disk Stats API Response

Field	Description
io-time-ms	Time spent doing I/Os 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.
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.
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 took to complete

time-per-write-ms	The average time a write operation took to complete
time-per-write-ms	The average time a write operation took to complete
pending-ops	The queue size of pending I/O operations

Example 2: disk-space

```
curl -k -v -u admin:Cisco123# -X GET
https://209.165.201.2/api/operational/system-monitoring/host/disk/stats/disk-space/5min?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/disk/stats/disk-space/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 05:59:38 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<disk-space xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <collect-start-date-time>2017-02-22T05:54:30-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <mount-point>
 <name>/boot</name>
 <free-GB>[0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33,
0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 0.33]</free-GB>
 <used-GB>[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
 0.1, 0.1, 0.1, 0.1]</used-GB>
```

```
 <reserved-GB>[0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03,
 0.03, 0.03, 0.03, 0.03, 0.03, 0.03]</reserved-GB>
 </mount-point>
 <mount-point>
 <name>/data</name>
 <free-GB>[849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26,
 849.26,
849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26, 849.26]</free-GB>
 <used-GB>[3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09, 3.09,
3.09, 3.09, 3.09, 3.09, 3.09, 3.09]</used-GB>
 <reserved-GB>[45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64,
 45.64,
45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64, 45.64]</reserved-GB>
 </mount-point>
 <mount-point>
 <name>/home</name>
 <free-GB>[0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43,
0.43, 0.43, 0.43, 0.43, 0.43, 0.43]</free-GB>
 <used-GB>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0]</used-GB>
 <reserved-GB>[0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03, 0.03,
 0.03, 0.03, 0.03, 0.03, 0.03, 0.03]</reserved-GB>
 </mount-point>
 <mount-point>
 <name>/</name>
 <free-GB>[5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69, 5.69,
5.69, 5.69, 5.69, 5.69, 5.69, 5.69]</free-GB>
 <used-GB>[1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65, 1.65,
1.65, 1.65, 1.65, 1.65, 1.65, 1.65]</used-GB>
 <reserved-GB>[0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42, 0.42,
 0.42, 0.42, 0.42, 0.42, 0.42, 0.42]</reserved-GB>
 </mount-point>
 <mount-point>
 <name>/var</name>
 <free-GB>[1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55, 1.55,
1.55, 1.55, 1.55, 1.55, 1.55, 1.55]</free-GB>
 <used-GB>[0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24,
0.24, 0.24, 0.24, 0.24, 0.24, 0.24]</used-GB>
 <reserved-GB>[0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12, 0.12,
 0.12, 0.12, 0.12, 0.12, 0.12, 0.12]</reserved-GB>
 </mount-point>
</disk-space>
```

This API response provides information about the disk name and data for various disk usage types.

Table 61: Field Description for Host Disk Stats API Response

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

Host Memory Stats APIs

Table 62: Host Memory Statistics APIs

Action	Method	Payload Required	API
To get the host memory utilization	GET	No	• /api/operational/system-monitoring/host/memory • /api/operational/system-monitoring/host/memory?deep • /api/operational/system-monitoring/host/memory/stats/mem-usage • /api/operational/system-monitoring/host/memory/stats/mem-usage?deep • /api/operational/system-monitoring/host /memory/stats/mem-usage/<duration>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host Memory Stats API

```
curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/system-monitoring/host/memory/stats/mem-usage/5min?deep'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 3 05:02:29 2017 GMT
* expire date: Feb 1 05:02:29 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/memory/stats/mem-usage/5min?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
```

```
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 07 Feb 2017 04:24:45 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<mem-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <collect-start-date-time>2017-02-07T04:19:40-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <buffered-MB>[16.2, 16.21, 16.22, 16.24, 16.26, 16.27, 16.28, 16.3, 16.32, 16.33, 16.35,
16.36, 16.38, 16.39, 16.4, 16.41, 16.44, 16.46, 16.47, 16.48, 16.5, 16.52, 16.53, 16.54,
16.56, 16.57, 16.58, 16.6, 16.61, 16.62]</buffered-MB>
 <cached-MB>[3730.54, 3730.55, 3730.55, 3730.56, 3730.56, 3730.57, 3730.58, 3730.58, 3730.58,
 3730.59, 3730.59, 3730.6, 3730.6, 3730.61, 3730.62, 3730.62, 3730.62, 3730.63, 3730.63,
3730.64, 3730.66, 3730.81, 3730.92, 3730.94, 3731.07, 3731.18, 3731.24, 3731.3, 3731.36,
3731.38]</cached-MB>
 <free-MB>[54090.05, 54089.9, 54089.84, 54089.93, 54089.81, 54089.7, 54089.67, 54089.67,
54089.7, 54089.62, 54089.66, 54089.72, 54089.63, 54089.51, 54089.44, 54089.36, 54089.46,
54089.57, 54089.14, 54088.85, 54088.3, 54087.94, 54088.17, 54076.76, 54080.71, 54088.02,
54087.82, 54087.59, 54087.54, 54087.69]</free-MB>
 <used-MB>[6086.81, 6086.9, 6086.98, 6086.8, 6086.76, 6086.8, 6086.78, 6086.85, 6086.86,
6086.83, 6086.67, 6086.55, 6086.68, 6086.83, 6086.86, 6086.84, 6086.75, 6086.67, 6087.09,
6087.36, 6087.83, 6088.08, 6087.79, 6099.19, 6095.12, 6087.67, 6087.74, 6087.86, 6087.84,
6087.66]</used-MB>
 <slab-recl-MB>[186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79,
186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79,
186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79, 186.79]</slab-recl-MB>
 <slab-unrecl-MB>[52.04, 52.08, 52.05, 52.11, 52.24, 52.3, 52.33, 52.24, 52.17, 52.26,
52.37, 52.41, 52.35, 52.3, 52.32, 52.4, 52.37, 52.3, 52.31, 52.31, 52.35, 52.28, 52.22,
52.2, 52.18, 52.2, 52.25, 52.29, 52.29]</slab-unrecl-MB>
</mem-usage>
```

This API response provides usage information for the following memory types:

- Buffered
- Cached
- Free
- Used
- Slab recl
- Slab unrecl

Host Memory Table APIs

Table 63: Host Memory Table APIs

Action	Method	Payload Required	API

To get the host memory utilization in tabular format (minimum, maximum, and average) for each memory type	GET	No	• /api/operational/system-monitoring/host/memory/table • /api/operational/system-monitoring/host/memory/table?deep • /api/operational/system-monitoring/host/memory/table/mem-usage • /api/operational/system-monitoring/host/memory/table/mem-usage?deep • /api/operational/system-monitoring/host/memory/table /mem-usage/<duration>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

.

Example: GET Host Memory Table APIs

```
curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/system-monitoring/host/memory/table/mem-usage/1h?deep'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 3 05:02:29 2017 GMT
* expire date: Feb 1 05:02:29 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/memory/table/mem-usage/1h?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Tue, 07 Feb 2017 04:27:22 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
```

```
< Pragma: no-cache
<


<mem-usage
 xmlns="http://www.cisco.com/nfvos/system-monitoring"
 xmlns:y="http://tail-f.com/ns/rest"
 xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>1h</duration>
 <memory>
  <type>buffered-MB</type>
  <min>11.41</min>
  <max>16.83</max>
  <average>14.13</average>
 </memory>
 <memory>
  <type>cached-MB</type>
  <min>3711.17</min>
  <max>3731.85</max>
  <average>3719.28</average>
 </memory>
 <memory>
  <type>free-MB</type>
  <min>54076.76</min>
  <max>54166.76</max>
  <average>54127.47</average>
 </memory>
 <memory>
  <type>slab-recl-MB</type>
  <min>186.78</min>
  <max>186.79</max>
  <average>186.79</average>
 </memory>
 <memory>
  <type>slab-unrecl-MB</type>
  <min>52.03</min>
  <max>52.84</max>
  <average>52.26</average>
 </memory>
 <memory>
  <type>used-MB</type>
  <min>6032.55</min>
  <max>6099.19</max>
  <average>6062.51</average>
 </memory>
</mem-usage>
```

Host Port Stats APIs

Table 64: Host Port Stats APIs

Action	Method	Payload Required	API

To get the packet counts information (error-rx, error-tx, error-total, packets-rx, packets-tx, and packets-total) on all host interfaces	GET	No	• /api/operational/system-monitoring/host/port • /api/operational/system-monitoring/host/port/stats • /api/operational/system-monitoring/host/port/stats?deep • /api/operational/system-monitoring/host/port /stats/port-usage/<duration>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host Port Stats API

```
curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/system-monitoring/host/port/stats/port-usage/5min?deep'
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/port/stats/port-usage/5min?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 05:43:42 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<port-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <collect-start-date-time>2017-02-22T05:38:40-00:00</collect-start-date-time>
```

```
 <collect-interval-seconds>10</collect-interval-seconds>
 <port>
 <name>eth0</name>
 <total-packets-per-sec>[38.8, 24.38, 34.9, 37.94, 21.64, 20.84, 31.72, 36.22, 22.44, 28.16,
 31.04, 33.24, 20.56, 21.02, 20.72, 22.64, 21.98, 27.14]</total-packets-per-sec>
 <rx-packets-per-sec>[36.66, 22.02, 32.72, 35.4, 19.88, 18.92, 29.26, 34.4, 18.64, 23.0,
28.88, 30.02, 17.56, 19.12, 18.46, 20.46, 19.74, 25.24]</rx-packets-per-sec>
 <tx-packets-per-sec>[2.14, 2.36, 2.18, 2.54, 1.76, 1.92, 2.46, 1.82, 3.8, 5.16, 2.16, 3.22,
 3.0, 1.9, 2.26, 2.18, 2.24, 1.9]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <name>eth1</name>
 <total-packets-per-sec>[34.58, 19.66, 30.5, 32.92, 18.3, 17.08, 26.88, 32.52, 14.2, 16.88,
 26.68, 26.7, 14.46, 17.12, 16.26, 18.42, 17.44, 23.46, 26.9]</total-packets-per-sec>
 <rx-packets-per-sec>[34.44, 19.54, 30.46, 32.78, 18.18, 17.0, 26.72, 32.46, 14.12, 16.72,
 26.62, 26.62, 14.3, 17.06, 16.18, 18.26, 17.38, 23.34, 26.72]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.14, 0.12, 0.04, 0.14, 0.12, 0.08, 0.16, 0.06, 0.08, 0.16, 0.06,
0.08, 0.16, 0.06, 0.08, 0.16, 0.06, 0.12, 0.18]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <name>eth2</name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <name>eth3</name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <name>eth4</name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <name>eth5</name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
</port-usage>
```


Table 65: Field Description for Host Port Statistics API Response

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

Host Port Table APIs

Table 66: Host Port Table APIs

Action	Method	Payload Required	API

To get statistics information about all ports	GET	No	• /api/operational/system-monitoring/host/port • /api/operational/system-monitoring/host/port/table • /api/operational/system-monitoring/host/port/table?deep • /api/operational/system-monitoring/host/port /table/port-usage/<duration>,<name>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET Host Port Table API

```
curl -k -v -u admin:Cisco123# -X GET
'https://172.19.162.209/api/operational/system-monitoring/host/port/table?deep'
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/host/port/table?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 05:50:53 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<table xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <port-usage>
```

```
<duration>5min</duration>
<name>eth0</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
<status>up</status>
<ip-address>NA</ip-address>
<rx-packets>9117</rx-packets>
<tx-packets>620</tx-packets>
<rx-packets-per-sec>31.44</rx-packets-per-sec>
<tx-packets-per-sec>2.14</tx-packets-per-sec>
</port-usage>
<port-usage>
<duration>5min</duration>
<name>eth1</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
<status>up</status>
<ip-address>NA</ip-address>
<rx-packets>8491</rx-packets>
<tx-packets>17</tx-packets>
<rx-packets-per-sec>29.28</rx-packets-per-sec>
<tx-packets-per-sec>0.06</tx-packets-per-sec>
</port-usage>
<port-usage>
<duration>5min</duration>
<name>eth2</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
<status>down</status>
<ip-address>NA</ip-address>
<rx-packets>0</rx-packets>
<tx-packets>0</tx-packets>
<rx-packets-per-sec>0.0</rx-packets-per-sec>
<tx-packets-per-sec>0.0</tx-packets-per-sec>
</port-usage>
<port-usage>
<duration>5min</duration>
<name>eth3</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
<status>down</status>
<ip-address>NA</ip-address>
<rx-packets>0</rx-packets>
<tx-packets>0</tx-packets>
<rx-packets-per-sec>0.0</rx-packets-per-sec>
<tx-packets-per-sec>0.0</tx-packets-per-sec>
</port-usage>
<port-usage>
<duration>5min</duration>
<name>eth4</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
<status>down</status>
<ip-address>NA</ip-address>
<rx-packets>0</rx-packets>
<tx-packets>0</tx-packets>
<rx-packets-per-sec>0.0</rx-packets-per-sec>
<tx-packets-per-sec>0.0</tx-packets-per-sec>
</port-usage>
<port-usage>
<duration>5min</duration>
<name>eth5</name>
<collect-start-date-time>2017-02-22T05:45:50-00:00</collect-start-date-time>
<collect-interval-seconds>10</collect-interval-seconds>
```

```
 <status>down</status>
 <ip-address>NA</ip-address>
 <rx-packets>0</rx-packets>
 <tx-packets>0</tx-packets>
 <rx-packets-per-sec>0.0</rx-packets-per-sec>
 <tx-packets-per-sec>0.0</tx-packets-per-sec>
 </port-usage>
</table>
```

Table 67: Field Description for Host Port Table API Response

Field	Description
Name	Name of the host interface or port
collect-start-date-time	The actual start date and time of this collection
duration	The duration of this collection
Status	Port status
IP_Address	IP address of this interface
collect-interval-seconds	Time interval of the collection
rx-packets	Received packets
tx-packets	Transmitted packets
rx-packets-per-sec	Received packet rate (packets/second)
tx-packets-per-sec	Transmitted packet rate (packets/second)

VNF CPU Stats APIs

Table 68: VNF CPU Stats APIs

Action	Method	Payload Required	API

To get CPU statistics information of VMs	GET	No	• /api/operational/system-monitoring/vnf/vcpu • /api/operational/system-monitoring/vnf/vcpu/stats • /api/operational/system-monitoring/vnf/vcpu/stats?deep • /api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage • /api/operational/system-monitoring/vnf/vcpu /stats/vcpu-usage?deep • /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

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET VNF CPU Stats API

This example is for all VNFs.

```
curl -k -v -u admin:Cisco123# -X GET
https://209.165.201.2/api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/5min?deep

Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Mar 8 19:19:56 2017 GMT
* expire date: Mar 6 19:19:56 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/vnf/vcpu/stats/vcpu-usage/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
```

```
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Thu, 09 Mar 2017 20:37:13 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<vcpu-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <vnf>
 <name>1489003560.ROUTER</name>
 <collect-start-date-time>2017-03-09T20:32:10-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <total-percentage>[15.62, 16.25, 16.28, 15.35, 15.3, 15.28, 15.27, 15.24, 15.12, 15.06,
15.05, 15.05, 15.29,
15.37, 14.81, 14.77, 14.75, 14.7, 14.59, 14.54, 14.51, 14.42, 14.39, 14.39, 14.34, 14.22,
14.15, 14.2, 14.17]</total-percentage>
 <vcpu>
 <id>0</id>
 <vcpu-percentage>[7.06, 9.82, 10.22, 6.5, 6.5, 6.54, 6.6, 6.48, 6.3, 6.3, 6.38, 6.46, 7.76,
 8.44, 6.4, 6.4, 6.44, 6.5,
6.42, 6.38, 6.46, 6.48, 6.6, 6.64, 6.66, 6.44, 6.36, 6.52]</vcpu-percentage>
 </vcpu>
 <vcpu>
 <id>1</id>
 <vcpu-percentage>[14.02, 13.98, 14.14, 14.16, 14.1, 14.1, 14.06, 14.04, 14.1, 14.1, 14.1,
 14.1, 14.1, 14.06, 14.0,
14.04, 14.1, 14.06, 14.0, 14.04, 14.06, 13.96, 13.94, 13.96, 13.98, 14.02, 13.94,
13.96]</vcpu-percentage>
 </vcpu>
 <vcpu>
 <id>2</id>
 <vcpu-percentage>[10.6, 10.68, 10.72, 10.6, 10.6, 10.64, 10.66, 10.6, 10.6, 10.64, 10.7,
10.66, 10.64, 10.7, 10.7, 10.7,
10.7, 10.7, 10.7, 10.7, 10.74, 10.76, 10.74, 10.76, 10.7, 10.74, 10.8,
10.76]</vcpu-percentage>
 </vcpu>
 <vcpu>
 <id>3</id>
 <vcpu-percentage>[30.78, 30.36, 30.12, 30.14, 29.9, 29.82, 29.74, 29.76, 29.54, 29.18,
28.96, 28.9, 28.7, 28.32, 28.08, 27.9,
27.82, 27.46, 27.06, 26.96, 26.78, 26.56, 26.38, 26.12, 25.92, 26.2, 26.52,
25.1]</vcpu-percentage>
 </vcpu>
 </vnf>
 <vnf>
 <name>1489002218.OTHER</name>
 <collect-start-date-time>2017-03-09T20:32:10-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <total-percentage>[0.36, 0.3, 0.3, 0.18, 0.16, 0.32, 0.2, 0.2, 0.2, 0.2, 0.16, 0.18, 0.26,
 0.2, 0.24, 0.22, 0.18, 0.3, 0.26, 0.2,
0.2, 0.2, 0.24, 0.3, 0.26, 0.24, 0.26, 0.2]</total-percentage>
 <vcpu>
 <id>0</id>
 <vcpu-percentage>[0.36, 0.26, 0.24, 0.18, 0.16, 0.32, 0.2, 0.2, 0.2, 0.2, 0.16, 0.18, 0.22,
 0.18, 0.3, 0.22, 0.14, 0.24,
0.26, 0.2, 0.2, 0.2, 0.24, 0.26, 0.2, 0.28, 0.28, 0.14, 0.2]</vcpu-percentage>
 </vcpu>
 </vnf>
```

```
</vcpu-usage>
```

VNF Disk Stats APIs

Table 69: VNF Disk Stats APIs

Action	Method	Payload Required	API
To get the VNF disk statistics	GET	No	• /api/operational/system-monitoring/vnf/disk • /api/operational/system-monitoring/vnf/disk/stats • /api/operational/system-monitoring/vnf/disk/stats?deep • /api/operational/system-monitoring/vnf/disk /stats/disk-operations • /api/operational/system-monitoring/vnf/disk /stats/disk-operations?deep • /api/operational/system-monitoring/vnf /disk/stats/disk-operations/<duration>?deep • /api/operational/system-monitoring/vnf /disk/stats/disk-operations/<duration>/vnf?deep • /api/operational/system-monitoring/vnf /disk/stats/disk-operations/<duration>/vnf/<vnf-name>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET VNF Disk Stats API

This example is for all VMs.

```
curl -k -v -u admin:Cisco123# -X GET
https://209.165.201.2/api/operational/system-monitoring/vnf/disk/stats/disk-operations/5min?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
```

```
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/vnf/disk/stats/disk-operations/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 06:17:48 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<disk-operations xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest" xmlns:system_monitoring="http://www.cisco.com
/nfvos/system-monitoring">
 <duration>5min</duration>
 <vnf>
 <name>1487397034.OTHER</name>
 <collect-start-date-time>2017-02-22T06:12:40-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <disk>
 <disk-name>vda</disk-name>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</writes-per-sec>
 </disk>
 </vnf>
 <vnf>
 <name>1487399314.ROUTER2</name>
 <collect-start-date-time>2017-02-22T06:12:40-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <disk>
 <disk-name>hdd</disk-name>
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</writes-per-sec>
 </disk>
 <disk>
 <disk-name>vda</disk-name>
```

```
 <bytes-read-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0]</bytes-read-per-sec>
 <bytes-written-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3276.8, 4915.2, 0.0, 655.36, 983.04,
 0.0, 0.0, 0.0, 0.0, 0.0, 122.88]</bytes-written-per-sec>
 <reads-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0]</reads-per-sec>
 <writes-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.36, 0.54, 0.0, 0.12, 0.18, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.12, 0.18]</writes-per-sec>
 </disk>
 </vnf>
</disk-operations>
```

