NSX for vSphere API Guide

Version: 6.3

Part-Number: EN-002339-02

Table of Contents

Introduction

This manual, the NSX for vSphere API Guide, describes how to install, configure, monitor, and maintain the VMware® NSX system by using REST API requests.

Intended Audience

This manual is intended for anyone who wants to use REST API to programmatically control NSX in a VMware vSphere environment. The information in this manual is written for experienced developers who are familiar with virtual machine technology, virtualized datacenter operations, and REST APIs. This manual also assumes familiarity with NSX for vSphere.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation go to http://www.vmware.com/support/pubs.

Technical Documentation and Product Updates

You can find the most up-to-date technical documentation on the VMware Web site at: http://www.vmware.com/support/.

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to: .

Using the NSX REST API

To use the NSX REST API, you must configure a REST client, verify the required ports are open between your REST client and the NSX Manager, and understand the general RESTful workflow.

Ports Required for the NSX REST API

The NSX Manager requires port 443/TCP for REST API requests.

Configuring REST Clients for the NSX REST API

Some browser-based clients include the Chrome app, Postman, or the Firefox add-on, RESTClient. Curl is a command-line tool that can function as a REST client. The details of REST client configuration will vary from client to client, but this general information should help you configure your REST client correctly.

- **The NSX REST API uses basic authentication.**
  You must configure your REST client to send the NSX Manager authentication credentials using basic authentication.
- **You must use https to send API requests to the NSX Manager.**
  You might need to import the certificate from the NSX Manager to your REST client to allow it to connect to the NSX Manager.
- **When you submit an API request with an XML request body, you must include the Content-Type: application/xml header.**
  Some requests require additional headers, for example, firewall configuration changes require the If-Match header. This is noted on each method description.
- **To ensure you always receive XML response bodies, set the Accept: application/xml header.**
  Some API methods respond with JSON output, which is an experimental feature. Setting the Accept header ensures you always get XML output. **Note:** some methods, for example, the central CLI method, POST /1.0/nsx/cli, might require a different Accept header.

The following API method will return a response on a newly deployed NSX Manager appliance, even if you have not made any configuration changes. You can use this as a test to verify that your REST client is configured correctly to communicate with the NSX Manager API.

```
GET /api/2.0/services/usermgmt/user/admin
```

URI and Query Parameters

Some methods have URI or query parameters. URI parameters are values that you include in the request URL. You use a question mark (**?**) to join the request URL and the query parameters. Multiple query parameters can be combined by using ampersands (**&**).

For example, you can use this method to get a list of logical switches on a transport zone:

```
GET /api/2.0/vdn/scopes/{scopeId}/virtualwires
```

scopeId is a URI parameter that represents a transport zone.

The **startindex** and **pagesize** query parameters control how this information is displayed. **startindex** determines which logical switch to begin the list with, and **pagesize** determines how many logical switches to list.

To view the first 20 logical switches on transport zone vdnscope-1, use the following parameters:
- **scopeId** URI parameter set to *vdnscope-1*.
- **startindex** query parameter set to *0*.
- **pagesize** query parameter set to *20*.

These parameters are combined to create this request:

```
GET https://192.168.110.42/api/2.0/vdn/scopes/vdnscope-1/virtualwires?startindex=0&pagesize=20
```

RESTful Workflow Patterns

All RESTful workflows fall into a pattern that includes only two fundamental operations, which you repeat in this order for as long as necessary.
- **Make an HTTP request (GET, PUT, POST, or DELETE).**
  The target of this request is either a well-known URL (such as NSX Manager) or a link obtained from the response to a previous request. For example, a GET request to an Org URL returns links to vDC objects contained by the Org.
- **Examine the response, which can be an XML document or an HTTP response code.**
  If the response is an XML document, it might contain links or other information about the state of an object. If the response is an HTTP response code, it indicates whether the request succeeded or failed, and might be accompanied by a URL that points to a location from which additional information can be retrieved.

Finding vCenter Object IDs

Many API methods reference vCenter object IDs in URI parameters, query parameters, request bodies, and response bodies. You can find vCenter object IDs via the vCenter Managed Object Browser.

Find Datacenter MOID

1. In a web browser, enter the vCenter Managed Object Browser URL: http://vCenter-IP-Address/mob.
2. Click **content**.
3. Find **rootFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-d1*.
4. Find the **childEntity** in the Name column, and the corresponding Value column entry is the datacenter MOID. For example, *datacenter-21*.

Find Cluster or Host MOID

1. In a web browser, enter the vCenter Managed Object Browser URL: http://vCenter-IP-Address/mob.
2. Click **content**.
3. Find **rootFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-d1*.

4 Find **childEntity** in the Name column, and click the corresponding link in the Value column. For example, *datacenter-21*.

5 Find **hostFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-h23*.

6 Find **childEntity** in the Name column. The corresponding Value column lists the host clusters. For example, *domain-c33*.

7 To find the MOID of a host in a cluster, click the appropriate host cluster link located in the previous step.

8 Find *host* in the Name column. The corresponding Value column lists the hosts in that cluster by vCenter MOID and hostname. For example, *host-32 (esx-02a.corp.local)*.

Find Portgroup MOID

1 In a web browser, enter the vCenter Managed Object Browser URL: http://vCenter-IP-Address/mob.

2 Click **content**.

3 Find **rootFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-d1*.

4 Find **childEntity** in the Name column, and click the corresponding link in the Value column. For example, *datacenter-21*.

5 Find **hostFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-h23*.

6 Find **childEntity** in the Name column. The corresponding Value column contains links to host clusters. Click the appropriate host cluster link. For example, *domain-c33*.

7 Find **host** in the Name column. The corresponding Value column lists the hosts in that cluster by vCenter MOID and hostname. Click the appropriate host link, For example, host-32.

8 Find **network** in the Name column. The corresponding Value column lists the port groups on that host, For example, *dvportgroup-388*.

Find VM MOID or VM Instance UUID

1 In a web browser, enter the vCenter Managed Object Browser URL: http://vCenter-IP-Address/mob.

2 Click **content**.

3 Find **rootFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-d1*.

4 Find **childEntity** in the Name column, and click the corresponding link in the Value column. For example, *datacenter-21*.

5 Find **hostFolder** in the Name column, and click the corresponding link in the Value column. For example, *group-h23*.

6 Find **childEntity** in the Name column. The corresponding Value column contains links to host clusters. Click the appropriate host cluster link. For example, *domain-c33*.

7 Find **host** in the Name column. The corresponding Value column lists the hosts in that cluster by vCenter MOID and hostname. Click the appropriate host link, For example, *host-32*.

8 Find **vm** in the Name column. The corresponding Value column lists the virtual machines by vCenter MOID and hostname. For example, *vm-216 (web-01a)*.

9 To find the instance UUID of a VM, click the VM MOID link located in the previous step. Click the config link in the Value column.

10 Find **instanceUuid** in the Name column. The corresponding Value column lists the VM instance UUID. For example, *502e71fa-1a00-759b-e40f-ce778e915f16*.

Endpoints

https://{nsxmanager}/api

Base URI Parameters:

nsxmanager (required)	Hostname or IP address of the NSX Manager.

Working with vSphere Distributed Switches

GET /api/2.0/vdn/switches

Description:
Retrieve information about all vSphere Distributed Switches.

Responses:
Status Code: 200
Body: application/xml

```
<vdsContexts>
 <vdsContext>
   <switch>
     <objectId>dvs-35</objectId>
     <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
     <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
     <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
     <revision>10</revision>
     <type>
       <typeName>VmwareDistributedVirtualSwitch</typeName>
     </type>
     <name>vds-site-a</name>
     <scope>
       <id>datacenter-21</id>
       <objectTypeName>Datacenter</objectTypeName>
       <name>Datacenter Site A</name>
     </scope>
     <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
     <isUniversal>false</isUniversal>
     <universalRevision>0</universalRevision>
   </switch>
   <mtu>1600</mtu>
   <teaming>FAILOVER_ORDER</teaming>
   <uplinkPortName>Uplink 4</uplinkPortName>
   <promiscuousMode>false</promiscuousMode>
 </vdsContext>
 <vdsContext>
   <switch>
     <objectId>dvs-47</objectId>
     <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
     ***
   </switch>
   ***
 </vdsContext>
</vdsContexts>
```

`POST /api/2.0/vdn/switches`

Description:
Prepare a vSphere Distributed Switch.

The MTU is the maximum amount of data that can be transmitted in one packet before it is divided into smaller packets. VXLAN frames are slightly larger in size because of the traffic encapsulation, so the MTU required is higher than the standard MTU. You must set the MTU for each switch to 1602 or higher.

Request:
Body: application/xml

```
<vdsContext>
 <switch>
   <objectId>dvs-26</objectId>
   <type>
     <typeName>DistributedVirtualSwitch</typeName>
   </type>
   <name></name>
   <revision>0</revision>
   <objectTypeName>DistributedVirtualSwitch</objectTypeName>
 </switch>
 <teaming>ETHER_CHANNEL</teaming>
 <mtu>mtu-value</mtu>
</vdsContext>
```

Working with vSphere Distributed Switches in a Datacenter

`GET /api/2.0/vdn/switches/datacenter/{datacenterID}`

URI Parameters:

`datacenterID` (required)	A valid datacenter ID (e.g. datacenter-21)

Description:
Retrieve information about all vSphere Distributed Switches in the specified datacenter.

Responses:
Status Code: 200
Body: application/xml

```
<vdsContexts>
 <vdsContext>
   <switch>
     <objectId>dvs-35</objectId>
     <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
     <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
     <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
     <revision>10</revision>
     <type>
```

```
      <typeName>VmwareDistributedVirtualSwitch</typeName>
    </type>
    <name>vds-site-a</name>
    <scope>
      <id>datacenter-21</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>Datacenter Site A</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </switch>
  <mtu>1600</mtu>
  <teaming>FAILOVER_ORDER</teaming>
  <uplinkPortName>Uplink 4</uplinkPortName>
  <promiscuousMode>false</promiscuousMode>
 </vdsContext>
 <vdsContext>
  <switch>
    <objectId>dvs-47</objectId>
    <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
    ***
  </switch>
  ***
 </vdsContext>
</vdsContexts>
```

Working With a Specific vSphere Distributed Switch

`GET /api/2.0/vdn/switches/{vdsId}`

URI Parameters:

vdsId (required)	A valid vSphere Distributed Switch ID (e.g. dvs-35)

Description:
Retrieve information about the specified vSphere Distributed Switch.

Responses:
Status Code: 200
Body: application/xml

```
<vdsContext>
 <switch>
  <objectId>dvs-35</objectId>
  <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
  <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
  <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
  <revision>10</revision>
  <type>
    <typeName>VmwareDistributedVirtualSwitch</typeName>
  </type>
```

```
    <name>vds-site-a</name>
    <scope>
      <id>datacenter-21</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>Datacenter Site A</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </switch>
  <mtu>1600</mtu>
  <teaming>FAILOVER_ORDER</teaming>
  <uplinkPortName>Uplink 4</uplinkPortName>
  <promiscuousMode>false</promiscuousMode>
</vdsContext>
```

`DELETE /api/2.0/vdn/switches/{vdsId}`

URI Parameters:

vdsId (required)	A valid vSphere Distributed Switch ID (e.g. dvs-35)

Description:
Delete the specified vSphere Distributed Switch.

Working with Segement ID Pools and Multicast Ranges

Working With Segment ID Pools

Segment ID pools (also called segment ID ranges) provide virtual network identifiers (VNIs) to logical switches.

You must configure a segment ID pool for each NSX Manager. You can have more than one segment ID pool. The segment ID pool includes the beginning and ending IDs.

You should not configure more than 10,000 VNIs in a single vCenter server because vCenter limits the number of dvPortgroups to 10,000.

If any of your transport zones will use multicast or hybrid replication mode, you must also configure a multicast address range.

`GET /api/2.0/vdn/config/segments`

Description:
Retrieve information about all segment ID pools.

Responses:
Status Code: 200
Body: application/xml

```
<segmentRanges>
 <segmentRange>
   <id>1</id>
   <name>Local Segments</name>
   <desc>Local Segment ID pool</desc>
   <begin>5000</begin>
   <end>5999</end>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
 </segmentRange>
 <segmentRange>
   <id>3</id>
   <name>Universal-Segments</name>
   <desc>Universal segment ID pool</desc>
   <begin>200000</begin>
   <end>201000</end>
   <isUniversal>true</isUniversal>
   <universalRevision>2</universalRevision>
 </segmentRange>
</segmentRanges>
```

`POST /api/2.0/vdn/config/segments`

Query Parameters:

`isUniversal` (optional)	Set to *true* when creating a universal segment ID pool.

Description:
Add a segment ID pool.

- **name** - Required property.
- **desc** - Optional property.
- **begin** - Required property. Minimum value is *5000*
- **end** - Required property. Maximum value is *16777216*

Request:
Body: application/xml

```
<segmentRange>
 <name>Segment 1</name>
 <desc>Segment Range 1</desc>
 <begin>5000</begin>
 <end>12999</end>
</segmentRange>
```

Working With a Specific Segment ID Pool

`GET /api/2.0/vdn/config/segments/{segmentPoolId}`

URI Parameters:

segmentPoolId (required)	A valid *segmentPoolId*

Description:
Retrieve information about the specified segment ID pool.

Responses:
Status Code: 200
Body: application/xml

```
<segmentRange>
 <id>1</id>
 <name>Local Segments</name>
 <desc>Local Segment ID pool</desc>
 <begin>5000</begin>
 <end>5999</end>
 <isUniversal>false</isUniversal>
 <universalRevision>0</universalRevision>
</segmentRange>
```

`PUT /api/2.0/vdn/config/segments/{segmentPoolId}`

URI Parameters:

segmentPoolId (required)	A valid *segmentPoolId*

Description:
Update the specified segment ID pool.

If the segment ID pool is universal you must send the API request to the primary NSX Manager.

Request:
Body: application/xml

```
<segmentRange>
 <desc>Local Segment ID pool expanded</desc>
 <end>6999</end>
</segmentRange>
```

`DELETE /api/2.0/vdn/config/segments/{segmentPoolId}`

URI Parameters:

segmentPoolId (required)	A valid *segmentPoolId*

Description:
Delete the specified segment ID pool.

If the segment ID pool is universal you must send the API request to the primary NSX Manager.

Working with Multicast Address Ranges

If any of your transport zones will use multicast or hybrid replication mode, you must add a multicast address range (also called a multicast address pool). Specifying a multicast address range helps in spreading traffic across your network to avoid overloading a single multicast address.

`GET /api/2.0/vdn/config/multicasts`

Description:
Retrieve information about all configured multicast address ranges.

Universal multicast address ranges have the property isUniversal set to *true*.

Responses:
Status Code: 200
Body: application/xml

```
<multicastRanges>
 <multicastRange>
   <id>5</id>
   <name>239.0.0.0-239.255.255.255</name>
   <begin>239.0.0.0</begin>
   <end>239.255.255.255</end>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
 </multicastRange>
 <multicastRange>
   <id>10</id>
   <name>Range 2</name>
   <begin>237.0.0.0</begin>
```

```
   <end>237.255.255.255</end>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
 </multicastRange>
</multicastRanges>
```

`POST /api/2.0/vdn/config/multicasts`

Query Parameters:

`isUniversal` (optional)	Set to *true* when creating a universal multicast address range.

Description:
Add a multicast address range for logical switches.

The address range includes the beginning and ending addresses.

Request:
Body: application/xml

```
<multicastRange>
 <name>Range 2</name>
 <begin>237.0.0.0</begin>
 <end>237.255.255.255</end>
</multicastRange>
```

Working With a Specific Multicast Address Range

`GET /api/2.0/vdn/config/multicasts/{multicastAddresssRangeId}`

URI Parameters:

`multicastAddresssRangeId` (required)	A valid multicast address range ID

Description:
Retrieve information about the specified multicast address range.

Responses:
Status Code: 200
Body: application/xml

```
<multicastRange>
 <id>5</id>
 <name>239.0.0.0-239.255.255.255</name>
 <begin>239.0.0.0</begin>
 <end>239.255.255.255</end>
 <isUniversal>false</isUniversal>
 <universalRevision>0</universalRevision>
```

```
</multicastRange>
```

PUT /api/2.0/vdn/config/multicasts/{multicastAddresssRangeId}

URI Parameters:

multicastAddresssRangeId (required)	A valid multicast address range ID

Description:
Update the specified multicast address range.

If the multicast address range is universal you must send the API request to the primary NSX Manager.

Request:
Body: application/xml

```
<multicastRange>
 <name>Extended range 2</name>
 <desc>Extended range 2</desc>
 <end>238.255.255.255</end>
</multicastRange>
```

DELETE /api/2.0/vdn/config/multicasts/{multicastAddresssRangeId}

URI Parameters:

multicastAddresssRangeId (required)	A valid multicast address range ID

Description:
Delete the specified multicast address range.

If the multicast address range is universal you must send the API request to the primary NSX Manager.

Working with the VXLAN Port Configuration

GET /api/2.0/vdn/config/vxlan/udp/port

Description:
Retrieve the UDP port configured for VXLAN traffic.

Responses:
Status Code: 200
Body: application/xml

```
<int>4789</int>
```

Update the VXLAN Port Configuration

`PUT /api/2.0/vdn/config/vxlan/udp/port/{portNumber}`

URI Parameters:

`portNumber` (required)	A valid UDP port for VXLAN

Query Parameters:

`force` (optional)	Set to *true* to force the change in VXLAN port. This updates the port configuration on the hosts directly, and might cause a disruption in VXLAN traffic. In a cross-vCenter NSX environment, this does not change the port on all NSX Managers.

Description:

Update the VXLAN port configuration to use port *portNumber*.

This method changes the VXLAN port in a three phrase process, avoiding disruption of VXLAN traffic. In a cross-vCenter NSX environment, change the VXLAN port on the primary NSX Manager to propagate this change on all NSX Managers and hosts in the cross-vCenter NSX environment.

Method history:

Release	Modification
6.2.3	Method updated. Port change is now non-disruptive, and propagates to secondary NSX Managers if performed on the primary NSX Manager. Force parameter added.

VXLAN Port Configuration Update Status

`GET /api/2.0/vdn/config/vxlan/udp/port/taskStatus`

Description:

Retrieve the status of the VXLAN port configuration update.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:

Status Code: 200

Body: application/xml

```
<vxlanPortUpdatingStatus>
 <prevPort>8472</prevPort>
```

```
  <targetPort>4789</targetPort>
  <taskPhase>PHASE_TWO</taskPhase>
  <taskStatus>PAUSED</taskStatus>
</vxlanPortUpdatingStatus>
```

Working with Allocated Resources

`GET /api/2.0/vdn/config/resources/allocated`

Query Parameters:

`type`	set to **segmentId** or **multicastAddress**
`pagesize`	The number of results to return. Range is 1-1024.
`startindex`	The starting point for returning results.

Description:
Retrieve information about allocated segment IDs or multicast addresses.

Resolving Missing VXLAN VMKernel Adapters

`POST /api/2.0/vdn/config/host/{hostId}/vxlan/vteps`

Query Parameters:

`action` (required)	• *remediate*: Use the *remediate* action to recreate the missing VXLAN VMKernel adapter on the host. This action removes the adapter using the *resync* action, then recreates the adapter. • *resync*: If the VXLAN VMKernel adapter is no longer needed, you can use the *resync* action to remove the missing VXLAN VMKernel adapter from the NSX Manager configuration database.

Description:
Resolve missing VXLAN VMKernel adapters.

Method history:

Release	Modification
6.2.3	Method introduced.

Working with Transport Zones

`GET /api/2.0/vdn/scopes`

Description:
Retrieve information about all transport zones (also known as network scopes).

CDO mode state parameters (read-only)

The CDO mode state shows the most recent CDO operation, and the status of that operation. The status can be: *UNKNOWN*, *PENDING*, *IN_PROGRESS*, *COMPLETED*, or *FAILED*.

Operation Type	Description
ENABLE	Enable CDO mode on all distributed switches in the transport zone.
DISABLE	Disable CDO mode on all distributed switches in the transport zone.
EXPAND	Enable CDO mode on newly joined distributed swithes.
SHRINK	Disable CDO mode on removed distributed switches.
CLEAN_UP	Transport zone removed, clean up the CDO mode configuration from all distributed switches in the transport zone.
SYNC_ENABLE	Repush CDO mode configuration data to all distributed switches in the scope
SYNC_DISABLE	Remove CDO mode configuration from all distributed switches in the transport zone.

Method history:

Release	Modification
6.3.0	Method updated. Output includes information about CDO mode. See *Working With Transport Zone CDO Mode* for more information.

Responses:
Status Code: 200
Body: application/xml

```
<vdnScopes>
 <vdnScope>
   <objectId>universalvdnscope</objectId>
   <objectTypeName>VdnScope</objectTypeName>
   <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
   <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
   <revision>5</revision>
   <type>
     <typeName>VdnScope</typeName>
   </type>
   <name>Universal-Transport-Zone</name>
   <clientHandle></clientHandle>
   <extendedAttributes></extendedAttributes>
   <isUniversal>true</isUniversal>
   <universalRevision>0</universalRevision>
```

```
  <id>universalvdnscope</id>
  <clusters>
    <cluster>
      <cluster>
        <objectId>domain-c33</objectId>
        <objectTypeName>ClusterComputeResource</objectTypeName>
        <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
        <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
        <revision>20</revision>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>Compute Cluster A</name>
        <scope>
          <id>datacenter-21</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>Datacenter Site A</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </cluster>
    </cluster>
    <cluster>
      <cluster>
        <objectId>domain-c41</objectId>
        <objectTypeName>ClusterComputeResource</objectTypeName>
        <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
        <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
        <revision>16</revision>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>Management &amp;amp; Edge Cluster</name>
        <scope>
          <id>datacenter-21</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>Datacenter Site A</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </cluster>
    </cluster>
  </clusters>
  <virtualWireCount>5</virtualWireCount>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
  <cdoModeEnabled>false</cdoModeEnabled>
  <cdoModeState>
    <jobId>jobdata-23061</jobId>
    <operationType>SYNC_DISABLE</operationType>
    <status>COMPLETED</status>
    <errorCode>0</errorCode>
  </cdoModeState>
</vdnScope>
<vdnScope>
  <objectId>vdnscope-1</objectId>
  <objectTypeName>VdnScope</objectTypeName>
  <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
  <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
```

```
  <revision>1</revision>
  <type>
    <typeName>VdnScope</typeName>
  </type>
  <name>Local-Transport-Zone-A test</name>
  <description></description>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <isUniversal>false</isUniversal>
  <universalRevision>0</universalRevision>
  <id>vdnscope-1</id>
  <clusters>
    <cluster>
      <cluster>
        <objectId>domain-c33</objectId>
        <objectTypeName>ClusterComputeResource</objectTypeName>
        <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
        <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
        <revision>20</revision>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>Compute Cluster A</name>
        <scope>
          <id>datacenter-21</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>Datacenter Site A</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </cluster>
    </cluster>
    <cluster>
      <cluster>
        <objectId>domain-c41</objectId>
        <objectTypeName>ClusterComputeResource</objectTypeName>
        <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
        <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
        <revision>16</revision>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>Management &amp;amp; Edge Cluster</name>
        <scope>
          <id>datacenter-21</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>Datacenter Site A</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </cluster>
    </cluster>
  </clusters>
  <virtualWireCount>4</virtualWireCount>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
  <cdoModeEnabled>false</cdoModeEnabled>
 </vdnScope>
</vdnScopes>
```

`POST /api/2.0/vdn/scopes`

Query Parameters:

`isUniversal` (optional)	Set the **isUniversal** property to *true* when creating a universal transport zone.

Description:
Create a transport zone.

Request body parameters:
- **name** - Required. The name of the transport zone.
- **description** - Optional. Description of the transport zone.
- **objectId** - Required. The cluster object ID from vSphere. One or more are required.
- **controlPlaneMode** - Optional. The control plane mode. It can be one of the following:
  - *UNICAST_MODE*
  - *HYBRID_MODE*
  - *MULTICAST_MODE*

Request:
Body: application/xml

```
<vdnScope>
 <name>Local-Transport-Zone-B</name>
 <clusters>
   <cluster>
     <cluster>
       <objectId>domain-c7</objectId>
     </cluster>
   </cluster>
 </clusters>
 <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
</vdnScope>
```

Working with a Specific Transport Zone

`GET /api/2.0/vdn/scopes/{scopeId}`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId)

Description:
Retrieve information about the specified transport zone.

Method history:

Release	Modification

6.3.0	Method updated. Output includes information about CDO mode. See *Working With Transport Zone CDO Mode* for more information.