VNF Memory Stats API

Table 70: VNF Memory Stats APIs

Action	Method	Payload Required	API
To get the memory statistics of VMs	GET	No	• /api/operational/system-monitoring/vnf/memory • /api/operational/system-monitoring/vnf/memory/stats • /api/operational/system-monitoring/vnf/memory/stats?deep • /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

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET VNF Memory Stats API

This example is for all VMs.

```
curl -k -v -u "admin:admin" -X GET
https://209.165.201.2/api/operational/system-monitoring/vnf/memory/stats/mem-usage/5min?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
```

```
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/vnf/memory/stats/mem-usage/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 06:35:09 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<mem-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <vnf>
 <name>1487397034.OTHER</name>
 <collect-start-date-time>2017-02-22T06:30:00-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <total-MB>[256.0, 256.0, 256.0, 256.0, 256.0, 256.0, 256.0, 256.0, 256.0, 256.0, 256.0,
256.0, 256.0,
256.0, 256.0, 256.0, 256.0, 256.0, 256.0]</total-MB>
 <rss-MB>[116.29, 116.29, 116.29, 116.29, 116.29, 116.29, 116.29, 116.29, 116.29, 116.29,
116.29, 116.29,
116.29, 116.29, 116.29, 116.29, 116.29, 116.29, 116.29]</rss-MB>
 </vnf>
 <vnf>
 <name>1487399314.ROUTER2</name>
 <collect-start-date-time>2017-02-22T06:30:00-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <total-MB>[4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0,
 4096.0, 4096.0,
4096.0, 4096.0, 4096.0, 4096.0, 4096.0, 4096.0]</total-MB>
 <rss-MB>[4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93,
 4179.93, 4179.93,
4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93, 4179.93]</rss-MB>
 </vnf>
</mem-usage>
```

Table 71: Field Description for VNF Memory Stats API Response

Field	Description
total-MB	Total memory of the VNF in MB
rss-MB	Resident Set Size of the VNF in MB

VNF Port Stats APIs

Table 72: VNF Port Stats APIs

Action	Method	Payload Required	API
To get the VNF port statistics	GET	No	• /api/operational/system-monitoring/vnf/port • /api/operational/system-monitoring/vnf/port/stats • /api/operational/system-monitoring/vnf/port/stats?deep • /api/operational/system-monitoring/vnf/port/stats/port-usage • /api/operational/system-monitoring/vnf/port/stats/port-usage?deep • /api/operational/system-monitoring /vnf/port/stats/port-usage/<duration>?deep • /api/operational/system-monitoring /vnf/port/stats/port-usage/<duration>/vnf?deep • /api/operational/system-monitoring/vnf/port /stats/port-usage/<duration>/vnf/<vnf-name>?deep

The valid duration can be: 1min, 5min, 15min, 30min, 1h, 1H, 6h, 6H, 1d, 1D, 5d, 5D, 30d, and 30D

Example: GET VNF Port Stats API

This example is for all VMs.

```
curl -k -v -u admin:Cisco123# -X GET https://209.165.201.2/api/operational/system-monitoring
/vnf/port/stats/port-usage/5min?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 209.165.201.2...
* Connected to 209.165.201.2 (209.165.201.2) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
```

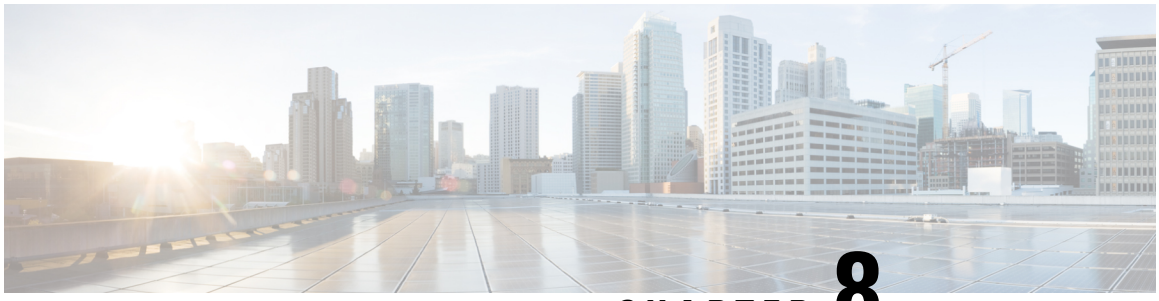
```
* start date: Feb 18 12:04:07 2017 GMT
* expire date: Feb 16 12:04:07 2027 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system-monitoring/vnf/port/stats/port-usage/5min?deep HTTP/1.1
> Host: 209.165.201.2
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 06:14:09 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<port-usage xmlns="http://www.cisco.com/nfvos/system-monitoring"
xmlns:y="http://tail-f.com/ns/rest"
xmlns:system_monitoring="http://www.cisco.com/nfvos/system-monitoring">
 <duration>5min</duration>
 <vnf>
 <name>1487397034.OTHER</name>
 <collect-start-date-time>2017-02-22T06:09:00-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <port>
 <port-name>vnic0</port-name>
 <total-packets-per-sec>[23.04, 16.26, 14.38, 13.38, 14.98, 14.5, 14.34, 14.46, 15.44,
14.86, 22.08, 25.78, 12.74,
15.02, 14.1, 15.84, 18.54, 16.56, 15.36, 18.64, 19.32, 24.2, 30.32, 19.06,
15.68]</total-packets-per-sec>
 <rx-packets-per-sec>[22.96, 16.1, 14.32, 13.3, 14.82, 14.44, 14.22, 14.28, 15.44, 14.74,
21.9, 25.78, 12.62,
14.84, 14.1, 15.72, 18.36, 16.52, 15.22, 18.52, 19.28, 24.06, 30.2, 19.02,
15.54]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.08, 0.16, 0.06, 0.08, 0.16, 0.06, 0.12, 0.18, 0.0, 0.12, 0.18, 0.0,
 0.12, 0.18, 0.0, 0.12,
0.18, 0.04, 0.14, 0.12, 0.04, 0.14, 0.12, 0.04, 0.14]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <port-name>vnic1</port-name>
 <total-packets-per-sec>[23.04, 16.26, 14.38, 13.38, 14.98, 14.5, 14.34, 14.46, 15.44,
14.86, 22.08, 25.78,
12.74, 15.02, 14.1, 15.84, 18.54, 16.56, 15.36, 18.64, 19.32, 24.2, 30.32, 19.06,
15.68]</total-packets-per-sec>
 <rx-packets-per-sec>[22.96, 16.1, 14.32, 13.3, 14.82, 14.44, 14.22, 14.28, 15.44, 14.74,
21.9, 25.78, 12.62, 14.84,
14.1, 15.72, 18.36, 16.52, 15.22, 18.52, 19.28, 24.06, 30.2, 19.02,
15.54]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.08, 0.16, 0.06, 0.08, 0.16, 0.06, 0.12, 0.18, 0.0, 0.12, 0.18, 0.0,
 0.12, 0.18, 0.0, 0.12,
0.18, 0.04, 0.14, 0.12, 0.04, 0.14, 0.12, 0.04, 0.14]</tx-packets-per-sec>
```

```
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 </vnf>
 <vnf>
 <name>1487399314.ROUTER2</name>
 <collect-start-date-time>2017-02-22T06:09:00-00:00</collect-start-date-time>
 <collect-interval-seconds>10</collect-interval-seconds>
 <port>
 <port-name>vnic2</port-name>
 <total-packets-per-sec>[0.68, 0.72, 0.68, 0.8, 0.72, 0.68, 0.8, 0.72, 0.6, 0.76, 0.84,
0.6, 0.76, 0.84, 0.6, 0.68, 0.8,
0.72, 0.68, 0.72, 0.68, 0.8, 0.72, 0.68, 0.8]</total-packets-per-sec>
 <rx-packets-per-sec>[0.34, 0.36, 0.34, 0.4, 0.36, 0.34, 0.4, 0.36, 0.3, 0.38, 0.42, 0.3,
0.38, 0.42, 0.3, 0.34, 0.4, 0.36,
0.34, 0.36, 0.34, 0.4, 0.36, 0.34, 0.4]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.34, 0.36, 0.34, 0.4, 0.36, 0.34, 0.4, 0.36, 0.3, 0.38, 0.42, 0.3,
0.38, 0.42, 0.3, 0.34, 0.4, 0.36,
0.34, 0.36, 0.34, 0.4, 0.36, 0.34, 0.4]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <port-name>vnic4</port-name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 <port>
 <port-name>vnic3</port-name>
 <total-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</total-packets-per-sec>
 <rx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
0.0, 0.0, 0.0, 0.0]</rx-packets-per-sec>
 <tx-packets-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0]</tx-packets-per-sec>
 <total-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</total-errors-per-sec>
 <rx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</rx-errors-per-sec>
 <tx-errors-per-sec>[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]</tx-errors-per-sec>
 </port>
 </vnf>
</port-usage>
```

Table 73: Field Description for VNF Port Stats API Response

Field	Description
total-packets-per-sec	Total packets received and sent per second
rx-packets-per-sec	Packets received per second
tx-packets-per-sec	Packets sent 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 sending packets



CHAPTER 8

System Operations APIs

External Disks API

Table 74: External Disks API

Action	Method	Payload Required	API
To get a list of external disks	GET	No	/api/operational/system/ext-disks

Example: GET External Disks API

```
curl -X GET -v -k -u admin:admin https://1.2.3.4/api/operational/system/ext-disks
    *   Trying 172.19.147.237...
    * Connected to 172.19.147.237 (172.19.147.237) port 443 (#0)
    * TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    * Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
    * Server auth using Basic with user 'admin'
    > GET /api/operational/system/ext-disks HTTP/1.1
    > Host: 172.19.147.237
    > Authorization: Basic YWRtaW46YWRtaW4=
    > User-Agent: curl/7.43.0
    > Accept: */*
    >
    < HTTP/1.1 204 No Conten
    < Server: nginx/1.6.3
    < Date: Fri, 26 Aug 2016 23:03:50 GMT
    < Content-Type: application/vnd.yang.collection+xml
    < Content-Length: 0
    < Connection: keep-alive
```

```
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<name>extdatastore1</data>
```

File List APIs

Use the File List APIs to get information about all files under the "/mnt-usb" (USB) and "/data/upload1" (local) folders.

Table 75: File List APIs

Action	Method	Payload Required	API
To get a list of VM images available for registration on the USB	GET	No	/api/operational/system/file-list/disk/usb
To get a list of VM images available for registration on the local system	GET	No	api/operational/system/file-list/disk/local

Example: GET File List APIs

```
curl -k -v -u "admin:Cisco#123" -H "Accept:application/vnd.yang.collection+json" -H
"Content-Type:application/vnd.yang.collection+json" -X
GET https://209.165.201.1/api/operational/system/file-list/disk/usb
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/system/file-list/disk/usb HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28jMTIz
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.collection+json
> Content-Type:application/vnd.yang.collection+json
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 12:12:11 GMT
< Content-Type: application/vnd.yang.collection+json
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
{
  "collection": {
    "system:usb": [
      {
        "name": "win2k.tar.gz",
        "path": "/mnt-usb/usb1/Win2k",
```

```
  "size": "5.1G",
  "type": "VM Package",
  "date-modified": "2016-04-06 12:07:52"
},
{
  "name": "CentOS-7-x86_64-Everything-1511.tar.gz",
  "path": "/mnt-usb/usb1/OtherLin",
  "size": "439M",
  "type": "VM Package",
  "date-modified": "2016-01-19 12:47:38"
},
{
  "name": "ubuntu-14.04.3-server-amd64-disk1.tar.gz",
  "path": "/mnt-usb/usb1/OtherLin",
  "size": "527M",
  "type": "VM Package",
  "date-modified": "2016-01-19 12:46:30"
},
{
  "name": "Cisco_NFVIS-3.4.0-454-20160927_022810.iso",
  "path": "/mnt-usb/usb1",
  "size": "1.8G",
  "type": "Other",
  "date-modified": "2016-09-27 02:06:48"
},
{
  "name": "asav961.tar.gz",
  "path": "/mnt-usb/usb1",
  "size": "164M",
  "type": "VM Package",
  "date-modified": "2016-10-07 14:20:52"
},
{
  "name": "Cisco-KVM-vWAAS-2500-6.2.1-b-11.tar.gz",
  "path": "/mnt-usb/usb1",
  "size": "919M",
  "type": "VM Package",
  "date-modified": "2016-10-07 14:19:24"
},
{
  "name": "TinyLinux.tar.gz",
  "path": "/mnt-usb/usb1",
  "size": "17M",
  "type": "VM Package",
  "date-modified": "2016-01-19 11:23:14"
},
{
  "name": "Cisco-KVM-vWAAS-2500-6.3.0-b98.tar.gz",
  "path": "/mnt-usb/usb1",
  "size": "979M",
  "type": "VM Package",
  "date-modified": "2016-12-05 10:29:52"
},
{
  "name": "IndexerVolumeGuid",
  "path": "/mnt-usb/usb1/System Volume Information",
  "size": "76",
  "type": "Other",
  "date-modified": "2017-02-06 11:05:38"
},
{
  "name": "isrv-universalk9.16.03.01.tar.gz",
  "path": "/mnt-usb/usb2",
  "size": "1.1G",
```

```
        "type": "VM Package",
        "date-modified": "2016-08-18 10:45:04"
      }
    ]
  }
}
```

```
curl -k -v -u "admin:Cisco#123" -H "Accept:application/vnd.yang.collection+json" -H
"Content-Type:application/vnd.yang.collection+json" -X
GET https://209.165.201.1/api/operational/system/file-list/disk/local
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/system/file-list/disk/local HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28jMTIz
> User-Agent: curl/7.43.0
> Accept:application/vnd.yang.collection+json
> Content-Type:application/vnd.yang.collection+json
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 22 Feb 2017 12:32:17 GMT
< Content-Type: application/vnd.yang.collection+json
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
{
  "collection": {
    "system:local": [
      {
        "name": "IndexerVolumeGuid",
        "path": "/data/upload1",
        "size": "76",
        "type": "Other",
        "date-modified": "2017-02-22 12:31:38"
      },
      {
        "name": "Cisco_NFVIS-3.4.0-454-20160927_022810.iso",
        "path": "/data/upload1",
        "size": "1.8G",
        "type": "Other",
        "date-modified": "2017-02-22 12:31:47"
      }
    ]
  }
}
```

File Delete API

Table 76: File Delete API

Action	Method	Payload Required	API

Delete one or more files from the host server (/data/upload1/)	POST	Yes	/api/operations/system/file-delete/file

Example for File Delete Payload

```
<input><name><xyz.txt></name></input>
```

Table 77: File Delete Payload Description

Property	Type	Description	Mandatory/Default Value
name	string	Name of the file that you want to delete.	Yes

Example: File Delete API

```
 curl -k -v -u "admin:admin" -H content-type:application/vnd.yang.data+json -X
 POST https://209.165.201.1/api/operations/system/file-delete/file -d
"<input><name>xyz.txt</name></input>"
 Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: Oct 21 07:43:27 2016 GMT
* expire date: Oct 19 07:43:27 2026 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/operations/system/file-delete/file HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.49.1
> Accept: */*
> content-type:application/vnd.yang.data+json
> Content-Length: 34
>
* upload completely sent off: 34 out of 34 bytes
< HTTP/1.1 204 No Content
```

```
< Server: nginx/1.6.3
< Date: Thu, 01 Dec 2016 07:37:28 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
```

USB Mount API

The supported USB formats are FAT32 and exFAT.

Table 78: USB Mount API

Action	Method	Payload Required	API
To mount a USB drive on a server that supports Cisco Enterprise NFVIS	POST	Yes	/api/operations/system/usb/mount
To unmount a USB drive from an NFVIS server	POST	No	/api/operations/system/usb/unmount
To view list of mount points	GET	No	/api/operational/system/usb/mnt-info

Example for USB Mount Payload

```
<mount>ACTIVE</mount>
```

Table 79: USB Mount Payload Description

Property	Type	Description	Mandatory/Default Value
mount	string	Mounts the USB drive. **Note** You can copy files from the USB drive only after mounting the USB drive.	Yes

Example: POST USB Mount API

```
Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
```

```
* Server auth using Basic with user 'admin'
> POST /api/operations/system/usb/mount HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Tue, 31 Jan 2017 22:25:38 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: POST USB Unmount API

```
Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> POST /api/operations/system/usb/unmount HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
```

```
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Tue, 31 Jan 2017 22:25:38 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET USB Mount Point

```
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* Server auth using Basic with user 'admin'
> GET /api/operational/system/usb/mnt-info HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.collection+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Tue, 31 Jan 2017 23:53:41 GMT
< Content-Type: application/vnd.yang.collection+xml
< Content-Length: 0
```

```
< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Pragma: no-cache

<

* Connection #0 to host 209.165.201.1 left intact
```

USB Copy API

Table 80: USB Copy API

Action	Method	Payload Required	API
Copy a single file from a mounted USB drive to the local folder of the server (/data/upload1/)	POST	Yes	/api/operations/system/file-copy/usb/file

Example for USB Copy Payload

```
<input><name><path_of_file_relative_to_usb/example_file.txt></name></input>
```

Table 81: USB Copy Payload Description

Property	Type	Description	Mandatory/Default Value
name	string	Name of the file with complete path relative to USB Path of the file within the USB drive. For example, if the file in the USB drive is like the following : *images/isrv.tar.gz*—The name parameter in payload must be "images/isrv.tar.gz". *asav.tar.gz*—The name parameter in payload must be "asav.tar.gz".	Yes

Example: POST USB Copy API

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+xml" -H
Content-Type:application/vnd.yang.data+xml -X
```

```
POST https://209.165.201.1/api/operations/system/file-copy/usb/file -d
"<input><name>asav.tat.gz</name></input>"
   Output Logs :
   * upload completely sent off: 21 out of 21 bytes
     < HTTP/1.1 204 No Content
     < Server: nginx/1.6.3
     < Date: Sat, 06 Aug 2016 10:05:48 GMT
     < Content-Type: text/html
     < Content-Length: 0
     < Connection: keep-alive
     < Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
     < Pragma: no-cache
```

Host Reboot API

Table 82: Host Reboot API

Action	Method	Payload Required	API
To reboot the host server	POST	No	/api/operations/hostaction/reboot

DHCP Renew API

Table 83: WAN DHCP Renew API

Action	Method	Payload Required	API
To renew the DHCP IP address on the WAN bridge	POST	No	/api/operations/hostaction/wan-dhcp-renew
To renew DHCP on bridge	POST	Yes	/api/operations/hostaction/bridge-dhcp-renew/bridge/<br_name>

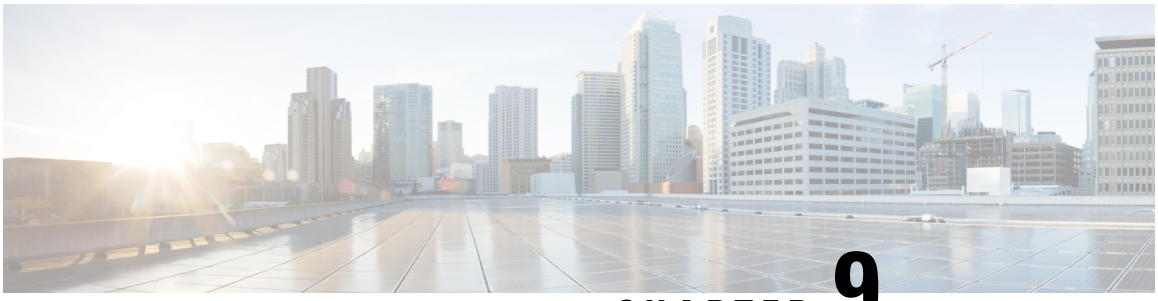
Example: POST WAN DHCP Renew API