Responses:
Status Code: 200
Body: application/xml

```
<vdnScope>
 <objectId>universalvdnscope</objectId>
 <objectTypeName>VdnScope</objectTypeName>
 <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
 <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
 <revision>4</revision>
 <type>
   <typeName>VdnScope</typeName>
 </type>
 <name>Universal-Transport-Zone</name>
 <clientHandle></clientHandle>
 <extendedAttributes></extendedAttributes>
 <isUniversal>true</isUniversal>
 <universalRevision>0</universalRevision>
 <id>universalvdnscope</id>
 <clusters>
   <cluster>
     <cluster>
       <objectId>domain-c33</objectId>
       <objectTypeName>ClusterComputeResource</objectTypeName>
       <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
       <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
       <revision>20</revision>
       <type>
         <typeName>ClusterComputeResource</typeName>
       </type>
       <name>Compute Cluster A</name>
       <scope>
         <id>datacenter-21</id>
         <objectTypeName>Datacenter</objectTypeName>
         <name>Datacenter Site A</name>
       </scope>
       <clientHandle></clientHandle>
       <extendedAttributes></extendedAttributes>
       <isUniversal>false</isUniversal>
       <universalRevision>0</universalRevision>
     </cluster>
   </cluster>
   <cluster>
     <cluster>
       <objectId>domain-c41</objectId>
       <objectTypeName>ClusterComputeResource</objectTypeName>
       <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
       <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
       <revision>16</revision>
       <type>
         <typeName>ClusterComputeResource</typeName>
       </type>
       <name>Management &amp;amp; Edge Cluster</name>
       <scope>
         <id>datacenter-21</id>
```

```
        <objectTypeName>Datacenter</objectTypeName>
        <name>Datacenter Site A</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
    </cluster>
  </cluster>
</clusters>
<virtualWireCount>5</virtualWireCount>
<controlPlaneMode>UNICAST_MODE</controlPlaneMode>
<cdoModeEnabled>true</cdoModeEnabled>
<cdoModeState>
  <jobId>jobdata-23057</jobId>
  <operationType>ENABLE</operationType>
  <status>COMPLETED</status>
  <errorCode>0</errorCode>
  <cdoLogicalSwitch>
    <objectId>universalcdologicalswitch-2</objectId>
    <objectTypeName>CdoLogicalSwitch</objectTypeName>
    <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
    <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
    <revision>0</revision>
    <type>
      <typeName>CdoLogicalSwitch</typeName>
    </type>
    <name>universalvdnscope-cdo-logical-switch</name>
    <description>The backing logical switch to support the cdo mode on universalvdnscope.</description>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>true</isUniversal>
    <universalRevision>0</universalRevision>
    <vdnId>200005</vdnId>
    <tenantId>cdo logical switch tenant</tenantId>
    <status>OK</status>
    <lSwitchUuid>30059c1e-8d79-4f20-99a3-3d49852835e4</lSwitchUuid>
  </cdoLogicalSwitch>
</cdoModeState>
</vdnScope>
```

`POST /api/2.0/vdn/scopes/{scopeId}`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId)

Query Parameters:

`action` (required)	The **action** parameter values are: • *expand* - add a cluster to a transport zone. • *shrink* - remove a cluster from a transport zone. • *repair* - recreate missing distributed port groups.

Description:

Update the specified transport zone.

You can add a cluster to or delete a cluster from a transport zone.

You can also repair missing portgroups. For every logical switch created, NSX creates a corresponding portgroup in vCenter. If the portgroup is lost for any reason, the logical switch will stop functioning. The repair action recreates any missing portgroups.

Request:
Body: application/xml

```
<vdnScope>
<objectId>universalvdnscope</objectId>
 <clusters>
  <cluster>
   <cluster>
    <objectId>domain-c7</objectId>
    </cluster>
  </cluster>
 </clusters>
</vdnScope>
```

`DELETE /api/2.0/vdn/scopes/{scopeId}`

URI Parameters:

scopeId (required)	A valid transport zone ID (vdnScope objectId)

Description:
Delete the specified transport zone.

Working With Transport Zone Attributes

`PUT /api/2.0/vdn/scopes/{scopeId}/attributes`

URI Parameters:

scopeId (required)	A valid transport zone ID (vdnScope objectId)

Description:
Update the attributes of a transport zone.

For example, you can update the name, description, or control plane mode. You must include the cluster object IDs for the transport zone in the request body.

Request:
Body: application/xml

```
<vdnScope>
 <objectId>vdnscope-1</objectId>
 <name>Local-Transport-Zone Site-B</name>
 <clusters>
   <cluster>
     <cluster>
```

```
      <objectId>domain-c7</objectId>
    </cluster>
  </cluster>
 </clusters>
</vdnScope>
```

Working With Transport Zone CDO Mode

`POST /api/2.0/vdn/scopes/{scopeId}/cdo`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId)

Query Parameters:

`action` (required)	• *enable* to enable CDO mode configuration. • *disable* to disable CDO mode configuration. • *force_sync* to manually push the CDO configuration to all distributed switches in the transport zone.

Description:

Enable or disable CDO mode for the specified transport zone.

Controller Disconnected Operation (CDO) mode ensures that the data plane connectivity is unaffected when host lose connectivity with the controller.

If you want to enable CDO mode on the universal transport zone in a cross-vCenter NSX environment, you must do this from the primary NSX Manager. The universal synchronization service will propagate the CDO configuration to the secondary NSX Managers.

Method history:

Release	Modification
6.3.2	Method introduced. (Tech preview in 6.3.0).

Testing Multicast Group Connectivity

`POST /api/2.0/vdn/scopes/{scopeId}/conn-check/multicast`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId)

Description:

Test multicast group connectivity.

Test multicast group connectivity between two hosts connected to the specified transport zone.

Parameter **packetSizeMode** has one of the following values:
- *0* - VXLAN standard packet size
- *1* - minimum packet size

- *2* - customized packet size. If you set **packetSizeMode** to *2*, you must specify the size using the **packetSize** parameter.

Request:
Body: application/xml

```
<testParameters>
 <gateway>172.23.233.1</gateway>
 <packetSizeMode>0</packetSizeMode>
 <packetSize>1600</packetSize>
 <sourceHost>
   <hostId>host-9</hostId>
   <switchId>dvs-22</switchId>
   <vlanId>54</vlanId>
 </sourceHost>
 <destinationHost>
   <hostId>host-92</hostId>
   <switchId>dvs-22</switchId>
   <vlanId>54</vlanId>
 </destinationHost>
</testParameters>
```

Working with Logical Switches in a Specific Transport Zone

`GET /api/2.0/vdn/scopes/{scopeId}/virtualwires`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId).

Query Parameters:

`startindex`	The starting point for returning results.
`pagesize`	The number of results to return. Range is 1-1024.

Description:
Retrieve information about all logical switches in the specified transport zone (network scope).

`POST /api/2.0/vdn/scopes/{scopeId}/virtualwires`

URI Parameters:

`scopeId` (required)	A valid transport zone ID (vdnScope objectId).

Description:
Create a logical switch.

To create a universal logical switch use *universalvdnscope* as the scopeId in the URI and send the request to the primary NSX Manager. Request body parameters:

- **name** - Optional. The name of the logical switch.
- **description** - Optional. Description of the logical switch.
- **tenantId** - Required.
- **controlPlaneMode** - Optional. The control plane mode. If not specified, the **controlPlaneMode** of the transport zone is used. It can be one of the following:
  - *UNICAST_MODE*
  - *HYBRID_MODE*
  - *MULTICAST_MODE*
- **guestVlanAllowed** - Optional. Default is *false*.

Request:
Body: application/xml

```
<virtualWireCreateSpec>
 <name>Web-Tier-01</name>
 <description>Web tier network</description>
 <tenantId>virtual wire tenant</tenantId>
 <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
 <guestVlanAllowed>false</guestVlanAllowed>
</virtualWireCreateSpec>
```

Working with Traceflow

For Traceflow to work as expected, make sure that the controller cluster is connected and in healthy state. The Traceflow operation requires active communication between vCenter, NSX Manager, controller cluster, and netcpa User World Agents (UWA) on the host. Traceflow observes marked packet as it traverses overlay network. Each packet is delivered to host VM and monitored as it crosses overlay network until it reaches the destination VM. The packet is never delivered to the destination guest VM. This means that Traceflow packet delivery is successful even when the guest VM is powered down. Unknown L2 Packets are always be sent to the bridge. Typically, the bridge forwards these packets to a VLAN and reports the Traceflow packet as delivered. The packet which is reported as delivered need not necessarily mean that the trace packet was delivered to the destination specified. You should conclude only after validating the observations.vdl2 serves ARP proxy for ARP packets coming from VMs. However, traceflow bypasses this process, hence vdl2 may broadcast the traceflow packet out.

`POST /api/api/2.0/vdn/traceflow`

Description:
Create a traceflow.

Request:
Body: application/xml

```
<traceflowRequest>
 <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
 <timeout>10000</timeout>
 <routed>true</routed>
 <packet class="fieldsPacketData">
   <resourceType>FieldsPacketData</resourceType>
   <ethHeader>
     <srcMac>00:50:56:83:7e:87</srcMac>
     <dstMac>00:50:56:83:fa:6c</dstMac>
     <ethType>2048</ethType>
   </ethHeader>
   <ipHeader>
     <ttl>64</ttl>
     <srcIp>172.32.1.5</srcIp>
     <dstIp>172.34.1.5</dstIp>
   </ipHeader>
 </packet>
</traceflowRequest>
```

Working with a Specific Traceflow

`GET /api/api/2.0/vdn/traceflow/{traceflowId}`

URI Parameters:

`traceflowId`	Traceflow ID.

Description:
Query a specific traceflow by *tracflowId* which is the value returned after executing the create Traceflow API call.

Responses:

Status Code: 200
Body: application/xml

```
<traceflowDto>
 <operState>COMPLETE</operState>
 <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
 <id>00000000-0000-0000-0000-000056b5dec3</id>
 <receivedCount>2</receivedCount>
 <forwardedCount>1</forwardedCount>
 <deliveredCount>1</deliveredCount>
 <logicalReceivedCount>4</logicalReceivedCount>
 <logicalDroppedCount>0</logicalDroppedCount>
 <logicalForwardedCount>4</logicalForwardedCount>
 <timeout>10000</timeout>
 <completeAvailable>true</completeAvailable>
 <result>SUCCESS</result>
 <resultSummary>Traceflow delivered observation(s) reported</resultSummary>
 <srcIp>172.32.1.5</srcIp>
 <srcMac>00:50:56:83:7e:87</srcMac>
 <dstMac>172.34.1.5</dstMac>
 <lifMac>00:50:56:83:fa:6c</lifMac>
</traceflowDto>
```

Traceflow Observations

`GET /api/api/2.0/vdn/traceflow/{traceflowId}/observations`

URI Parameters:

traceflowId	Traceflow ID.

Description:
Retrieve traceflow observations.

Responses:
Status Code: 200
Body: application/xml

```
<traceflowObservations>
 <traceflowObservationsDataPage>
   <pagingInfo>
     <pageSize>100</pageSize>
     <startIndex>0</startIndex>
     <totalCount>12</totalCount>
     <sortOrderAscending>true</sortOrderAscending>
     <sortBy></sortBy>
   </pagingInfo>
   <traceflowObservationReceived>
     <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
     <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
     <hostName>10.146.104.42</hostName>
     <hostId>host-22</hostId>
```

```
    <component>PHYS</component>
    <compDisplayName>vNIC</compDisplayName>
    <hopCount>0</hopCount>
  </traceflowObservationReceived>
  <traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>1</hopCount>
  </traceflowObservationLogicalReceived>
  <traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>2</hopCount>
    <ruleId>1001</ruleId>
  </traceflowObservationLogicalForwarded>
  <traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LS</component>
    <compDisplayName>1-switch-3</compDisplayName>
    <hopCount>3</hopCount>
    <vni>10000</vni>
    <logicalCompId>universalwire-1</logicalCompId>
    <logicalCompName>1-switch-3</logicalCompName>
  </traceflowObservationLogicalForwarded>
  <traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LR</component>
    <compDisplayName>1-vm-3</compDisplayName>
    <hopCount>4</hopCount>
    <vni>10000</vni>
    <lifName>27100000000a</lifName>
    <compId>10000</compId>
    <srcNsxManager>4204ad55-71ec-927b-ca1b-aabfa36863ad</srcNsxManager>
    <srcGlobal>true</srcGlobal>
    <compName>default+edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</compName>
    <logicalCompId>edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</logicalCompId>
    <logicalCompName>1-vm-3</logicalCompName>
    <otherLogicalCompId>universalwire-1</otherLogicalCompId>
    <otherLogicalCompName>1-switch-3</otherLogicalCompName>
  </traceflowObservationLogicalReceived>
  <traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LR</component>
    <compDisplayName>1-vm-3</compDisplayName>
    <hopCount>5</hopCount>
```

```
    <vni>10002</vni>
    <lifName>27100000000c</lifName>
    <compId>10000</compId>
    <compName>default+edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</compName>
    <srcNsxManager>4204ad55-71ec-927b-ca1b-aabfa36863ad</srcNsxManager>
    <srcGlobal>true</srcGlobal>
    <logicalCompId>edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</logicalCompId>
    <logicalCompName>1-vm-3</logicalCompName>
    <otherLogicalCompId>universalwire-3</otherLogicalCompId>
    <otherLogicalCompName>3-switch-98</otherLogicalCompName>
  </traceflowObservationLogicalForwarded>
  <traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LS</component>
    <compDisplayName>3-switch-98</compDisplayName>
    <hopCount>6</hopCount>
    <vni>10002</vni>
    <logicalCompId>universalwire-3</logicalCompId>
    <logicalCompName>3-switch-98</logicalCompName>
  </traceflowObservationLogicalReceived>
  <traceflowObservationForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>PHYS</component>
    <compDisplayName>10.146.104.42</compDisplayName>
    <hopCount>7</hopCount>
    <remoteIpAddress>172.19.172.142</remoteIpAddress>
    <context>5109430534275084</context>
  </traceflowObservationForwarded>
  <traceflowObservationReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>PHYS</component>
    <compDisplayName>10.146.103.3</compDisplayName>
    <hopCount>8</hopCount>
  </traceflowObservationReceived>
  <traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>9</hopCount>
  </traceflowObservationLogicalReceived>
  <traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>10</hopCount>
    <ruleId>1001</ruleId>
  </traceflowObservationLogicalForwarded>
```

```
   <traceflowObservationDelivered>
     <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
     <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
     <hostName>10.146.103.3</hostName>
     <hostId>host-20</hostId>
     <component>PHYS</component>
     <compDisplayName>vNIC</compDisplayName>
     <hopCount>11</hopCount>
     <vlanId>0</vlanId>
   </traceflowObservationDelivered>
 </traceflowObservationsDataPage>
</traceflowObservations>
```

Working with Logical Switches in All Transport Zones

`GET /api/2.0/vdn/virtualwires`

Query Parameters:

`startindex`	The starting point for returning results.
`pagesize`	The number of results to return. Range is 1-1024.

Description:
Retrieve information about all logical switches in all transport zones.

Responses:
Status Code: 200
Body: application/xml

```
<virtualWires>
 <dataPage>
   <pagingInfo>
     <pageSize>20</pageSize>
     <startIndex>0</startIndex>
     <totalCount>13</totalCount>
     <sortOrderAscending>true</sortOrderAscending>
   </pagingInfo>
   <virtualWire>
     <objectId>virtualwire-1</objectId>
     <objectTypeName>VirtualWire</objectTypeName>
     <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
     <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
     <revision>3</revision>
     <type>
       <typeName>VirtualWire</typeName>
     </type>
     <name>Transit-Network-01</name>
     <description></description>
     <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
     <isUniversal>false</isUniversal>
     <universalRevision>0</universalRevision>
     <tenantId>virtual wire tenant</tenantId>
     <vdnScopeId>vdnscope-1</vdnScopeId>
     <vdsContextWithBacking>
       <switch>
         <objectId>dvs-47</objectId>
         <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
         <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
         <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
         <revision>29</revision>
         <type>
           <typeName>VmwareDistributedVirtualSwitch</typeName>
         </type>
         <name>vds-mgt-edge</name>
         <scope>
           <id>datacenter-21</id>
           <objectTypeName>Datacenter</objectTypeName>
           <name>Datacenter Site A</name>
         </scope>
```

```
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </switch>
      <mtu>1600</mtu>
      <promiscuousMode>false</promiscuousMode>
      <backingType>portgroup</backingType>
      <backingValue>dvportgroup-355</backingValue>
      <missingOnVc>false</missingOnVc>
    </vdsContextWithBacking>
    <vdsContextWithBacking>
      <switch>
        <objectId>dvs-35</objectId>
        <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
        ***
      </switch>
      <mtu>1600</mtu>
      <promiscuousMode>false</promiscuousMode>
      <backingType>portgroup</backingType>
      <backingValue>dvportgroup-354</backingValue>
      <missingOnVc>false</missingOnVc>
    </vdsContextWithBacking>
    <vdnId>5000</vdnId>
    <guestVlanAllowed>false</guestVlanAllowed>
    <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
    <ctrlLsUuid>7ad8bc71-5857-475c-af2a-a9e5337b0944</ctrlLsUuid>
    <macLearningEnabled>false</macLearningEnabled>
  </virtualWire>
  <virtualWire>
    <objectId>virtualwire-2</objectId>
    ***
  </virtualWire>
  <virtualWire>
    <objectId>virtualwire-3</objectId>
    ***
  </virtualWire>
  <virtualWire>
    <objectId>virtualwire-4</objectId>
    ***
  </virtualWire>
  <virtualWire>
    <objectId>universalwire-1</objectId>
    ***
  </virtualWire>
  ***
  <virtualWire>
    <objectId>virtualwire-9</objectId>
    ***
  </virtualWire>
 </dataPage>
</virtualWires>
```

Working Virtual Machine Connections to Logical Switches

`POST /api/2.0/vdn/virtualwires/vm/vnic`

Description:
Attach a VM vNIC to, or detach a VM vNIC from a logical switch.

Specify the logical switch ID in the **portgroupId** parameter. To detach a VM vNIC from a logical switch, leave the **portgroupId** parameter empty.

To find the ID of a VM vNIC, do the following:

1 In the vSphere MOB, navigate to the VM you want to connect or disconnect.

2 Click **config** and take note of the **instanceUuid**.

3 Click **hardware** and take note of the last three digits of the appropriate network interface device.

Use these two values to form the VM vNIC ID. For example, if the **instanceUuid** is *502e71fa-1a00-759b-e40f-ce778e915f16* and the appropriate **device** value is *device[4000]*, the **objectId** and **vnicUuid** are both *502e71fa-1a00-759b-e40f-ce778e915f16.000*.

Request:
Body: application/xml

```
<com.vmware.vshield.vsm.inventory.dto.VnicDto>
 <objectId>502e71fa-1a00-759b-e40f-ce778e915f16.000</objectId>
 <vnicUuid>502e71fa-1a00-759b-e40f-ce778e915f16.000</vnicUuid>
 <portgroupId>virtualwire-2</portgroupId>
</com.vmware.vshield.vsm.inventory.dto.VnicDto>
```

Working With a Specific Logical Switch

`GET /api/2.0/vdn/virtualwires/{virtualWireID}`

URI Parameters:

`virtualWireID` (required)	A logical switch id, e.g. virtualwire-1002

Description:
Retrieve information about the specified logical switch.

If the switch is a universal logical switch the **isUniversal** parameter is set to true in the response body.

Responses:
Status Code: 200
Body: application/xml

```
<virtualWire>
 <objectId>universalwire-2</objectId>
 <objectTypeName>VirtualWire</objectTypeName>
 <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
 <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
 <revision>3</revision>
 <type>
   <typeName>VirtualWire</typeName>
 </type>
```

```
<name>ULS-Web-Tier-02</name>
<clientHandle></clientHandle>
<extendedAttributes></extendedAttributes>
<isUniversal>true</isUniversal>
<universalRevision>2</universalRevision>
<tenantId>ULS-Tenant</tenantId>
<vdnScopeId>universalvdnscope</vdnScopeId>
<vdsContextWithBacking>
  <switch>
    <objectId>dvs-35</objectId>
    <objectTypeName>VmwareDistributedVirtualSwitch</objectTypeName>
    <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
    <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
    <revision>29</revision>
    <type>
      <typeName>VmwareDistributedVirtualSwitch</typeName>
    </type>
    <name>vds-site-a</name>
    <scope>
      <id>datacenter-21</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>Datacenter Site A</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </switch>
  <mtu>1600</mtu>
  <promiscuousMode>false</promiscuousMode>
  <backingType>portgroup</backingType>
  <backingValue>dvportgroup-397</backingValue>
  <missingOnVc>false</missingOnVc>
</vdsContextWithBacking>
<vdsContextWithBacking>
  ***
</vdsContextWithBacking>
<vdnId>200001</vdnId>
<guestVlanAllowed>false</guestVlanAllowed>
<controlPlaneMode>UNICAST_MODE</controlPlaneMode>
<ctrlLsUuid>f360d6e5-c709-4aca-b8d1-37de500a867a</ctrlLsUuid>
<macLearningEnabled>false</macLearningEnabled>
</virtualWire>
```

PUT /api/2.0/vdn/virtualwires/{virtualWireID}

URI Parameters:

virtualWireID (required)	A logical switch id, e.g. virtualwire-1002

Description:

Update the specified logical switch.

For example, you can update the name, description, or control plane mode.

Request:

Body: application/xml

```
<virtualWire>
 <name>ULS-Web-Tier-02 </name>
 <description>Universal Web Logical Switch</description>
 <tenantId>virtual wire tenant</tenantId>
 <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
</virtualWire>
```

`DELETE /api/2.0/vdn/virtualwires/{virtualWireID}`

URI Parameters:

`virtualWireID` (required)	A logical switch id, e.g. virtualwire-1002

Description:
Delete the specified logical switch.

Resolving Missing Portgroups for a Logical Switch

`POST /api/2.0/vdn/virtualwires/{virtualWireID}/backing`

URI Parameters:

`virtualWireID` (required)	A logical switch id, e.g. virtualwire-1002

Query Parameters:

`action` (required)	• *remediate*: The *remediate* action performs the *resync* action and then creates a new backing port group for the logical switch. Under normal operations, you should need the *remediate* action only. • *resync*: The *resync* action removes the association between the backing port group and the logical switch in the NSX Manager configuration.

Description:
For every logical switch created, NSX creates a corresponding port group in vCenter. If the port group is missing, the logical switch will stop functioning.

If the port group backing a logical switch is deleted, you can recreate a new backing port group for the logical switch.

Method history:

Release	Modification
6.2.3	Method introduced.

Testing Host Connectivity

`POST /api/2.0/vdn/virtualwires/{virtualWireID}/conn-check/multicast`

URI Parameters:

virtualWireID (required)	A logical switch id, e.g. virtualwire-1002

Description:

Test multicast group connectivity.

Test multicast group connectivity between two hosts connected to the specified logical switch.

Parameter **packetSizeMode** has one of the following values:

- *0* - VXLAN standard packet size
- *1* - minimum packet size
- *2* - customized packet size. If you set **packetSizeMode** to *2*, you must specify the size using the **packetSize** parameter.

Request:

Body: application/xml

```
<testParameters>
 <gateway></gateway>
 <packetSizeMode></packetSizeMode>
 <packetSize></packetSize>
 <sourceHost>
   <hostId></hostId>
   <switchId></switchId>
   <vlanId></vlanId>
 </sourceHost>
 <destinationHost>
   <hostId></hostId>
   <switchId></switchId>
   <vlanId></vlanId>
 </destinationHost>
</testParameters>
```

Testing Point-to-Point Connectivity

`POST /api/2.0/vdn/virtualwires/{virtualWireID}/conn-check/p2p`

URI Parameters:

virtualWireID (required)	A logical switch id, e.g. virtualwire-1002

Description:

Test point-to-point connectivity.

Test point-to-point connectivity between two hosts connected to the specified logical switch.

Parameter **packetSizeMode** has one of the following values:

- *0* - VXLAN standard packet size
- *1* - minimum packet size
- *2* - customized packet size. If you set **packetSizeMode** to *2*, you must specify the size using the **packetSize** parameter.

Request:

Body: application/xml

```
<testParameters>
 <gateway></gateway>
 <packetSizeMode></packetSizeMode>
 <packetSize></packetSize>
 <sourceHost>
   <hostId></hostId>
   <switchId></switchId>
   <vlanId></vlanId>
 </sourceHost>
 <destinationHost>
   <hostId></hostId>
   <switchId></switchId>
   <vlanId></vlanId>
 </destinationHost>
</testParameters>
```

Working with Hardware Gateway Bindings for a Specific Logical Switch

`GET /api/2.0/vdn/virtualwires/{virtualWireID}/hardwaregateways`

URI Parameters:

`virtualWireID` (required)	A logical switch id, e.g. virtualwire-1002

Description:
Retrieve hardware gateway bindings for the specified logical switch.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<list>
 <hardwareGatewayBinding>
   <id>torbinding-2</id>
   <hardwareGatewayId>torgateway-1</hardwareGatewayId>
   <switchName>1-switch-579</switchName>
   <portname>p1</portname>
   <vlan>0</vlan>
   <virtualWire>virtualwire-1</virtualWire>
   <vni>5342</vni>
 </hardwareGatewayBinding>
 <hardwareGatewayBinding>
   <id>torbinding-1</id>
   <hardwareGatewayId>torgateway-2</hardwareGatewayId>
   <switchName>1-switch-104</switchName>
```

```
   <portname>p1</portname>
   <vlan>0</vlan>
   <virtualWire>virtualwire-1</virtualWire>
   <vni>5342</vni>
 </hardwareGatewayBinding>
</list>
```

Working with Connections Between Hardware Gateways and Logical Switches

POST
/api/2.0/vdn/virtualwires/{virtualWireID}/hardwaregateways/{hardwareGatewayBindingId}

URI Parameters:

hardwareGatewayBindingId	Hardware Gateway Binding ID.
virtualWireID (required)	A logical switch id, e.g. virtualwire-1002

Query Parameters:

action (optional)	Specify *attach* to attach a hardware gateway to a logical switch. Specify *detach* to detach a hardware gateway from a logical switch.

Description:
Manage the connection between a hardware gateway and a logical switch.

Attach a hardware gateway to a logical switch and create a new binding with the information provided

`POST /api/2.0/vdn/virtualwires/{virtualwireid}/hardwaregateways`

```
<hardwareGatewayBinding>
 <hardwareGatewayId>hardwarewgateway1</hardwareGatewayId>
 <vlan>v1</vlan>
 <switchName>s1</switchName>
 <portName>s1</portName>
</hardwareGatewayBinding>
```

Attach a hardware gateway to a logical switch, specifying an existing binding by ID

POST /api/2.0/vdn/virtualwires/<virtualwireId>/hardwaregateways/{bindingId}?action=attach

```
<virtualWire>
 ***
 <hardwareGatewayBindings>
   <hardwareGatewayBinding>
     <id>binding id</id>
   </hardwareGatewayBinding>
 </hardwareGatewayBindings>
</virtualWire>
```

Detach a hardware gateway from a logical switch

POST /api/2.0/vdn/virtualwires/<virtualwireId>/hardwaregateways/{bindingId}?action=detach

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewayBinding>
 <hardwareGatewayId>hardwarewgateway1</hardwareGatewayId>
 <vlan>v1</vlan>
 <switchName>s1</switchName>
 <portName>s1</portName>
</hardwareGatewayBinding>
```

Working with IP Discovery and MAC Learning for Logical Switches

You can enable IP discovery (ARP suppression) and MAC learning for logical switches or dvPortGroup. Enabling MAC learning builds a VLAN - MAC pair learning table on each vNic.

This table is stored as part of the dvfilter data. During vMotion, dvfilter saves/restores the table at the new location. The switch then issues RARPs for all the VLAN - MAC entries in the table.

Enabling this feature avoids possible traffic loss during vMotion in the following cases:
- the vNic is in VLAN trunk mode
- the VM is using more than one unicast MAC address. Since Etherswitch supports only one unicast MAC per vNic, RARP is not processed.

When a logical switch is created using the API, IP discovery is enabled, and MAC learning is disabled.

In cross-vCenter NSX, the following applies:
- The MAC learning setting for a universal logical switch is managed on the primary NSX Manager. Any changes are synchronized to all secondary NSX Managers.
- The IP discovery setting for a universal logical switch is managed separately on each NSX Manager.

Note: In NSX 6.2.2 and earlier you cannot disable IP discovery for universal logical switches on secondary NSX Managers.

```
GET /api/2.0/xvs/networks/{ID}/features
```

URI Parameters:

ID (required)	dvPortGroup MOID or logical switch (virtual wire) ID.

Description:
Retrieve IP discovery and MAC learning information.

```
PUT /api/2.0/xvs/networks/{ID}/features
```

URI Parameters:

ID (required)	dvPortGroup MOID or logical switch (virtual wire) ID.

Description:
Enable or disable IP discovery and MAC learning.

Method history:

Release	Modification
6.2.3	Method updated. IP discovery can be disabled on secondary NSX Managers.

Request:
Body: application/xml

```
<networkFeatureConfig>
 <ipDiscoveryConfig>
   <enabled></enabled>
 </ipDiscoveryConfig>
 <macLearningConfig>
   <enabled></enabled>
```

```
 </macLearningConfig>
</networkFeatureConfig>
```

Working with NSX Controllers

For the unicast or hybrid control plane mode, you must add an NSX controller to manage overlay transport and provide East-West routing. The controller optimizes virtual machine broadcast (ARP only) traffic, and the learning is stored on the host and the controller.

`GET /api/2.0/vdn/controller`

Description:

Retrieves details and runtime status for all controllers. Runtime status can be one of the following:

- **Deploying** - controller is being deployed and the procedure has not completed yet.
- **Removing** - controller is being removed and the procedure has not completed yet.
- **Running** - controller has been deployed and can respond to API invocation.
- **Unknown** - controller has been deployed but fails to respond to API invocation.

Responses:

Status Code: 200

Body: application/xml

```
<controllers>
 <controller>
   <id></id>
   <name></name>
   <description></description>
   <ipAddress></ipAddress>
   <status></status>
 </controller>
</controllers>
```

`POST /api/2.0/vdn/controller`

Description:

Adds a new NSX controller on the specified given cluster. The *hostId* parameter is optional. The *resourcePoolId* can be either the *clusterId* or *resourcePoolId*.

The IP address of the controller node will be allocated from the specified IP pool. The *deployType* property determines the controller node memory size and can be small, medium, or large. However, different controller deployment types are not currently supported because the OVF overrides it and different OVF types require changes in the manager build scripts. Despite not being supported, an arbitrary *deployType* size must still be specified or an error will be returned. Request without body to upgrade controller cluster.

Request:

Body: application/xml

```
<controllerSpec>
 <name></name>
 <description></description>
 <ipPoolId></ipPoolId>
 <resourcePoolId></resourcePoolId>
 <hostId></hostId>
 <datastoreId></datastoreId>
 <deployType></deployType>
 <networkId></networkId>
```

```
 <password></password>
</controllerSpec>
```

Working With Controller Upgrade Availability

`GET /api/2.0/vdn/controller/upgrade-available`

Description:
Retrieve controller upgrade availability.

Working With of Controller Job Status

`GET /api/2.0/vdn/controller/progress/{jobId}`

URI Parameters:

jobId (required)	Specified job Id

Description:
Retrieves status of controller creation or removal. The progress gives a percentage indication of current deploy / remove procedure.

Responses:
Status Code: 200
Body: application/xml

```
<controllerDeploymentInfo>
 <vmId></vmId>
 <progress></progress>
 <status></status>
 <exceptionMessage></exceptionMessage>
</controllerDeploymentInfo>
```

Working with a Specific Controller

`POST /api/2.0/vdn/controller/{controllerId}`

URI Parameters:

`controllerId` (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Query Parameters:

`action` (required)	Specify *remediate* to recover from controller shutdown or deletion.

Description:
If you power off or delete a controller from vCenter, NSX Manager detects the change in controller status. You can remediate the controller, which will power on a powered off controller, or remove the controller from the NSX Manager database if the controller is deleted.

Method history:

Release	Modification
6.2.3	Method introduced.

```
DELETE /api/2.0/vdn/controller/{controllerId}
```

URI Parameters:

`controllerId` (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Query Parameters:

`forceRemoval` (required)	Specify whether to force removal of controller. Must be set to true to remove last controller of the controller cluster.

Description:
Delete the NSX controller.

Working With NSX Controller System Statistics

```
GET /api/2.0/vdn/controller/{controllerId}/systemStats
```

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Retrieve NSX Controller system statistics.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<controllerNodeStatus>
 <id>controller-2</id>
 <ipAddress>192.168.110.32</ipAddress>
 <syncTime>1490991545530</syncTime>
 <cpuCoreCount>2</cpuCoreCount>
 <cpuLoadInfo>
   <interval>1</interval>
   <averageLoad>0.17</averageLoad>
 </cpuLoadInfo>
 <cpuLoadInfo>
   <interval>5</interval>
   <averageLoad>0.6</averageLoad>
 </cpuLoadInfo>
 <cpuLoadInfo>
   <interval>15</interval>
   <averageLoad>0.4</averageLoad>
 </cpuLoadInfo>
 <totalMemory>1924280</totalMemory>
 <usedMemory>1542524</usedMemory>
 <cachedMemory>589196</cachedMemory>
 <totalSwap>4190204</totalSwap>
 <usedSwap>0</usedSwap>
 <systemTime>1490991545521</systemTime>
 <upTime>433880</upTime>
 <nodeFailoverReady>false</nodeFailoverReady>
 <nodeDiskLatencyStatus>
   <deviceName>sda</deviceName>
   <refreshTime>1490991404000</refreshTime>
   <latencyType>w_await</latencyType>
   <lastLatency>97.0</lastLatency>
   <avgLatency>28.572</avgLatency>
```

```
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
  <nodeDiskLatencyStatus>
    <deviceName>sda</deviceName>
    <refreshTime>1490991186000</refreshTime>
    <latencyType>r_await</latencyType>
    <lastLatency>9.18</lastLatency>
    <avgLatency>0.0</avgLatency>
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
  <nodeDiskLatencyStatus>
    <deviceName>dm-1</deviceName>
    <refreshTime>1490991185000</refreshTime>
    <latencyType>w_await</latencyType>
    <lastLatency>0.0</lastLatency>
    <avgLatency>0.0</avgLatency>
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
  <nodeDiskLatencyStatus>
    <deviceName>dm-1</deviceName>
    <refreshTime>1490991185000</refreshTime>
    <latencyType>r_await</latencyType>
    <lastLatency>51.51</lastLatency>
    <avgLatency>0.0</avgLatency>
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
  <nodeDiskLatencyStatus>
    <deviceName>dm-0</deviceName>
    <refreshTime>1490991404000</refreshTime>
    <latencyType>w_await</latencyType>
    <lastLatency>129.33</lastLatency>
    <avgLatency>34.16</avgLatency>
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
  <nodeDiskLatencyStatus>
    <deviceName>dm-0</deviceName>
    <refreshTime>1490991225000</refreshTime>
    <latencyType>r_await</latencyType>
    <lastLatency>0.0</lastLatency>
    <avgLatency>12.678</avgLatency>
    <alertEnabled>false</alertEnabled>
  </nodeDiskLatencyStatus>
</controllerNodeStatus>
```

Working with Controller Tech Support Logs

`GET /api/2.0/vdn/controller/{controllerId}/techsupportlogs`

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Retrieve controller logs. Response content type is application/octet-stream and response header is filename. This streams a fairly large bundle back (possibly hundreds of MB).

Working with Controller Syslog

`GET /api/2.0/vdn/controller/{controllerId}/syslog`

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Retrieve details about the syslog exporter on the controller.

Responses:
Status Code: 200
Body: application/xml

```
<controllerSyslogServer>
 <syslogServer></syslogServer>
 <port></port>
 <protocol></protocol>
 <level></level>
</controllerSyslogServer>
```

`POST /api/2.0/vdn/controller/{controllerId}/syslog`

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Add controller syslog exporter on the controller.

Request:
Body: application/xml

```
<controllerSyslogServer>
 <syslogServer></syslogServer>
 <port></port>
 <protocol></protocol>
 <level></level>
</controllerSyslogServer>
```

`DELETE /api/2.0/vdn/controller/{controllerId}/syslog`

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Deletes syslog exporter on the specified controller node.

Working with Controller Cluster Snapshots

`GET /api/2.0/vdn/controller/{controllerId}/snapshot`

URI Parameters:

controllerId (required)	Specified controller ID. To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation > Management, and view the **NSX Controller nodes** section. The controller ID is listed in the **Controller ID** or **Controller Node** column, depending on NSX version. An example controller ID is *controller-1*. In a cross-vCenter NSX environment, retrieve the controller IDs from rows where the NSX Manager column contains the primary NSX Manager IP address.

Description:
Take a snapshot of the control cluster from the specified controller node.

Working with the NSX Controller Cluster Configuration

`GET /api/2.0/vdn/controller/cluster`

Description:
Retrieve cluster wide configuration information for controller.

Responses:
Status Code: 200
Body: application/xml

```
<controllerConfig>
 <sslEnabled></sslEnabled>
</controllerConfig>
```

`PUT /api/2.0/vdn/controller/cluster`

Description:
Modify cluster wide configuration information for controller.

Request:
Body: application/xml

```
<controllerConfig>
 <sslEnabled></sslEnabled>
</controllerConfig>
```

Working with the NSX Controller Password

PUT /api/2.0/vdn/controller/credential

Description:
Change the NSX controller password.

Request:
Body: application/xml

```
<controllerCredential>
 <apiPassword></apiPassword>
</controllerCredential>
```

Working with Services Grouping Objects

Retrieve Services from a Specific Scope

`GET /api/2.0/services/application/scope/{scopeId}`

URI Parameters:

`scopeId` (required)	Can be "globalroot-0", "universalroot-0" or datacenterId in upgrade use cases.

Description:
Retrieve services that have been created on the specified scope.

Create a Service on a Specific Scope

`POST /api/2.0/services/application/{scopeId}`

Description:
Create a new service on the specified scope.

Request:
Body: application/xml

```
<application>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <element>
   <applicationProtocol>mandatory</applicationProtocol>
   <value>mandatory</value>
 </element>
</application>
```

Working With a Specified Service

`GET /api/2.0/services/application/{applicationId}`

URI Parameters:

applicationId (required)	Application ID. You can get a list of application IDs from GET /api/2.0/services/application/scope/{scopeId}.

Description:
Retrieve details about the specified service. The scopeId can be "globalroot-0", or datacenterId in upgrade use cases.

`PUT /api/2.0/services/application/{applicationId}`

URI Parameters:

applicationId (required)	Application ID. You can get a list of application IDs from GET /api/2.0/services/application/scope/{scopeId}.

Description:
Modify the name, description, applicationProtocol, or port value of a service.

Request:
Body: application/xml

```
<application>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <element>
   <applicationProtocol></applicationProtocol>
   <value></value>
 </element>
</application>
```

`DELETE /api/2.0/services/application/{applicationId}`

URI Parameters:

applicationId (required)	Application ID. You can get a list of application IDs from GET /api/2.0/services/application/scope/{scopeId}.

Query Parameters:

`force` (optional)	Determines if the delete should be forced or unforced. Default is *false*. If *true*, the object is deleted even if it is in use in other places such as firewall rules, which invalidates other configurations referring to the deleted object. If *false*, the object is deleted only if it is not being used by any other configuration.

Description:

Delete the specified service.You can delete a service by specifying its .

Working with Service Groups Grouping Objects

Working with Service Groups on a Specific Scope

`GET /api/2.0/services/applicationgroup/scope/{scopeId}`

URI Parameters:

scopeId (required)	The scopeId can be "globalroot-0", "universalroot-0" or datacenterId in upgrade use cases

Description:
Retrieve a list of service groups that have been created on the scope.

`POST /api/2.0/services/applicationgroup/scope/{scopeId}`

URI Parameters:

scopeId (required)	The scopeId can be "globalroot-0", "universalroot-0" or datacenterId in upgrade use cases

Description:
Create a new service group on the specified scope.

Request:
Body: application/xml

```
<applicationGroup>
 <description></description>
 <name></name>
 <revision></revision>
 <inheritanceAllowed></inheritanceAllowed>
</applicationGroup>
```

Working with a Specific Service Group

`GET /api/2.0/services/applicationgroup/{applicationgroupId}`

URI Parameters:

applicationgroupId (required)	Application group ID

Description:
Retrieve details about the specified service group.

PUT /api/2.0/services/applicationgroup/{applicationgroupId}

URI Parameters:

applicationgroupId (required)	Application group ID

Description:
Modify the name, description, applicationProtocol, or port value of the specified service group.

Request:
Body: application/xml

```
<applicationGroup>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <name></name>
 <description></description>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <scope>
   <id></id>
   <objectTypeName></objectTypeName>
   <name></name>
 </scope>
 <extendedAttributes></extendedAttributes>
 <inheritanceAllowed></inheritanceAllowed>
 <member>
   <objectId></objectId>
   <type>
     <typeName></typeName>
   </type>
   <name></name>
   <revision></revision>
   <objectTypeName></objectTypeName>
   <scope>
     <id></id>
     <objectTypeName></objectTypeName>
     <name></name>
   </scope>
 </member>
</applicationGroup>
```

DELETE /api/2.0/services/applicationgroup/{applicationgroupId}

URI Parameters:

applicationgroupId (required)	Application group ID

Query Parameters:

force (optional)	Determines if the delete should be forced or unforced. Default is *false*. If *true*, the object is deleted even if it is in use in other places such as firewall rules, which invalidates other configurations referring to the deleted object. If *false*, the object is deleted only if it is not being used by any other configuration.

Description:
Delete the specified service group (application group) from a scope.

Working with a Specific Service Group Member

`PUT /api/2.0/services/applicationgroup/{applicationgroupId}/members/{moref}`

URI Parameters:

moref (required)	Managed object reference to the member.
applicationgroupId (required)	Application group ID

Description:
Add a member to the service group.

`DELETE /api/2.0/services/applicationgroup/{applicationgroupId}/members/{moref}`

URI Parameters:

moref (required)	Managed object reference to the member.
applicationgroupId (required)	Application group ID

Description:
Delete a member from the service group.

Working with Service Group Members on a Specific Scope

`GET /api/2.0/services/applicationgroup/scope/{scopeId}/members`

URI Parameters:

scopeId (required)	globalroot-0 or datacenterId in upgrade use cases

Description:
Get a list of member elements that can be added to the service groups created on a particular scope.

Working with IP Pool Grouping Objects

Working with IP Pools on a Specific Scope

`GET /api/2.0/services/ipam/pools/scope/{scopeId}`

URI Parameters:

scopeId (required)	For *scopeID* use globalroot-0 or *datacenterId* in upgrade use cases.

Description:
Retrieves all IP pools on the specified scope where the *scopeID* is the reference to the desired scope. An example of the *scopeID* is globalroot-0.

Responses:
Status Code: 200
Body: application/xml

```
<ipamAddressPool>
 <objectId>ipaddresspool-1</objectId>
 <objectTypeName>IpAddressPool</objectTypeName>
 <vsmUuid>4237BA90-C373-A71A-9827-1673BFA29498</vsmUuid>
 <revision>1</revision>
 <type>
   <typeName>IpAddressPool</typeName>
 </type>
 <name>rest-ip-pool-1</name>
 <extendedAttributes></extendedAttributes>
 <prefixLength>23</prefixLength>
 <gateway>192.168.1.1</gateway>
 <dnsSuffix>example.com</dnsSuffix>
 <dnsServer1>10.11.0.1</dnsServer1>
 <dnsServer2>10.11.0.2</dnsServer2>
 <ipRanges>
   <ipRangeDto>
     <id>iprange-1</id>
     <startAddress>192.168.1.2</startAddress>
     <endAddress>192.168.1.3</endAddress>
   </ipRangeDto>
 </ipRanges>
 <totalAddressCount>2</totalAddressCount>
 <usedAddressCount>0</usedAddressCount>
 <usedPercentage>0</usedPercentage>
</ipamAddressPool>
```

`POST /api/2.0/services/ipam/pools/scope/{scopeId}`

URI Parameters:

scopeId (required)	For *scopeID* use globalroot-0 or *datacenterId* in upgrade use cases.

Description:
Create a pool of IP addresses. For *scopeId* use globalroot-0 or the *datacenterId* in upgrade use cases.

Request:
Body: application/xml

```
<ipamAddressPool>
 <name></name>
 <prefixLength></prefixLength>
 <gateway></gateway>
 <dnsSuffix></dnsSuffix>
 <dnsServer1></dnsServer1>
 <dnsServer2></dnsServer2>
 <ipRanges>
   <ipRangeDto>
     <startAddress></startAddress>
     <endAddress></endAddress>
   </ipRangeDto>
 </ipRanges>
</ipamAddressPool>
```

Working with a Specific IP Pool

`GET /api/2.0/services/ipam/pools/{poolId}`

URI Parameters:

poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
Retrieve details about a specific IP pool.

Responses:
Status Code: 200
Body: application/xml

```
<ipamAddressPool>
 <objectId>ipaddresspool-1</objectId>
 <objectTypeName>IpAddressPool</objectTypeName>
 <vsmUuid>4237BA90-C373-A71A-9827-1673BFA29498</vsmUuid>
 <revision>1</revision>
 <type>
   <typeName>IpAddressPool</typeName>
 </type>
 <name>rest-ip-pool-1</name>
 <extendedAttributes></extendedAttributes>
 <prefixLength>23</prefixLength>
 <gateway>192.168.1.1</gateway>
```

```
 <dnsSuffix>example.com</dnsSuffix>
 <dnsServer1>10.11.0.1</dnsServer1>
 <dnsServer2>10.11.0.2</dnsServer2>
 <ipRanges>
   <ipRangeDto>
     <id>iprange-1</id>
     <startAddress>192.168.1.2</startAddress>
     <endAddress>192.168.1.3</endAddress>
   </ipRangeDto>
 </ipRanges>
 <totalAddressCount>2</totalAddressCount>
 <usedAddressCount>0</usedAddressCount>
 <usedPercentage>0</usedPercentage>
</ipamAddressPool>
```

PUT /api/2.0/services/ipam/pools/{poolId}

URI Parameters:

poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
To modify an IP pool, query the IP pool first. Then modify the output and send it back as the request body.

Request:
Body: application/xml

```
<ipamAddressPool>
 <objectId></objectId>
 <objectTypeName></objectTypeName>
 <vsmUuid></vsmUuid>
 <revision></revision>
 <type>
   <typeName></typeName>
 </type>
 <name></name>
 <extendedAttributes></extendedAttributes>
 <prefixLength></prefixLength>
 <gateway></gateway>
 <dnsSuffix></dnsSuffix>
 <dnsServer1></dnsServer1>
 <dnsServer2></dnsServer2>
 <ipRanges>
   <ipRangeDto>
     <id></id>
     <startAddress></startAddress>
     <endAddress></endAddress>
   </ipRangeDto>
 </ipRanges>
</ipamAddressPool>
```

DELETE /api/2.0/services/ipam/pools/{poolId}

URI Parameters:

poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
Delete an IP pool.

Working with IP Pool Address Allocations

`GET /api/2.0/services/ipam/pools/{poolId}/ipaddresses`

URI Parameters:

poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
Retrieves all allocated IP addresses from the specified pool.

Responses:
Status Code: 200
Body: application/xml

```
<allocatedIpAddresses>
 <allocatedIpAddress>
   <id>allocatedipaddress-4</id>
   <ipAddress>192.168.1.2</ipAddress>
   <gateway>192.168.1.1</gateway>
   <prefixLength>23</prefixLength>
   <dnsServer1>10.112.0.1</dnsServer1>
   <dnsServer2>10.112.0.2</dnsServer2>
   <dnsSuffix>eng.vmware.com</dnsSuffix>
   <allocationNote>sample note</allocationNote>
 </allocatedIpAddress>
</allocatedIpAddresses>
```

`POST /api/2.0/services/ipam/pools/{poolId}/ipaddresses`

URI Parameters:

poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
Allocate an IP Address from the pool. Use *ALLOCATE* in the **allocationMode** field in the body to allocate the next available IP. To allocate a specific one use *RESERVE* and pass the IP to reserve in the **ipAddress** fields in the body.

Request:
Body: application/xml

```
<ipAddressRequest>
 <allocationMode>ALLOCATE</allocationMode>
 <ipAddress>192.168.1.2</ipAddress>
```

```
</ipAddressRequest>
```

Responses:
Status Code: 200
Body: application/xml

```
<allocatedIpAddress>
 <id>allocatedipaddress-1</id>
 <ipAddress>192.168.1.2</ipAddress>
 <gateway>192.168.1.1</gateway>
 <prefixLength>23</prefixLength>
 <dnsServer1>10.112.0.1</dnsServer1>
 <dnsServer2>10.112.0.2</dnsServer2>
 <dnsSuffix>eng.vmware.com</dnsSuffix>
 <allocationNote>sample note</allocationNote>
</allocatedIpAddress>
```

Working with Specific IPs Allocated to an IP Pool

`DELETE /api/2.0/services/ipam/pools/{poolId}/ipaddresses/{ipAddress}`

URI Parameters:

ipAddress (required)	The IP address to release, e.g. '192.168.10.10'
poolId (required)	Specifiy the pool ID as *poolId* in the URI.

Description:
Release an IP address allocation in the pool.

Working with Licensing Capacity

The licensing capacity usage API command reports usage of CPUs, VMs and concurrent users for the distributed firewall and VXLAN.

`GET /api/2.0/services/licensing/capacityusage`

Description:

Retrieve capacity usage information on the usage of CPUs, VMs and concurrent users for the distributed firewall and VXLAN.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:

Status Code: 200

Body: application/xml

```
<featureCapacityUsageList>
 <featureCapacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>CPU_CAPACITY_TYPE</capacityType>
     <usageCount>16</usageCount>
   </capacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>VM_CAPACITY_TYPE</capacityType>
     <usageCount>3</usageCount>
   </capacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>CONCURRENT_USER_CAPACITY_TYPE</capacityType>
     <usageCount>3</usageCount>
   </capacityUsageInfo>
   <feature>dfw</feature>
 </featureCapacityUsageInfo>
 <featureCapacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>CPU_CAPACITY_TYPE</capacityType>
     <usageCount>16</usageCount>
   </capacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>VM_CAPACITY_TYPE</capacityType>
     <usageCount>3</usageCount>
   </capacityUsageInfo>
   <capacityUsageInfo>
     <capacityType>CONCURRENT_USER_CAPACITY_TYPE</capacityType>
     <usageCount>3</usageCount>
   </capacityUsageInfo>
   <feature>vxlan</feature>
 </featureCapacityUsageInfo>
</featureCapacityUsageList>
```

Working with Security Tags

You can manage security tags and their virtual machine assignments. For example, you can create a user defined security tag, assign tags to a virtual machine, view tags assigned to virtual machines, and view virtual machines that have a specific tag assigned.

Managing Security Tags

`GET /api/2.0/services/securitytags/tag`

Query Parameters:

`isUniversal (optional)`	Set to *true* to view universal security tags only. Set to *false* to view security tags local to that NSX Manager only. To view all tags (tags local to that NSX Manager plus universal tags), omit the action parameter.

Description:
Retrieve all security tags.

Method history:

Release	Modification
6.3.0	Method updated. Added **isUniversal** query parameter to filter universal security tags.

`POST /api/2.0/services/securitytags/tag`

Description:
Create a new security tag.

Method history:

Release	Modification
6.3.0	Method updated. **isUniversal** parameter can be set to create a universal security tag.

Request:
Body: application/xml

```
<securityTag>
 <objectTypeName>SecurityTag</objectTypeName>
 <type>
   <typeName>SecurityTag</typeName>
 </type>
 <name>TAG_NAME</name>
 <isUniversal>true</isUniversal>
 <description>description of the tag</description>
 <extendedAttributes></extendedAttributes>
</securityTag>
```

Delete a Security Tag

`DELETE /api/2.0/services/securitytags/tag/{tagId}`

URI Parameters:

`tagId` (required)	Specified security tag.

Description:
Delete the specified security tag.

Working With Virtual Machines on a Specific Security Tag

`GET /api/2.0/services/securitytags/tag/{tagId}/vm`

URI Parameters:

`tagId` (required)	Specified security tag.

Description:
Retrieve the list of VMs that have the specified tag attached to them.

`POST /api/2.0/services/securitytags/tag/{tagId}/vm`

URI Parameters:

`tagId` (required)	Specified security tag.

Query Parameters:

`action` (required)	Action to perform: *attach* or *detach* specified security tag from the VMs listed in the request body.

Description:
Attach or detach a security tag to a virtual machine.

This operation does not check that the virtual machine exists in the local inventory. This allows you to attach a universal security tag to a virtual machine that is connected to a secondary NSX Manager (and therefore is not connected to the primary NSX Manager where the call is sent).

Possible keys for the tagParameter are:
- instance_uuid
- bios_uuid
- vmname

Method history:

Release	Modification
6.3.0	Method introduced.

Request:
Body: application/xml