```
curl -k -v -u admin:Cisco123# -H content-type: application/vnd.yang.data+xml -H
Accept:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/operations/hostaction/wan-dhcp-renew
* Hostname was NOT found in DNS cache
* Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
* CAfile: none
 CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
```

```
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
* subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* start date: 2016-04-17 23:36:59 GMT
* expire date: 2026-04-15 23:36:59 GMT
* issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/operations/hostaction/wan-dhcp-renew HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> content-type: application/vnd.yang.data+xml
> Accept:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
* Server nginx/1.6.3 is not blacklisted
< Server: nginx/1.6.3
< Date: Wed, 20 Apr 2016 23:05:05 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
```

Example: Bridge DHCP Renew

```
curl -k -v -u admin:admin -H "Accept:application/vnd.yang.data+json" -H
"Content-Type:application/vnd.yang.data+json" -X POST
https://localhost/api/operations/hostaction/bridge-dhcp-renew/bridge/test-br
```



CHAPTER 9

SPAN Session and Packet Capture APIs

Table 84: SPAN Session APIs

Action	Method	Payload Required	API
To create a SPAN session	POST	Yes	/api/config/monitor
To get the SPAN monitor session status	GET	No	/api/operational/monitor\?deep
To get the SPAN session configuration details	GET	No	/api/config/monitor\?deep
to get the SPAN session operational status	GET	No	• /api/operational/system/monitor/session • /api/operational/system/monitor/session\?deep

Example for a SPAN Session Payload

```
<session>
    <number>20</number>
    <destination>
        <vm-vnic>
            <vm-name>Linux2</vm-name>
            <vnic-id>0</vnic-id>
        </vm-vnic>
    </destination>
    <source>
    <interfaces>
        <vm-vnic>
            <vm-name>Linux1</vm-name>
            <vnic-id>0</vnic-id>
            <direction>both</direction>
       </vm-vnic>
        <interface>
            <name>GE0-0</name>
            <direction>both</direction>
        </interface>
```

```
    </interfaces>
    </source>
</session>
```

Table 85: SPAN Session Payload Description

Property	Type	Description	Mandatory/Default Value
number	Integer	SPAN session number	Yes
destination	String	Destination for the mirrored traffic	Yes
vm-name	String	Name of the VM	Yes
vnic-id	String	Virtual network interface controller ID	Yes In the case of virtio net or SRIOV VF, you have to specify the NIC ID of the VM interface.
source	String	Source the mirrored traffic	Yes
direction	String	Direction of the traffic	Yes
interface	String	Source or destination interface.	Yes

- Example: POST SPAN Session API, on page 162
- Example: GET SPAN Session APIs, on page 163
- Packet Capture APIs, on page 166

Example: POST SPAN Session API

```
curl -v -u admin:XXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
POST https://209.165.201.1/api/config/monitor -d '<session><number>20</number>
<destination><vm-vnic><vm-name>Linux2</vm-name><vnic-id>0</vnic-id></vm-vnic></destination>
<source><interfaces><vm-vnic><vm-name>Linux1</vm-name><vnic-id>0</vnic-id><direction>both</direction></vm-vnic>
<interface><name>GE0-0</name><direction>both</direction></interface></interfaces></source></session>'

Note: Unnecessary use of -X or --request, POST is already inferred.
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
```

```
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/monitor HTTP/1.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 330
>
* upload completely sent off: 330 out of 330 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 02:42:25 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 15 Mar 2017 02:42:25 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-545745-460682
< Pragma: no-cache
<
sj22lab-as2:145>
```

Example: GET SPAN Session APIs

Use this operational API to get the SPAN monitor session status.

```
curl -v -u admin:XXXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
GET https://209.165.201.1/api/operational/monitor\?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
```

```
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/monitor?deep HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 04:43:15 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache

<

<monitor xmlns="http://www.cisco.com/nfv/span_session" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:span-session="http://www.cisco.com/nfv/span_session">
  <session>
    <number>20</number>
    <source>
      <interfaces>
        <vm-vnic>
          <vm-name>Linux1</vm-name>
          <vnic-id>0</vnic-id>
          <direction>both</direction>
        </vm-vnic>
        <interface>
          <name>GE0-0</name>
          <direction>both</direction>
        </interface>
      </interfaces>
    </source>
    <destination>
      <vm-vnic>
        <vm-name>Linux2</vm-name>
        <vnic-id>0</vnic-id>
      </vm-vnic>
    </destination>
    <status>CREATE_SUCCESS</status>
  </session>
</monitor>
```

Use this GET API to get the SPAN session configuration details.

```
curl -v -u admin:XXXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
GET https://209.165.201.1/api/config/monitor\?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
```

```
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/config/monitor?deep HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 04:39:29 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Wed, 15 Mar 2017 02:42:25 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-545745-460682
< Pragma: no-cache
<

<monitor xmlns="http://www.cisco.com/nfv/span_session" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:span-session="http://www.cisco.com/nfv/span_session">
  <session>
    <number>20</number>
    <source>
      <interfaces>
        <vm-vnic>
          <vm-name>Linux1</vm-name>
          <vnic-id>0</vnic-id>
          <direction>both</direction>
        </vm-vnic>
        <interface>
          <name>GE0-0</name>
          <direction>both</direction>
        </interface>
      </interfaces>
    </source>
    <destination>
      <vm-vnic>
        <vm-name>Linux2</vm-name>
        <vnic-id>0</vnic-id>
      </vm-vnic>
    </destination>
  </session>
</monitor>
```

Packet Capture APIs

Table 86: Packet Capture APIs

Action	Method	Payload Required	API
To configure packet capture on a physical or virtual network interface controller	POST	Yes	api/operations/packet-capture/tcpdump

Example for the Packet Capture Payload for a Physical Port

```
<input>
    <port>eth0</port>
    <time>10</time><
/input>
```

Example for the Packet Capture Payload for a vNIC

```
<input>
    <vnic>
        <tenant-name>admin</tenant-name>
        <deployment-name>1489084431</deployment-name>
        <vm-name>ROUTER</vm-name>
        <vnic-id>0</vnic-id>
    </vnic>
    <time>10</time>
</input>
```

Table 87: Packet Capture Payload Description

Property	Type	Description	Mandatory/Default Value
port	String	Physical or virtual network interface controller	Yes
time	String	Time period over which packets are captured. The default value is 60 seconds.	Yes
tenant-name	String	Name of the tenant	Yes
deployment-name	String	Name of the VM deployment	Yes
vm-name	String	Name of the VM	Yes
vnic-id	Integer	Virtual network interface controller ID	Yes

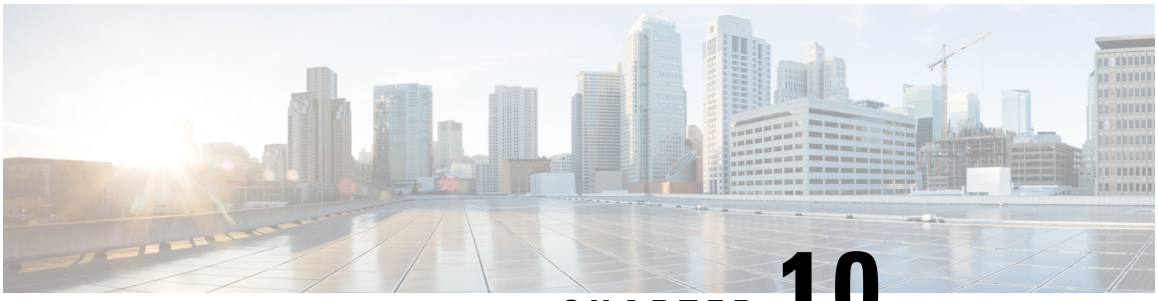
Example: POST Packet Capture APIs

Use this POST API to configure packet capture on a physical port.

```
curl -v -k -u admin:Cisco123# -H "Content-Type: application/vnd.yang.data+xml"  -H "Accept:
 application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/packet-capture/tcpdump  -d
'<input><port>eth0</port><time>10</time></input>'
*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> POST /api/operations/packet-capture/tcpdump HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMj

> User-Agent: curl/7.43.0

> Content-Type: application/vnd.yang.data+xml

> Accept: application/vnd.yang.data+xml

> Content-Length: 47

>

* upload completely sent off: 47 out of 47 bytes

< HTTP/1.1 200 OK

< Server: nginx/1.10.1

< Date: Wed, 29 Mar 2017 20:35:50 GMT

< Content-Type: application/vnd.yang.operation+xml

< Content-Length: 151

< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Vary: Accept-Encoding

< Pragma: no-cache

<

<output xmlns='http://www.cisco.com/nfvos/packet_capture'>

  <pcap-location>/data/intdatastore/pktcaptures/tcpdump_eth0.pcap</pcap-location>

</output>

* Connection #0 to host 209.165.201.1 left intact
```

Use this POST API to configure packet capture on a vNIC.

```
curl -v -k -u admin:Cisco123# -H "Content-Type: application/vnd.yang.data+xml"  -H "Accept:
 application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/packet-capture/tcpdump  -d
'<input><vnic><tenant-name>admin</tenant-name>
<deployment-name>1489084431</deployment-name><vm-name>ROUTER</vm-name><vnic-id>0</vnic-id></vnic><time>10</time></input>'

*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> POST /api/operations/packet-capture/tcpdump HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMj

> User-Agent: curl/7.43.0

> Content-Type: application/vnd.yang.data+xml

> Accept: application/vnd.yang.data+xml

> Content-Length: 47

>

* upload completely sent off: 47 out of 47 bytes

< HTTP/1.1 200 OK

< Server: nginx/1.10.1

< Date: Wed, 29 Mar 2017 20:35:50 GMT

< Content-Type: application/vnd.yang.operation+xml

< Content-Length: 151

< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Vary: Accept-Encoding

< Pragma: no-cache

<

<output
xmlns='/data/intdatastore/pktcaptures/1489084431_ROUTER_vnic0.pcaphttp://www.cisco.com/nfvos/packet_capture'>
<pcap-location>/data/intdatastore/pktcaptures/1489084431_ROUTER_vnic0.pcap</pcap-location>
</output>

* Connection #0 to host 209.165.201.1 left intact
```



CHAPTER 10

Upgrade Package APIs

Upgrade Package Register API

Table 88: Upgrade Package Register API

Action	Method	Payload Required	API
To register a package for upgrade	POST	Yes	/api/config/system/upgrade
To view registered packages	GET	No	• /api/operational/system/upgrade/reg-info • /api/operational/system/upgrade /reg-info\?deep • /api/config/system/upgrade
To delete a registered package	DELETE	No	/api/config/system/upgrade/image-name

Example for Upgrade Package Register Payload

```
<image-name>
 <name>test3</name>
 <location>/data/intdatastore/uploads/package/upgrade package filename(.nfvispkg)</location>

</image-name>
```

Table 89: Upgrade Package Register Payload Description

Property	Type	Description	Mandatory/Default Value
image-name	String	Name of the image	Yes

location	String	Location of the image	Yes
/data/intdatastore/uploads	String	Default path of the image	Yes
upgrade package filename(.nfvispkg)	String	Full path with the package name.	• If only one upgrade-package (.nfvispkg) exists on NFVIS /data/intdatastore/uploads directory, it's not necessary to specify the upgrade package name, after path /data/intdatastore/uploads. • When multiple upgrade-packages (.nfvispkg) exist on NFVIS /data/intdatastore/uploads, users need to specify specific upgrade-package to be registered, after path /data/intdatastore/uploads.

Example: POST Upgrade Package Register API

```
curl -k -v -u admin:admin -H content-type:application/vnd.yang.data+json -X
POST https://209.165.201.1/api/config/system/upgrade --data '<image-name>
<name>nfvis-3.6.1</name>
<location>/data/intdatastore/uploads/Cisco_NFVIS_Upgrade-3.6.1-FC3.nfvispkg</location></image-name>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
```

```
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug  5 15:38:14 2016 GMT
*  expire date: Aug  3 15:38:14 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/system/upgrade HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.49.1
> Accept: */*
> content-type:application/vnd.yang.data+json
> Content-Length: 33
>
* upload completely sent off: 33 out of 33 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.6.3
< Date: Fri, 05 Aug 2016 17:34:35 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://209.165.201.1/api/config/system/upgrade/image-name/test1
< Connection: keep-alive
< Last-Modified: Fri, 05 Aug 2016 17:34:33 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1470-418473-704300
< Pragma: no-cache
```

Example: GET Upgrade Package Register API

```
curl -k -v -u admin:admin -H content-type:application/vnd.yang.data+json -X
GET https://209.165.201.1/api/operational/system/upgrade/reg-info

Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 31 02:47:22 2017 GMT
*  expire date: Mar 29 02:47:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/upgrade/reg-info HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
```

```
> User-Agent: curl/7.50.1
> Accept: */*
> content-type:application/vnd.yang.data+json
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Fri, 31 Mar 2017 22:34:27 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<reg-info xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
  <name>
    <name>Cisco_NFVIS_Upgrade-3.6.1-693-20170329_022604.nfvispkg</name>
  </name>
</reg-info>
* Connection #0 to host 209.165.201.1 left intact
```

Example: DELETE Upgrade Package Register API

```
curl -k -v -u admin:admin -X DELETE
https://209.165.201.1/api/config/system/upgrade/image-name/nfvis-3.3.1

*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug  5 15:38:14 2016 GMT
*  expire date: Aug  3 15:38:14 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/system/upgrade/image-name/nfvis-3.3.1 HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 05 Aug 2016 19:36:57 GMT
< Content-Type: text/html
< Content-Length: 0
```

```
< Connection: keep-alive
< Last-Modified: Fri, 05 Aug 2016 19:36:57 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1470-425817-671801
< Pragma: no-cache
```

Upgrade Apply-Image APIs

Table 90: Upgrade Apply-Image API

Action	Method	Payload Required	API
To upgrade the existing image to a newly registered image	POST	Yes	/api/config/system/upgrade
To verify the upgrade status	GET	No	• /api/operational/system/upgrade/apply-image • /api/operational/system/upgrade/apply-image\?deep
To delete the upgraded image	DELETE	No	/api/config/system/upgrade/ /apply-image/<image-name>

Example for Upgrade Apply-Image Payload

```
<apply-image>
 <name>nfvis-3.3.1</name>
 <scheduled-time>24</scheduled-time>
</apply-image>
```

Table 91: Upgrade Apply-Image Payload Description

Property	Type	Description	Mandatory/Default Value
name	string	Name of the image for the upgrade	Yes
scheduled-time	Numeric	Scheduled time in hours. Valid range: 0-24. Zero means immediate upgrade.	Yes

Example: POST Upgrade Apply-Image API

```
curl -k -v -u admin:admin -H content-type:application/vnd.yang.data+json -X
POST https://209.165.201.1/api/config/system/upgrade --data '<apply-image>
<name>nfvis-3.3.1</name> <scheduled-time>24</scheduled-time> </apply-image>'
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
```

```
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug  5 15:38:14 2016 GMT
*  expire date: Aug  3 15:38:14 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/system/upgrade HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.49.1
> Accept: */*
> content-type:application/vnd.yang.data+json
> Content-Length: 53
>
* upload completely sent off: 53 out of 53 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.6.3
< Date: Fri, 05 Aug 2016 18:41:02 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://209.165.201.1/api/config/system/upgrade/apply-image/nfvis-3.3.1
< Connection: keep-alive
< Last-Modified: Fri, 05 Aug 2016 18:41:02 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1470-422462-89670
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```

Example: GET Upgrade Apply-Image API

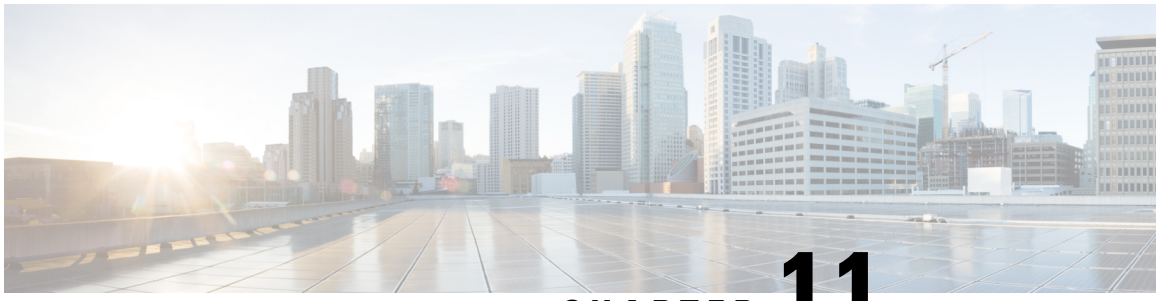
```
curl -k -v -u admin:admin -H content-type:application/vnd.yang.data+json -X
GET https://209.165.201.1/api/operational/system/upgrade/apply-image

Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
```

```
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 31 02:47:22 2017 GMT
*  expire date: Mar 29 02:47:22 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/upgrade/apply-image HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> content-type:application/vnd.yang.data+json
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Fri, 31 Mar 2017 22:34:49 GMT
< Content-Type: application/vnd.yang.collection+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<
<collection xmlns:y="http://tail-f.com/ns/rest">
  <apply-image xmlns="http://www.cisco.com/nfv">
    <name>Cisco_NFVIS_Upgrade-3.6.1-693-20170329_022604.nfvispkg</name>
    <scheduled-time>24</scheduled-time>
    <status>SCHEDULED</status>
  </apply-image>
</collection>
* Connection #0 to host 209.165.201.1 left intact
```

Example: DELETE Upgrade Apply-Image API

```
curl -k -v -u admin:admin -X DELETE
https://209.165.201.1/api/config/system/upgrade/apply-image/nfvis-3.3.1

*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt  CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Aug  5 15:38:14 2016 GMT
*  expire date: Aug  3 15:38:14 2026 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
```

```
* Server auth using Basic with user 'admin'
> DELETE /api/config/system/upgrade/apply-image/nfvis-3.3.1 HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 204 No Content
< Server: nginx/1.6.3
< Date: Fri, 05 Aug 2016 19:57:32 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Fri, 05 Aug 2016 19:57:32 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1470-427052-771331
< Pragma: no-cache
<
* Connection #0 to host 209.165.201.1 left intact
```



CHAPTER 11

Factory Default Reset APIs

Table 92: Factory Default Reset APIs

Action	Method	Payload Required	API
To reset to factory default	POST	No	• /api/operations/factory-default-reset/all • /api/operations/factory-default-reset /all-except-images • /api/operations/factory-default-reset /all-except-images-connectivity

- Example: POST Factory Default Reset All, on page 177
- Example: POST Factory Default Reset All Except Images, on page 178
- Example: POST Factory Default Reset All Except Images Connectivity, on page 179

Example: POST Factory Default Reset All

```
curl  -v -u 'admin:Admin123$' -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/operations/factory-default-reset/all
* About to connect() to 209.165.201.1:443

* Connected to 209.165.201.1 (209.165.201.1) port 443

* SSL connection using EDH-RSA-DES-CBC3-SHA

* Server certificate:

*   subject: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*   start date: 2017-02-21 20:10:51 GMT

*   expire date: 2027-02-19 20:10:51 GMT

*   common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate (does not match
'209.165.201.1')
```