```
<securityTagAssignment>
 <tagParameter>
  <key>instance_uuid</key>
  <value>123e4567-e89b-12d3-a456-426655440000</value>
 </tagParameter>
</securityTagAssignment>
```

Manage a Security Tag on a Virtual Machine

`PUT /api/2.0/services/securitytags/tag/{tagId}/vm/{vmId}`

URI Parameters:

vmId (required)	Specify VM using VM managed object ID or VM instance UUID.
tagId (required)	Specified security tag.

Description:
Apply a security tag to the specified virtual machine.

Note: this method can attach a universal security tag to a virtual machine. However, this method checks that the VM exists on the NSX Manager to which the API call is sent. In a cross-vCenter active active environment, the VM might exist on a secondary NSX Manager, and so the call would fail.

You can instead use the `POST /api/2.0/services/securitytags/tag/{tagId}/vm?action=attach` method to attach universal security tags to a VM that is not local to the primary NSX Manager. This method does not check that the VM is local to the NSX Manager.

`DELETE /api/2.0/services/securitytags/tag/{tagId}/vm/{vmId}`

URI Parameters:

vmId (required)	Specify VM using VM managed object ID or VM instance UUID.
tagId (required)	Specified security tag.

Description:
Detach a security tag from the specified virtual machine.

Working with Virtual Machine Details for a Specific Security Tag

`GET /api/2.0/services/securitytags/tag/{tagId}/vmDetail`

URI Parameters:

tagId (required)	Specified security tag.

Description:
Retrieve details about the VMs that are attached to the specified security tag.

Method history:

Release	Modification
6.3.0	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<attachedVMList>
 <attachedVM>
   <objectId>vm-17</objectId>
   <objectTypeName>VirtualMachine</objectTypeName>
   <vsmUuid>564D5E43-1A21-9061-CE62-16E4E64FBC52</vsmUuid>
   <revision>1</revision>
   <type>
     <typeName>VirtualMachine</typeName>
   </type>
   <name>Ubuntu2</name>
   <scope>
     <id>domain-c7</id>
     <objectTypeName>ClusterComputeResource</objectTypeName>
     <name>sp cluster</name>
   </scope>
   <clientHandle></clientHandle>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
   <instanceUUID>520932b3-b531-7b4a-d3fe-76f0fdd82736</instanceUUID>
   <biosUUID>423f7c14-6463-8ebc-d06d-2a284b24cabb</biosUUID>
 </attachedVM>
 <attachedVM>
   <objectId>vm-59</objectId>
   <objectTypeName>VirtualMachine</objectTypeName>
   <vsmUuid>564D5E43-1A21-9061-CE62-16E4E64FBC52</vsmUuid>
   <revision>1</revision>
   <type>
     <typeName>VirtualMachine</typeName>
   </type>
   <name>vShield-FW (1)</name>
   <scope>
     <id>domain-c7</id>
     <objectTypeName>ClusterComputeResource</objectTypeName>
     <name>sp cluster</name>
   </scope>
   <clientHandle></clientHandle>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
   <instanceUUID>502777a8-a4b0-6b1e-1af1-6ab43f3417a0</instanceUUID>
   <biosUUID>42278ffc-021c-cd1f-1413-978f34079593</biosUUID>
 </attachedVM>
</attachedVMList>
```

Working With Security Tags on a Specific Virtual Machine

`GET /api/2.0/services/securitytags/vm/{vmId}`

URI Parameters:

`vmId`	Specify VM using VM managed object ID or VM instance UUID.

Description:
Retrieve all security tags associated with the specified virtual machine.

`POST /api/2.0/services/securitytags/vm/{vmId}`

URI Parameters:

`vmId`	Specify VM using VM managed object ID or VM instance UUID.

Query Parameters:

`action` (required)	Action to perform. *ASSIGN_TAGS* or *CLEAR_ALL_TAGS*.

Description:
Update security tags associated with the specified virtual machine.

You can assign multiple tags at a time to the specified VM, or clear all assigned tags from the specified VM.

Method history:

Release	Modification
6.3.0	Method introduced.

Request:
Body: application/xml

```
<securityTags>
 <securityTag>
   <objectId>securitytag-12</objectId>
 </securityTag>
 <securityTag>
   <objectId>securitytag-13</objectId>
 </securityTag>
 <securityTag>
   <objectId>securitytag-14</objectId>
 </securityTag>
</securityTags>
```

Working with Security Tags Unique ID Selection Criteria

In NSX versions before 6.3.0, security tags are local to a NSX Manager, and are mapped to VMs using the VM's managed object ID.

In NSX 6.3.0 and later, you can create universal security tags to use in all NSX Managers in a cross-vCenter NSX environment.

In an active standby environment, the managed object ID for a given VM might not be the same in the active and standby datacenters. NSX 6.3.x introduces a Unique ID Selection Criteria on the primary NSX Manager to use to identify VMs when attaching them to universal security tags only. You can use them singly or in combination. The VM instance UUID is the recommended selection criteria. See the descriptions for more information.

The default value for the selection criteria is null and must be set before assigning a universal security tag to a VM. The selection criteria can be set only on the primary NSX manager and is read-only on secondary NSX Managers.

Security Tag Assignment Metadata Parameter	Description
instance_uuid	The VM instance UUID is generally unique within a vCenter domain, however there are exceptions such as when deployments are made through snapshots. If the VM instance UUID is not unique, you can use the VM BIOS UUID in combination with the VM name.
bios_uuid	The BIOS UUID is not guaranteed to be unique within a vCenter domain, but it is always preserved in case of disaster. Use BIOS UUID in combination with VM name to reduce the chance of a duplicate ID.
vmname	If all of the VM names in an environment are unique, then VM name can be used to identify a VM across vCenters. Use VM name in combination with VM BIOS UUID to reduce the chance of a duplicate ID.

`GET /api/2.0/services/securitytags/selection-criteria`

Description:
Retrieve unique ID section criteria configuration.

Method history:

Release	Modification
6.3.0	Method introduced.

`PUT /api/2.0/services/securitytags/selection-criteria`

Description:
Configure the unique ID section criteria configuration.

If you set the selection criteria and assign security tags to VMs, you must remove all security tags from VMs before you can change the selection criteria.

Method history:

Release	Modification
6.3.0	Method introduced.

Request:
Body: application/xml

```
<securityTagAssignmentMetadata>
 <metadata>instance_uuid</metadata>
</securityTagAssignmentMetadata>
```

Working with NSX Manager SSO Registration

`GET /api/2.0/services/ssoconfig`

Description:
Retrieve SSO Configuration.

Responses:
Status Code: 200
Body: application/xml

```
<ssoConfig>
 <ssoLookupServiceUrl>https://vc-l-01a.corp.local:443/lookservice/sdk</ssoLookupServiceUrl>
 <ssoAdminUsername>administrator@vsphere.local</ssoAdminUsername>
</ssoConfig>
```

`POST /api/2.0/services/ssoconfig`

Description:
Register NSX Manager to SSO Services.

Request:
Body: application/xml

```
<ssoConfig>
 <ssoLookupServiceUrl></ssoLookupServiceUrl>
 <ssoAdminUsername></ssoAdminUsername>
 <ssoAdminUserpassword></ssoAdminUserpassword>
 <certificateThumbprint></certificateThumbprint>
</ssoConfig>
```

`DELETE /api/2.0/services/ssoconfig`

Description:
Deletes the NSX Manager SSO Configuration.

Working with SSO Configuration Status

`GET /api/2.0/services/ssoconfig/status`

Description:
Retrieve the SSO configuration status of NSX Manager.

Working with User Management

Manage Users on NSX Manager

`GET /api/2.0/services/usermgmt/user/{userId}`

URI Parameters:

userId (required)	user ID

Description:
Get information about a user.

`DELETE /api/2.0/services/usermgmt/user/{userId}`

URI Parameters:

userId (required)	user ID

Description:
Remove the NSX role for a vCenter user.

Manage NSX Roles for Users

`GET /api/2.0/services/usermgmt/role/{userId}`

URI Parameters:

userId (required)	User to retrieve role information from.

Description:
Retrieve a user's role (possible roles are super_user, vshield_admin, enterprise_admin, security_admin, and audit).

`PUT /api/2.0/services/usermgmt/role/{userId}`

URI Parameters:

userId (required)	User to retrieve role information from.

Description:
Change a user's role.

Request:
Body: application/xml

```
<accessControlEntry>
 <role></role>
 <resource>
   <resourceId></resourceId>
 </resource>
</accessControlEntry>
```

`POST /api/2.0/services/usermgmt/role/{userId}`

URI Parameters:

userId (required)	User to retrieve role information from.

Query Parameters:

isGroup (required)	Set to "true" to apply to a group; set to "false" to apply to an individual user

Description:
Add role and resources for a user.

Request:
Body: application/xml

```
<accessControlEntry>
 <role></role>
 <resource>
   <resourceId></resourceId>
 </resource>
</accessControlEntry>
```

`DELETE /api/2.0/services/usermgmt/role/{userId}`

URI Parameters:

userId (required)	User to retrieve role information from.

Description:
Delete the role assignment for specified vCenter user. Once this role is deleted, the user is removed from NSX Manager. You cannot delete the role for a local user.

Working with User Account State

`PUT /api/2.0/services/usermgmt/enablestate/{value}`

URI Parameters:

value (required)	value can be 0 to disable, or 1 to enable.

Description:

Enable or disable a user account.

Working with NSX Manager Role Assignment

`GET /api/2.0/services/usermgmt/users/vsm`

Description:
Get information about users who have been assigned a NSX Manager role (local users as well as vCenter users with NSX Manager role).

Working with Available NSX Manager Roles

`GET /api/2.0/services/usermgmt/roles`

Description:
Read all possible roles in NSX Manager

Working With Scoping Objects

`GET /api/2.0/services/usermgmt/scopingobjects`

Description:
Retrieve a list of objects that can be used to define a user's access scope.

Working with Security Group Grouping Objects

A security group is a collection of assets or grouping objects from your vSphere inventory.

Creating New Security Groups With Members

`POST /api/2.0/services/securitygroup/bulk/{scopeId}`

URI Parameters:

`scopeId` (required)	For the scopeId use *globalroot-0* for non-universal security groups and *universalroot-0* for universal security groups.

Description:

Create a new security group on a global scope or universal scope with membership information.

Universal security groups are read-only when querying a secondary NSX manager.

When you create a universal security group (on scope *universalroot-0*) by default **localMembersOnly** is set to *false* which indicates that the universal security group will contain members across the cross-vCenter NSX environment. This is the case in an active active environment. You can add the following objects to a universal security group with *localMembersOnly=false* (active active):

- IP Address Set
- MAC Address Set
- Universal Security Groups with *localMembersOnly=false*

When you create a universal security group (on scope *universalroot-0*) you can set the extendedAttribute **localMembersOnly** to *true* to indicate that the universal security group will contain members local to that NSX Manager only. This is the case in an active standby environment, because only one NSX environment is active at a time, and the same VMs are present in each NSX environment. You can add the following objects to a universal security group with *localMembersOnly=true* (active standby):

- Universal Security Tag
- IP Address Set
- MAC Address Set
- Universal Security Groups with *localMembersOnly=true*
- Dynamic criteria using VM name

You can set the **localMembersOnly** attribute only when the universal security group is created, it cannot be modified afterwards.

Method history:

Release	Modification
6.3.0	Extended attribute **localMembersOnly** introduced.

Request:

Body: application/xml

```
<securitygroup>
 <objectId></objectId>
 <objectTypeName></objectTypeName>
 <vsmUuid></vsmUuid>
 <revision></revision>
 <type>
```

```
  <typeName></typeName>
</type>
<name></name>
<scope>
  <id></id>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <name></name>
  <revision></revision>
</scope>
<clientHandle></clientHandle>
<extendedAttributes>
  <extendedAttribute>
    <name>localMembersOnly</name>
    <value>true</value>
  </extendedAttribute>
</extendedAttributes>
<member>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name></name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
</member>
<excludeMember>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name></name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
</excludeMember>
<dynamicMemberDefinition>
  <dynamicSet>
    <operator></operator>
    <dynamicCriteria>
      <operator></operator>
      <key></key>
      <criteria></criteria>
      <value></value>
    </dynamicCriteria>
  </dynamicSet>
</dynamicMemberDefinition>
```

```
</securitygroup>
```

Creating New Security Groups Without Members

`POST /api/2.0/services/securitygroup/{scopeId}`

URI Parameters:

`scopeId` (required)	For the scopeId use *globalroot-0* for non-universal security groups and *universalroot-0* for universal security groups.

Description:

Create a new security group, with no membership information specified. You can add members later with `PUT /2.0/services/securitygroup/bulk/{objectId}`

When you create a universal security group (on scope *universalroot-0*) by default **localMembersOnly** is set to *false* which indicates that the universal security group will contain members across the cross-vCenter NSX environment. This is the case in an active active environment. You can add the following objects to a universal security group with *localMembersOnly=false* (active active):

- IP Address Set
- MAC Address Set
- Universal Security Groups with *localMembersOnly=false*

When you create a universal security group (on scope *universalroot-0*) you can set the extendedAttribute **localMembersOnly** to *true* to indicate that the universal security group will contain members local to that NSX Manager only. This is the case in an active standby environment, because only one NSX environment is active at a time, and the same VMs are present in each NSX environment. You can add the following objects to a universal security group with *localMembersOnly=true* (active standby):

- Universal Security Tag
- IP Address Set
- MAC Address Set
- Universal Security Groups with *localMembersOnly=true*
- Dynamic criteria using VM name

You can set the **localMembersOnly** attribute only when the universal security group is created, it cannot be modified afterwards.

Method history:

Release	Modification
6.3.0	Extended attribute **localMembersOnly** introduced.

Request:

Body: application/xml

```
<securitygroup>
 <name></name>
 <extendedAttributes>
   <extendedAttribute>
     <name>localMembersOnly</name>
     <value>true</value>
   </extendedAttribute>
 </extendedAttributes>
```

```
</securitygroup>
```

Updating a Specific Security Group Including Membership

`PUT /api/2.0/services/securitygroup/bulk/{objectId}`

URI Parameters:

objectId (required)	Security group ID.

Description:
Update configuration for the specified security group, including membership information.

Request:
Body: application/xml

```
<securitygroup>
 <objectId></objectId>
 <objectTypeName></objectTypeName>
 <vsmUuid></vsmUuid>
 <revision></revision>
 <type>
   <typeName></typeName>
 </type>
 <name></name>
 <scope>
   <id></id>
   <objectTypeName></objectTypeName>
   <vsmUuid></vsmUuid>
   <name></name>
   <revision></revision>
 </scope>
 <clientHandle></clientHandle>
 <extendedAttributes></extendedAttributes>
 <member>
   <objectId></objectId>
   <objectTypeName></objectTypeName>
   <vsmUuid></vsmUuid>
   <revision></revision>
   <type>
     <typeName></typeName>
   </type>
   <name></name>
   <scope>
     <id></id>
     <objectTypeName></objectTypeName>
     <name></name>
   </scope>
   <clientHandle></clientHandle>
   <extendedAttributes></extendedAttributes>
 </member>
 <excludeMember>
   <objectId></objectId>
```

```
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </excludeMember>
  <dynamicMemberDefinition>
    <dynamicSet>
      <operator></operator>
      <dynamicCriteria>
        <operator></operator>
        <key></key>
        <criteria></criteria>
        <value></value>
      </dynamicCriteria>
    </dynamicSet>
  </dynamicMemberDefinition>
</securitygroup>
```

Working with a Specific Security Group

`GET /api/2.0/services/securitygroup/{objectId}`

URI Parameters:

objectId (required)	Security group ID.

Description:
Retrieve all members of the specified security group.

`PUT /api/2.0/services/securitygroup/{objectId}`

URI Parameters:

objectId (required)	Security group ID.

Description:
Update configuration for the specified security group. Members are not updated. You must use `PUT /2.0/services/securitygroup/bulk/{objectId}` to update a security group membership.

Request:
Body: application/xml

```
<securitygroup>
 <objectId></objectId>
 <objectTypeName></objectTypeName>
 <revision></revision>
 <type>
   <typeName></typeName>
 </type>
 <name></name>
 <scope>
   <id></id>
   <objectTypeName></objectTypeName>
   <name></name>
 </scope>
 <clientHandle></clientHandle>
 <extendedAttributes></extendedAttributes>
 <isUniversal></isUniversal>
 <inheritanceAllowed></inheritanceAllowed>
</securitygroup>
```

`DELETE /api/2.0/services/securitygroup/{objectId}`

URI Parameters:

objectId (required)	Security group ID.

Query Parameters:

force (optional)	Use *force=true* to force removal of security group that is in use in other configurations.

Description:

Delete an existing security group.

If *force=true* is specified, the object is deleted even if used in other configurations, such as firewall rules. If *force=true* is not specified, the object is deleted only if it is not used by other configuration; otherwise the delete fails.

Working with Members of a Specific Security Group

`PUT /api/2.0/services/securitygroup/{objectId}/members/{memberId}`

URI Parameters:

memberId (required)	Security group member, can be a vSphere managed object ID or NSX object ID.
objectId (required)	Security group ID.

Query Parameters:

failIfExists (optional)	Default is true. If failIfExists=false: • If the member is not already present in the SecurityGroup, the API adds the member to the SecurityGroup. • If the member is already present in the SecurityGroup, the API will be a no-op and will return silently. If failIfExists=true: • If the member is not already present in the SecurityGroup, the API adds the member to the SecurityGroup. • If the member is already present in the SecurityGroup, the API call fails with the below error:<error> <details>The object vm-1000 is already present in the system.</details> <errorCode>203</errorCode> <moduleName>core-services</moduleName> </error>

Description:
Add a new member to the specified security group.

```
DELETE /api/2.0/services/securitygroup/{objectId}/members/{memberId}
```

URI Parameters:

memberId (required)	Security group member, can be a vSphere managed object ID or NSX object ID.
objectId (required)	Security group ID.

Query Parameters:

failIfAbsent (optional)	Default is true. If failIfAbsent=false: • If the member is present in the SecurityGroup, the API removes the member from the SecurityGroup. • If the member is not present in the SecurityGroup, the API call will be a no-op and will return silently. If failIfExists=true: • If the member is present in the SecurityGroup, the API removes the member from the SecurityGroup. • If the member is not present in the SecurityGroup, the API call fails with the below error:<error> <details>The requested object : vm-1000 could not be found. Object identifiers are case sensitive.</details> <errorCode>202</errorCode> <moduleName>core-services</moduleName> </error>

Description:
Delete member from the specified security group.

Working with Virtual Machines in a Security Group

```
GET /api/2.0/services/securitygroup/{objectId}/translation/virtualmachines
```

URI Parameters:

objectId (required)	Security group ID.

Description:
Retrieve effective membership of a security group in terms of virtual machines. The effective membership is calculated using all the three membership components of a security group - static include, static exclude, and dynamic using the following formula:

Effective membership virtual machines = [(VMs resulting from static include component + VMs resulting from dynamic component) - (VMs resulting from static exclude component)]

Working with IP Addresses in a Security Group

```
GET /api/2.0/services/securitygroup/{objectId}/translation/ipaddresses
```

URI Parameters:

objectId (required)	Security group ID.

Description:
Retrieve list of IP addresses that belong to a specific security group.

Working with MAC Addresses in a Security Group

```
GET /api/2.0/services/securitygroup/{objectId}/translation/macaddresses
```

URI Parameters:

objectId (required)	Security group ID.

Description:
Retrieve list of MAC addresses that belong to a specific security group.

Working with vNICs in a Security Group

```
GET /api/2.0/services/securitygroup/{objectId}/translation/vnics
```

URI Parameters:

objectId (required)	Security group ID.

Description:

Retrieve list of vNICs that belong to a specific security group.

Working with Virtual Machine Security Group Membership

`GET /api/2.0/services/securitygroup/lookup/virtualmachine/{virtualMachineId}`

URI Parameters:

`virtualMachineId` (required)	Specified virtual machine

Description:

Retrieves the collection of security groups to which a virtual machine is a direct or indirect member. Indirect membership involves nesting of security groups.

Working with Internal Security Groups

`GET /api/2.0/services/securitygroup/internal/scope/{scopeId}`

URI Parameters:

`scopeId` (required)	Specified transport zone (scope)

Description:

Retrieve all internal security groups on the NSX Manager. These are used internally by the system and should not be created or modified by end users.

Working with Security Groups on a Specific Scope

`GET /api/2.0/services/securitygroup/scope/{scopeId}`

URI Parameters:

`scopeId` (required)	scopeId can be "globalroot-0", "universalroot-0" or datacenterID / portgroupID in upgrade use cases

Description:

List all the security groups created on a specific scope.

Working with Security Group Member Types

`GET /api/2.0/services/securitygroup/scope/{scopeId}/memberTypes`

URI Parameters:

scopeId (required)	scopeId can be "globalroot-0", "universalroot-0" or datacenterID / portgroupID in upgrade use cases

Description:
Retrieve a list of valid elements that can be added to a security group.

Working with a Specific Security Group Member Type

`GET /api/2.0/services/securitygroup/scope/{scopeId}/members/{memberType}`

URI Parameters:

memberType (required)	Specific member type
scopeId (required)	scopeId can be "globalroot-0", "universalroot-0" or datacenterID / portgroupID in upgrade use cases

Description:
Retrieve members of a specific type in the specified scope.

Working with IP Set Grouping Objects

Working with IP Sets on a Specific Scope

`GET /api/2.0/services/ipset/scope/{scopeMoref}`

URI Parameters:

scopeMoref (required)	For scopeMoref use "globalroot-0" for non-universal IP sets and use "universalroot-0" for universal IP sets.

Description:
Retrieve all configured IPSets

Creating New IP Sets

`POST /api/2.0/services/ipset/{scopeMoref}`

URI Parameters:

scopeMoref (required)	For scopeMoref use "globalroot-0" for non-universal IP sets and use "universalroot-0" for universal IP sets.

Description:
Create a new IP set.

Request:
Body: application/xml

```
<ipset>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <value></value>
 <inheritanceAllowed></inheritanceAllowed>
</ipset>
```

Working with a Specific IP Set

GET /api/2.0/services/ipset/{ipsetId}

URI Parameters:

ipsetId (required)	The IP set to be queried or changed.

Description:
Retrieve an individual IP set.

PUT /api/2.0/services/ipset/{ipsetId}

URI Parameters:

ipsetId (required)	The IP set to be queried or changed.

Description:
Modify an existing IP set.

Request:
Body: application/xml

```
<ipset>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <objectTypeName></objectTypeName>
 <value></value>
</ipset>
```

DELETE /api/2.0/services/ipset/{ipsetId}

URI Parameters:

ipsetId (required)	The IP set to be queried or changed.

Query Parameters:

force (optional)	Set to "true" when forcing the removal of an IP set.

Description:
Delete an IP set.

Configuring NSX Manager with vCenter Server

You can synchronize NSX Manager with a vCenter Server, which enables the Networking and Security tab in the vCenter Web Client to display your VMware Infrastructure inventory.

vCenter Config Parameters

Parameter | Comments ipAddress | FQDN or IP address of vCenter server. userName | Required. password | Required. certificateThumbprint | Required. Must be colon (:) delimited hexadecimal. assignRoleToUser | Optional. *true* or *false*. pluginDownloadServer | Optional. pluginDownloadPort | Optional.

`GET /api/2.0/services/vcconfig`

Description:
Get vCenter Server configuration details on NSX Manager.

Responses:
Status Code: 200
Body: application/xml

```
<vcInfo>
 <ipAddress>vcsa-01a.corp.local</ipAddress>
 <userName>administrator@vsphere.local</userName>
 <certificateThumbprint>D2:75:61:24:52:CA:B2:8D:D3:25:3F:78:11:2A:8F:94:5A:30:57:0D</certificateThumbprin t>
 <assignRoleToUser>true</assignRoleToUser>
 <vcInventoryLastUpdateTime>1492567224920</vcInventoryLastUpdateTime>
</vcInfo>
```

`PUT /api/2.0/services/vcconfig`

Description:
Synchronize NSX Manager with vCenter server.

Request:
Body: application/xml

```
<vcInfo>
 <ipAddress>vc-l-01a.corp.local</ipAddress>
 <userName>administrator@vsphere.local</userName>
 <password>VMware123</password>
 <certificateThumbprint>D2:75:61:24:52:CA:B2:8D:D3:25:3F:78:11:2A:8F:94:5A:30:57:0D</certificateThumbprin t>
 <assignRoleToUser>true</assignRoleToUser>
 <pluginDownloadServer></pluginDownloadServer>
 <pluginDownloadPort></pluginDownloadPort>
</vcInfo>
```

Connection Status for vCenter Server

GET /api/2.0/services/vcconfig/status

Description:
Get default vCenter Server connection status.

Responses:
Status Code: 200
Body: application/xml

```
<vcConfigStatus>
 <connected>true</connected>
 <lastInventorySyncTime>1492568145678</lastInventorySyncTime>
</vcConfigStatus>
```

Working with Universal Sync Configuration in Cross-vCenter NSX

Working with Universal Sync Configuration Roles

You can set the role of an NSX Manager to primary, secondary, or standalone. If you set an NSX Manager's role to primary, then use it to create universal objects, and then set the role to standalone, the role will be set as transit. In the transit role, the universal objects will still exist, but cannot be modified, other than being deleted.

`GET /api/2.0/universalsync/configuration/role`

Description:
Retrieve the universal sync configuration role.

`POST /api/2.0/universalsync/configuration/role`

Query Parameters:

`action`	Set the role of the NSX manager. Possible values are *set-as-standalone*, or *set-as-primary*. To set an NSX Manager as secondary, use the `POST /api/2.0/universalsync/configuration/nsxmanagers` method on the primary NSX Manager.

Description:
Set the universal sync configuration role.

Working with Universal Sync Configuration of NSX Managers

`GET /api/2.0/universalsync/configuration/nsxmanagers`

Description:
If run on a primary NSX Manager, it will list secondary NSX Managers configured on the primary NSX Manager.

If run on a secondary NSX Manager, it will list information about the secondary NSX Manager and the primary NSX Manager it is associated with.

`POST /api/2.0/universalsync/configuration/nsxmanagers`

Description:
Add a secondary NSX manager.

Run this method on the primary NSX Manager, providing details of the secondary NSX Manager.

Retrieve the certificate thumbprint of the secondary NSX Manager using the `GET /api/1.0/appliance-management/certificatemanager/certificates/nsx` method. The **sha1Hash** parameter contains the thumbprint.

Request:
Body: application/xml

```
<nsxManagerInfo>
 <nsxManagerIp></nsxManagerIp>
 <nsxManagerUsername></nsxManagerUsername>
 <nsxManagerPassword></nsxManagerPassword>
 <certificateThumbprint></certificateThumbprint>
 <isPrimary></isPrimary>
</nsxManagerInfo>
```

`DELETE /api/2.0/universalsync/configuration/nsxmanagers`

Description:
Delete secondary NSX manager configuration.

Universal Sync Configuration of a Specific NSX Manager

`GET /api/2.0/universalsync/configuration/nsxmanagers/{nsxManagerID}`

URI Parameters:

nsxManagerID	NSX Manager UUID.

Description:
Retrieve information about the specified secondary NSX Manager.

`PUT /api/2.0/universalsync/configuration/nsxmanagers/{nsxManagerID}`

URI Parameters:

nsxManagerID	NSX Manager UUID.

Description:
Update the the specified secondary NSX manager IP or thumbprint in the universal sync configuration.

Request:
Body: application/xml

```
<nsxManagerInfo>
 <uuid></uuid>
 <nsxManagerIp></nsxManagerIp>
 <certificateThumbprint></certificateThumbprint>
</nsxManagerInfo>
```

`DELETE /api/2.0/universalsync/configuration/nsxmanagers/{nsxManagerID}`

URI Parameters:

nsxManagerID	NSX Manager UUID.

Query Parameters:

forceRemoval (optional)	Force removal of a secondary NSX manager. Options are true and false.

Description:
Delete the specified secondary NSX Manager.

NSX Manager Synchronization

`POST /api/2.0/universalsync/sync`

Query Parameters:

action	Use *invoke* to sync all objects on the NSX Manager.

Description:
Sync all objects on the NSX Manager.

Working with Universal Sync Entities

`GET /api/2.0/universalsync/entitystatus`

Query Parameters:

objectType	Specifiy the object type. For example "VdnScope"
objectId	Specify the objectID. For example "globalvdnscope"

Description:
Retrieve the status of a universal sync entity.

Working With Universal Sync Status

`GET /api/2.0/universalsync/status`

Description:
Retrieve the universal sync status.

Working with the Appliance Manager

With the appliance management tool, you can manage:
- System configurations like network configuration, syslog, time settings, and certificate management etc.
- Components of appliance such as NSX Manager, Postgres, SSH component, Rabbitmq service etc.
- Overall support related features such as tech support logs, backup restore, status, and summary reports of appliance health.

Global Information for NSX Manager

`GET /api/1.0/appliance-management/global/info`

Description:
Retrieve global information containing version information as well as current logged in user.

Responses:
Status Code: 200
Body: application/xml

```
<globalInfo>
 <currentLoggedInUser>admin</currentLoggedInUser>
 <versionInfo>
   <majorVersion>6</majorVersion>
   <minorVersion>2</minorVersion>
   <patchVersion>5</patchVersion>
   <buildNumber>4818372</buildNumber>
 </versionInfo>
</globalInfo>
```

Summary Information for NSX Manager

`GET /api/1.0/appliance-management/summary/system`

Description:
Retrieve system summary info such as address, dns name, version, CPU, memory and storage.

Responses:
Status Code: 200
Body: application/xml

```
<systemSummary>
 <ipv4Address>192.168.110.15</ipv4Address>
 <dnsName>nsxmgr-01a</dnsName>
 <hostName>nsxmgr-01a</hostName>
 <applianceName>vShield Virtual Appliance Management</applianceName>
 <versionInfo>
```

```
    <majorVersion>6</majorVersion>
    <minorVersion>2</minorVersion>
    <patchVersion>5</patchVersion>
    <buildNumber>4818372</buildNumber>
  </versionInfo>
  <uptime>25 days, 21 hours, 51 minutes</uptime>
  <cpuInfoDto>
    <totalNoOfCPUs>4</totalNoOfCPUs>
    <capacity>2799 MHZ</capacity>
    <usedCapacity>49 MHZ</usedCapacity>
    <freeCapacity>2750 MHZ</freeCapacity>
    <usedPercentage>2</usedPercentage>
  </cpuInfoDto>
  <memInfoDto>
    <totalMemory>16025 MB</totalMemory>
    <usedMemory>5633 MB</usedMemory>
    <freeMemory>10392 MB</freeMemory>
    <usedPercentage>35</usedPercentage>
  </memInfoDto>
  <storageInfoDto>
    <totalStorage>86G</totalStorage>
    <usedStorage>22G</usedStorage>
    <freeStorage>64G</freeStorage>
    <usedPercentage>25</usedPercentage>
  </storageInfoDto>
  <currentSystemDate>Wednesday, 19 April 2017 06:02:32 AM UTC</currentSystemDate>
</systemSummary>
```

Component Information for NSX Manager

`GET /api/1.0/appliance-management/summary/components`

Description:
Retrieve summary of all available components and their status info.

Responses:
Status Code: 200
Body: application/xml

```
<componentsSummary>
  <componentsByGroup class="tree-map">
    <entry>
      <string>COMMON</string>
      <components>
        <component>
          <componentId>VPOSTGRES</componentId>
          <name>vPostgres</name>
          <description>vPostgres - Database service</description>
          <status>RUNNING</status>
          <enabled>true</enabled>
          <showTechSupportLogs>false</showTechSupportLogs>
          <usedBy>
            <string>NSX</string>
```

```
        </usedBy>
        <componentGroup>COMMON</componentGroup>
      </component>
      <component>
        <componentId>RABBITMQ</componentId>
        <name>RabbitMQ</name>
        <description>RabbitMQ - Messaging service</description>
        <status>RUNNING</status>
        <enabled>true</enabled>
        <showTechSupportLogs>false</showTechSupportLogs>
        <usedBy>
          <string>NSX</string>
        </usedBy>
        <componentGroup>COMMON</componentGroup>
      </component>
    </components>
  </entry>
  <entry>
    <string>NSXGRP</string>
    <components>
      <component>
        <componentId>NSXREPLICATOR</componentId>
        <name>NSX Replicator</name>
        <description>NSX Replicator</description>
        <status>RUNNING</status>
        <enabled>true</enabled>
        <showTechSupportLogs>false</showTechSupportLogs>
        <uses>
          <string>NSX</string>
        </uses>
        <usedBy></usedBy>
        <componentGroup>NSXGRP</componentGroup>
        <versionInfo>
          <majorVersion>6</majorVersion>
          <minorVersion>2</minorVersion>
          <patchVersion>5</patchVersion>
          <buildNumber>4818383</buildNumber>
        </versionInfo>
      </component>
      <component>
        <componentId>NSX</componentId>
        <name>NSX Manager</name>
        <description>NSX Manager</description>
        <status>RUNNING</status>
        <enabled>true</enabled>
        <showTechSupportLogs>true</showTechSupportLogs>
        <uses>
          <string>VPOSTGRES</string>
          <string>RABBITMQ</string>
        </uses>
        <usedBy>
          <string>NSXREPLICATOR</string>
        </usedBy>
        <componentGroup>NSXGRP</componentGroup>
        <versionInfo>
          <majorVersion>6</majorVersion>
          <minorVersion>2</minorVersion>
          <patchVersion>5</patchVersion>
          <buildNumber>4818372</buildNumber>
        </versionInfo>
      </component>
    </components>
```

```
    </entry>
    <entry>
      <string>SYSTEM</string>
      <components>
        <component>
          <componentId>SSH</componentId>
          <name>SSH Service</name>
          <description>Secure Shell</description>
          <status>RUNNING</status>
          <enabled>true</enabled>
          <showTechSupportLogs>false</showTechSupportLogs>
          <usedBy></usedBy>
          <componentGroup>SYSTEM</componentGroup>
        </component>
      </components>
    </entry>
  </componentsByGroup>
</componentsSummary>
```

Reboot NSX Manager

`POST /api/1.0/appliance-management/system/restart`

Description:
Reboot the NSX Manager appliance.

NSX Manager CPU Information

`GET /api/1.0/appliance-management/system/cpuinfo`

Description:
Retrieve NSX Manager CPU information.

Responses:
Status Code: 200
Body: application/xml

```
<cpuInfo>
 <totalNoOfCPUs>4</totalNoOfCPUs>
 <capacity>2799 MHZ</capacity>
 <usedCapacity>47 MHZ</usedCapacity>
 <freeCapacity>2752 MHZ</freeCapacity>
 <usedPercentage>2</usedPercentage>
</cpuInfo>
```

NSX Manager Appliance Uptime Information

`GET /api/1.0/appliance-management/system/uptime`

Description:
Retrieve NSX Manager uptime information.

Example response:

```
25 days, 22 hours, 11 minutes
```

NSX Manager Appliance Memory Information

`GET /api/1.0/appliance-management/system/meminfo`

Description:
Retrieve NSX Manager memory information.

Responses:
Status Code: 200
Body: application/xml

```
<memInfo>
 <totalMemory>16025 MB</totalMemory>
 <usedMemory>5633 MB</usedMemory>
 <freeMemory>10392 MB</freeMemory>
 <usedPercentage>35</usedPercentage>
</memInfo>
```

NSX Manager Appliance Storage Information

`GET /api/1.0/appliance-management/system/storageinfo`

Description:
Retrieve NSX Manager storage information.

Responses:
Status Code: 200
Body: application/xml

```
<storageInfo>
 <totalStorage>86G</totalStorage>
 <usedStorage>22G</usedStorage>
 <freeStorage>64G</freeStorage>
 <usedPercentage>25</usedPercentage>
</storageInfo>
```

NSX Manager Appliance Network Settings

`GET /api/1.0/appliance-management/system/network`

Description:
Retrieve network information for the NSX Manager appliance. i.e. host name, IP address, DNS settings

Responses:
Status Code: 200
Body: application/xml

```
<network>
 <hostName>nsxmgr-01a</hostName>
 <networkIPv4AddressDto>
   <ipv4Address>192.168.110.15</ipv4Address>
   <ipv4NetMask>255.255.255.0</ipv4NetMask>
   <ipv4Gateway>192.168.110.1</ipv4Gateway>
 </networkIPv4AddressDto>
 <dns>
   <ipv4Address>192.168.110.10</ipv4Address>
   <domainList>corp.local</domainList>
 </dns>
</network>
```

`PUT /api/1.0/appliance-management/system/network`

Description:
Update network information for the NSX Manager appliance.

Request:
Body: application/xml

```
<network>
 <hostName>nsxmgr-01a</hostName>
 <networkIPv4AddressDto>
   <ipv4Address>192.168.110.15</ipv4Address>
   <ipv4NetMask>255.255.255.0</ipv4NetMask>
   <ipv4Gateway>192.168.110.1</ipv4Gateway>
 </networkIPv4AddressDto>
 <networkIPv6AddressDto>
   <ipv6Address>fdd1:0ebc:b724:d2f1:0000:8a7e:0360:5332</ipv6Address>
```

```
    <ipv6PrefixLength>64</ipv6PrefixLength>
    <ipv6Gateway>fdd1:0ebc:b724:d2f1:0000:8a7e:0360:0002</ipv6Gateway>
  </networkIPv6AddressDto>
  <dns>
    <ipv4Address>192.168.110.10</ipv4Address>
    <ipv6Address>fdd1:0ebc:b724:d2f1:0000:8a7e:0360:0010</ipv6Address>
    <domainList>corp.local</domainList>
  </dns>
</network>
```

Working with DNS Configuration

`PUT /api/1.0/appliance-management/system/network/dns`

Description:
Configure DNS.

Request:
Body: application/xml

```
<dns>
 <ipv4Address></ipv4Address>
 <ipv6Address></ipv6Address>
 <domainList></domainList>
</dns>
```

`DELETE /api/1.0/appliance-management/system/network/dns`

Description:
Delete DNS server configuration.

Working with Security Settings

`GET /api/1.0/appliance-management/system/securitysettings`

Description:
Retrieve the NSX Manager FIPS and TLS settings.

Method history:

Release	Modification
6.3.0	Method introduced.

Responses:
Status Code: 200

Body: application/xml

```
<securitySettings>
 <fipsEnabled>false</fipsEnabled>
 <tlsSettings>
   <serverEnabledProtocols>TLSv1,TLSv1.1,TLSv1.2</serverEnabledProtocols>
   <clientEnabledProtocols>TLSv1,TLSv1.1,TLSv1.2</clientEnabledProtocols>
 </tlsSettings>
</securitySettings>
```

POST /api/1.0/appliance-management/system/securitysettings

Description:
Update the NSX Manager security settings, including FIPS and TLS.

Do not enable FIPS until you have upgraded all NSX components to NSX 6.3.0 or later. Enable FIPS on NSX Edges before enabling it on the NSX Manager.

Changing the FIPS mode will reboot the NSX Manager appliance.

Method history:

Release	Modification
6.3.0	Method introduced.

Request:
Body: application/xml

```
<securitySettings>
 <fipsEnabled>true</fipsEnabled>
 <tlsSettings>
   <serverEnabledProtocols>TLSv1.1,TLSv1.2</serverEnabledProtocols>
   <clientEnabledProtocols>TLSv1.1,TLSv1.2</clientEnabledProtocols>
 </tlsSettings>
</securitySettings>
```

Working with TLS Settings

GET /api/1.0/appliance-management/system/tlssettings

Description:
Retrieve TLS settings.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:

Status Code: 200
Body: application/xml

```
<tlsSettings>
 <serverEnabledProtocols>TLSv1,TLSv1.1,TLSv1.2</serverEnabledProtocols>
 <clientEnabledProtocols>TLSv1,TLSv1.1,TLSv1.2</clientEnabledProtocols>
</tlsSettings>
```

`POST /api/1.0/appliance-management/system/tlssettings`

Description:
Update TLS settings.

Include a comma separated list of the TLS versions you want to enable, for both server and client.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<tlsSettings>
 <serverEnabledProtocols>TLSv1.1,TLSv1.2</serverEnabledProtocols>
 <clientEnabledProtocols>TLSv1.1,TLSv1.2</clientEnabledProtocols>
</tlsSettings>
```

Working with Time Settings

You can either configure time or specify the NTP server to be used for time synchronization.

`GET /api/1.0/appliance-management/system/timesettings`

Description:
Retrieve time settings, like timezone or current date and time with NTP server, if configured.

Responses:
Status Code: 200
Body: application/xml

```
<timeSettings>
 <ntpServer>
   <string>192.168.110.1</string>
 </ntpServer>
 <datetime>04/19/2017 06:53:57</datetime>
 <timezone>UTC</timezone>
</timeSettings>
```

PUT /api/1.0/appliance-management/system/timesettings

Description:
Configure time or specify the NTP server to use for time synchronization.

Request:
Body: application/xml

```
<timeSettings>
 <ntpServer>
   <string>192.168.110.1</string>
 </ntpServer>
 <datetime>04/19/2017 06:53:57</datetime>
 <timezone>UTC</timezone>
</timeSettings>
```

Working with NTP Settings

DELETE /api/1.0/appliance-management/system/timesettings/ntp

Description:
Delete NTP server.

Configure System Locale

GET /api/1.0/appliance-management/system/locale

Description:
Retrieve locale info.

Responses:
Status Code: 200
Body: application/xml

```
<locale>
 <language>en</language>
 <country>US</country>
</locale>
```

PUT /api/1.0/appliance-management/system/locale

Description:
Configure locale.

Request:
Body: application/xml

```
<locale>
 <language>ja</language>
 <country>JP</country>
</locale>
```

Working with Syslog Servers

`GET /api/1.0/appliance-management/system/syslogserver`

Description:
Retrieve syslog servers.

Responses:
Status Code: 200
Body: application/xml

```
<syslogserver>
 <syslogServer>192.168.110.20</syslogServer>
 <port>514</port>
 <protocol>UDP</protocol>
</syslogserver>
```

`PUT /api/1.0/appliance-management/system/syslogserver`

Description:
Configure syslog servers.

Request:
Body: application/xml

```
<syslogserver>
 <syslogServer>192.168.110.20</syslogServer>
 <port>514</port>
 <protocol>UDP</protocol>
</syslogserver>
```

`DELETE /api/1.0/appliance-management/system/syslogserver`

Description:

Delete syslog servers.

Working with Components

The NSX Manager appliance has the following components:

Component	Description
NSX	NSX Manager
NSXREPLICATOR	Universal Synchronization Service
RABBITMQ	RabbitMQ - Messaging service
SSH	SSH Service
VPOSTGRES	vPostgres - Database service

`GET /api/1.0/appliance-management/components`

Description:
Retrieve all appliance manager components.

Responses:
Status Code: 200
Body: application/xml

```
<components>
 <component>
   <componentId>SSH</componentId>
   <name>SSH Service</name>
   <description>Secure Shell</description>
   <status>RUNNING</status>
   <enabled>true</enabled>
   <showTechSupportLogs>false</showTechSupportLogs>
   <usedBy></usedBy>
   <componentGroup>SYSTEM</componentGroup>
 </component>
 <component>
   <componentId>VPOSTGRES</componentId>
   <name>vPostgres</name>
   <description>vPostgres - Database service</description>
   <status>RUNNING</status>
   <enabled>true</enabled>
   <showTechSupportLogs>false</showTechSupportLogs>
   <usedBy>
     <string>NSX</string>
   </usedBy>
   <componentGroup>COMMON</componentGroup>
 </component>
 <component>
   <componentId>NSXREPLICATOR</componentId>
   <name>NSX Replicator</name>
   <description>NSX Replicator</description>
   <status>RUNNING</status>
   <enabled>true</enabled>
   <showTechSupportLogs>false</showTechSupportLogs>
```

```
    <uses>
      <string>NSX</string>
    </uses>
    <usedBy></usedBy>
    <componentGroup>NSXGRP</componentGroup>
    <versionInfo>
      <majorVersion>6</majorVersion>
      <minorVersion>2</minorVersion>
      <patchVersion>5</patchVersion>
      <buildNumber>4818383</buildNumber>
    </versionInfo>
  </component>
  <component>
    <componentId>RABBITMQ</componentId>
    <name>RabbitMQ</name>
    <description>RabbitMQ - Messaging service</description>
    <status>RUNNING</status>
    <enabled>true</enabled>
    <showTechSupportLogs>false</showTechSupportLogs>
    <usedBy>
      <string>NSX</string>
    </usedBy>
    <componentGroup>COMMON</componentGroup>
  </component>
  <component>
    <componentId>NSX</componentId>
    <name>NSX Manager</name>
    <description>NSX Manager</description>
    <status>RUNNING</status>
    <enabled>true</enabled>
    <showTechSupportLogs>true</showTechSupportLogs>
    <uses>
      <string>VPOSTGRES</string>
      <string>RABBITMQ</string>
    </uses>
    <usedBy>
      <string>NSXREPLICATOR</string>
    </usedBy>
    <componentGroup>NSXGRP</componentGroup>
    <versionInfo>
      <majorVersion>6</majorVersion>
      <minorVersion>2</minorVersion>
      <patchVersion>5</patchVersion>
      <buildNumber>4818372</buildNumber>
    </versionInfo>
  </component>
</components>
```

Working with a Specific Component

`GET /api/1.0/appliance-management/components/component/{componentID}`

URI Parameters:

componentID (required)	Specified component ID.

Description:
Retrieve details for the specified component.

Responses:
Status Code: 200
Body: application/xml

```
<component>
 <componentId>NSX</componentId>
 <name>NSX Manager</name>
 <description>NSX Manager</description>
 <status>RUNNING</status>
 <enabled>true</enabled>
 <showTechSupportLogs>true</showTechSupportLogs>
 <uses>
   <string>VPOSTGRES</string>
   <string>RABBITMQ</string>
 </uses>
 <usedBy>
   <string>NSXREPLICATOR</string>
 </usedBy>
 <componentGroup>NSXGRP</componentGroup>
 <versionInfo>
   <majorVersion>6</majorVersion>
   <minorVersion>2</minorVersion>
   <patchVersion>5</patchVersion>
   <buildNumber>4818372</buildNumber>
 </versionInfo>
</component>
```

Working with Component Dependencies

`GET /api/1.0/appliance-management/components/component/{componentID}/dependencies`

URI Parameters:

componentID (required)	Specified component ID.

Description:
Retrieve dependency details for the specified component.

Responses:
Status Code: 200
Body: application/xml

```
<list>
 <string>VPOSTGRES</string>
 <string>RABBITMQ</string>
</list>
```

Working with Component Dependents

`GET /api/1.0/appliance-management/components/component/{componentID}/dependents`

URI Parameters:

componentID (required)	Specified component ID.

Description:
Retrieve dependents for the specified component.

Responses:
Status Code: 200
Body: application/xml

```
<list>
 <string>NSXREPLICATOR</string>
</list>
```

Working with Component Status

`GET /api/1.0/appliance-management/components/component/{componentID}/status`

URI Parameters:

componentID (required)	Specified component ID.

Description:
Retrieve current status for the specified component.

Responses:
Status Code: 200
Body: application/xml

```
<result>
 <result class="status">RUNNING</result>
 <operationStatus>SUCCESS</operationStatus>
</result>
```

Toggle Component Status

POST
/api/1.0/appliance-management/components/component/{componentID}/toggleStatus/{command}

URI Parameters:

command (required)	Use **command** parameter *start* or *stop*.
componentID (required)	Specified component ID.

Description:
Start or stop a component.

Working With the Appliance Management Web Application

POST /api/1.0/appliance-management/components/component/APPMGMT/restart

Description:
Restart the appliance management web application.

NSX Manager Appliance Backup Settings

You can back up and restore your NSX Manager data, which can include system configuration, events, and audit log tables. Configuration tables are included in every backup. Backups are saved to a remote location that must be accessible by the NSX Manager.

Parameters for the NSX Manager appliance backup:
- **transferProtocol**: *FTP, SFTP*
- **frequency**: *weekly, daily, hourly*
- **dayOfWeek**: *SUNDAY, MONDAY, ..., SATURDAY*
- **hourOfDay**: [*0-24*]
- **minuteOfHour**: [*0-60*]
- **excludeTables**: *AUDIT_LOG, SYSTEM_EVENTS, FLOW_RECORDS*
  The tables specified in the **excludeTables** parameter are not backed up.

You must set a **passPhrase** for the backups. The passphrase is used to create and read backup files. If you do not set a passphrase, backups will fail. If you forget the passphrase set on a backup file, you cannot restore that backup file.

GET /api/1.0/appliance-management/backuprestore/backupsettings

Description:
Retrieve backup settings.

Responses:
Status Code: 200
Body: application/xml

```
<backupRestoreSettings>
 <ftpSettings>
   <transferProtocol>SFTP</transferProtocol>
```

```
    <hostNameIPAddress>192.168.110.30</hostNameIPAddress>
    <port>22</port>
    <userName>admin</userName>
    <password>testing123</password>
    <backupDirectory>/backups</backupDirectory>
    <filenamePrefix>nsxmgr-01a</filenamePrefix>
    <passPhrase>testing456</passPhrase>
    <passiveMode>true</passiveMode>
    <useEPRT>false</useEPRT>
    <useEPSV>true</useEPSV>
  </ftpSettings>
  <backupFrequency>
    <frequency>DAILY</frequency>
    <hourOfDay>2</hourOfDay>
    <minuteOfHour>15</minuteOfHour>
  </backupFrequency>
  <excludeTables>
    <excludeTable>AUDIT_LOGS</excludeTable>
    <excludeTable>SYSTEM_EVENTS</excludeTable>
    <excludeTable>FLOW_RECORDS</excludeTable>
  </excludeTables>
</backupRestoreSettings>
```

PUT /api/1.0/appliance-management/backuprestore/backupsettings

Description:
Configure backups on the appliance manager.

Request:
Body: application/xml

```
<backupRestoreSettings>
  <ftpSettings>
    <transferProtocol>SFTP</transferProtocol>
    <hostNameIPAddress>192.168.110.30</hostNameIPAddress>
    <port>22</port>
    <userName>admin</userName>
    <password>testing123</password>
    <backupDirectory>/backups</backupDirectory>
    <filenamePrefix>nsxmgr-01a</filenamePrefix>
    <passPhrase>testing456</passPhrase>
    <passiveMode>true</passiveMode>
    <useEPRT>false</useEPRT>
    <useEPSV>true</useEPSV>
  </ftpSettings>
  <backupFrequency>
    <frequency>WEEKLY</frequency>
    <dayOfWeek>SUNDAY</dayOfWeek>
    <hourOfDay>2</hourOfDay>
    <minuteOfHour>15</minuteOfHour>
  </backupFrequency>
  <excludeTables>
    <excludeTable>AUDIT_LOGS</excludeTable>
  </excludeTables>
</backupRestoreSettings>
```

```
DELETE /api/1.0/appliance-management/backuprestore/backupsettings
```

Description:
Delete appliance manager backup configuration.

NSX Manager Appliance Backup FTP Settings

See *NSX Manager Appliance Backup Settings* for details.