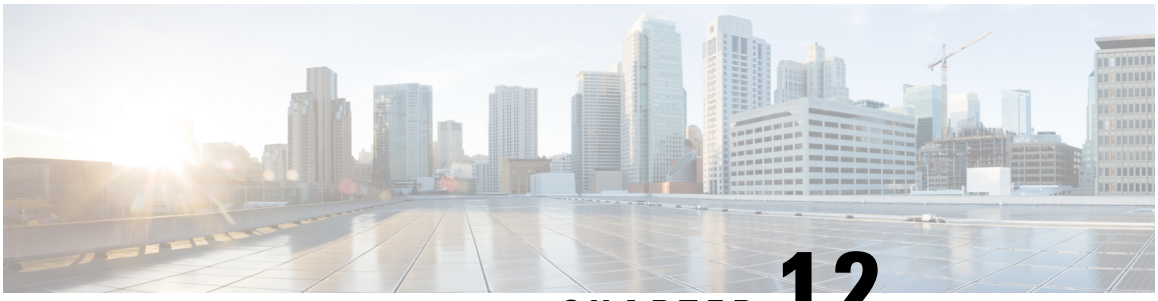
```
*   issuer: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

> POST /api/operations/factory-default-reset/all HTTP/1.1

Authorization: Basic YWRtaW46QWRtaW4xMjMk

User-Agent: curl/7.9.6 (i686-pc-linux-gnu) libcurl 7.9.6 (OpenSSL 0.9.6)

Host: 209.165.201.1

Pragma: no-cache

Accept:application/vnd.yang.data+xml

Content-Type:application/vnd.yang.data+xml
```

Example: POST Factory Default Reset All Except Images

```
curl  -v -u 'admin:Admin123$' -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/operations/factory-default-reset/all-except-images
* About to connect() to 209.165.201.1:443

* Connected to 209.165.201.1 (209.165.201.1) port 443

* SSL connection using EDH-RSA-DES-CBC3-SHA

* Server certificate:

*   subject: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*   start date: 2017-02-21 20:10:51 GMT

*   expire date: 2027-02-19 20:10:51 GMT

*   common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate (does not match
'209.165.201.1')

*   issuer: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

> POST /api/operations/factory-default-reset/all-except-images HTTP/1.1

Authorization: Basic YWRtaW46QWRtaW4xMjMk

User-Agent: curl/7.9.6 (i686-pc-linux-gnu) libcurl 7.9.6 (OpenSSL 0.9.6)

Host: 209.165.201.1

Pragma: no-cache

Accept:application/vnd.yang.data+xml

Content-Type:application/vnd.yang.data+xml
```

Example: POST Factory Default Reset All Except Images Connectivity

```
curl  -v -u 'admin:Admin123$' -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X
POST https://209.165.201.1/api/operations/factory-default-reset/all-except-images-connectivity
* About to connect() to 209.165.201.1:443

* Connected to 209.165.201.1 (209.165.201.1) port 443

* SSL connection using EDH-RSA-DES-CBC3-SHA

* Server certificate:

*   subject: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

*   start date: 2017-02-21 20:10:51 GMT

*   expire date: 2027-02-19 20:10:51 GMT

*   common name: Cisco-Enterprise-NFVIS-Self-Signed-Certificate (does not match
'209.165.201.1')

*   issuer: /CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate

> POST /api/operations/factory-default-reset/all-except-images-connectivity HTTP/1.1

Authorization: Basic YWRtaW46QWRtaW4xMjMk

User-Agent: curl/7.9.6 (i686-pc-linux-gnu) libcurl 7.9.6 (OpenSSL 0.9.6)

Host: 209.165.201.1

Pragma: no-cache

Accept:application/vnd.yang.data+xml

Content-Type:application/vnd.yang.data+xml
```



CHAPTER 12

Syslog Support APIs

Table 93: Syslog Support APIs

Action	Method	Payload Required	API
To configure syslog server	POST	Yes	/api/config/system/settings/logging
To update the syslog server information	PUT	Yes	/api/config/system/settings/logging/host/<host-address>
To remove the syslog server information	DELETE	No	/api/config/system/settings/logging/host/<host-address>
To configure syslog severity	PUT	Yes	/api/config/system/settings/logging/severity
To configure syslog facility	PUT	Yes	/api/config/system/settings/logging/facility
To view syslog configuration	GET	No	/api/operational/system/settings/logging

Example: POST Syslog Server

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X POST
https://172.19.162.209/api/config/system/settings/logging -d
'<host><host>172.19.162.143</host><transport><udp/></transport><port>525</port></host>'

Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
```

```
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/system/settings/logging HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 85
>
* upload completely sent off: 85 out of 85 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:01:40 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://172.19.162.209/api/config/system/settings/logging/host/172.19.162.143
< Connection: keep-alive
< Last-Modified: Thu, 08 Jun 2017 09:01:40 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1496-912500-289655
< Pragma: no-cache
<
```

Example: PUT Remote Logging Host Configuration

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X PUT
https://172.19.162.209/api/config/system/settings/logging/host/172.19.162.143 -d
'<host><host>172.19.162.143</host><transport><tcp/></transport><port>1525</port></host>'

* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
```

```
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/settings/logging/host/172.19.162.143 HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 86
>
* upload completely sent off: 86 out of 86 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:11:58 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 08 Jun 2017 09:11:58 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1496-913118-15836
< Pragma: no-cache
<
```

Example: DELETE Remote Logging Host Configuration

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X DELETE
https://172.19.162.209/api/config/system/settings/logging/host/172.19.162.143

* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
```

```
* Server auth using Basic with user 'admin'
> DELETE /api/config/system/settings/logging/host/172.19.162.143 HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:18:10 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 08 Jun 2017 09:18:10 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1496-913490-383818
< Pragma: no-cache
<
```

Example: PUT Syslog Severity

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X PUT
https://172.19.162.209/api/config/system/settings/logging/severity -d
'<severity>error</severity>'

* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/settings/logging/severity HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 26
>
* upload completely sent off: 26 out of 26 bytes
< HTTP/1.1 204 No Content
```

```
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:33:01 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 08 Jun 2017 09:33:01 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1496-914381-204591
< Pragma: no-cache
<
```

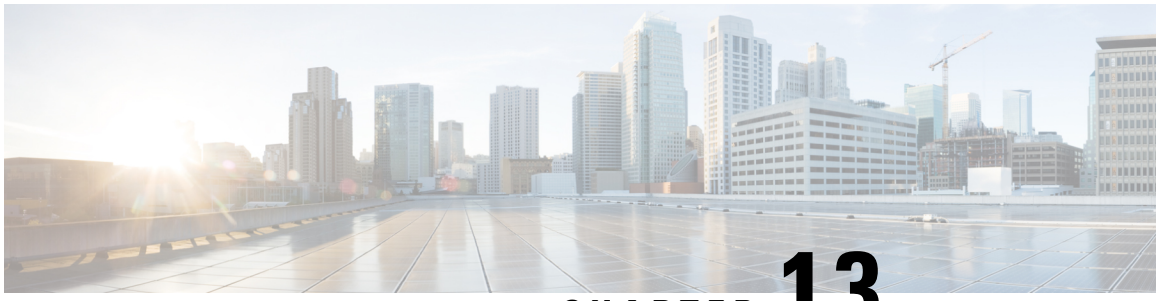
Example: PUT Syslog Facility

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X PUT
https://172.19.162.209/api/config/system/settings/logging/facility -d
'<facility>local5</facility>'
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/system/settings/logging/facility HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 27
>
* upload completely sent off: 27 out of 27 bytes
< HTTP/1.1 204 No Content
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:20:21 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Thu, 08 Jun 2017 09:20:21 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1496-913621-135440
< Pragma: no-cache
<
```

Example: GET Remote Logging Host

```
curl -k -v -u admin:Cisco123# -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml-X GET
'https://172.19.162.209/api/operational/system/settings/logging?deep'

Note: Unnecessary use of -X or --request, GET is already inferred.
* Trying 172.19.162.209...
* Connected to 172.19.162.209 (172.19.162.209) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
 CApath: none
* TLSv1.0 (OUT), TLS handshake, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Server hello (2):
* TLSv1.0 (IN), TLS handshake, Certificate (11):
* TLSv1.0 (IN), TLS handshake, Server key exchange (12):
* TLSv1.0 (IN), TLS handshake, Server finished (14):
* TLSv1.0 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.0 (OUT), TLS change cipher, Client hello (1):
* TLSv1.0 (OUT), TLS handshake, Finished (20):
* TLSv1.0 (IN), TLS change cipher, Client hello (1):
* TLSv1.0 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.0 / DHE-RSA-AES256-SHA
* Server certificate:
* subject: CN=nfvis
* start date: Jun 2 18:39:33 2017 GMT
* expire date: May 31 18:39:33 2027 GMT
* issuer: CN=nfvis
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/system/settings/logging?deep HTTP/1.1
> Host: 172.19.162.209
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.49.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Thu, 08 Jun 2017 09:03:37 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<logging xmlns="http://www.cisco.com/nfv" xmlns:y="http://tail-f.com/ns/rest"
xmlns:system="http://www.cisco.com/nfv">
 <host>
   <host>172.19.162.117</host>
   <transport>
    <tcp/>
   </transport>
   <port>1635</port>
 </host>
 <host>
   <host>172.19.162.111</host>
   <transport>
     <udp/>
```

```
    </transport>
    <port>163</port>
  </host>
  <host>
    <host>172.19.162.112</host>
      <port>1523</port>
    </host>
  <host>
    <host>172.19.162.143</host>
    <transport>
      <udp/>
    </transport>
    <port>525</port>
  </host>
</logging>
```



CHAPTER 13

SNMP Support APIs

Table 94: SNMP Support APIs

Action	Method	Payload Required	API
To configure communities	POST	Yes	/api/snmp/communities
To enable SNMP traps	POST	Yes	/api/config/snmp/enable/traps
To configure SNMP hosts	POST	Yes	/api/config/snmp/hosts
To configure SNMP users	POST	Yes	/api/config/snmp/users
To configure SNMP groups	POST	Yes	/api/config/snmp/groups
To view SNMP configuration	GET	No	/api/config/snmp/agent /api/snmp/communities /api/config/snmp/enable/traps /api/config/snmp/hosts /api/config/snmp/users /api/config/snmp/groups

Note SNMP Agent is enabled by default.

If the curl command is not working, run the curl commands with **-i** option. For example:

```
curl -k -i -v -u <USER>:<PASSWORD> -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X DELETE https://<IP-ADDR>/api/config/snmp/hosts/
```


- Example: POST Configuring SNMP Communities, on page 190
- Example: POST SNMP Traps, on page 190
- Example: POST SNMP Host, on page 190
- Example: POST SNMP Users, on page 191

• Example: POST SNMP Groups, on page 191
• Example: GET SNMP Configurations, on page 191

Example: POST Configuring SNMP Communities

```
curl -k -v -u "admin:Cisco123#" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/snmp/communities -d '
<community>
 <community-name>test3</community-name>
 <community-access>readOnly</community-access>
</community>'
```

Table 95: Field Descriptions for SNMP Communities

Field	Description
community-name	Upto 32 char alphanumeric string (including _ and -)
community-access	Read-only

Example: POST SNMP Traps

```
curl -k -v -u "admin:XXX" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/config/snmp/enable/traps -d '
<trap-type>linkDown</trap-type>'
```

Table 96: Field Description for SNMP trap API

Field	Description
trap-type	linkUp or linkDown

Example: POST SNMP Host

```
curl -k -v -u "admin:XXX" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/config/snmp/hosts -d '
<host>
 <host-name>listen_host7</host-name>
 <host-port>162</host-port>
 <host-ip-address>10.32.172.190</host-ip-address>
 <host-version>2</host-version>
 <host-security-level>noAuthNoPriv</host-security-level>
 <host-user-name>user1</host-user-name>
</host>'
```

```
curl -k -v -u "admin:XXX" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/config/snmp/hosts -d '
<host>
 <host-name>listen_host7</host-name>
 <host-port>162</host-port>
 <host-ip-address>10.32.172.190</host-ip-address>
 <host-version>3</host-version>
 <host-security-level>authPriv</host-security-level>
 <host-user-name>user1</host-user-name>
</host>
```

Example: POST SNMP Users

```
curl -k -v -u "admin:XXXX" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/config/snmp/users -d '
<user>
 <user-name>test_user1</user-name>
 <user-version>2</user-version>
 <user-group>public</user-group>
</user>'
```

Table 97: Field Description for SNMP User API

Field	Description
passphrase	Alphanumeric string with 8 character mininum length.
auth-protocol	md5 or sha
priv-protocol	aes or des

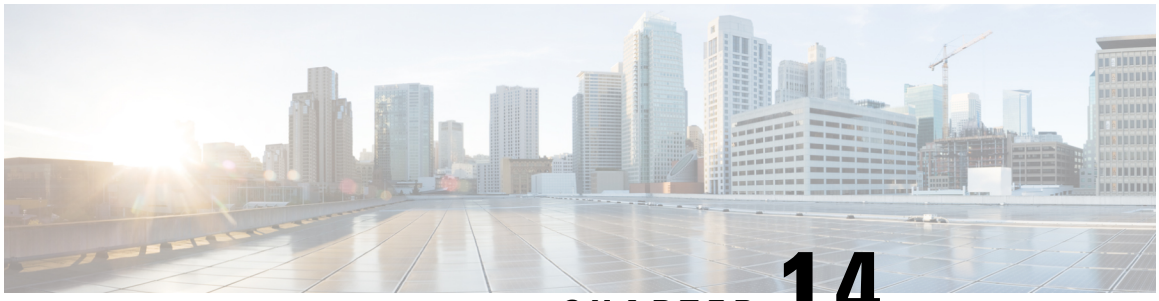
Example: POST SNMP Groups

```
curl -k -v -u "admin:XXX" -H "Accept:application/vnd.yang.data+xml" -H
"Content-Type:application/vnd.yang.data+xml" -X POST
https://172.19.162.235/api/config/snmp/groups -d '
<group>
 <group-name>testgroup2</group-name>
 <group-context-prefix>snmp</group-context-prefix>
 <group-version>2</group-version>
 <security-level>noAuthNoPriv</security-level>
 <read>read-access</read>
 <write>write-access</write>
 <notify>notify-access</notify>
</group>'
```

Example: GET SNMP Configurations

```
curl -k -v -u "admin:XXX" -H "Accept:application/vnd.yang.data+xml" -H
```

```
"Content-Type:application/vnd.yang.data+xml" -X GET
https://172.19.162.235/api/config/snmp/hosts
```



CHAPTER 14

TACACS and RADIUS Support APIs

TACACS Support APIs

Table 98: TACACS Support APIs

Action	Method	Payload Required	API
To configure TACACS server	POST	Yes	/api/config/security_servers/tacacs-server/
To configure TACACS server	PUT	Yes	/api/config/security_servers/tacacs-server/
To configure TACACS server	DELETE	No	/api/config/security_servers/tacacs-server/
To view TACACS server configuration	GET	No	/api/config/security_servers/tacacs-server/

Example: POST TACACS Server

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H Content-

Type:application/vnd.yang.data+json -X POST

https://209.165.201.1/api/config/security_servers/tacacs-server -d '{"host":

{"server":"10.10.10.10", "secret": {"key": "0", "shared-secret": "heyworld", "admin-priv":
 "14",

"oper-priv": "10"}}}'
* Hostname was NOT found in DNS cache
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
```

```
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/security_servers/tacacs-server HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
> Content-Length: 122
>
* upload completely sent off: 122 out of 122 bytes
< HTTP/1.1 201 Created
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:14:46 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://209.165.201.1/api/config/security_servers/tacacs-server/host/5.5.5.5
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:14:46 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219286-189602
< Pragma: no-cache
<
```

Example: PUT TACACS Server

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H Content-

Type:application/vnd.yang.data+json -X PUT

https://209.165.201.1/api/config/security_servers/tacacs-server/host/5.5.5.5 -d '{"host":

{"server":"5.5.5.5", "secret": {"shared-secret":"helloworld", "admin-priv": "15"}}}'
* Hostname was NOT found in DNS cache
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
```

```
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/security_servers/tacacs-server/host/5.5.5.5 HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
> Content-Length: 92
>
* upload completely sent off: 92 out of 92 bytes
< HTTP/1.1 204 No Content
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:20:13 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:20:13 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219613-571277
< Pragma: no-cache
<
```

Example: GET TACACS Server API

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+json -X
GET https://209.165.201.1/api/config/security_servers/tacacs-server?deep
* Hostname was NOT found in DNS cache
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
```

```
> GET /api/config/security_servers/tacacs-server?deep HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
>
< HTTP/1.1 200 OK
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:07:49 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Fri, 24 Feb 2017 01:13:51 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1487-898831-958028
< Pragma: no-cache
<

<tacacs-server xmlns="http://www.cisco.com/ns/test/security" xmlns:y="http://tail-

f.com/ns/rest"  xmlns:security="http://www.cisco.com/ns/test/security">
  <host>
    <server>10.2.2.2</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
  <host>
    <server>10.3.3.3</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
  <host>
    <server>10.1.1.1</server>
    <secret>
      <key>0</key>
      <shared-secret>tac22</shared-secret>
    </secret>
  </host>
</tacacs-server>
```

Example: DELETE TACACS Server

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H Content-

Type:application/vnd.yang.data+json -X DELETE

https://209.165.201.1/api/config/security_servers/tacacs-server/host/5.5.5.5
* Hostname was NOT found in DNS cache
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: none
  CApath: /etc/ssl/certs
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
```

```
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*        subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        start date: 2017-01-13 23:47:41 GMT
*        expire date: 2027-01-11 23:47:41 GMT
*        issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*        SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/security_servers/tacacs-server/host/5.5.5.5 HTTP/1.1
> Authorization: Basic YWRtaW46Y2lzY28xMjM=
> User-Agent: curl/7.35.0
> Host: 209.165.201.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+json
>
< HTTP/1.1 204 No Content
* Server nginx/1.10.1 is not blacklisted
< Server: nginx/1.10.1
< Date: Mon, 27 Feb 2017 18:21:30 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Mon, 27 Feb 2017 18:21:30 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1488-219690-404414
< Pragma: no-cache
```

RADIUS Support APIs

Table 99: RADIUS Support APIs

Action	Method	Payload Required	API
To configure RADIUS server	POST	Yes	/api/config/security_servers/radius-server/
To update configurations on RADIUS server	PUT	Yes	/api/config/security_servers/radius-server/
To delete configurations on RADIUS server	DELETE	No	/api/config/security_servers/radius-server/
To view RADIUS server configuration	GET	No	/api/config/security_servers/radius-server/

Example: GET RADIUS Server

```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H Content-
Type:application/vnd.yang.data+json -X GET
https://209.165.201.1/api/config/security_servers/radius-server?deep
```

Example: POST RADIUS Server

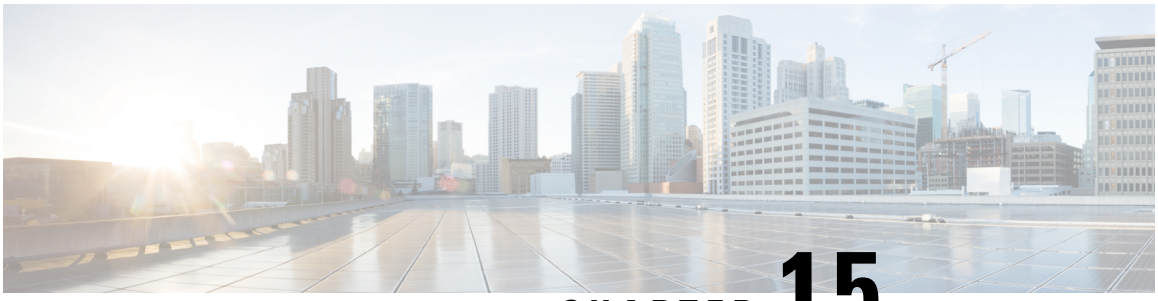
```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H Content-
Type:application/vnd.yang.data+json -X POST
https://209.165.201.1/api/config/security_servers/radius-server -d '{"host":
{"server":"5.5.5.5", "secret": {"key": "0", "shared-secret": "heyworld", "admin-priv": "14",
"oper-priv": "10"}}}'
```

Example: PUT RADIUS Server

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H Content-
Type:application/vnd.yang.data+json -X PUT
https://209.165.201.1/api/config/security_servers/radius-server/host/5.5.5.5 -d '{"host":
{"server":"5.5.5.5", "secret": {"shared-secret":"helloworld", "admin-priv": "15"}}}'
```

Example: DELETE RADIUS Server

```
curl -k -v -u "admin:cisco123" -H Accept:application/vnd.yang.data+xml -H Content-
Type:application/vnd.yang.data+json -X DELETE
https://209.165.201.1/api/config/security_servers/radius-server/host/5.5.5.5
```



CHAPTER 15

Port and Port Channel APIs

Action	Method	Payload Required	API
To show information about all ports including port channels	GET	No	/api/operational/pnics /api/operational/pnics?deep
To create a port channel	POST	Yes	/api/config/pnics
To add a port to a port channel	PUT	Yes	/api/config/pnics/pnic/*name*/member_of
To add a port channel to a new bridge	POST	Yes	/api/config/bridges/
To add a port channel to an existing bridge	PUT	Yes	/api/config/bridges/bridge/*bridgename*
To configure the LACP mode of a port channel	PUT	Yes	/api/config/pnics/pnic/*portchannel_name*/lacp_type
To configure the bond mode of a port channel	PUT	Yes	/api/config/pnics/pnic/*portchannel_name*/bond_mode
To configure trunks on a port channel	PUT	Yes	/api/config/pnics/pnic/*portchannelname*/trunks
To remove a port from a port channel	DELETE	Yes	/api/config/pnics/pnic/*portname*/member_of
To remove a port channel from a bridge	PUT	Yes	/api/config/bridges/bridge/*bridgename*

Action	Method	Payload Required	API
To delete a port channel **Note** Before deleting a port channel, you must remove all members assigned to the port channel. If the port channel is configured on the bridge, you must remove the port channel from the bridge.	DELETE	No	/api/config/pnics/pnic/*portchannelname*
To enable or disable LLDP **Note** LLDP cannot be enabled on a port channel.	PUT	Yes	/api/config/pnics/pnic/*portname*/lldp
To show LLDP neighbors and stats	GET	No	/api/operational/lldp /api/operational/lldp?deep
To configure the port admin status **Note** Adminstrator status is not supported on a port channel.	PUT	Yes	/api/config/pnics/pnic/*portname*/adminstatus
To get information about the admin status of a port	GET	No	/api/config/pnics/pnic/*portname*/adminstatus

Table 100: Ports and Port Channels APIs Payload Description

Property	Type	Description	Mandatory	Example
pnic name	String	Name of the port or port channel.	Yes	`<pnic><name>pc</name>` `<type>port_channel</type></pnic>`
type	String	Type of the port. Valid values are **ethernet** and **port_channel**. To create a port channel, you must specify the value as **port_channel**.	Yes	

Property	Type	Description	Mandatory	Example
lacp_type	String	The LACP type for a port channel. Valid values are **off**, **active**, and **passive**. Default is **off**.	No	`<lacp_type>active</lacp_type>`
bond_mode	String	The bond mode for a port channel. Valid values are **active-backup**, **balance-slb**, and **balance-tcp**. Default is **balance-tcp**	No	`<bond_mode>balance-tcp</bond_mode>`
trunks	Integer	VLAN IDs. Valid range is from 1 to 4096. Default is VLAN 1. Enter VLANs separated by commas, VLAN ranges separated by dashes, or a combination of both.	No	`<trunks>10,20</trunks>`
member_of	String	The name of the port channel to which you want to add a port or from which you want to remove a port.	Yes	`<member_of>pc</member_of>`
port name	String	The name of the port channel that you want to add to the bridge or remove from the bridge.	Yes	`<port><name>pc</name></port>`
bridge name	String	The name of the bridge from which you want to remove the port channel or to which you want to add the port channel.	Yes	`<bridge><name>test-br</name></bridge>`
lldp	String	Enables or disables LLDP on a port. Valid values are **enable** and **disable**. Default is **disable**.	No	`<lldp>enabled</lldp>`
adminstatus	String	Shuts down or brings up a port administratively. Valid values are **up** and **down**.	No	`<adminstatus>up</adminstatus>`

- Example: GET Port and Port Channel Information API, on page 202
- Example: POST Create a Port Channel API, on page 206
- Example: PUT Add a Port to a Port Channel API, on page 207
- Example: PUT Add a Port Channel to an Existing Bridge API, on page 208
- Example: PUT Configure the LACP Mode of a Port Channel API, on page 209
- Example: PUT Configure the Bond Mode of a Port Channel API, on page 210
- Example: PUT Configure Trunks on a Port Channel API, on page 211
- Example: DELETE Remove a Port from a Port Channel API, on page 212
- Example: PUT Remove a Port Channel from a Bridge API, on page 213
- Example: DELETE Delete a Port Channel API, on page 214
- Example: GET LLDP Information API, on page 214
- Example: PUT Enable LLDP Configuration API, on page 217
- Example: GET Port Admin Status API, on page 218

• Example: PUT Configure Port Admin Status API, on page 219
• Speed, Autoneg and Duplex APIs, on page 220

Example: GET Port and Port Channel Information API

```
curl -v -k -u admin:Admin#123 -X GET https://198.51.100.11/api/operational/pnics
* About to connect() to 198.51.100.11 port 443 (#0)
*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec 05 15:26:32 2017 GMT
*  expire date: Dec 03 15:26:32 2027 GMT
*  common name: nfvis
*  issuer: CN=nfvis
* Server auth using Basic with user 'admin'
> GET /api/operational/pnics HTTP/1.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.29.0
> Host: 198.51.100.11
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Wed, 06 Dec 2017 17:45:17 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<pnics xmlns="http://www.cisco.com/nfvis/pnic" xmlns:y="http://tail-f.com/ns/rest"
xmlns:pnic="http://www.cisco.com/nfvis/pnic">
  <pnic>
    <name>eth0</name>
  </pnic>
  <pnic>
    <name>eth1</name>
  </pnic>
  <pnic>
    <name>eth2</name>
  </pnic>
  <pnic>
    <name>eth3</name>
  </pnic>
  <pnic>
    <name>eth4</name>
  </pnic>
  <pnic>
    <name>eth5</name>
  </pnic>
</pnics>
* Connection #0 to host 198.51.100.11 left intact


curl -v -k -u admin:Admin#123 -X GET https://198.51.100.11/api/operational/pnics?deep
* About to connect() to 198.51.100.11 port 443 (#0)
*   Trying 198.51.100.11...
```

```
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec 05 15:26:32 2017 GMT
*  expire date: Dec 03 15:26:32 2027 GMT
*  common name: nfvis
*  issuer: CN=nfvis
* Server auth using Basic with user 'admin'
> GET /api/operational/pnics?deep HTTP/1.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.29.0
> Host: 198.51.100.11
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Wed, 06 Dec 2017 17:47:14 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<

<pnics xmlns="http://www.cisco.com/nfvis/pnic" xmlns:y="http://tail-f.com/ns/rest"
xmlns:pnic="http://www.cisco.com/nfvis/pnic">
  <pnic>
    <name>eth0</name>
    <speed>1G</speed>
    <operational-speed>1000</operational-speed>
    <link_state>up</link_state>
    <mac_address>58:ac:78:59:ca:66</mac_address>
    <mtu>9216</mtu>
    <refcnt>0</refcnt>
    <stats>
    <receive>
      <bytes>62837229</bytes>
      <packets>496647</packets>
      <errors>0</errors>
      <dropped>0</dropped>
      <broadcast>60009</broadcast>
      <multicast>408172</multicast>
    </receive>
    <transmit>
      <bytes>517791</bytes>
      <packets>1565</packets>
      <errors>0</errors>
      <dropped>0</dropped>
      <broadcast>1462</broadcast>
      <multicast>12</multicast>
    </transmit>
  </stats>
</pnic>
<pnic>
  <name>eth1</name>
  <speed>1G</speed>
  <operational-speed>0</operational-speed>
  <link_state>down</link_state>
  <mac_address>58:ac:78:59:ca:67</mac_address>
  <mtu>9216</mtu>
  <refcnt>1</refcnt>
```

```
  <stats>
  <receive>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </receive>
  <transmit>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </transmit>
</stats>
</pnic>
<pnic>
  <name>eth2</name>
  <speed>1G</speed>
  <operational-speed>0</operational-speed>
  <link_state>down</link_state>
  <mac_address>a0:36:9f:7b:87:9c</mac_address>
  <mtu>9216</mtu>
  <refcnt>2</refcnt>
  <stats>
  <receive>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </receive>
  <transmit>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </transmit>
</stats>
</pnic>
<pnic>
  <name>eth3</name>
  <speed>1G</speed>
  <operational-speed>0</operational-speed>
  <link_state>down</link_state>
  <mac_address>a0:36:9f:7b:87:9d</mac_address>
  <mtu>9216</mtu>
  <refcnt>3</refcnt>
  <stats>
  <receive>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </receive>
  <transmit>
```

```
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </transmit>
</stats>
</pnic>
<pnic>
  <name>eth4</name>
  <speed>1G</speed>
  <operational-speed>0</operational-speed>
  <link_state>down</link_state>
  <mac_address>a0:36:9f:7b:87:9e</mac_address>
  <mtu>9216</mtu>
  <refcnt>4</refcnt>
  <stats>
  <receive>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </receive>
  <transmit>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </transmit>
</stats>
</pnic>
<pnic>
  <name>eth5</name>
  <speed>1G</speed>
  <operational-speed>0</operational-speed>
  <link_state>down</link_state>
  <mac_address>a0:36:9f:7b:87:9f</mac_address>
  <mtu>9216</mtu>
  <refcnt>5</refcnt>
  <stats>
  <receive>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </receive>
  <transmit>
    <bytes>0</bytes>
    <packets>0</packets>
    <errors>0</errors>
    <dropped>0</dropped>
    <broadcast>0</broadcast>
    <multicast>0</multicast>
  </transmit>
</stats>
</pnic>
```

```
</pnics>
* Connection #0 to host 198.51.100.11 left intact
```

Example: POST Create a Port Channel API

```
curl -k -v -u admin:Admin#123 -X POST -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics
 --data '<pnic><name>pc</name><type>port_channel</type></pnic>'
* About to connect() to 198.51.100.11 port 443 (#0)
*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec 05 15:26:32 2017 GMT
*  expire date: Dec 03 15:26:32 2027 GMT
*  common name: nfvis
*  issuer: CN=nfvis
* Server auth using Basic with user 'admin'
> POST /api/config/pnics HTTP/1.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.29.0
> Host: 198.51.100.11
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 53
>
* upload completely sent off: 53 out of 53 bytes
< HTTP/1.1 201 Created
< Server: nginx
< Date: Wed, 06 Dec 2017 17:48:44 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://198.51.100.11/api/config/pnics/pnic/pc
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 17:48:44 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-582524-631443
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

```
curl -k -v -u  admin:Admin#123 -X POST  -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/
pnics --data
'<pnic><name>pc</name><type>port_channel</type><lacp_type>active</lacp_type><bond_mode>balance-tcp</bond_mode>
<trunks>10,20</trunks></pnic>'

Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
```

```
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/pnics HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 138
>
* upload completely sent off: 138 out of 138 bytes
< HTTP/1.1 201 Created
< Server: nginx
< Date: Wed, 06 Dec 2017 17:59:50 GMT
< Content-Type: text/html
< Content-Length: 0
< Location: https://198.51.100.11/api/config/pnics/pnic/pc
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 17:59:50 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-583190-180622
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Add a Port to a Port Channel API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/eth1/member_of --data '<member_of>pc</member_of>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
```

```
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/eth1/member_of HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 25
>
* upload completely sent off: 25 out of 25 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:12:42 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:12:42 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-583962-225643
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Add a Port Channel to an Existing Bridge API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/bridges/bridge/test-br --data
'<bridge><name>test-br</name><port><name>pc</name></port></bridge>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
```

```
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/bridges/bridge/test-br
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 66
>
* upload completely sent off: 66 out of 66 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:32:49 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:32:48 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-734699-373894
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Configure the LACP Mode of a Port Channel API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/pc/lacp_type --data
'<lacp_type>active</lacp_type>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/pc/lacp_type HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
```

```
> Content-type:application/vnd.yang.data+xml
> Content-Length: 29
>
* upload completely sent off: 29 out of 29 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:40:30 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:32:48 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-585168-972196
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Configure the Bond Mode of a Port Channel API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/pc/bond_mode --data
'<bond_mode>balance-tcp</bond_mode>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/pc/bond_mode HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 34
>
* upload completely sent off: 34 out of 34 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:41:11 GMT
```

```
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:32:48 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-585168-972196
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Configure Trunks on a Port Channel API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/pc/trunks --data '<trunks>10,20</trunks>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/pc/trunks HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 22
>
* upload completely sent off: 22 out of 22 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:54:53 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:32:48 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-585168-972196
< Pragma: no-cache
```

```
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: DELETE Remove a Port from a Port Channel API

```
curl -k -v -u  admin:Admin#123 -X DELETE -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/eth1/member_of --data '<member_of>pc</member_of>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/pnics/pnic/eth1/member_of HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 25
>
* upload completely sent off: 25 out of 25 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 18:55:32 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 18:55:32 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-586532-509745
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: PUT Remove a Port Channel from a Bridge API

```
curl -k -v -u  admin:Admin#123 -X PUT -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/bridges/bridge/testbridge --data
'<bridge><name>test-br</name></bridge>'


*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/bridges/bridge/test-br HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 37
>
* upload completely sent off: 37 out of 37 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 19:09:06 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 19:09:05 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-587345-710932
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: DELETE Delete a Port Channel API

```
curl -k -v -u admin:Admin#123 -X DELETE -H Content-type:application/vnd.yang.data+xml
https://198.51.100.11/api/config/pnics/pnic/pc

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> DELETE /api/config/pnics/pnic/pc HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
>
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Wed, 06 Dec 2017 19:11:24 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 19:11:24 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-587484-283024
< Pragma: no-cache
<
* Connection #0 to host 198.51.100.11 left intact
```

Example: GET LLDP Information API

```
curl -k -v -u admin:Cisco123# -X GET -H Content-type:application/vnd.yang.data+xml
'https://172.19.162.231/api/operational/lldp?deep'
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 172.19.162.231...
```

```
* Connected to 172.19.162.231 (172.19.162.231) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=nfvis
*  start date: Jul  2 00:53:36 2017 GMT
*  expire date: Jun 30 00:53:36 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/lldp?deep HTTP/1.1
> Host: 172.19.162.231
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
>< HTTP/1.1 200 OK
< Server: nginx
< Date: Mon, 03 Jul 2017 04:30:47 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
<


<lldp
 xmlns="http://www.cisco.com/nfvis/pnic"
 xmlns:y="http://tail-f.com/ns/rest"
 xmlns:pnic="http://www.cisco.com/nfvis/pnic">
 <neighbors>
  <name>eth0</name>
  <device_id>Switch1623</device_id>
  <holdtime>120</holdtime>
  <caps>Bridge, Router</caps>
  <platform>Cisco IOS Software, Catalyst L3 Switch Software (CAT3K_CAA-UNIVERSALK9-M),
Version 15.0(1)EX3, RELEASE SOFTWARE (fc2)</platform>
  <portid>Ifname: Gi1/0/4</portid>
  <description>GigabitEthernet1/0/4</description>
 </neighbors>
 <neighbors>
  <name>eth1</name>
  <device_id>None</device_id>
  <holdtime>0</holdtime>
  <caps>None</caps>
  <platform>None</platform>
  <portid>None</portid>
  <description>None</description>
```

```
</neighbors>
<neighbors>
 <name>eth2</name>
 <device_id>None</device_id>
 <holdtime>0</holdtime>
 <caps>None</caps>
 <platform>None</platform>
 <portid>None</portid>
 <description>None</description>
</neighbors>
<neighbors>
 <name>eth3</name>
 <device_id>None</device_id>
 <holdtime>0</holdtime>
 <caps>None</caps>
 <platform>None</platform>
 <portid>None</portid>
 <description>None</description>
</neighbors>
<neighbors>
 <name>eth4</name>
 <device_id>None</device_id>
 <holdtime>0</holdtime>
 <caps>None</caps>
 <platform>None</platform>
 <portid>None</portid>
 <description>None</description>
</neighbors>
<neighbors>
 <name>eth5</name>
 <device_id>None</device_id>
 <holdtime>0</holdtime>
 <caps>None</caps>
 <platform>None</platform>
 <portid>None</portid>
 <description>None</description>
</neighbors>
<stats>
 <name>eth0</name>
 <tx_frames>10</tx_frames>
 <discard_rx>0</discard_rx>
 <error_rx>0</error_rx>
 <rx_frames>19589</rx_frames>
 <discarded_tlvs>0</discarded_tlvs>
 <unrec_tlvs>0</unrec_tlvs>
 <ageouts>0</ageouts>
</stats>
<stats>
 <name>eth1</name>
 <tx_frames>0</tx_frames>
 <discard_rx>0</discard_rx>
 <error_rx>0</error_rx>
 <rx_frames>0</rx_frames>
 <discarded_tlvs>0</discarded_tlvs>
 <unrec_tlvs>0</unrec_tlvs>
 <ageouts>0</ageouts>
</stats>
<stats>
 <name>eth2</name>
 <tx_frames>0</tx_frames>
 <discard_rx>0</discard_rx>
 <error_rx>0</error_rx>
 <rx_frames>0</rx_frames>
 <discarded_tlvs>0</discarded_tlvs>
```

```
  <unrec_tlvs>0</unrec_tlvs>
  <ageouts>0</ageouts>
 </stats>
 <stats>
  <name>eth3</name>
  <tx_frames>0</tx_frames>
  <discard_rx>0</discard_rx>
  <error_rx>0</error_rx>
  <rx_frames>0</rx_frames>
  <discarded_tlvs>0</discarded_tlvs>
  <unrec_tlvs>0</unrec_tlvs>
  <ageouts>0</ageouts>
 </stats>
 <stats>
  <name>eth4</name>
  <tx_frames>0</tx_frames>
  <discard_rx>0</discard_rx>
  <error_rx>0</error_rx>
  <rx_frames>0</rx_frames>
  <discarded_tlvs>0</discarded_tlvs>
  <unrec_tlvs>0</unrec_tlvs>
  <ageouts>0</ageouts>
 </stats>
 <stats>
  <name>eth5</name>
  <tx_frames>0</tx_frames>
  <discard_rx>0</discard_rx>
  <error_rx>0</error_rx>
  <rx_frames>0</rx_frames>
  <discarded_tlvs>0</discarded_tlvs>
  <unrec_tlvs>0</unrec_tlvs>
  <ageouts>0</ageouts>
 </stats>
</lldp>
* Connection #0 to host 172.19.162.231 left intact
```