```
PUT /api/1.0/appliance-management/backuprestore/backupsettings/ftpsettings
```

Description:
Configure FTP settings.

NSX Manager Appliance Backup Exclusion Settings

See *NSX Manager Appliance Backup Settings* for details.

```
PUT /api/1.0/appliance-management/backuprestore/backupsettings/excludedata
```

Description:
Specify tables that need not be backed up.

NSX Manager Appliance Backup Schedule Settings

See *NSX Manager Appliance Backup Settings* for details.

```
PUT /api/1.0/appliance-management/backuprestore/backupsettings/schedule
```

Description:
Set backup schedule.

```
DELETE /api/1.0/appliance-management/backuprestore/backupsettings/schedule
```

Description:
Delete backup schedule.

NSX Manager Appliance On-Demand Backup

```
POST /api/1.0/appliance-management/backuprestore/backup
```

Description:
Start an on-demand NSX backup.

Working with NSX Manager Appliance Backup Files

`GET /api/1.0/appliance-management/backuprestore/backups`

Description:
Retrieve list of all backups available at configured backup location.

Responses:
Status Code: 200
Body: application/xml

```
<list>
 <backupFileProperties>
   <fileName></fileName>
   <fileSize></fileSize>
   <creationTime></creationTime>
 </backupFileProperties>
</list>
```

Restoring Data from an NSX Manager Appliance Backup File

`POST /api/1.0/appliance-management/backuprestore/restore`

Query Parameters:

`restoreFile` (required)	File name of restore file.

Description:
Restore data from a backup file.

Retrive a list of restore files using `GET /api/1.0/appliance-management/backuprestore/backups`.

Working with Tech Support Logs by Component

`POST /api/1.0/appliance-management/techsupportlogs/{componentID}`

URI Parameters:

`componentID` (required)	Specified component to generate tech support logs. For example, *NSX*.

Description:
Generate tech support logs. The location response header contains the location of the created tech support file.

Working with Tech Support Log Files

`GET /api/1.0/appliance-management/techsupportlogs/{filename}`

URI Parameters:

`filename` (required)	Name of log file to download.

Description:
Download tech support logs

Working with Support Notifications

`GET /api/1.0/appliance-management/notifications`

Description:
Retrieve all system generated notifications.

`DELETE /api/1.0/appliance-management/notifications`

Description:
Delete all notifications.

Acknowledge Notifications

`POST /api/1.0/appliance-management/notifications/{ID}/acknowledge`

URI Parameters:

`ID` (required)	Notification ID.

Description:
Acknowledge a notification. The notification is then deleted from the system.

Upgrading NSX Manager Appliance

To upgrade NSX Manager, you must do the following:

- upload an upgrade bundle
  POST /api/1.0/appliance-management/upgrade/uploadbundle/{componentID}
- retrieve the upgrade information
  GET /api/1.0/appliance-management/upgrade/information/{componentID}
- edit the **preUpgradeQuestionsAnswers** section of the upgrade information response, if needed
- start the upgrade, providing the edited **preUpgradeQuestionsAnswers** section as the request body
  POST /api/1.0/appliance-management/upgrade/start/{componentID}

Upload an NSX Manager Upgrade Bundle

You must upload the upgrade bundle using the form-data content-type. Consult the documentation for your REST client for instructions.

Do not set other Content-type headers in your request, for example, *Content-type: application/xml*.

When you upload a file as form-data, you must provide a **key** and a **value** for the file. The **key** is *file*, and the **value** is the location of the upgrade bundle file.

Example using curl

```
/usr/bin/curl -v -k -i -F file=@/tmp/VMware-NSX-Manager-upgrade-bundle-6.2.7-5343628.tar.gz -H
'Authorization: Basic YWRtaW46ZGXXXXXXXX=='
https://192.168.110.42/api/1.0/appliance-management/upgrade/uploadbundle/NSX
```

`POST /api/1.0/appliance-management/upgrade/uploadbundle/{componentID}`

URI Parameters:

componentID (required)	Component ID.

Description:
Upload upgrade bundle.

Prepare for NSX Manager Upgrade

`GET /api/1.0/appliance-management/upgrade/information/{componentID}`

Description:
Once you have uploaded an upgrade bundle, you must retrieve information about the upgrade. This request contains pre-upgrade validation warnings and error messages, along with pre-upgrade questions with default answers. Review the information and edit the answers in the **preUpgradeQuestionsAnswers** section if needed before providing the section as the request body to the `POST /api/1.0/appliance-management/upgrade/start/{componentID}` method.

Responses:
Status Code: 200
Body: application/xml

```
<upgradeInformation>
 <fromVersion>6.2.5</fromVersion>
 <toVersion>6.2.7.5343628</toVersion>
```

```
 <upgradeBundleDescription>Upgrade to 6.2.7 5343628</upgradeBundleDescription>
 <preUpgradeQuestionsAnswers>
   <preUpgradeQuestionAnswer>
     <questionId>preUpgradeChecks1:Q1</questionId>
     <question>Do you want to enable SSH ?</question>
     <questionAnserType>YESNO</questionAnserType>
     <defaultSelection>NO</defaultSelection>
   </preUpgradeQuestionAnswer>
   <preUpgradeQuestionAnswer>
     <questionId>preUpgradeChecks1:Q2</questionId>
     <question>This product participates in VMware's Customer Experience Improvement Program ("CEIP").
The CEIP provides VMware with information that enables VMware to improve its products and services, to
fix problems, and to advise you on
       how best to deploy and use our products. As part of the CEIP, VMware collects technical
information about your organization's use of VMware products and services on a regular basis in
association with your organization's VMware license
       key(s). This information does not personally identify any individual. For additional information
regarding the CEIP, please see the Trust and Assurance Center at
http://www.vmware.com/trustvmware/ceip.html. You can select your participation
       preferences below. Do you want to join the VMware Customer Experience Improvement Program
?</question>
     <questionAnserType>YESNO</questionAnserType>
     <defaultSelection>YES</defaultSelection>
   </preUpgradeQuestionAnswer>
 </preUpgradeQuestionsAnswers>
 <upgradeStepsDto>
   <step>
     <stepId>ValidationStep</stepId>
     <stepLabel>Upgrade Bundle Validation</stepLabel>
     <description>Upgrade bundle will be validated before the actual upgrade process.</description>
   </step>
   <step>
     <stepId>UpgradeStep</stepId>
     <stepLabel>Upgrade NSX manager</stepLabel>
     <description>Upgrade process for NSX Manager will begin.</description>
   </step>
 </upgradeStepsDto>
</upgradeInformation>
```

Start the NSX Manager Upgrade

`POST /api/1.0/appliance-management/upgrade/start/{componentID}`

URI Parameters:

componentID (required)	Component ID

Description:
Start upgrade process.

Request:
Body: application/xml

```
<preUpgradeQuestionsAnswers>
 <preUpgradeQuestionAnswer>
   <questionId>preUpgradeChecks1:Q1</questionId>
   <question>Do you want to enable SSH ?</question>
   <questionAnserType>YESNO</questionAnserType>
   <answer>YES</answer>
 </preUpgradeQuestionAnswer>
 <preUpgradeQuestionAnswer>
   <questionId>preUpgradeChecks1:Q2</questionId>
   <question>This product participates in VMware's Customer Experience Improvement Program ("CEIP"). The
CEIP provides VMware with information that enables VMware to improve its products and services, to fix
problems, and to advise you on
     how best to deploy and use our products. As part of the CEIP, VMware collects technical information
about your organization's use of VMware products and services on a regular basis in association with your
organization's VMware license
     key(s). This information does not personally identify any individual. For additional information
regarding the CEIP, please see the Trust and Assurance Center at
http://www.vmware.com/trustvmware/ceip.html. You can select your participation
     preferences below. Do you want to join the VMware Customer Experience Improvement Program
?</question>
   <questionAnserType>YESNO</questionAnserType>
   <answer>YES</answer>
 </preUpgradeQuestionAnswer>
</preUpgradeQuestionsAnswers>
```

NSX Manager Upgrade Status

`GET /api/1.0/appliance-management/upgrade/status/{componentID}`

URI Parameters:

componentID (required)	Component ID.

Description:
Query upgrade status.

Working with Certificates on the NSX Manager Appliance

Working with Keystore Files

`POST /api/1.0/appliance-management/certificatemanager/pkcs12keystore/nsx`

Query Parameters:

password	Password.

Description:

Upload keystore file.

Input is PKCS#12 formatted NSX file along with password.

NSX Manager Certificate Manager

`GET /api/1.0/appliance-management/certificatemanager/certificates/nsx`

Description:
Retrieve certificate information from NSX Manager.

Responses:
Status Code: 200
Body: application/xml

```
<x509Certificates>
 <x509certificate>
   <subjectCn></subjectCn>
   <issuerCn></issuerCn>
   <version></version>
   <serialNumber></serialNumber>
   <signatureAlgo></signatureAlgo>
   <signature></signature>
   <notBefore></notBefore>
   <notAfter></notAfter>
   <issuer></issuer>
   <subject></subject>
   <publicKeyAlgo></publicKeyAlgo>
   <publicKeyLength></publicKeyLength>
   <rsaPublicKeyModulus></rsaPublicKeyModulus>
   <rsaPublicKeyExponent></rsaPublicKeyExponent>
   <sha1Hash></sha1Hash>
   <md5Hash></md5Hash>
   <isCa></isCa>
   <isValid></isValid>
 </x509certificate>
 ***
</x509Certificates>
```

Working with Certificate Signing Requests

`GET /api/1.0/appliance-management/certificatemanager/csr/nsx`

Description:
Retrieve generated certificate signing request (CSR).

`POST /api/1.0/appliance-management/certificatemanager/csr/nsx`

Description:
Create a certificate signing request (CSR) for NSX Manager.

The response header contains the created file location.

Method history:

Release	Modification
6.2.3	Method introduced. Replaces `PUT /api/1.0/appliance-management/certificatemanager/csr/nsx`.

Request:
Body: application/xml

```
<csr>
 <algorithm></algorithm>
 <keySize></keySize>
 <subjectDto>
   <commonName></commonName>
   <organizationUnit></organizationUnit>
   <organizationName></organizationName>
   <localityName></localityName>
   <stateName></stateName>
   <countryCode></countryCode>
 </subjectDto>
</csr>
```

Working with Certificate Chains

`POST /api/1.0/appliance-management/certificatemanager/uploadchain/nsx`

Description:
Upload certificate chain.

Input is certificate chain file which is a PEM encoded chain of certificates received from the CA after signing a CSR.

Working with NSX Manager System Events

`GET /api/2.0/systemevent`

Query Parameters:

`startIndex` (optional)	The starting point for returning results.
`pageSize` (optional)	The number of results to return. Range is 1-1024.

Description:
Get NSX Manager system events

Working with NSX Manager Audit Logs

`GET /api/2.0/auditlog`

Query Parameters:

`startIndex` (optional)	The starting point for returning results.
`pageSize` (optional)	The number of results to return. Range is 1-1024.

Description:
Get NSX Manager audit logs

Working with Network Fabric Configuration

Working with Network Virtualization Components and VXLAN

Cluster preparation can be broken down into the following:
- Install VIB and non-VIB related action: Before any per-host config a VIB must be installed on the host. The feature can use this time to perform other bootstrapping tasks which do not depend on VIB-installation. e.g. VXLAN creates the vmknic-pg and sets up some opaque data.
- Post-VIB install: Prepare each host for the feature. In the case of VXLAN, create vmknics.

`PUT /api/2.0/nwfabric/configure`

Description:

Upgrade Network virtualization components. _ This API call can be used to upgrade network virtualization components. After NSX Manager is upgraded, previously prepared clusters must have the 6.x network virtualization components installed.

Request:

Body: application/xml

```
<nwFabricFeatureConfig>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
 </resourceConfig>
</nwFabricFeatureConfig>
```

`POST /api/2.0/nwfabric/configure`

Query Parameters:

`action`	Specify *synchronize* to reset communication between NSX Manager and a host or cluster.

Description:

Install network fabric or VXLAN.

This method can be used to perform the following tasks:
- Install Network Virtualization Components
- Configure VXLAN
- Configure VXLAN with LACPv2
- Reset Communication Between NSX Manager and a Host or Cluster

Parameter Information

Name	Comments
resourceId	vCenter MOB ID of cluster. For example, *domain-7*. A host can be specified when resetting communication. For example, *host-24*.

featureId	Feature to act upon. Omit for network virtualization components operations. Use *com.vmware.vshield.vsm.vxlan* for VXLAN operations, *com.vmware.vshield.vsm.messagingInfra* for message bus operations.
ipPoolId	Used for VXLAN installation. If not specified, DHCP is used for VTEP address assignment.
teaming	Used for VXLAN installation. Options are *FAILOVER_ORDER*, *ETHER_CHANNEL*, *LACP_ACTIVE*, *LACP_PASSIVE*, *LOADBALANCE_LOADBASED*, *LOADBALANCE_SRCID*, *LOADBALANCE_SRCMAC*, *LACP_V2*
uplinkPortName	The *uplinkPortName* as specified in vCenter.

Install Network Virtualization Components

```
POST /api/2.0/nwfabric/configure
```

```
<nwFabricFeatureConfig>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Configure VXLAN

```
POST /api/2.0/nwfabric/configure
```

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
   <configSpec class="clusterMappingSpec">
     <switch>
       <objectId>DVS MOID</objectId></switch>
       <vlanId>0</vlanId>
       <vmknicCount>1</vmknicCount>
       <ipPoolId>IPADDRESSPOOL ID</ipPoolId>
   </configSpec>
 </resourceConfig>
 <resourceConfig>
   <resourceId>DVS MOID</resourceId>
   <configSpec class="vdsContext">
     <switch>
         <objectId>DVS MOID</objectId>
     </switch>
     <mtu>1600</mtu>
     <teaming>ETHER_CHANNEL</teaming>
   </configSpec>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Configure VXLAN with LACPv2

`POST /api/2.0/nwfabric/configure`

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
   <configSpec class="clusterMappingSpec">
     <switch>
       <objectId>DVS MOID</objectId>
     </switch>
     <vlanId>0</vlanId>
     <vmknicCount>1</vmknicCount>
   </configSpec>
 </resourceConfig>
 <resourceConfig>
   <resourceId>DVS MOID</resourceId>
   <configSpec class="vdsContext">
     <switch>
       <objectId>DVS MOID</objectId>
     </switch>
     <mtu>1600</mtu>
     <teaming>LACP_V2</teaming>
     <uplinkPortName>LAG NAME</uplinkPortName>
   </configSpec>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Reset Communication Between NSX Manager and a Host or Cluster

`POST /api/2.0/nwfabric/configure?action=synchronize`

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.messagingInfra</featureId>
 <resourceConfig>
   <resourceId>resourceId</resourceId>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Request:
Body: application/xml

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
   <configSpec class="clusterMappingSpec">
     <switch>
       <objectId>DVS MOID</objectId></switch>
       <vlanId>0</vlanId>
       <vmknicCount>1</vmknicCount>
       <ipPoolId>IPADDRESSPOOL ID</ipPoolId>
   </configSpec>
 </resourceConfig>
</nwFabricFeatureConfig>
```

```
DELETE /api/2.0/nwfabric/configure
```

Description:

Remove VXLAN or network virtualization components.

Removing network virtualization components removes previously installed VIBs, tears down NSX Manager to ESXi messaging, and removes any other network fabric dependent features such as logical switches. If a feature such as logical switches is being used in your environment, this call fails.

Removing VXLAN does not remove the network virtualization components from the cluster.

Name	Comments
resourceId	vCenter MOB ID of cluster. For example, domain-7.
featureId	Feature to act upon. Omit for network virtualization components operations. Use *com.vmware.vshield.vsm.vxlan* for VXLAN operations.

Remove Network Virtualization Components

```
<nwFabricFeatureConfig>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Remove VXLAN

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>
</nwFabricFeatureConfig>
```

Remove VXLAN with vDS context

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
   <configSpec class="map">
     <entry>
       <keyclass="java.lang.String">vxlan</key>
       <valueclass="java.lang.String">cascadeDeleteVdsContext</value>
     </entry>
   </configSpec>
 </resourceConfig>
</nwFabricFeatureConfig>
```

Request:

Body: application/xml

```
<nwFabricFeatureConfig>
 <featureId>com.vmware.vshield.vsm.vxlan</featureId>
 <resourceConfig>
   <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>
</nwFabricFeatureConfig>
```

`GET /api/2.0/nwfabric/features`

Description:
Retrieves all network fabric features available on the cluster. Multiple **featureInfo** sections may be returned.

Responses:
Status Code: 200
Body: application/xml

```
<featureInfos>
 <featureInfo>
   <name>FEATURE NAME</name>
   <featureId>FEATURE ID</featureId>
   <version>FEATURE VERSION</version>
 </featureInfo>
</featureInfos>
```

Working With Network Fabric Status

`GET /api/2.0/nwfabric/status`

Query Parameters:

`resource` (required)	Set **resource** to the correct *resourceId* which is a valid vCenter MOID (e.g. domain-c34 for a cluster).

Description:
Retrieve the network fabric status of the specified resource.

Responses:
Status Code: 200
Body: application/xml

```
<resourceStatuses>
 <resourceStatus>
   <resource>
     <objectId>resource-id</objectId>
     <objectTypeName>ClusterComputeResource</objectTypeName>
```

```
    <nsxmgrUuid>jfldj</nsxmgrUuid>
    <revision>2</revision>
    <type>
      <typeName>ClusterComputeResource</typeName>
    </type>
    <name>c-1</name>
    <scope>
      <id>datacenter-2</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>dc-1</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </resource>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>RED</status>
    <message></message>
    <installed>true</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.firewall</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
 </resourceStatus>
</resourceStatuses>
```

Working With Network Fabric Status of Child Resources

`GET /api/2.0/nwfabric/status/child/{parentResourceID}`

URI Parameters:

parentResourceID (required)	Parent resource ID

Description:

Retrieve the network fabric status of child resources of the specified resource.

Responses:
Status Code: 200
Body: application/xml

```
<resourceStatuses>
 <resourceStatus>
   <resource>
     <objectId>host-9</objectId>
     <objectTypeName>HostSystem</objectTypeName>
     <nsxmgrUuid>jfldj</nsxmgrUuid>
     <revision>4</revision>
     <type>
       <typeName>HostSystem</typeName>
     </type>
     <name>10.135.14.186</name>
     <scope>
       <id>domain-c34</id>
       <objectTypeName>ClusterComputeResource</objectTypeName>
       <name>c-1</name>
     </scope>
     <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
   </resource>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>RED</status>
     <message></message>
     <installed>true</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>UNKNOWN</status>
     <installed>false</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>UNKNOWN</status>
     <installed>false</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.firewall</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>UNKNOWN</status>
     <installed>false</installed>
   </nwFabricFeatureStatus>
 </resourceStatus>
</resourceStatuses>
```

Working With Status of Resources by Criterion

`GET /api/2.0/nwfabric/status/alleligible/{resourceType}`

URI Parameters:

`resourceType` (required)	Valid resource type

Description:
Retrieve status of resources by criterion.

Responses:
Status Code: 200
Body: application/xml

```
<resourceStatuses>
 <resourceStatus>
   <resource>
     <objectId>domain-c34</objectId>
     <objectTypeName>ClusterComputeResource</objectTypeName>
     <nsxmgrUuid>jfldj</nsxmgrUuid>
     <revision>2</revision>
     <type>
       <typeName>ClusterComputeResource</typeName>
     </type>
     <name>c-1</name>
     <scope>
       <id>datacenter-2</id>
       <objectTypeName>Datacenter</objectTypeName>
       <name>dc-1</name>
     </scope>
     <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
   </resource>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>RED</status>
     <message></message>
     <installed>true</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>UNKNOWN</status>
     <installed>false</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
     <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
     <featureVersion>5.5</featureVersion>
     <updateAvailable>false</updateAvailable>
     <status>UNKNOWN</status>
     <installed>false</installed>
   </nwFabricFeatureStatus>
   <nwFabricFeatureStatus>
```

```
    <featureId>com.vmware.vshield.firewall</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
 </resourceStatus>
 <resourceStatus>
  <resource>
    <objectId>domain-c32</objectId>
    <objectTypeName>ClusterComputeResource</objectTypeName>
    <nsxmgrUuid>jfldj</nsxmgrUuid>
    <revision>1</revision>
    <type>
      <typeName>ClusterComputeResource</typeName>
    </type>
    <name>c-2</name>
    <scope>
      <id>datacenter-12</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>dc-2</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </resource>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
  <nwFabricFeatureStatus>
    <featureId>com.vmware.vshield.firewall</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>false</installed>
  </nwFabricFeatureStatus>
 </resourceStatus>
</resourceStatuses>
```

Working With Locale ID Configuration For Clusters

`GET /api/2.0/nwfabric/clusters/{clusterID}`

URI Parameters:

clusterID	Cluster ID.

Description:
Retrieve the locale ID for the specified cluster.

`PUT /api/2.0/nwfabric/clusters/{clusterID}`

URI Parameters:

clusterID	Cluster ID.

Description:
Update the locale ID for the specified cluster.

Request:
Body: application/xml

```
<nwFabricClusterConfig>
 <localeId></localeId>
</nwFabricClusterConfig>
```

`DELETE /api/2.0/nwfabric/clusters/{clusterID}`

URI Parameters:

clusterID	Cluster ID.

Description:
Delete locale ID for the specified cluster.

Working With Locale ID Configuration for Hosts

`GET /api/2.0/nwfabric/hosts/{hostID}`

URI Parameters:

hostID	Host ID.

Description:
Retrieve the locale ID for the specified host.

`PUT /api/2.0/nwfabric/hosts/{hostID}`

URI Parameters:

hostID	Host ID.

Description:
Update the locale ID for the specified host.

Request:
Body: application/xml

```
<nwFabricHostConfig>
 <localeId></localeId>
</nwFabricHostConfig>
```

`DELETE /api/2.0/nwfabric/hosts/{hostID}`

URI Parameters:

hostID	Host ID.

Description:
Delete the locale ID for the specified host.

Working with Security Fabric and Security Services

The security fabric simplifies and automates deployment of security services and provide a platform for configuration of the elements that are required to provide security to workloads. These elements include:

Internal components:
- Guest Introspection Universal Service Virtual Machine
- Guest Introspection Mux
- Logical Firewall

External components:
- Partner OVFs / VIBs
- Partner vendor policy templates

For partner services, the overall workflow begins with registration of services by partner consoles, followed by deployment of the services by the administrator.

Subsequent workflow is as follows:

1. Select the clusters on which to deploy the security fabric (Mux, Traffic filter, USVM).
2. Specify an IP pool to be used with the SVMs (available only if the partner registration indicates requirement of static IPs)
3. Select portgroup (DVPG) to be used for each cluster (a default is pre-populated for the user).
4. Select datastore to be used for each cluster (a default is pre-populated for the user).
5. NSX Manager deploys the components on all hosts of the selected clusters.

Once you deploy the security fabric, an agency defines the configuration needed to deploy agents (host components and appliances). An agency is created per cluster per deployment spec associated with services. Agents are deployed on the selected clusters, and events / hooks for all the relevant actions are generated.

Request parameters

Parameter	Description
dataStore	Needs to be specified only in POST call. In PUT call, it should be left empty.
dvPortGroup	Optional. If not specified, then user will set the Agent using vCenter Server.
ipPool	Optional. If not specified, IP address is assigned through DHCP.

```
PUT /api/2.0/si/deploy
```

Query Parameters:

`startTime (optional)`	Specify time to start upgrade.

Description:
Upgrade service to recent version.

The datastore, dvPortGroup, and ipPool variables should either not be specified or have same value as provided at time of deployment.

Request:
Body: application/xml

```
<clusterDeploymentConfigs>
 <clusterDeploymentConfig>
```

```
   <clusterId></clusterId>
   <datastore></datastore>
   <services>
     <serviceDeploymentConfig>
       <serviceId></serviceId>
       <serviceInstanceId></serviceInstanceId>
       <dvPortGroup></dvPortGroup>
       <ipPool></ipPool>
     </serviceDeploymentConfig>
   </services>
 </clusterDeploymentConfig>
</clusterDeploymentConfigs>
```

`POST /api/2.0/si/deploy`

Query Parameters:

`startTime`	Time to start deployment task. If not specified, deploy immediately.

Description:
Deploy security fabric.

Request:
Body: application/xml

```
<clusterDeploymentConfigs>
 <clusterDeploymentConfig>
   <clusterId></clusterId>
   <datastore></datastore>
   <services>
     <serviceDeploymentConfig>
       <serviceId></serviceId>
       <dvPortGroup></dvPortGroup>
       <ipPool></ipPool>
     </serviceDeploymentConfig>
   </services>
 </clusterDeploymentConfig>
</clusterDeploymentConfigs>
```

Working With a Specified Service

`GET /api/2.0/si/deploy/service/{serviceID}`

URI Parameters:

`serviceID` (required)	Specified service.

Description:
Retrieve all clusters on which the service is installed.

```
DELETE /api/2.0/si/deploy/service/{serviceID}
```

URI Parameters:

serviceID (required)	Specified service.

Query Parameters:

clusters	Comma-separated list of cluster IDs from which to uninstall the service.
startTime	Time for uninstall to be scheduled. If not specified, uninstall immediately.