Example: PUT Enable LLDP Configuration API

```
curl -v -k -u admin:Admin#123 -X PUT  -H Content-type:application/vnd.yang.data+xml
Content-type:application/vnd.yang.data+xml
'https://198.51.100.1/api/config/pnics/pnic/eth0/lldp --data
'<lldp>enabled</lldp>'

*   Trying 198.51.100.1...
* Connected to 198.51.100.1 (198.51.100.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
```

```
* Server certificate:
*  subject: CN=nfvis
*  start date: Jul  2 00:53:36 2017 GMT
*  expire date: Jun 30 00:53:36 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/eth0/lldp HTTP/1.1
> Host: 198.51.100.1
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 20
>
* upload completely sent off: 20 out of 20 bytes
< HTTP/1.1 204 No Content
< Server: nginx
< Date: Mon, 03 Jul 2017 04:09:38 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Sun, 02 Jul 2017 01:30:26 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1498-959026-423491
< Pragma: no-cache
* Connection #0 to host 198.51.100.1 left intact
```

Example: GET Port Admin Status API

```
curl -k -v -u admin:Cisco123# -X GET
https://198.51.100.1/api/config/pnics/pnic/eth5/adminstatus
*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/config/pnics/pnic/eth5/adminstatus HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
```

```
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Wed, 06 Dec 2017 19:15:23 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 19:14:09 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-587649-439226
< Pragma: no-cache
<

<adminstatus xmlns="http://www.cisco.com/nfvis/pnic" xmlns:y="http://tail-f.com/ns/rest"
xmlns:pnic="http://www.cisco.com/nfvis/
pnic">up</adminstatus>
* Connection #0 to host 198.51.100.11 left intact
```

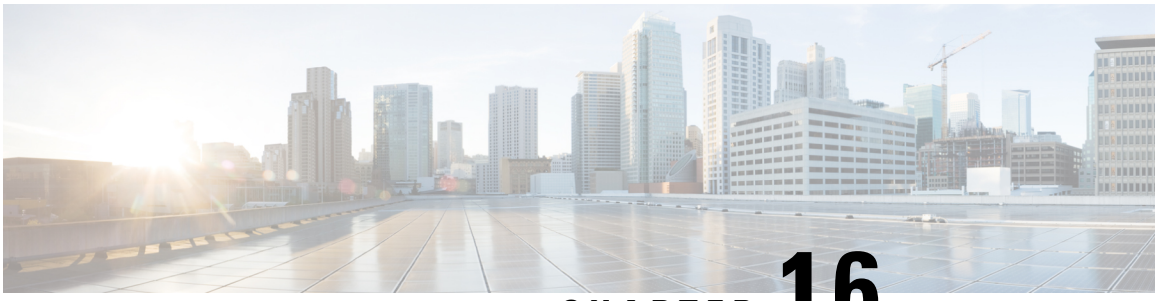
Example: PUT Configure Port Admin Status API

```
curl -v -k -u admin:Admin#123 -X PUT  -H Content-type:application/vnd.yang.data+xml
Content-type:application/vnd.yang.data+xml
'https://198.51.100.11/api/config/pnics/pnic/eth5/adminstatus --data
'<adminstatus>up</adminstatus>'

*   Trying 198.51.100.11...
* Connected to 198.51.100.11 (198.51.100.11) port 443 (#1)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / DHE-RSA-AES256-SHA
* Server certificate:
*  subject: CN=nfvis
*  start date: Dec  5 15:26:32 2017 GMT
*  expire date: Dec  3 15:26:32 2027 GMT
*  issuer: CN=nfvis
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> PUT /api/config/pnics/pnic/eth5/adminstatus HTTP/1.1
> Host: 198.51.100.11
> Authorization: Basic YWRtaW46Q2lzY28xMjMj
> User-Agent: curl/7.50.1
> Accept: */*
> Content-type:application/vnd.yang.data+xml
> Content-Length: 29
>
* upload completely sent off: 29 out of 29 bytes
< HTTP/1.1 204 No Content
```

```
< Server: nginx
< Date: Wed, 06 Dec 2017 19:14:09 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 06 Dec 2017 19:14:09 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1512-587649-439226
< Pragma: no-cache
<
* Connection #1 to host 198.51.100.11 left intact
```

Speed, Autoneg and Duplex APIs

Action	Method	Payload Required	API
To configure speed	PUT	Yes	/api/config/pnics/pnic/GE0-0/speed
To configure duplex	PUT	Yes	/api/config/pnics/pnic/GE0-0/duplex
To configure speed and duplex	PUT	Yes	/api/config/pnics/pnic/GE0-0/
To get the perational speed	GET	No	/api/config/pnics/pnic/GE0-0/operational-speed
To get the operational duplex	GET	No	/api/config/pnics/pnic/GE0-0/operational-duplex
To get the operational autoneg	GET	No	/api/config/pnics/pnic/GE0-0/operational-duplex



CHAPTER 16

Port Security APIs

Example: Max Number of MAC Addresses

```
https://209.165.201.1/api/running/switch/interface/gigabitEthernet


Payload:
<gigabitEthernet>
<name>1/0</name>
<port-security>
<max>2</max>
</port-security>
</gigabitEthernet>
```

Example: Violation Discard

```
https://209.165.201.1/api/running/switch/interface/gigabitEthernet


Payload:

<gigabitEthernet>
<name>1/0</name>
<port-security>
<violation>discard</violation>
</port-security>
</gigabitEthernet>
```

Example: Enable Port Security

```
https://209.165.201.1/api/running/switch/interface/gigabitEthernet


Payload:
<gigabitEthernet>
<name>1/0</name>
<port-security>
 <enable/>
</port-security>
</gigabitEthernet>
```

Example: Secure Static MAC

```
https://209.165.201.1/api/running/switch/mac


Payload:
     <mac>
        <address-table>
          <static>
            <mac-entries>
              <mac-addr>18:65:90:cb:e6:10</mac-addr>
              <vlan>1</vlan>
              <interface>
                <gigabitEthernet>1/0</gigabitEthernet>
              </interface>
              <type>secure</type>
            </mac-entries>
          </static>
        </address-table>
      </mac>
```

Example: show switch interface port-security

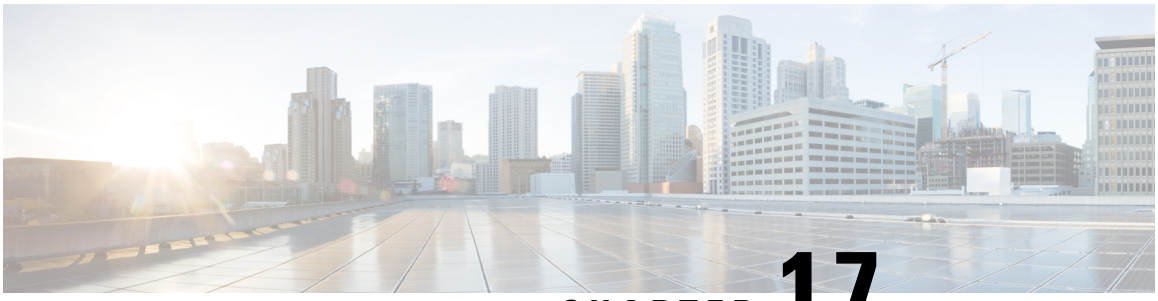
```
GET https://209.165.201.1/api/operational/switch/interface/port-security?deep
```

Example: no port security enable

```
DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/"1/0"/port-security/enable
```

Example: no mac address-table static

```
DELETE https://209.165.201.1/api/running/switch/mac/address-table/static/mac-entries
```



CHAPTER 17

Secure Overlay APIs

Table 101: Secure Overlay APIs

Action	Method	Payload Required	API
To create secure overlay configuration	POST	Yes	/api/config/secure-overlays
To get secure overlay configuration	GET	No	/api/config/secure-overlays?deep
To delete secure overlay configuration	DELETE	No	/api/config/secure-overlays
To get secure overlay state data	GET	No	/api/operational/secure-overlays

Example for secure overlay payload

```
<secure-overlay>
 <name>mgmthub</name>
 <local-bridge>wan-br</local-bridge>
 <local-system-ip-addr>34.34.34.4</local-system-ip-addr>
 <remote-interface-ip-addr>10.85.189.36</remote-interface-ip-addr>
 <remote-system-ip-addr>10.19.18.251</remote-system-ip-addr>
 <remote-id>mgmt-hub.cloudvpn.com</remote-id>
 <psk>
  <local-psk>Cisco1234Admin</local-psk>
  <remote-psk>Cisco1234Admin</remote-psk>
 </psk>
</secure-overlay>
```

Table 102: Description for Secure Overlay Payloads

Property	Type	Description	Mandatory
name	String	Name of secure overlay connection.	Yes
description	String	Description of secure overlay connection	No

Property	Type	Description	Mandatory
local-bridge	String	Local bridge name for overlay (default wan-br)	No
local-system-ip-addr	String	Local overlay system IPv4 address.	No
local-system-ip-subnet	String	Local overlay subnet. H.H.H.H/N Default is /32	No
remote-interface-ip-addr	String	Remote interface IPv4 address; FQDN	Yes
remote-system-ip-addr	List of strings max element 2	Remote system IPv4 address	No
remote-system-ip-subnet	List of strings max element 2	List of remote system IPv4 subnets Default is /24 for each remote system IP address.	No
remote-id	List of strings max element 2	Remote id for overlay - IP, FQDN, Distinguished Name, or email domain (default remote-interface-ip-addr)	No
ike-cipher	String	IKE algorithms. Possible values: aes128-sha1-modp1536, aes256-sha512-modp2048, aes256-sha512-modp4096 Default: aes128-sha1-modp1536	No
esp-cipher	String	ESP algorithms. Possible values: aes128-sha1, aes256-sha512, aes256-sha512-modp2048, aes256-sha512-modp4096, es128-sha1-modp1536, aes256-sha1-modp2048, and aes256-sha256-modp2048 Default: aes128-sha1	No

Property	Type	Description	Mandatory
psk	String	Pre-shared-key for authentication	No
psk local-psk	String	Local pre-shared-key	Yes if PSK
psk remote-psk	String	Remote pre-shared-key	Yes if PSK
local-id	String	Local id for overlay - IP, FQDN, or email domain (default local-bridge IP address)	No
dual-local-bridge	String	Secondary local bridge name for overlay in case of dual WAN interface	No
local-system-ip-bridge	String	Internal management network bridge used for private tunnel endpoint. If configured, must be int-mgmt-net.	No
eap	String	Extensible Authentication Protocol for authentication	No
cacert	String	EAP CA Server certificate location	Yes if EAP
method	String	EAP hash method Possible values: eap-md5 Default: eap-md5	No
username	String	EAP local identity	Yes if EAP
password	String	EAP local identity password	Yes if EAP
bgp-neighbor-name	String	Name tag corresponding to BGP neighbor used over secure overlay.	No

Note When you configure a list of two remote system IP addresses, subnets, and remote IDs, each list must have consistent order for the remote system configurations provided.

Example: POST Secure Overlay APIs

```
curl -k -v -u "admin:123#Admin" -H Accept:application/vnd.yang.data+xml -H
```

```
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/secure-overlays -d '
<secure-overlay>
 <name>mgmthub</name>
 <local-bridge>wan-br</local-bridge>
 <local-system-ip-addr>10.0.0.1</local-system-ip-addr>
 <remote-interface-ip-addr>10.0.0.1</remote-interface-ip-addr>
 <remote-system-ip-addr>10.0.0.2</remote-system-ip-addr>
 <remote-id>mgmt-hub.cloudvpn.com</remote-id>
 <psk>
  <local-psk>1234Admin</local-psk>
  <remote-psk>1234Admin</remote-psk>
 </psk>
</secure-overlay>'
```

Example: POST create secure overlaywith int-mgmt-net ip as local system ip address

```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/secure-overlays -d '
<secure-overlay>
<name>mgmthub</name>
 <local-bridge>wan-br</local-bridge>
 <local-system-ip-addr>10.0.0.4</local-system-ip-addr>
 <local-system-ip-bridge>int-mgmt-net</local-system-ip-bridge>
 <remote-interface-ip-addr>10.0.0.1</remote-interface-ip-addr>
 <remote-system-ip-addr>10.0.0.2</remote-system-ip-addr>
 <remote-id>mgmt-hub.cloudvpn.com</remote-id>
 <psk>
  <local-psk>Cisco1234Admin</local-psk>
  <remote-psk>Cisco1234Admin</remote-psk>
 </psk>
</secure-overlay>'
```

Example: GET Secure Overlay APIs

```
curl -k -v -u "admin:123#Admin" -X GET "https://209.165.201.1/api/config/secure-overlays?deep"
```

Example: GET Secure Overlay APIs

```
curl -k -v -u "admin:123#Admin" -X GET
"https://209.165.201.1/api/operational/secure-overlays?deep"
```

Example: DELETE Secure Overlay APIs

```
curl -k -v -u "admin:123#Admin" -X DELETE "https://209.165.201.1/api/config/secure-overlays"
```

- Single IP Configuration APIs, on page 226

Single IP Configuration APIs

Table 103: Secure Overlay APIs

Action	Method	Payload Required	API
To create single IP configuration	POST	Yes	/api/config/single-ip-mode

To get single IP configuration	GET	No	/api/config/single-ip-mode
To delete single IP configuration	DELETE	No	/api/config/single-ip-mode
To get single IP configuration state information	GET	No	/api/operational/single-ip-mode

Example for single IP configuration payload

```
<single-ip-mode>
 <vm-name>ROUTER.ROUTER</vm-name>
</single-ip-mode>"
```

Table 104: Description for Single IP Payload

Property	Type	Description	Mandatory
vm-name	String	Name of VM taking the public IP.	Yes

Example: POST Single IP configuration APIs

```
curl -k -v -u "admin:123#Admin" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X PUT
https://209.165.201.1/api/config/single-ip-mode -d "
<single-ip-mode>
 <vm-name>ROUTER.ROUTER</vm-name>
</single-ip-mode>"
```

Example: GET Single IP configuration APIs

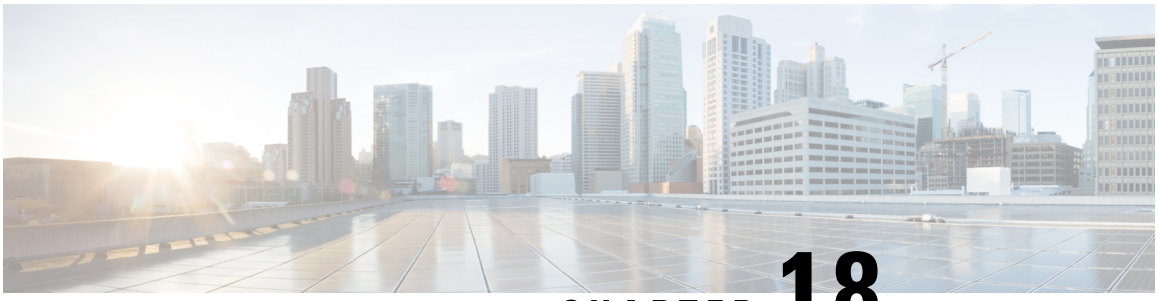
```
curl -k -v -u "admin:123#Admin" -X GET "https://209.165.201.1/api/config/single-ip-mode"
```

Example: GET Single IP configuration APIs

```
curl -k -v -u "admin:123#Admin" -X GET "https://209.165.201.1/api/operational/single-ip-mode"
```

Example: DELETE Single IP configuration APIs

```
curl -k -v -u "admin:123#Admin" -X DELETE "https://209.165.201.1/api/config/single-ip-mode"
```



CHAPTER 18

BGP Support APIs

Table 105: BGP APIs

Action	Method	Payload Required	API
To configure BGP	POST	Yes	/api/config/routers/router/bgp/<as-number>
To update BGP	PUT	Yes	/api/config/routers/router/bgp/<as-number>
To delete BGP	DELETE	No	/api/config/routers/router/bgp/<as-number>
To get BGP configuration	GET	No	/api/config/routers/router/bgp?deep

Example for BGP payload

```
<bgp xmlns="http://www.cisco.com/nfvis/router">
  <as>200</as>
  <router-id>10.20.0.1</router-id>
     <address-family>
            <protocol>ipv4</protocol>
            <routing-scheme>unicast</routing-scheme>
            <network>
                <network>10.20.0.0</network>
                <mask>255.255.255.0</mask>
            </network>
        <neighbor>
            <remote-ip>bgp-neighbor</remote-ip>
            <activate/>
        </neighbor>
      </address-family>
  <neighbor xmlns="http://www.cisco.com/nfvis/router">
    <remote-ip>bgp-neighbor</remote-ip>
    <remote-as>65000</remote-as>
  </neighbor>
</bgp>
```

Table 106: Description for BGP Payloads

Property	Type	Description	Mandatory
remote-ip	String	Neighbor IPv4 address or bgp-neighbor-name matching secure-overlay configuration	Yes
as	String	Local autonomous system number	Yes
remote-as	String	Remote autonomous system number	Yes
router-id	String	Local router id IP address	No
network	String	IP address of announced network	No
mask	String	Network mask (e.g. 255.255.255.0)	No
routing-scheme	String	Routing scheme used for announcing networks. Only unicast supported	No
activate		Activate neighbor under address family	No
protocol	String	Protocol to be used for the announcements over BGP. Only ipv4 supported	No

Example: POST create BGP local session

```
curl -k -v -u "admin:admin" -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -X POST
https://209.165.201.1/api/config/routers/router/bgp/200 -d '
<bgp
 xmlns="http://www.cisco.com/nfvis/router">
 <as>200</as>
 <router-id>10.20.0.1</router-id>
 <neighbor
  xmlns="http://www.cisco.com/nfvis/router">
  <remote-ip>90.90.90.1</remote-ip>
  <remote-as>65000</remote-as>
 </neighbor>
</bgp>'
```

Example: PUT update BGP

```
curl -k -v -u "admin:admin" -X GET "https://209.165.201.1/api/config/routers/router/bgp?deep"
```

Example: DELETE BGP configuration

```
curl -k -v -u "admin:admin" -X DELETE "https://209.165.201.1/api/config/routers/router/bgp"
```

• BGP over MPLS APIs, on page 231

BGP over MPLS APIs

Table 107: BGP APIs

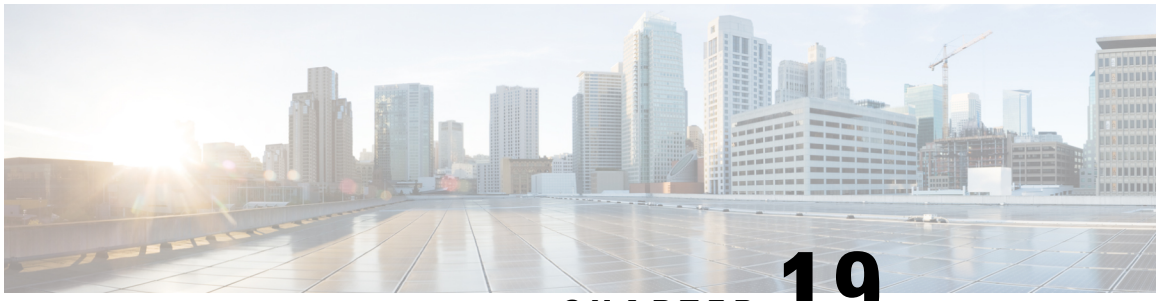
Action	Method	Payload Required	API
Add subnet to routes announced in address-family	POST	Yes	/api/config/router/bgp/<as_number>/address-family/network
Activate neighbor within address-family to announce routes specified	POST	Yes	/api/config/router/bgp/<as_number>/address-family/neighbor
Retrieve BGP status information	GET	No	/api/operational/bgp/<protocol>,<routing-scheme>/

Example for BGP payload

```
<router:bgp>
  <as>200</as>
  <address-family>
   <protocol>ipv4</protocol>
   <routing-scheme>unicast</routing-scheme>
   <network>
    <network>10.20.0.0</network>
    <mask>255.255.255.0</mask>
   </network>
   <neighbor>
    <remote-ip>90.90.90.1</remote-ip>
    <activate />
   </neighbor>
  </address-family>
  <neighbor>
   <remote-ip>90.90.90.1</remote-ip>
   <remote-as>65000</remote-as>
   <description>This is the CSR Headend</description>
  </neighbor>
 </router:bgp>
```