Description:
Uninstall specified service from specified clusters.

Working with Service Dependencies

Services installed through the security fabric may be dependent on other services. When an internal service is registered, a dependencyMap is maintained with the service-id and implementation type of the internal service.

When partner registers a new service, the security fabric looks up its implementation type in the dependencyMap to identify the service it depends on, if any. Accordingly, a new field in Service object called dependsOn-service-id is populated.

```
GET /api/2.0/si/deploy/service/{serviceID}/dependsOn
```

URI Parameters:

serviceID (required)	Specified service.

Description:
Retrieve service on which the specified service depends.

Working With Installed Services on a Cluster

```
GET /api/2.0/si/deploy/cluster/{clusterID}
```

URI Parameters:

clusterID (required)	Cluster ID

Description:
Retrieve all services deployed along with their status.

Responses:
Status Code: 200
Body: application/xml

```
<deployedServices>
 <deployedService>
   <deploymentUnitId>deploymentunit-1</deploymentUnitId>
   <serviceId>service-3</serviceId>
   <cluster>
     <objectId>domain-c41</objectId>
     <objectTypeName>ClusterComputeResource</objectTypeName>
     <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
     <revision>2</revision>
     <type>
       <typeName>ClusterComputeResource</typeName>
     </type>
     <name>Cluster-1</name>
     <scope>
       <id>datacenter-21</id>
       <objectTypeName>Datacenter</objectTypeName>
       <name>nasingh-dc</name>
     </scope>
      <extendedAttributes></extendedAttributes>
   </cluster>
   <serviceName>domain-c41_service-3</serviceName>
   <datastore>
     <objectId>datastore-29</objectId>
     <objectTypeName>Datastore</objectTypeName>
     <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
     <revision>1</revision>
     <type>
       <typeName>Datastore</typeName>
     </type>
     <name>datastore1</name>
      <extendedAttributes></extendedAttributes>
   </datastore>
   <dvPortGroup>
     <objectId>dvportgroup-45</objectId>
     <objectTypeName>DistributedVirtualPortgroup</objectTypeName>
     <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
     <revision>2</revision>
     <type>
       <typeName>DistributedVirtualPortgroup</typeName>
     </type>
     <name>dvPortGroup</name>
     <scope>
       <id>datacenter-21</id>
       <objectTypeName>Datacenter</objectTypeName>
       <name>nasingh-dc</name>
     </scope>
      <extendedAttributes></extendedAttributes>
   </dvPortGroup>
   <serviceStatus>SUCCEEDED</serviceStatus>
 </deployedService>
</deployedServices>
```

`DELETE /api/2.0/si/deploy/cluster/{clusterID}`

URI Parameters:

clusterID (required)	Cluster ID

Query Parameters:

services (optional)	Comma-separated list of service IDs to specify which services to uninstall. If this is not specified then all the services are uninstalled.
startTime	Time for uninstall to be scheduled. If not specified, do immediately.

Description:
Uninstall a service. Fails if you try to remove a service that another service depends on.

In order to uninstall services in any order, set parameter ignoreDependency to true.

Working with a Specific Service on a Cluster

`GET /api/2.0/si/deploy/cluster/{clusterID}/service/{serviceID}`

URI Parameters:

serviceID	Service ID on cluster
clusterID (required)	Cluster ID

Description:
Retrieve detailed information about the service.

Working with Data Collection for Activity Monitoring

Activity Monitoring provides visibility into your virtual network to ensure that security policies at your organization are being enforced correctly.

A Security policy may mandate who is allowed access to what applications. The Cloud administrator can generate Activity Monitoring reports to see if the IP based firewall rule that they set is doing the intended work. By providing user and application level detail, Activity Monitoring translates high level security policies to low level IP address and network based implementation.

Once you enable data collection for Activity Monitoring, you can run reports to view inbound traffic (such as virtual machines being accessed by users) as well as outbound traffic (resource utilization, interaction between inventory containers, and AD groups that accessed a server).

You must enable data collection for one or more virtual machines on a vCenter Server before running an Activity Monitoring report. Before running a report, ensure that the enabled virtual machines are active and are generating network traffic.

You should also register NSX Manager with the AD Domain Controller. See "Working with Domains".

Note that only active connections are tracked by Activity Monitoring. Virtual machine traffic blocked by firewall rules at the vNIC level is not reflected in reports.

In case of an emergency such as a network overload, you can turn off data collection at a global level. This overrides all other data collection settings.

Some API calls may require the VMID, which is the MOID of the guest virtual machine. You can retrieve this by queuing the vCenter mob structure (https:VC-IP-Address/mob). The VMID is listed under host structure.

Working With Data Collection on a Specific Virtual Machine

You must enable data collection at least five minutes before running an Activity Monitoring report.

`POST /api/1.0/eventcontrol/vm/{vmID}/request`

URI Parameters:

vmID (required)	MOID of the guest vm

Description:
Enable or disable data collection on a virtual machine

Set **value** to *enabled* or *disabled*.

Request:
Body: application/xml

```
<perVmConfig>
 <actions>
   <action>
     <type>per_vm_config</type>
     <value>enabled</value>
   </action>
 </actions>
</perVmConfig>
```

Override Data Collection

`POST /api/1.0/eventcontrol/eventcontrol-root/request`

Description:
Turn data collection on or off at the global level.

In case of an emergency such as a network overload, you can turn off data collection at a global level (kill switch). This overrides all other data collection settings.

Set **value** to *enabled* or *disabled*.

Request:
Body: application/xml

```
<request>
 <actions>
   <action>
     <type>global_switch</type>
     <value>disabled</value>
   </action>
 </actions>
</request>
```

Retrieve Data Collection Configuration for a Specific Virtual Machine

When reporting per virtual machine configuration, current kill switch status is also reported too. The effective configuration of a virtual machine is determined by both kill switch config and per virtual machine configuration. If kill switch is on, event collection is effectively disabled regardless of what its per virtual machine configuration is; if kill switch is off, per virtual machine configuration determines whether event collection should be performed for this virtual machine.

`GET /api/1.0/eventcontrol/config/vm/{vmID}`

URI Parameters:

vmID (required)	MOID of the guest vm

Description:
Retrieve per VM configuration for data collection.

Responses:
Status Code: 200
Body: application/xml

```
<perVmConfig>
 <actions>
   <action>
     <type>global_switch</type>
     <value>disabled</value>
```

```
   </action>
   <action>
     <type>per_vm_config</type>
     <value>enabled</value>
   </action>
 </actions>
</perVmConfig>
```

Working with Activity Monitoring

Working With Aggregated User Activity

Get aggregated user activity (action records) using parameters. Requires that NSX Guest Introspection is configured, NSX Manager must be registered with Active Directory, and data collection is enabled on one or more VMs.

```
GET /api/3.0/ai/records
```

Query Parameters:

`query` (required)	Name of report (resource,adg,containers,sam,vma).
`interval` (required)	Relative time to current time (number followed by either m,h,d,s).
`stime` (optional)	Start time for query. **interval** is used if **stime** and **etime** are not specified.
`etime` (optional)	End time for query. **interval** is used if **stime** and **etime** are not specified. example: 2012-02-29T21:00
`param`	Parameter to be applied to query <param-name>:<param-type>:<comma-separated-values>:<operator>
`pagesize`	The number of results to return. Recommended range is 100-2000.
`startindex`	The starting point for returning results.

Description:

View Outbound Activity

You can view what applications are being run by a security group or desktop pool and then drill down into the report to find out which client applications are making outbound connections by a particular group of users. You can also discover all user groups and users who are accessing a particular application, which can help you determine if you need to adjust identity firewall in your environment.

- query=*resource*
- param=<param-name>:<param-type>:<comma-separated-values>:<operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
    - *app*
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
    - for dest: *VIRTUAL_MACHINE*
    - for app: *SRC_APP*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE* (default is *INCLUDE*).

Example: View user activities to VM ID 1 originating from application ID 1

```
GET /api/3.0/ai/records?query=resource&interval=60m&param=src:DIRECTORY_GROUP
&param=dest:VIRTUAL_MACHINE:1&param=app:SRC_APP:1
```

View Inbound Activity

You can view all inbound activity to a server by desktop pool, security group, or AD group.

- query=*sam*
- param=<param-name>:<param-type>:<comma-separated-values>:<operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
    - *app*
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
    - for dest: *VIRTUAL_MACHINE*
    - for app: *DEST_APP*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE*, *NOT* (default is *INCLUDE*).

Example: View user activities to VM ID 1 originating from application ID 1

```
GET /api/3.0/ai/records?query=containers&interval=60m&param=dest:SECURITY_GROUP:1:EXCLUDE
&param=src:SECURITY_GROUP:1
```

View Interaction between Inventory Containers

You can view the traffic passing between defined containers such as AD groups, security groups and/or desktop pools. This can help you identify and configure access to shared services and to resolve misconfigured relationships between Inventory container definitions, desktop pools and AD groups.

- query=*containers*
- param=<param-name>:<param-type>:<comma-separated-values>:<operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
    - for dest: *SECURITY_GROUP*, DESKTOP_POOL*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE*, or *NOT* (default *is* INCLUDE*).

Example: View interaction between inventory containers

```
GET /api/3.0/ai/records?query=containers&interval=60m&param=dest:SECURITY_GROUP:1:EXCLUDE
&param=src:SECURITY_GROUP:1
```

View Outbound AD Group Activity

You can view the traffic between members of defined Active Directory groups and can use this data to fine tune your firewall rules.

- query=*adg*
- param=<param-name>:<param-type>:<comma-separated-values>:<operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *adg*
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DESKTOP_POOL*
    - for adg: *USER*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE* (default *is* INCLUDE*).

Example: View outbound AD group activity

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/records?query=adg&interval=24h&param=adg:USER:1:I NCLUDE
&param=src:SECURITY_GROUP:1:EXCLUDE
```

Working with User Details

`GET /api/3.0/ai/userdetails`

Query Parameters:

`query` (required)	Name of report (resource,adg,containers,sam,vma)
`interval` (required)	Relative time to current time (number followed by either m,h,d,s)
`stime`	Start time for query
`etime`	End time for query
`param`	Parameter to be applied to query <param-name>:<param-type>:<comma-separated-values>:<operator>
`pagesize`	The number of results to return. Recommended range is 100-2000.
`startindex`	The starting point for returning results.

Description:

View Outbound Activity

You can view what applications are being run by a security group or desktop pool and then drill down into the report to find out which client applications are making outbound connections by a particular group of users. You can also discover all user groups and users who are accessing a particular application, which can help you determine if you need to adjust identity firewall in your environment.

- query=*resource*
- param=<param-name><param-type><comma-separated-values><operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
    - *app*
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
    - for dest: *IP* - a valid IP address in dot notation, xx.xx.xx.xx
    - for app: *SRC_APP*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE* (default is *INCLUDE*).

Example: View user activities to VM ID 1 originating from application ID 1

```
GET /api/3.0/ai/userdetails?query=resource&stime=2012-10-15T00:00:00&etime=2012-10-20T00:00:00
&param=src:DIRECTORY_GROUP:2&param=app:SRC_APP:16&param=dest:IP:172.16.4.52
```

View Inbound Activity

You can view all inbound activity to a server by desktop pool, security group, or AD group.

- query=*sam*
- param=<param-name><param-type><comma-separated-values><operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
    - *app* (required)

- <param-type> is one of:
  - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
  - for dest: *VIRTUAL_MACHINE*
  - for app: *DEST_APP*
- <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> is one of *INCLUDE*, *EXCLUDE*, *NOT* (default is *INCLUDE*).

Example: View user activities to VM ID 1 originating from application ID 1

```
GET /api/3.0/userdetails?query=sam&interval=60m&param=app:DEST_APP:1:EXCLUDE
&param=dest:IP:1:EXCLUDE&param=src:SECURITY_GROUP:1:EXCLUDE
```

View Interaction between Inventory Containers

You can view the traffic passing between defined containers such as AD groups, security groups and/or desktop pools. This can help you identify and configure access to shared services and to resolve misconfigured relationships between Inventory container definitions, desktop pools and AD groups.

- query=*containers*
- param=<param-name><param-type><comma-separated-values><operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *dest* (required)
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DIRECTORY_GROUP*, *DESKTOP_POOL*
    - for dest: *SECURITY_GROUP*, DESKTOP_POOL*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE*, or *NOT* (default *is* INCLUDE*).

Example: View interaction between inventory containers

```
GET /api/3.0/ai/userdetails?query=containers&interval=60m&param=dest:SECURITY_GROUP:1:EXCLUDE
&param=src:SECURITY_GROUP:1
```

View Outbound AD Group Activity

You can view the traffic between members of defined Active Directory groups and can use this data to fine tune your firewall rules.

- query=*adg*
- param=<param-name><param-type><comma-separated-values><operator>, where:
  - <param-name> is one of:
    - *src* (required)
    - *adg*
  - <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DESKTOP_POOL*
    - for adg: *USER*
  - <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
  - <operator> is one of *INCLUDE*, *EXCLUDE* (default is *INCLUDE*).

Example: View outbound AD group activity

```
GET /api/3.0/ai/userdetails?query=adg&interval=24h&param=adg:USER:1:INCLUDE
&param=src:SECURITY_GROUP:1:EXCLUDE
```

View Virtual Machine Activity Report

- query=*vma*
- param=<param-name><param-type><comma-separated-values><operator>, where:
  - <param-name> is one of:
    - *src*

    - *dst*
    - *app*
    - If no parameters are passed, then this would show all SAM activities
- <param-type> is one of:
    - for src: *SECURITY_GROUP*, *DESKTOP_POOL*
    - for dst: *VIRTUAL_MACHINE*, *VM_UUID*
    - for app - *SRC_APP* or *DEST_APP*
- <comma-separated-values> is a comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> is one of *INCLUDE*, *EXCLUDE* (default is *INCLUDE*).

Example: View outbound AD group activity

```
GET /api/3.0/ai/userdetails?query=vma&interval=60m&param=dest:VIRTUAL_MACHINE:1
&param=app:DEST_APP:16
```

Working With a Specific User

```
GET /api/3.0/ai/user/{userID}
```

URI Parameters:

`userID` (required)	User ID

Description:
Retrieve details for a specific user.

Working With Applications

```
GET /api/3.0/ai/app
```

Description:
Retrieve app details.

Working with a Specific Application

```
GET /api/3.0/ai/app/{appID}
```

URI Parameters:

`appID` (required)	Specified app ID.

Description:
Retrieve details for specific app.

Working With Discovered Hosts

`GET /api/3.0/ai/host`

Description:
Retrieve list of all discovered hosts (both by agent introspection and LDAP Sync) and their detail.

Working with a Specific Discovered Host

`GET /api/3.0/ai/host/{hostID}`

URI Parameters:

`hostID` (required)	Specified host ID.

Description:
Get host details.

Working With Desktop Pools

`GET /api/3.0/ai/desktoppool`

Description:
Retrieve list of all discovered desktop pools by agent introspection.

Working with a Specific Desktop Pool

`GET /api/3.0/ai/desktoppool/{desktoppoolID}`

URI Parameters:

`desktoppoolID` (required)	Specified desktop pool.

Description:
Retrieve specific desktop pool details.

Working with Virtual Machines

`GET /api/3.0/ai/vm`

Description:
Retrieve list of all discovered VMs.

Working with a Specific Virtual Machine

`GET /api/3.0/ai/vm/{vmID}`

URI Parameters:

vmID (required)	VM ID

Description:
Retrieve details about a specific virtual machine.

Working with LDAP Directory Groups

`GET /api/3.0/ai/directorygroup`

Description:
Retrieve list of all discovered (and configured) LDAP directory groups.

Working with a Specific LDAP Directory Group

`GET /api/3.0/ai/directorygroup/{directorygroupID}`

URI Parameters:

directorygroupID (required)	Specified directory group.

Description:
Retrieve details about a specific directory group.

Working with a Specific User's Active Directory Groups

`GET /api/3.0/ai/directorygroup/user/{userID}`

URI Parameters:

`userID` (required)	User ID.

Description:
Retrieve Active Directory groups that user belongs to.

Working with Security Groups

`GET /api/3.0/ai/securitygroup`

Description:
Retrieve list of all observed security groups.

Observed entities are the ones that are reported by the agents. For example, if a host activity is reported by an agent and if that host belongs to a security group then that security group would reported as observed in SAM database.

Working with a Specific Security Group

`GET /api/3.0/ai/securitygroup/{secgroupID}`

URI Parameters:

`secgroupID` (required)	Specified security group.

Description:
Retrieve details about specific security group.

Working with Domains

After you create a domain, you can apply a security policy to it and run queries to view the applications and virtual machines being accessed by the users of a domain.

Registering Domains

You can a register one or more Windows domains with an NSX Manager and associated vCenter server. NSX Manager gets group and user information as well as the relationship between them from each domain that it is registered with. NSX Manager also retrieves Active Directory credentials. You can apply security policies on an Active Directory domain and run queries to get information on virtual machines and applications accessed by users within an Active Directory domain.

Parameter Values for Registering or Updating a Domain

Parameter Name	Description	Mandatory?
ID	Domain id. If you want to create a new domain, do not provide this value. Otherwise, the system will find an existing domain object by this ID and update it.	true if update existing domain
name	Domain name. This should be domain's full qualified name. In case agent discovered, this will be NetBIOS name, so you need to update it to FQN in order to support LDAP sync and event log reader.	true if creating a new domain.
description	Domain description	false
type	Domain type. Valid value include: AGENT_DISCOVERED, ActiveDirectory, SPECIAL (Do NOT modify SPECIAL domain). For LDAP sync and event log reader work, this need to be set to ActiveDirectory.	true if creating a new domain
netbiosName	NetBIOS name of domain. This is Domain's NetBIOS name. Check windows domain setting, for value of it. Normally Agent report domain name is NetBIOS name. But confirm from Windows domain setting.	false
baseDn	Domain's Base DN (for LDAP sync). Base DN is REQUIRED for LDAP Sync. If you have a domain like: w2k3.vshield.vmware.com, the base DN is very likely to be: DC=w2k3,DC=vshield,DC=vmware,DC=com. Another example is: domain name is: vs4.net, the base DN should be: DC=vs4,DC=net. You can use a LDAP client and connect to domain controller to find the domain's base DN.	false
rootDn	LDAP Sync root DN. Specify where should LDAP sync start from LDAP tree. This could be absolute path, for example: OU=Engineer,DC=vs4,DC=net, or relative path (relate to Base DN), for example: OU=Engineer.	false
securityId	Domain's Security ID (SID). This should be filled by LDAP sync process, and should not need to be modified.	false

username	Domain's User name (Used for LDAP Sync and/or Event Log reader)	false
password	User password	false
eventLogUsername	Domain's event log reader username (will use above username if this is NULL)	false
eventLogPassword	Domain's event log reader password	false

`POST /api/1.0/directory/updateDomain`

Description:
Register or update a domain with NSX Manager

Request:
Body: application/xml

```
<DirectoryDomain>
 <name>example.com</name>
 <netbiosName>Example</netbiosName>
 <username>Administrator</username>
 <password>xxx</password>
</DirectoryDomain>
```

Retrieve LDAP Domains

`GET /api/1.0/directory/listDomains`

Description:
Retrieve all agent discovered (or configured) LDAP domains.

Responses:
Status Code: 200
Body: application/xml

```
<DirectoryDomains>
 <DirectoryDomain>
   <id>2</id>
   <name>vs4.net</name>
   <type>ActiveDirectory</type>
   <netbiosName>VS4</netbiosName>
   <username>Administrator</username>
   <baseDn>DC=vs4,DC=net</baseDn>
 </DirectoryDomain>
</DirectoryDomains>
```

Delete a Specific Domain

`DELETE /api/1.0/directory/deleteDomain/{ID}`

URI Parameters:

ID (required)	Domain ID.

Description:
Delete domain.

Create LDAP Server

`POST /api/1.0/directory/updateLdapServer`

Description:
Create LDAP server.

Request:
Body: application/xml

```
<LDAPServer>
 <domainId>4</domainId>
 <hostName>10.142.72.70</hostName>
 <enabled>true</enabled>
</LDAPServer>
```

Query LDAP Servers for a Domain

`GET /api/1.0/directory/listLdapServersForDomain/{domainID}`

URI Parameters:

domainID (required)	Specified domain.

Description:
Query LDAP servers for a domain.

Start LDAP Full Sync

`PUT /api/1.0/directory/fullSync/{domainID}`

URI Parameters:

domainID (required)	Specified domain.

Description:
Start LDAP full sync.

Start LDAP Delta Sync

`PUT /api/1.0/directory/deltaSync/{domainID}`

URI Parameters:

domainID (required)	Specified domain.

Description:
Start LDAP delta sync.

Delete LDAP Server

`DELETE /api/1.0/directory/deleteLdapServer/{serverID}`

URI Parameters:

serverID (required)	Specified LDAP server.

Description:
Delete LDAP server.

EventLog Server

`POST /api/1.0/directory/updateEventLogServer`

Description:
Create EventLog server.

Request:
Body: application/xml

```
<EventlogServer>
 <id>1</id>
 <domainId>4</domainId>
 <hostName>10.142.72.70</hostName>
 <enabled>false</enabled>
```

```
</EventlogServer>
```

Working with EventLog Servers for a Domain

```
GET /api/1.0/directory/listEventLogServersForDomain/{domainID}
```

URI Parameters:

domainID (required)	Specified domain.

Description:
Query EventLog servers for a domain.

Delete EventLog Server

```
DELETE /api/1.0/directory/deleteEventLogServer/{serverID}
```

URI Parameters:

serverID (required)	Specified EventLog server ID.

Description:
Delete EventLog server.

Working with Mapping Lists

Working With User to IP Mappings

`GET /api/1.0/identity/userIpMapping`

Description:
Query user-to-ip mapping list from database.

Working With Host to IP Mappings

`GET /api/1.0/identity/hostIpMapping`

Description:
Query host-to-ip mapping list from database.

Working With IP to User Mappings

`GET /api/1.0/identity/ipToUserMapping`

Description:
Retrieve set of users associated with a given set of IP addresses during a specified time period. Since more than one user can be associated with a single IP address during the specified time period, each IP address can be associated with zero or more (i.e a SET of) users.

Working With User Domain Groups

`GET /api/1.0/identity/directoryGroupsForUser`

Description:
Query set of Windows Domain Groups (AD Groups) to which the specified user belongs.

Working with a Specific Static User Mapping

`POST /api/1.0/identity/staticUserMapping/{userID}/{IP}`

URI Parameters:

`userID` (required)	User ID
`IP` (required)	IP address

Description:
Create static user IP mapping.

Working with Static User Mappings

`GET /api/1.0/identity/staticUserMappings`

Description:
Query static user IP mapping list.

Working with Static User IP Mappings for a Specific User

`GET /api/1.0/identity/staticUserMappingsbyUser/{userID}`

URI Parameters:

`userID` (required)	User ID

Description:
Query static user IP mapping for specified user.

`DELETE /api/1.0/identity/staticUserMappingsbyUser/{userID}`

URI Parameters:

`userID` (required)	User ID

Description:
Delete static user IP mapping for specified user.

Working With Static User IP Mappings for a Specific IP

`GET /api/1.0/identity/staticUserMappingsbyIP/{IP}`

URI Parameters:

`IP` (required)	IP address

Description:

Query static user IP mapping for specified IP.