PART II

Switch Related APIs



CHAPTER 19

DOT1x APIs

Table 108: DOT1x APIs

Action	Method	Payload Required	API
To view the dot1x summary	GET	No	/api/operational/switch/dot1x/summary
To view the dot1x configuration	GET	No	/api/config/switch/dot1x

Example: GET DOT1x APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/operational/switch/dot1x/summary
```

Example: GET DOT1x APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/config/switch/dot1x
```


DOT1x guest-vlan Timeout Value APIs

Table 109: DOT1x guest-vlan Timeout Value APIs

Action	Method	Payload Required	API
To enable unauthorized users on the access interface to the guest VLAN	POST	Yes	/api/config/switch/dot1x/guest-vlan

Action	Method	Payload Required	API
To set the time delay between enabling Dot1X and adding a port to the guest VLAN	PUT	Yes	/api/config/switch/dot1x/guest-vlan/timeout
To restore the default configuration	DELETE	No	/api/config/switch/dot1x/guest-vlan/timeout
To get the VLAN timeout value	GET	No	/api/config/switch/dot1x/guest-vlan/timeout

Example for DOT1x guest-vlan Timeout Value APIs Payload

```
<timeout>30</timeout>
```

Table 110: Description for DOT1x guest-vlan Timeout Value APIs Payload

Property	Type	Description	Mandatory/Default Value
timeout	integer	Specifies the time delay in seconds between enabling dot1X and adding the port to the guest VLAN. (Range: 30–180)	Yes

Example: POST DOT1x guest-vlan Timeout Value APIs

```
curl -k -u admin:admin -d "<timeout>30</timeout>" -X POST
https://209.165.201.1/api/config/switch/dot1x/guest-vlan -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PUT DOT1x guest-vlan Timeout Value APIs

```
curl -k -u admin:admin -d "<timeout>40</timeout>" -X PUT
https://209.165.201.1/api/config/switch/dot1x/guest-vlan/timeout -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE DOT1x guest-vlan Timeout Value APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/config/switch/dot1x/guest-vlan/timeout
```

Example: GET DOT1x guest-vlan Timeout Value APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/config/switch/dot1x/guest-vlan/timeout
```

DOT1x Default authentication APIs

Table 111: DOT1x Default authentication APIs

Action	Method	Payload Required	API
To enable authentication methods on a port	POST	Yes	/api/config/switch/dot1x/authentication
To restore the default configuration	GET	No	/api/config/switch/dot1x/authentication/default
To delete the authentication configuration	DELETE	No	/api/config/switch/dot1x/authentication

Example for DOT1x Default authentication APIs Payload

```
<default>radius</default>
```

Example: POST DOT1x Default authentication APIs

```
curl -k -u admin:admin -d "<default>radius</default>" -X POST
https://209.165.201.1/api/config/switch/dot1x/authentication -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: GET DOT1x Default authentication APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/config/switch/dot1x/authentication/default
```

Example: DELETE DOT1x Default authentication APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/config/switch/dot1x/authentication
```

DOT1x System Authentication Control APIs

Table 112: DOT1x System Authentication Control APIs

Action	Method	Payload Required	API
To enable dot1x globally	POST	Yes	/api/config/switch/dot1x
To get the configuration for system authentication control	GET	No	/api/config/switch/dot1x/system-auth-control
To restore default configuration	DELETE	No	/api/config/switch/dot1x/system-auth-control

Example for DOT1x System Authentication Control APIs Payload

```
<system-auth-control></system-auth-control>
```

Example: POST DOT1x System Authentication Control APIs

```
curl -k -u admin:admin -d "<system-auth-control></system-auth-control>" -X POST
https://209.165.201.1/api/config/switch/dot1x -H "Content-Type: application/vnd.yang.data+xml
```

Example: GET DOT1x System Authentication Control APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/config/switch/dot1x/system-auth-control
```

Example: DELETE DOT1x System Authentication Control APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/config/switch/dot1x/system-auth-control
```

RADIUS Source Interface Address APIs

Table 113: RADIUS Source Interface Address APIs

Action	Method	Payload Required	API
To enable RADIUS-based VLAN assignment	POST	Yes	/api/config/switch/ip/radius/source-interface
To get the RADIUS-based VLAN configurations	GET	No	/api/config/switch/ip/radius/source-interface/vlan
To replace RADIUS-based VLAN	PUT	Yes	/api/config/switch/ip/radius/source-interface/vlan
To disable RADIUS-based VLAN assignment	DELETE	No	/api/config/switch/ip/radius/source-interface/vlan

Example for RADIUS Source Interface Address APIs Payload

```
<vlan>505</vlan>
```

Example: POST RADIUS Source Interface Address APIs

```
curl -k -u admin:admin -d "<vlan>505</vlan>" -X POST
https://209.165.201.1/api/config/switch/ip/radius/source-interface -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: GET RADIUS Source Interface Address APIs

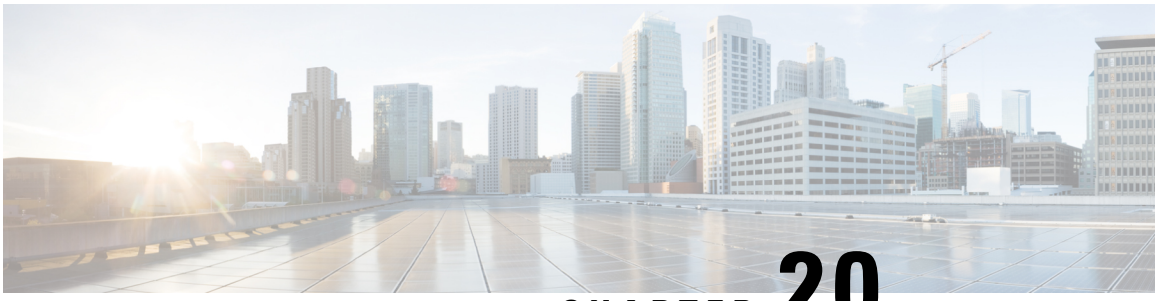
```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/config/switch/ip/radius/source-interface/vlan
```

Example: PUT RADIUS Source Interface Address APIs

```
curl -k -u admin:admin -d "<vlan>506</vlan>" -X PUT
https://209.165.201.1/api/config/switch/ip/radius/source-interface/vlan -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE RADIUS Source Interface Address APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/config/switch/ip/radius/source-interface/vlan
```



CHAPTER 20

IP Gateway APIs

Table 114: IP Gateway APIs

Action	Method	Payload Required	API
To define a default gateway	PUT	Yes	/api/running/switch/ip/default-gateway
To restore the default configuration	DELETE	No	/api/running/switch/ip/default-gateway
To show the default gateway	GET	No	api/running/switch/ip/default-gateway

Example for IP Gateway APIs Payload

```
<default-gateway><gateway>169.254.1.3</gateway></default-gateway>
```

Table 115: Description for IP Gateway APIs Payload

Property	Type	Description	Mandatory/Default Value
gateway	String	Specifies the default gateway IP address	Yes

Example: PUT IP Gateway APIs

```
curl -k -u admin:admin -d "<default-gateway><gateway>169.254.1.3</gateway></default-gateway>"
 -X PUT https://209.165.201.1/api/running/switch/ip/default-gateway -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE IP Gateway APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/ip/default-gateway
```

Example: GET IP Gateway APIs

```
curl -k -u admin:admin -X GET "https://209.165.201.1/api/running/switch/ip/default-gateway"
```

• IP Route APIs, on page 242

IP Route APIs

Table 116: IP Route APIs

Action	Method	Payload Required	API
To enable static routes	PUT	Yes	/api/running/switch/ip/routing
To add a static route	PUT	Yes	/api/running/switch/ip/route
To remove a static route	DELETE	No	/api/running/switch/ip/route/ip-route-forwarding-list

Example for IP Route APIs Payload

```
<routing></routing>
```

Example for IP Route APIs Payload

```
<route><ip-route-forwarding-list><prefix>2.2.2.2</prefix><mask>255.255.255.255</mask><forwarding-address>5.5.5.1</forwarding-address></ip-route-forwarding-list></route>
```

Example: PUT IP Route APIs

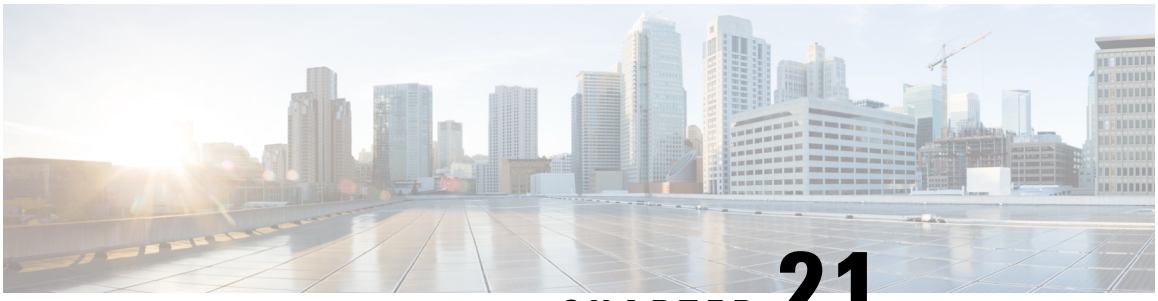
```
curl -k -u admin:admin -d "<routing></routing>" -X PUT
https://209.165.201.1/api/running/switch/ip/routing -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PUT IP Route APIs

```
curl -k -u admin:admin -d
"<route><ip-route-forwarding-list><prefix>2.2.2.2</prefix><mask>255.255.255.255</mask><forwarding-address>5.5.5.1</forwarding-address></ip-route-forwarding-list></route>"
 -X PUT https://209.165.201.1/api/running/switch/ip/route -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE IP Route APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/ip/route/ip-route-forwarding-list/2.2.2.2,255.255.255.255,5.5.5.1
```



CHAPTER 21

Spanning-Tree All or Individual Elements APIs

Table 117: Spanning-Tree Individual Elements APIs

Action	Method	Payload Required	API
To get the spanning tree configuration	GET	No	/api/running/switch/spanning-tree?deep
To define Bridge Protocol Data Unit (BPDU) handling when the spanning tree is disabled globally or on a single interface	GET	No	/api/running/switch/spanning-tree?select=bpdu
To enable the spanning-tree functionality	GET	No	/api/running/switch/spanning-tree?select=enable
To configure the spanning-tree bridge forward time, which is the amount of time a port remains in the listening and learning states before entering the forwarding state.	GET	No	/api/running/switch/spanning-tree?select=forward-time
To select the Spanning Tree Protocol (STP) protocol	GET	No	/api/running/switch/spanning-tree?select=mode
To configure the number of times Hello messages of the device is broadcasted to other devices.	GET	No	/api/running/switch/spanning-tree?select=hello-time
To configure the STP maximum age	GET	No	/api/running/switch/spanning-tree?select=max-age

Action	Method	Payload Required	API
To shutdown an interface if it receives a loopback BPDU	GET	No	/api/running/switch/spanning-tree?select=loopback-guard
To configure the path cost for MST calculations.	GET	No	/api/running/switch/spanning-tree/pathcost?deep
To set the default path cost method.	GET	No	/api/running/switch/spanning-tree/pathcost?select=method
To configure the device STP priority, which is used to determine which bridge is selected as the root bridge.	GET	No	/api/running/switch/spanning-tree?select=priority

Example: GET Spanning-Tree APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/running/switch/spanning-tree?deep
```

Example: GET Spanning-Tree bpdu APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=bpdu
```

Example: GET Spanning-Tree enable APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=enable
```

Example: GET Spanning-Tree forward-time APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=forward-time
```

Example: GET Spanning-Tree mode APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=mode
```

Example: GET Spanning-Tree hello-time APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=hello-time
```

Example: GET Spanning-Tree max-age APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=max-age
```

Example: GET Spanning-Tree loopback-guard APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=loopback-guard
```

Example: GET Spanning-Tree pathcost APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree/pathcost?deep
```

Example: GET Spanning-Tree pathcost method APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree/pathcost?select=method
```

Example: GET Spanning-Tree priority APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/spanning-tree?select=priority
```


- Create Spanning-Tree APIs, on page 245
- Modify Spanning-Tree APIs, on page 247
- Delete Spanning-Tree APIs, on page 248

Create Spanning-Tree APIs

Table 118: Create Spanning-Tree APIs

Action	Method	Payload Required	API
To create the spanning-tree elements	PATCH	Yes	/api/running/switch/spanning-tree

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><bpdu>filtering</bpdu></spanning-tree>
```

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><forward-time>18</forward-time></spanning-tree>
```

Table 119: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
forward-time	Integer	Specifies the spanning-tree forward time in seconds. (Range: 4–30)	Yes

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><mode>rstp</mode></spanning-tree>
```

Table 120: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
mode	String	Specifies the STP, RSTP or MSTP mode	Yes

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><hello-time>6</hello-time></spanning-tree>
```

Table 121: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
hello-time	Integer	Specifies the spanning-tree Hello time in seconds. (Range: 1–10)	Yes

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><max-age>24</max-age></spanning-tree>
```

Table 122: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
max-age	Integer	Specifies the spanning-tree bridge maximum age in seconds. (Range:6–40)	Yes

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><loopback-guard></loopback-guard></spanning-tree>
```

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><method>short</method></spanning-tree>
```

Table 123: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
method	String	Specifies the default port path costs	Yes

Example for Create Spanning-Tree APIs Payload

```
<spanning-tree><priority>8192</priority></spanning-tree>
```

Table 124: Description for Create Spanning-Tree APIs Payload

Property	Type	Description	Mandatory/Default Value
priority	String	Specifies the device priority for the specified spanning-tree instance.	yes

Example: PATCH Create Spanning-Tree bpdu APIs

```
curl -k -u admin:admin -d "<spanning-tree><bpdu>filtering</bpdu></spanning-tree>" -X PATCH
 https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree forward-time APIs

```
curl -k -u admin:admin -d "<spanning-tree><forward-time>18</forward-time></spanning-tree>"
 -X PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree mode APIs

```
curl -k -u admin:admin -d "<spanning-tree><mode>rstp</mode></spanning-tree>" -X PATCH
https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree hello-time APIs

```
curl -k -u admin:admin -d "<spanning-tree><hello-time>6</hello-time></spanning-tree>" -X
PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree max-age APIs

```
curl -k -u admin:admin -d "<spanning-tree><max-age>24</max-age></spanning-tree>" -X PATCH
https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree loopback-guard APIs

```
curl -k -u admin:admin -d "<spanning-tree><loopback-guard></loopback-guard></spanning-tree>"
 -X PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree method APIs

```
curl -k -u admin:admin -d "<spanning-tree><method>short</method></spanning-tree>" -X PATCH
 https://209.165.201.1/api/running/switch/spanning-tree/pathcost -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Create Spanning-Tree priority APIs

```
curl -k -u admin:admin -d "<spanning-tree><priority>8192</priority></spanning-tree>" -X
PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Modify Spanning-Tree APIs

Table 125: Modify Spanning-Tree APIs

Action	Method	Payload Required	API
To modify the spanning-tree elements	PATCH	Yes	/api/running/switch/spanning-tree

Example: PATCH Modify Spanning-Tree bpdu APIs

```
curl -k -u admin:admin -d "<spanning-tree><bpdu>filtering</bpdu></spanning-tree>" -X PATCH
 https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree forward-time APIs

```
curl -k -u admin:admin -d "<spanning-tree><forward-time>18</forward-time></spanning-tree>"
 -X PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree mode APIs

```
curl -k -u admin:admin -d "<spanning-tree><mode>rstp</mode></spanning-tree>" -X PATCH
https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree hello-time APIs

```
curl -k -u admin:admin -d "<spanning-tree><hello-time>6</hello-time></spanning-tree>" -X
PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree max-age APIs

```
curl -k -u admin:admin -d "<spanning-tree><max-age>24</max-age></spanning-tree>" -X PATCH
https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree loopback-guard APIs

```
curl -k -u admin:admin -d "<spanning-tree><loopback-guard></loopback-guard></spanning-tree>"
 -X PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree method APIs

```
curl -k -u admin:admin -d "<spanning-tree><method>short</method></spanning-tree>" -X PATCH
 https://209.165.201.1/api/running/switch/spanning-tree/pathcost -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH Modify Spanning-Tree priority APIs

```
curl -k -u admin:admin -d "<spanning-tree><priority>8192</priority></spanning-tree>" -X
PATCH https://209.165.201.1/api/running/switch/spanning-tree -H "Content-Type:
application/vnd.yang.data+xml"
```

Delete Spanning-Tree APIs

Table 126: Delete Spanning-Tree APIs

Action	Method	Payload Required	API
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/bpdu
To disable the spanning-tree functionality.	DELETE	No	/api/running/switch/spanning-tree/enable

Action	Method	Payload Required	API
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/forward-time
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/mode
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/hello-time
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/max-age
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/loopback-guard
To restore the default configuration	DELETE	No	/api/running/switch/spanning-tree/pathcost
To restore the default device spanning-tree priority.	DELETE	No	/api/running/switch/spanning-tree/priority

Example: DELETE Delete Spanning-Tree bpdu APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/spanning-tree/bpdu
```

Example: DELETE Delete Spanning-Tree APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/spanning-tree/enable
```

Example: DELETE Delete Spanning-Tree forward-time APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/forward-time
```

Example: DELETE Delete Spanning-Tree mode APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/spanning-tree/mode
```

Example: DELETE Delete Spanning-Tree hello-time APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/hello-time
```

Example: DELETE Delete Spanning-Tree max-age APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/max-age
```

Example: DELETE Delete Spanning-Tree loopback-guard APIs

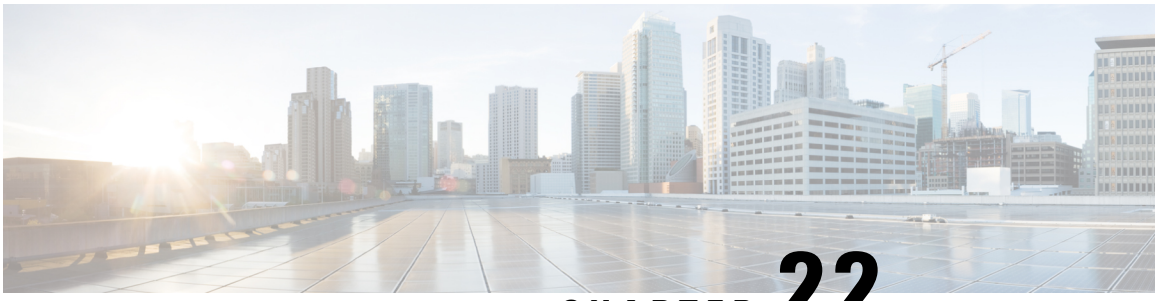
```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/loopback-guard
```

Example: DELETE Delete Spanning-Tree pathcost APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/pathcost
```

Example: DELETE Delete Spanning-Tree priority APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/spanning-tree/priority
```



CHAPTER 22

Interface Stat APIs

Table 127: Interface Stat APIs

Action	Method	Payload Required	API
To display the status of all interfaces or of a specific interface	GET	No	/api/operational/switch/interface/status/gigabitEthernet/\"1/0\"
To display traffic seen by all the physical interfaces or by a specific interface	GET	No	/api/operational/switch/interface/counters/gigabitEthernet/\"1/0\"
To display RMON Ethernet statistics	GET	No	/api/operational/switch/interface/rmon/gigabitEthernet/\"1/0\"
To display information about the inline power for all interfaces or for a specific interface	GET	No	/api/operational/switch/interface/inline-status/gigabitEthernet/\"1/0\"
To display the administrative and operational status of all interfaces or a specific interface.	GET	No	/api/operational/switch/interface/switchPort/gigabitEthernet/\"1/0\"
To display the configuration for all configured interfaces	GET	No	/api/running/switch/interface/gigabitEthernet
To display the configuration for a specific interface	GET	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"
To configure the speed of a given Ethernet interface when not using auto-negotiation	PATCH	Yes	/api/running/switch/interface/gigabitEthernet

Example for Interface Stat APIs Payload

```
<gigabitEthernet><name>1/0</name><speed>1000</speed></gigabitEthernet>
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://172.25.212.178/api/operational/switch/interface/status/gigabitEthernet/\"1/0\"
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://172.25.212.178/api/operational/switch/interface/counters/gigabitEthernet/\"1/0\"
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://172.25.212.178/api/operational/switch/interface/rmon/gigabitEthernet/\"1/0\"
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://172.25.212.178/api/operational/switch/interface/inline-status/gigabitEthernet/\"1/0\"
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://172.25.212.178/api/operational/switch/interface/switchPort/gigabitEthernet/\"1/0\"
```

Example: GET Interface Stat APIs

```
curl -i -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Accept:
application/vnd.yang.collection+xml"
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET http://209.165.201.1/api/running/switch/interface?deep
```

Example: GET Interface Stat APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"
```

Example: PATCH Interface Stat APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><speed>1000</speed></gigabitEthernet>" -X PATCH
https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
application/vnd.yang.data+xml"
```

- Interface Port APIs, on page 253

Interface Port APIs

Table 128: Interface Port APIs

Action	Method	Payload Required	API
To delete the interface speed configuration	DELETE	No	/api/running/switch/interface/gigabitEthernet/speed
To disable an interface	PATCH	Yes	/api/running/switch/interface/gigabitEthernet
To restart a disabled interface	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/shutdown
To delete the interface description	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/description

Example for Interface Port APIs Payload

```
<gigabitEthernet><name>1/0</name><shutdown/></gigabitEthernet>
```

Example: DELETE Interface Port APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/speed
```

Example: PATCH Interface Port APIs

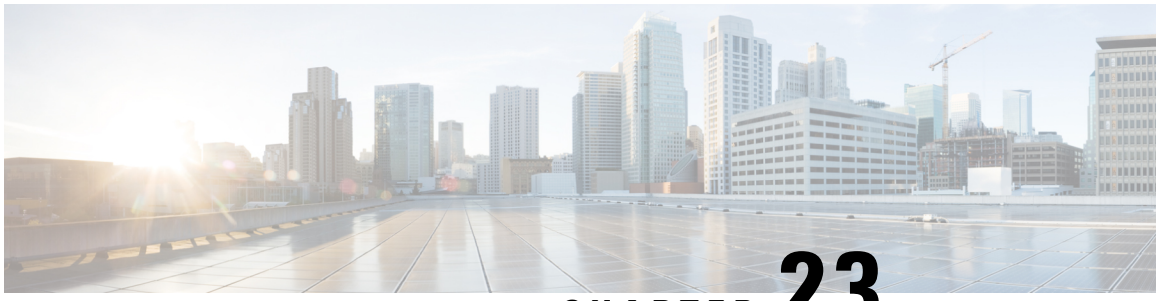
```
curl -k -u admin:admin -d "<gigabitEthernet><name>1/0</name><shutdown/></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: DELETE Interface Port APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/shutdown
```

Example: DELETE Interface Port APIs

```
curl -i -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/description
```



CHAPTER 23

Interface GigabitEthernet Switchport APIs

Table 129: Interface GigabitEthernet Switchport APIs

Action	Method	Payload Required	API
To retrieve interface switchport configuration	GET	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport
To configure interface switchport mode	PATCH	Yes	/api/running/switch/interface/gigabitEthernet
To replace interface switchport trunk allowed vlans for interface	PUT	Yes	/api/config/switch/interface/gigabitEthernet/\"1/0\"
To delete interface switchport protected-port	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/protected-port
To delete interface switchport mode	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/mode
To delete interface switchport trunk allowed vlan	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/trunk/allowed/vlan/

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><trunk><allowed><vlan><vlan-range>1-10,20-30</vlan-range></vlan></allowed></trunk></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><mode>trunk</mode></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><trunk><native><vlan>100</vlan></native></trunk></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><trunk><allowed><vlan><ids>502</ids><ids>503</ids></vlan></allowed></trunk></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><community>1</community></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><dot1q-tunnel><vlan>100</vlan></dot1q-tunnel></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><access><vlan>2</vlan></access></switchport></gigabitEthernet>
```

Example for Interface GigabitEthernet Switchport APIs Payload

```
<gigabitEthernet><name>1/0</name><switchport><protected-port></protected-port></switchport></gigabitEthernet>
```

Example: GET Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><mode>trunk</mode></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><trunk><native><vlan>100</vlan></native></trunk></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><trunk><allowed><vlan><ids>502</ids><ids>503</ids></vlan></allowed></trunk></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PUT Interface GigabitEthernet Switchport APIs

```
curl -k -v -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><trunk><allowed><vlan><ids>52</ids><ids>507</ids></vlan></allowed></trunk></switchport></gigabitEthernet>"
 -X  PUT  https://209.165.201.1/api/config/switch/interface/gigabitEthernet/\"1/0\" -H
"Content-Type: application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><community>1</community></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><dot1q-tunnel><vlan>100</vlan></dot1q-tunnel></switchport></gigabitEthernet>"
```

```
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><access><vlan>2</vlan></access></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><switchport><protected-port></protected-port></switchport></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: DELETE Interface GigabitEthernet Switchport APIs

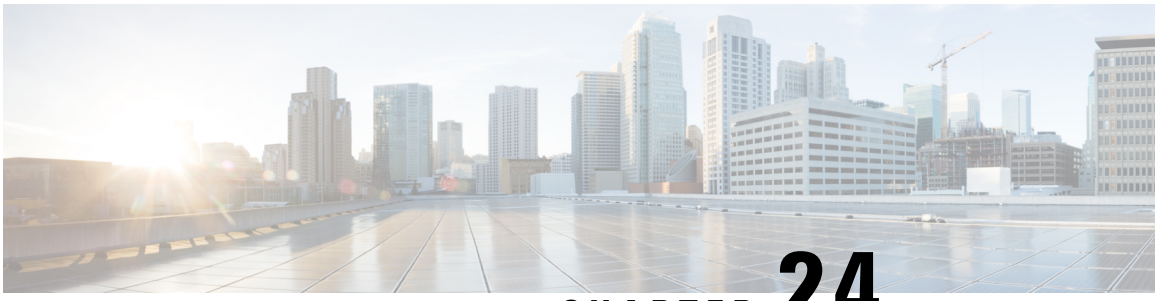
```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/protected-port
```

Example: DELETE Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/mode
```

Example: DELETE Interface GigabitEthernet Switchport APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/switchport/trunk/allowed/vlan/
```



CHAPTER 24

Interface GigabitEthernet Spanning-Tree APIs

Table 130: Interface GigabitEthernet Spanning-Tree APIs

Action	Method	Payload Required	API
To retrieve interface spanning-tree configuration	GET	No	/api/running/switch/interface/gigabitEthernet/"1/0"/spanning-tree
To configure interface spanning-tree element	PATCH	Yes	/api/running/switch/interface/gigabitEthernet
To configure the interface spanning-tree to guard the interface from becoming a root port	PUT	Yes	/api/running/switch/interface/gigabitEthernet/\"1/0\"
To delete the spanning-tree root guard	DELETE	No	/api/running/switch/interface/gigabitEthernet/"1/0"/spanning-tree/guard/root
To delete interface spanning-tree element	DELETE	No	/api/running/switch/interface/gigabitEthernet/\"1/0\"/spanning-tree/cost

Example for Interface GigabitEthernet Spanning-Tree APIs Payload

```
<gigabitEthernet><name>1/0</name><spanning-tree><cost>2000</cost></spanning-tree></gigabitEthernet>
```

Example for Interface GigabitEthernet Spanning-Tree APIs Payload

```
<gigabitEthernet><name>1/0</name><spanning-tree><guard><root></root></guard></spanning-tree></gigabitEthernet>
```

Example for Interface GigabitEthernet Spanning-Tree APIs Payload

```
<gigabitEthernet><name>1/0</name><description>GigabitEthernet_slot_1_port_0</description></gigabitEthernet>
```

Example for Interface GigabitEthernet Spanning-Tree APIs Payload

```
<gigabitEthernet><name>1/0</name><bridge><multicast><unregistered>filtering</unregistered></multicast></bridge></gigabitEthernet>
```

Example: GET Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -X GET
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/"1/0"/spanning-tree
```

Example: PATCH Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><spanning-tree><cost>2000</cost></spanning-tree></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><spanning-tree><guard><root></root></guard></spanning-tree></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: PUT Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><description>GigabitEthernet_slot_1_port_0</description></gigabitEthernet>"
 -X PUT https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\" -H
"Content-Type: application/vnd.yang.data+xml"
```

Example: PATCH Interface GigabitEthernet Spanning-Tree APIs

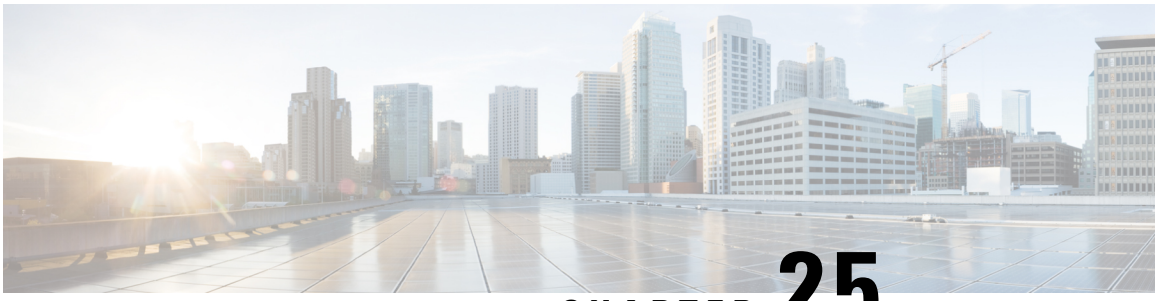
```
curl -k -u admin:admin -d
"<gigabitEthernet><name>1/0</name><bridge><multicast><unregistered>filtering</unregistered></multicast></bridge></gigabitEthernet>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/gigabitEthernet -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: DELETE Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/spanning-tree/guard/root
```

Example: DELETE Interface GigabitEthernet Spanning-Tree APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/interface/gigabitEthernet/\"1/0\"/spanning-tree/cost
```



CHAPTER 25

SPAN/RSPAN APIs

Table 131: SPAN/RSPAN APIs

Action	Method	Payload Required	API
To add source vlan to SPAN session	PATCH	Yes	/api/config/switch/monitor
To replace existing vlan with current source vlan to SPAN session	PUT	Yes	/api/config/switch/monitor
To delete source vlan in a SPAN session	DELETE	No	/api/config/switch/monitor/session/\"1\"/source/vlan
To delete source interface in a SPAN session	DELETE	No	/api/config/switch/monitor/session/\"1\"/source/interfaces/gigabitEthernet/\"1/0/"
To delete destination interface in a SPAN session	DELETE	No	/api/running/switch/monitor/session/\"2\"/destination
To delete source vlan to RSPAN session	DELETE	No	/api/config/switch/monitor/session/\"1\"/source/remote
To delete destination vlan and reflector-port to RSPAN session	DELETE	No	/api/config/switch/monitor
To show source session	GET	No	/api/running/switch/monitor/session/\"1\"/source
To show destination session	GET	No	/api/running/switch/monitor/session/\"4\"/destination

Example for SPAN/RSPAN APIs Payload

```
<monitor><session><session-id>1</session-id><source><vlan>5</vlan></source></session></monitor>
```

Example for SPAN/RSPAN APIs Payload

```
<monitor><session><session-id>1</session-id><source><interfaces><gigabitEthernet><name>1/0</name><direction>both</direction></gigabitEthernet></interfaces></source></session></monitor>
```

Example for SPAN/RSPAN APIs Payload

```
<monitor><session><session-id>1</session-id><source><remote><vlan>20</vlan></remote></source></session></monitor>
```

Example for SPAN/RSPAN APIs Payload

```
<monitor><session><session-id>4</session-id><destination><erspan-id>20</erspan-id><reflector-port></reflector-port><gigabitEthernet>1/4</gigabitEthernet><network></network></erspan></destination></session></monitor>
```

Example: PATCH SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>1</session-id><source><vlan>5</vlan></source></session></monitor>"
 -X PATCH https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PUT SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>1</session-id><source><vlan>6</vlan></source></session></monitor>"
 -X PUT https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>1</session-id><source><interfaces><gigabitEthernet><name>1/0</name><direction>both</direction></gigabitEthernet></interfaces></source></session></monitor>"
 -X PATCH https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>1</session-id><source><interfaces><gigabitEthernet><name>1/1</name><direction>x</direction></gigabitEthernet></interfaces></source></session></monitor>"
 -X PATCH https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>1</session-id><source><remote><vlan>20</vlan></remote></source></session></monitor>"
 -X PATCH https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: PATCH SPAN/RSPAN APIs

```
curl -k -v -u admin:admin -d
"<monitor><session><session-id>4</session-id><destination><erspan-id>20</erspan-id><reflector-port></reflector-port><gigabitEthernet>1/4</gigabitEthernet><network></network></erspan></destination></session></monitor>"
 -X PATCH https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE SPAN/RSPAN APIs

```
curl -k -v -u admin:admin  -X DELETE
https://209.165.201.1/api/config/switch/monitor/session/\"1\"/source/vlan -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: DELETE SPAN/RSPAN APIs

```
curl -k -v -u admin:admin  -X DELETE
https://209.165.201.1/api/config/switch/monitor/session/\"1\"/source/interfaces/gigabitEthernet/\"1\/0\"
```

Example: DELETE SPAN/RSPAN APIs

```
curl -k -v -u admin:admin  -X  DELETE
https://172.25.212.189/api/running/switch/monitor/session/\"2\"/destination
```

Example: DELETE SPAN/RSPAN APIs

```
curl -k -v -u admin:admin  -X DELETE
https://209.165.201.1/api/config/switch/monitor/session/\"1\"/source/remote -H "Content-Type:
 application/vnd.yang.data+xml"
```

Example: DELETE SPAN/RSPAN APIs

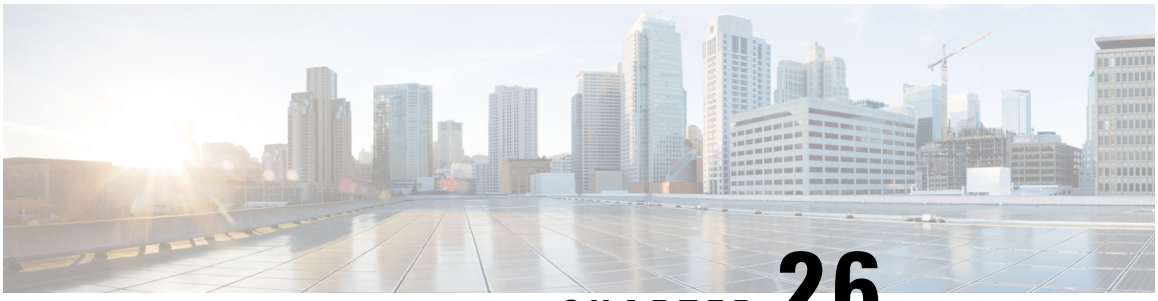
```
curl -k -v -u admin:admin   -d
"<session><session-id>1</session-id><destination><remote><vlan>20</vlan><reflector-port></reflector-port><gigabitEthernet>1/4/gigabitEthernet><network></network></remote></destination></session>"
 -X DELETE https://209.165.201.1/api/config/switch/monitor -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: GET SPAN/RSPAN APIs

```
curl -k -v -u admin:admin   -X GET
https://209.165.201.1/api/running/switch/monitor/session/\"1\"/source
```

Example: GET SPAN/RSPAN APIs

```
curl -k -v -u admin:admin   -X GET
https://209.165.201.1/api/running/switch/monitor/session/\"4\"/destination
```



CHAPTER 26

VLAN and interface VLAN related APIs

Table 132: VLAN and interface VLAN related APIs

Action	Method	Payload Required	API
To create VLAN	POST	Yes	/api/running/switch
To delete VLAN	DELETE	No	/api/running/switch/vlan/90
To create interface VLAN	POST	Yes	/api/running/switch/interface
To display the VLAN configuration	GET	No	/api/running/switch/vlan?deep
To displace all the interface VLAN configurations	GET	No	/api/running/switch/interface/vlan?deep
To display a specific interface VLAN configuration	GET	No	/api/running/switch/interface/vlan/90
To delete the VLAN interface	DELETE	No	/api/running/switch/interface/vlan/90
To configure an IP address for a VLAN interface	PATCH	Yes	/api/running/switch/interface/vlan
To delete an IP address for a VLAN interface	DELETE	No	/api/running/switch/ip/route/route-forward-list222.255.255.255,5551
To show the IP interface	GET	No	/api/operational/switch/ip/interface
To delete remote SPAN setting for VLAN	DELETE	No	/api/running/switch/interface/vlan/20/remote-span

Example for VLAN and interface VLAN related APIs Payload

```
<vlan><vlan-id>90</vlan-id></vlan>
```

Example for VLAN and interface VLAN related APIs Payload

```
<vlan><vlan-id>90</vlan-id><ip><address><primary><address>13.13.13.13</address><mask>255.255.255.0</mask></primary></address></ip></vlan>
```

Example for VLAN and interface VLAN related APIs Payload

```
<vlan><vlan-id>20</vlan-id><remote-span/></vlan>
```

Example: POST VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -d "<vlan><vlan-id>90</vlan-id></vlan>" -X POST
https://209.165.201.1/api/running/switch -H "Content-Type: application/vnd.yang.data+xml"
```

Example: DELETE VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/vlan/90
```

Example: POST VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -d "<vlan><vlan-id>90</vlan-id></vlan>" -X POST
https://209.165.201.1/api/running/switch/interface -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: GET VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/running/switch/vlan?deep
```

Example: GET VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/running/switch/interface/vlan?deep
```

Example: GET VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/running/switch/interface/vlan/90
```

Example: DELETE VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X DELETE https://209.165.201.1/api/running/switch/interface/vlan/90
```

Example: PATCH VLAN and interface VLAN related APIs

```
curl -k -v -u admin:admin  -d
"<vlan><vlan-id>90</vlan-id><ip><address><primary><address>13.13.13.13</address><mask>255.255.255.0</mask></primary></address></ip></vlan>"
 -X PATCH https://209.165.201.1/api/running/switch/interface/vlan -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X DELETE
https://209.165.201.1/api/running/switch/ip/route/ip-route-forwarding-list/2.2.2.2,255.255.255.255,5.5.5.1
```

Example: GET VLAN and interface VLAN related APIs

```
curl -k -u admin:admin -X GET https://209.165.201.1/api/operational/switch/ip/interface
```

Example: PATCH VLAN and interface VLAN related APIs

```
curl -k -v -u admin:admin -d "<vlan><vlan-id>20</vlan-id><remote-span/></vlan>" -X PATCH
https://209.165.201.1/api/running/switch/interface/vlan -H "Content-Type:
application/vnd.yang.data+xml"
```

Example: DELETE VLAN and interface VLAN related APIs

```
curl -k -v -u admin:admin  -X  DELETE
https://209.165.201.1/api/running/switch/interface/vlan/20/remote-span -H "Content-Type:
application/vnd.yang.data+xml"
```