```
DELETE /api/1.0/identity/staticUserMappingsbyIP/{IP}
```

URI Parameters:

IP (required)	IP address

Description:
Delete static user IP mapping for specified IP.

Working with Activity Monitoring Syslog Support

Enable Syslog Support

`POST /api/1.0/sam/syslog/enable`

Description:
Enable syslog support.

Disable Syslog Support

`POST /api/1.0/sam/syslog/disable`

Description:
Disable syslog support.

Working with Solution Integrations

Working With Agents on a Specific Host

`GET /api/2.0/si/host/{hostID}/agents`

URI Parameters:

hostID (required)	Specified host

Description:
Retrieves all agents on the specified host. The response body contains agent IDs for each agent, which you can use to retrieve details about that agent.

Responses:
Status Code: 200
Body: application/xml

```
<fabricAgents>
 <agent>
   <agentId>nsxmgragent-1</agentId>
   <agentName>agent name</agentName>
   <serviceId>service-6</serviceId>
   <serviceName>EndpointService</serviceName>
   <operationalStatus>ENABLED</operationalStatus>
   <progressStatus>IN_PROGRESS</progressStatus>
   <vmId>vm-92</vmId>
   <host>host-10</host>
   <allocatedIpAddress>
     <id>2</id>
     <ipAddress>10.112.5.182</ipAddress>
     <gateway>10.112.5.253</gateway>
     <prefixLength>23</prefixLength>
     <dnsServer1>10.112.0.1</dnsServer1>
     <dnsServer2>10.112.0.2</dnsServer2>
     <dnsSuffix></dnsSuffix>
     <subnetId>subnet-1</subnetId>
   </allocatedIpAddress>
   <serviceStatus>
     <status>WARNING</status>
     <errorId>partner_error</errorId>
     <errorDescription>partner_error</errorDescription>
   </serviceStatus>
   <hostInfo>
     <objectId>host-10</objectId>
     <objectTypeName>HostSystem</objectTypeName>
     <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
     <revision>1</revision>
     <type>
       <typeName>HostSystem</typeName>
     </type>
     <name>10.112.5.173</name>
     <scope>
```

```
      <id>domain-c7</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>Kaustubh-CL</name>
    </scope>
    <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
  </hostInfo>
  <initialData>partner data if present</initialData>
 </agent>
</fabricAgents>
```

Working with a Specific Agent

`GET /api/2.0/si/agent/{agentID}`

URI Parameters:

agentID (required)	Specified agent

Description:
Retrieve agent (host components and appliances) details.

Responses:
Status Code: 200
Body: application/xml

```
<agent>
 <agentId>nsxmgragent-1</agentId>
 <agentName>agent name</agentName>
 <serviceId>service-6</serviceId>
 <serviceName>EndpointService</serviceName>
 <operationalStatus>ENABLED</operationalStatus>
 <progressStatus>IN_PROGRESS</progressStatus>
 <vmId>vm-92</vmId>
 <host>host-10</host>
 <allocatedIpAddress>
   <id>2</id>
   <ipAddress>10.112.5.182</ipAddress>
   <gateway>10.112.5.253</gateway>
   <prefixLength>23</prefixLength>
   <dnsServer1>10.112.0.1</dnsServer1>
   <dnsServer2>10.112.0.2</dnsServer2>
   <dnsSuffix></dnsSuffix>
   <subnetId>subnet-1</subnetId>
 </allocatedIpAddress>
 <serviceStatus>
   <status>WARNING</status>
   <errorId>partner_error</errorId>
   <errorDescription>partner_error</errorDescription>
 </serviceStatus>
 <hostInfo>
   <objectId>host-10</objectId>
   <objectTypeName>HostSystem</objectTypeName>
```

```
   <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
   <revision>1</revision>
   <type>
     <typeName>HostSystem</typeName>
   </type>
   <name>10.112.5.173</name>
   <scope>
     <id>domain-c7</id>
     <objectTypeName>ClusterComputeResource</objectTypeName>
     <name>Kaustubh-CL</name>
   </scope>
   <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
 </hostInfo>
 <initialData>partner data if present</initialData>
</agent>
```

Working with Agents on a Specific Deployment

`GET /api/2.0/si/deployment/{deploymentunitID}/agents`

URI Parameters:

deploymentunitID (required)	Specified deployment.

Description:
Retrieve all agents for the specified deployment.

Responses:
Status Code: 200
Body: application/xml

```
<fabricAgents>
 <agent>
   <agentId>nsxmgragent-1</agentId>
   <agentName>agent name</agentName>
   <serviceId>service-6</serviceId>
   <serviceName>EndpointService</serviceName>
   <operationalStatus>ENABLED</operationalStatus>
   <progressStatus>IN_PROGRESS</progressStatus>
   <vmId>vm-92</vmId>
   <host>host-10</host>
   <allocatedIpAddress>
     <id>2</id>
     <ipAddress>10.112.5.182</ipAddress>
     <gateway>10.112.5.253</gateway>
     <prefixLength>23</prefixLength>
     <dnsServer1>10.112.0.1</dnsServer1>
     <dnsServer2>10.112.0.2</dnsServer2>
     <dnsSuffix></dnsSuffix>
     <subnetId>subnet-1</subnetId>
   </allocatedIpAddress>
   <serviceStatus>
```

```
    <status>WARNING</status>
    <errorId>partner_error</errorId>
    <errorDescription>partner_error</errorDescription>
  </serviceStatus>
  <hostInfo>
    <objectId>host-10</objectId>
    <objectTypeName>HostSystem</objectTypeName>
    <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
    <revision>1</revision>
    <type>
      <typeName>HostSystem</typeName>
    </type>
    <name>10.112.5.173</name>
    <scope>
      <id>domain-c7</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>Kaustubh-CL</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </hostInfo>
  <initialData>partner data</initialData>
 </agent>
</fabricAgents>
```

Working With Conflicting Agencies

When the NSX Manager database backup is restored to an older point in time, it is possible that deployment units for some EAM Agencies are missing. These methods help the administrator identify such EAM Agencies and take appropriate action.

`GET /api/2.0/si/fabric/sync/conflicts`

Description:
Retrieve conflicting deployment units and EAM agencies, if any, and the allowed operations on them.

Responses:
Status Code: 200
Body: application/xml

```
<fabricSyncConflictInfo>
 <conflictExist>true</conflictExist>
 <agencies>
   <agenciesInfo>
     <agencyConflictInfo>
       <agencyId>agency-150</agencyId>
       <agencyName>_VCNS_264_nasingh-cluster1_VMware Endpoint</agencyName>
     </agencyConflictInfo>
   </agenciesInfo>
   <allowedOperations>
     <conflictResolverOperation>DELETE</conflictResolverOperation>
     <conflictResolverOperation>RESTORE</conflictResolverOperation>
   </allowedOperations>
 </agencies>
```

```
</fabricSyncConflictInfo>
```

`PUT /api/2.0/si/fabric/sync/conflicts`

Description:
Create deployment units for conflicting EAM Agencies, delete conflicting EAM agencies, or delete deployment units for conflicting EAM agencies.

Create deployment units for conflicting EAM agencies

```
<conflictResolverInfo>
 <agencyAction>RESTORE</agencyAction>
</conflictResolverInfo>
```

Delete conflicting EAM agencies

```
<conflictResolverInfo>
 <agencyAction>DELETE</agencyAction>
</conflictResolverInfo>
```

Delete deployment units for conflicting EAM agencies

```
<conflictResolverInfo>
 <deploymentUnitAction>DELETE</deploymentUnitAction>
</conflictResolverInfo>
```

Request:
Body: application/xml

```
<conflictResolverInfo>
 <agencyAction></agencyAction>
</conflictResolverInfo>
```

Working with MAC Address Set Grouping Objects

You can create a MAC address set on the specified scope. On success, the API returns a string identifier for the new MAC address set.

Working With a Specific MAC Address Set

`GET /api/2.0/services/macset/{macsetId}`

URI Parameters:

macsetId (required)	Specified MAC address set ID (can be retrieved by listing the MAC address set on a scope). The revision parameter is incremented and the value parameter lists the updated IP addresses.

Description:
Retrieve details about a MAC address set.

Responses:
Status Code: 200
Body: application/xml

```
<macset>
 <objectId>macset-1</objectId>
 <objectTypeName>MACSet</objectTypeName>
 <vsmUuid>4226CACF-0558-AFF3-5D92-279B201C40E2</vsmUuid>
 <nodeId>72eee9ab-bb75-49ba-a782-d7dffedd180a</nodeId>
 <revision>4</revision>
 <type>
 <typeName>MACSet</typeName>
 </type>
 <name>system-generated-broadcast-macset</name>
 <scope>
   <id>globalroot-0</id>
   <objectTypeName>GlobalRoot</objectTypeName>
   <name>Global</name>
 </scope>
 <clientHandle></clientHandle>
 <extendedAttributes>
   <extendedAttribute>
     <name>isReadOnly</name>
     <value>true</value>
   </extendedAttribute>
   <extendedAttribute>
     <name>isHidden</name>
     <value>true</value>
   </extendedAttribute>
   <extendedAttribute>
     <name>facadeHidden</name>
     <value>true</value>
   </extendedAttribute>
 </extendedAttributes>
 <isUniversal>false</isUniversal>
```

```
 <universalRevision>0</universalRevision>
 <inheritanceAllowed>false</inheritanceAllowed>
 <value>FF:FF:FF:FF:FF:FF</value>
</macset>
```

PUT /api/2.0/services/macset/{macsetId}

URI Parameters:

macsetId (required)	Specified MAC address set ID (can be retrieved by listing the MAC address set on a scope). The revision parameter is incremented and the value parameter lists the updated IP addresses.

Description:
Modify an existing MAC address set.

Request:
Body: application/xml

```
<macset>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <value></value>
</macset>
```

DELETE /api/2.0/services/macset/{macsetId}

URI Parameters:

macsetId (required)	Specified MAC address set ID (can be retrieved by listing the MAC address set on a scope). The revision parameter is incremented and the value parameter lists the updated IP addresses.

Query Parameters:

force (optional)	Indicates forced or unforced delete. With forced delete, the object is deleted even if used in other places such as firewall rules, causing invalid referrals. For unforced delete, the object is deleted only if it is no used by other configurations; otherwise the delete fails.

Description:
Delete a MAC address set.

Working with MAC Address Sets on a Specific Scope

`GET /api/2.0/services/macset/scope/{scopeId}`

URI Parameters:

scopeId (required)	Can be "globalroot-0", "universalroot-0" or datacenterId in upgrade use cases. The value parameter can include a single MAC identifier or a comma separated set of MAC identifiers. Universal MAC address sets are read-only from secondary managers.

Description:
List MAC address sets on the specified scope.

Responses:
Status Code: 200
Body: application/xml

```
<list>
 <macset>
   <objectId>macset-1</objectId>
   <objectTypeName>MACSet</objectTypeName>
   <vsmUuid>4226CACF-0558-AFF3-5D92-279B201C40E2</vsmUuid>
   <nodeId>72eee9ab-bb75-49ba-a782-d7dffedd180a</nodeId>
   <revision>4</revision>
   <type>
     <typeName>MACSet</typeName>
   </type>
   <name>system-generated-broadcast-macset</name>
   <scope>
     <id>globalroot-0</id>
     <objectTypeName>GlobalRoot</objectTypeName>
     <name>Global</name>
   </scope>
   <clientHandle></clientHandle>
   <extendedAttributes>
     <extendedAttribute>
       <name>isReadOnly</name>
       <value>true</value>
     </extendedAttribute>
     <extendedAttribute>
       <name>isHidden</name>
       <value>true</value>
     </extendedAttribute>
     <extendedAttribute>
       <name>facadeHidden</name>
       <value>true</value>
     </extendedAttribute>
   </extendedAttributes>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
   <inheritanceAllowed>false</inheritanceAllowed>
   <value>FF:FF:FF:FF:FF:FF</value>
 </macset>
</list>
```

`POST /api/2.0/services/macset/scope/{scopeId}`

URI Parameters:

scopeId (required)	Can be "globalroot-0", "universalroot-0" or datacenterId in upgrade use cases. The value parameter can include a single MAC identifier or a comma separated set of MAC identifiers. Universal MAC address sets are read-only from secondary managers.

Description:
Create a MAC address set on the specified scope.

Request:
Body: application/xml

```
<macset>
 <objectId></objectId>
 <type>
   <typeName></typeName>
 </type>
 <description></description>
 <name></name>
 <revision></revision>
 <objectTypeName></objectTypeName>
 <value></value>
</macset>
```

Working with Alarms from a Specific Source

Some system alerts will show up as alarms in the NSX dashboard. You can view and resolve alarms from a specific source.

`GET /api/2.0/services/alarms/{sourceId}`

URI Parameters:

`sourceId`	ID of the object for which you want to manage alarms. *sourceId* can be the ID of a cluster, host, resource pool, security group, or edge.

Description:
Retrive all alarms from the specified source.

`POST /api/2.0/services/alarms/{sourceId}`

URI Parameters:

`sourceId`	ID of the object for which you want to manage alarms. *sourceId* can be the ID of a cluster, host, resource pool, security group, or edge.

Query Parameters:

`action`	Use *action=resolve* to resolve system alarms.

Description:
Resolve all alarms for the specified source.

Alarms will resolve automatically when the cause of the alarm is resolved. For example, if an NSX Edge appliance is powered off, this will trigger an alarm. If you power the NSX Edge appliance back on, the alarm will resolve. If however, you delete the NSX Edge appliance, the alarm will persist, because the alarm cause was never resolved. In this case, you may want to manually resolve the alarm. Resolving the alarms will clear them from the NSX dashboard.

Use `GET /api/2.0/services/alarms/{sourceId}` to retrieve the list of alarms for the source. Use this response as the request body for the `POST` call.

Request:
Body: application/xml

```
<systemAlarms>
   <systemAlarm>
       <eventId>79965</eventId>
       <timestamp>1485556529744</timestamp>
       <severity>High</severity>
       <eventSource>edge-3</eventSource>
       <eventCode>130027</eventCode>
       <message>NSX Edge VM (vmId : vm-430) is powered off. Please use vsphere client to power on Edge
VM</message>
       <module>NSX Edge Communication Agent</module>
       <objectId>edge-3</objectId>
       <reporterName>vShield Manager</reporterName>
       <reporterType>4</reporterType>
       <sourceType>4</sourceType>
       <isResourceUniversal>false</isResourceUniversal>
```

```
        <eventMetadata>
            <data>
                <key>edgeVmVcUUId</key>
                <value>502e05c2-380f-998c-35ec-1f48991fe7e0</value>
            </data>
        </eventMetadata>
        <resolutionAttempted>false</resolutionAttempted>
        <resolvable>true</resolvable>
        <alarmId>79965</alarmId>
        <alarmCode>130027</alarmCode>
        <alarmSource>edge-3</alarmSource>
        <target>
            <objectId>vm-430</objectId>
            <objectTypeName>VirtualMachine</objectTypeName>
            <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
            <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
            <revision>18</revision>
            <type>
                <typeName>VirtualMachine</typeName>
            </type>
            <name>Perimeter-Gateway-01-0</name>
            <scope>
                <id>domain-c41</id>
                <objectTypeName>ClusterComputeResource</objectTypeName>
                <name>Management &amp;amp; Edge Cluster</name>
            </scope>
            <clientHandle></clientHandle>
            <extendedAttributes></extendedAttributes>
            <isUniversal>false</isUniversal>
            <universalRevision>0</universalRevision>
        </target>
        <alarmBeingResolved>false</alarmBeingResolved>
        <alarmMetadata>
            <data>
                <key>edgeVmVcUUId</key>
                <value>502e05c2-380f-998c-35ec-1f48991fe7e0</value>
            </data>
        </alarmMetadata>
    </systemAlarm>
    <systemAlarm>
        <eventId>79967</eventId>
        <timestamp>1485556529774</timestamp>
        <severity>High</severity>
        <eventSource>edge-3</eventSource>
        <eventCode>130033</eventCode>
        <message>NSX Edge VM (vmId : vm-430) is not responding to NSX manager health check. Please check
NSX manager logs for details.</message>
        <module>NSX Edge Health Check</module>
        <objectId>edge-3</objectId>
        <reporterName>vShield Manager</reporterName>
        <reporterType>4</reporterType>
        <sourceType>4</sourceType>
        <isResourceUniversal>false</isResourceUniversal>
        <eventMetadata>
            <data>
                <key>edgeVmVcUUId</key>
                <value>502e05c2-380f-998c-35ec-1f48991fe7e0</value>
            </data>
            <data>
                <key>edgeId</key>
                <value>edge-3</value>
            </data>
```

```
            <data>
                <key>edgeVmName</key>
                <value>Perimeter-Gateway-01-0</value>
            </data>
            <data>
                <key>edgeVmId</key>
                <value>vm-430</value>
            </data>
        </eventMetadata>
        <resolutionAttempted>false</resolutionAttempted>
        <resolvable>true</resolvable>
        <alarmId>79967</alarmId>
        <alarmCode>130033</alarmCode>
        <alarmSource>edge-3</alarmSource>
        <target>
            <objectId>vm-430</objectId>
            <objectTypeName>VirtualMachine</objectTypeName>
            <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
            <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
            <revision>18</revision>
            <type>
                <typeName>VirtualMachine</typeName>
            </type>
            <name>Perimeter-Gateway-01-0</name>
            <scope>
                <id>domain-c41</id>
                <objectTypeName>ClusterComputeResource</objectTypeName>
                <name>Management &amp;amp; Edge Cluster</name>
            </scope>
            <clientHandle></clientHandle>
            <extendedAttributes></extendedAttributes>
            <isUniversal>false</isUniversal>
            <universalRevision>0</universalRevision>
        </target>
        <alarmBeingResolved>false</alarmBeingResolved>
        <alarmMetadata>
            <data>
                <key>edgeVmVcUUId</key>
                <value>502e05c2-380f-998c-35ec-1f48991fe7e0</value>
            </data>
            <data>
                <key>edgeId</key>
                <value>edge-3</value>
            </data>
            <data>
                <key>edgeVmName</key>
                <value>Perimeter-Gateway-01-0</value>
            </data>
            <data>
                <key>edgeVmId</key>
                <value>vm-430</value>
            </data>
        </alarmMetadata>
    </systemAlarm>
</systemAlarms>
```

Working with a Specific Alarm

Some system alerts will show up as alarms in the NSX dashboard. You can view and resolve alarms by alarm ID.

`GET /api/2.0/services/systemalarms/{alarmId}`

URI Parameters:

`alarmId`	The alarm ID you want to manage. Find the alarm ID using the `GET /api/2.0/services/alarms/{source-Id}` method.

Description:
Retrieve information about the specified alarm.

Method history:

Release	Modification
6.3.0	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<systemAlarm>
 <eventId>262</eventId>
 <timestamp>1479121141922</timestamp>
 <severity>High</severity>
 <eventSource>Policy</eventSource>
 <eventCode>300006</eventCode>
 <message>Service Composer is out of sync due to failure on sync on reboot operation</message>
 <module>Policy</module>
 <objectId>servicecomposer</objectId>
 <reporterName>NSX Manager</reporterName>
 <reporterType>1</reporterType>
 <sourceType>1</sourceType>
 <displayName>Service Composer</displayName>
 <isResourceUniversal>false</isResourceUniversal>
 <eventMetadata></eventMetadata>
 <resolutionAttempted>true</resolutionAttempted>
 <resolvable>true</resolvable>
 <alarmId>262</alarmId>
 <alarmCode>300006</alarmCode>
 <alarmSource>Policy</alarmSource>
 <alarmBeingResolved>false</alarmBeingResolved>
 <alarmMetadata></alarmMetadata>
</systemAlarm>
```

`POST /api/2.0/services/systemalarms/{alarmId}`

URI Parameters:

`alarmId`	The alarm ID you want to manage. Find the alarm ID using the `GET /api/2.0/services/alarms/{source-Id}` method.

Query Parameters:

`action`	Use *action=resolve* to resolve the specified alarm.

Description:

Resolve the specified alarm.

Alarms will resolve automatically when the cause of the alarm is resolved. For example, if an NSX Edge appliance is powered off, this will trigger an alarm. If you power the NSX Edge appliance back on, the alarm will resolve. If however, you delete the NSX Edge appliance, the alarm will persist, because the alarm cause was never resolved. In this case, you may want to manually resolve the alarm. Resolving the alarm will clear it from the NSX dashboard.

Method history:

Release	Modification
6.3.0	Method introduced.

Working with the Task Framework

Working with filtering criteria and paging information for jobs on the task framework.

`GET /api/2.0/services/taskservice/job`

Query Parameters:

`startIndex` (optional)	The starting point for returning results.
`pageSize` (optional)	The number of results to return.
`sortBy` (optional)	Always sorted by "startTime"
`sortOrderAscending` (optional)	Sort in ascending order of start time (true/false)

Description:
Query job instances by criterion.

Working With a Specific Job Instance

`GET /api/2.0/services/taskservice/job/{jobId}`

URI Parameters:

`jobId` (required)	Specified job ID.

Description:
Retrieve all job instances for the specified job ID.

Working with Guest Introspection and Third-party Endpoint Protection (Anti-virus) Solutions

About Guest Introspection and Endpoint Protection Solutions

VMware's Guest Introspection Service enables vendors to deliver an introspection-based, endpoint protection (anti-virus) solution that uses the hypervisor to scan guest virtual machines from the outside, with only a thin agent on each guest virtual machine.

Version Compatibility

Note: The management APIs listed in this section are to be used only with partner endpoint protection solutions that were developed with EPSec Partner Program 3.0 or earlier (for vShield 5.5 or earlier). These partner solutions are also supported on NSX 6.0 and need the APIs listed below. These APIs should not be used with partner solutions developed specifically for NSX 6.0 or later, as these newer solutions automate the registration and deployment process by using the new features introduced in NSX. Using these with newer NSX 6.0 based solutions could result in loss of features.

Register a Solution

To register a third-party solution with Guest Introspection, clients can use four REST calls to do the following:

1. Register the vendor.
2. Register one or more solutions.
3. Set the solution IP address and port (for all hosts).
4. Activate registered solutions per host.

Note: Steps 1 through 3 need to be performed once per solution. Step 4 needs to be performed for each host.

Unregister a Solution

To unregister a solution, clients perform these steps in reverse:

1. Deactivate solutions per host.
2. Unset a solution’s IP address and port.
3. Unregister solutions.
4. Unregister the vendor.

Updating Registration Information

To update registration information for a vendor or solution, clients must:

1. Unregister the vendor or solution.
2. Reregister the vendor or solution.

Register a Vendor and Solution with Guest Introspection

```
POST /api/2.0/endpointsecurity/registration
```

Description:
Register the vendor of an endpoint protection solution. Specify the following parameters in the request.

Name	Comments
vendorId	VMware-assigned ID for the vendor.
vendorTitle	Vendor-specified title.
vendorDescription	Vendor-specified description.

Request:
Body: application/xml

```
<VendorInfo>
 <id>vendorId</id>
 <title>vendorTitle</title>
 <description>vendorDescription</description>
</VendorInfo>
```

Working With Registered Guest Introspection Vendors

`GET /api/2.0/endpointsecurity/registration/vendors`

Description:
Retrieve the list of all registered Guest Introspection vendors.

Working With Guest Introspection Vendors and Endpoint Protection Solutions

`GET /api/2.0/endpointsecurity/registration/{vendorID}`

URI Parameters:

`vendorID` (required)	VMware-assigned ID for the vendor.

Description:
Retrieve registration information for a Guest Introspection vendor.

`POST /api/2.0/endpointsecurity/registration/{vendorID}`

URI Parameters:

`vendorID` (required)	VMware-assigned ID for the vendor.

Description:
Register an endpoint protection solution. Specify the following parameters in the request.

Name	Comments
solutionAltitude	VMware-assigned altitude for the solution. *Altitude* is a number that VMware assigns to uniquely identify the solution. The altitude describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
solutionTitle	Vendor-specified title for the solution.
solutionDescription	Vendor-specified description of the solution.

Request:
Body: application/xml

```
<SolutionInfo>
 <altitude>solutionAltitude</altitude>
 <title>solutionTitle</title>
 <description>solutionDescription</description>
</SolutionInfo>
```

`DELETE /api/2.0/endpointsecurity/registration/{vendorID}`

URI Parameters:

vendorID (required)	VMware-assigned ID for the vendor.

Description:
Unregister a Guest Introspection vendor.

Information About Registered Endpoint Protection Solutions

`GET /api/2.0/endpointsecurity/registration/{vendorID}/solutions`

URI Parameters:

vendorID (required)	VMware-assigned ID for the vendor.

Description:
Get registration information for all endpoint protection solutions for a Guest Introspection vendor.

Endpoint Protection Solution Registration Information

`GET /api/2.0/endpointsecurity/registration/{vendorID}/{altitude}`

URI Parameters:

altitude	VMware-assigned number that uniquely identifies a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
vendorID (required)	VMware-assigned ID for the vendor.

Description:
Get registration information for an endpoint protection solution.

```
DELETE /api/2.0/endpointsecurity/registration/{vendorID}/{altitude}
```

URI Parameters:

altitude	VMware-assigned number that uniquely identifies a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
vendorID (required)	VMware-assigned ID for the vendor.

Description:
Unregister an endpoint protection solution.

IP Address and Port For an Endpoint Protection Solution

To change the location of an endpoint protection solution:

1 Deactivate all security virtual machines.

2 Change the location.

3 Reactivate all security virtual machines.

```
GET /api/2.0/endpointsecurity/registration/{vendorID}/{altitude}/location
```

URI Parameters:

altitude	VMware-assigned number that uniquely identifies a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
vendorID (required)	VMware-assigned ID for the vendor.

Description:
Get the IP address and port on the vNIC host for an endpoint protection solution.

```
POST /api/2.0/endpointsecurity/registration/{vendorID}/{altitude}/location
```

URI Parameters:

altitude	VMware-assigned number that uniquely identifies a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
vendorID (required)	VMware-assigned ID for the vendor.

Description:
Set the IP address and port on the vNIC host for an endpoint protection solution.

Request:
Body: application/xml

```
<LocationInfo>
 <ip>solutionIpAddress</ip>
 <port>solutionIPPort</port>
</LocationInfo>
```

`DELETE /api/2.0/endpointsecurity/registration/{vendorID}/{altitude}/location`

URI Parameters:

altitude	VMware-assigned number that uniquely identifies a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.
vendorID (required)	VMware-assigned ID for the vendor.

Description:
Unset the IP address and port for an endpoint protection solution.

Activate an Endpoint Protection Solution

You can activate a solution that has been registered and located.

`GET /api/2.0/endpointsecurity/activation`

Query Parameters:

hostId (required)	Host ID associated with activated security VMs.

Description:
Retrieve activation information for all activated security VMs on the specified host.

Responses:
Status Code: 200
Body: application/xml

```
<ActivatedSVMs>
 <ActivationInfo>
   <moid>vm-819</moid>
   <hostMoid>host-9</hostMoid>
   <vmName>VMWARE-Solution-Name-XXX.XXX.XXX.XXX</vmName>
   <hostName>10.24.130.174</hostName>
```

```
   <clusterName>Dev</clusterName>
   <dcName>dev</dcName>
   <vendorId>VMWARE</vendorId>
   <solutionId>6341068275337723904</solutionId>
 </ActivationInfo>
 ***
</ActivatedSVMs>
```

Activated Security Virtual Machines

`GET /api/2.0/endpointsecurity/activation/{vendorID}/{solutionID}`

URI Parameters:

vendorID (required)	VMware-assigned ID for the vendor.
solutionID (required)	solution ID for the endpoint protection solution.

Description:
Retrieve a list of activated security VMs for an endpoint protection solution.

Responses:
Status Code: 200
Body: application/xml

```
<ActivatedSVMs>
 <ActivationInfo>
   <moid>vm-819</moid>
   <hostMoid>host-9</hostMoid>
   <vmName>VMWARE-Solution-Name-XXX.XXX.XXX.XXX</vmName>
   <hostName>10.24.130.174</hostName>
   <clusterName>Dev</clusterName>
   <dcName>dev</dcName>
   <vendorId>VMWARE</vendorId>
   <solutionId>6341068275337723904</solutionId>
 </ActivationInfo>
 ***
</ActivatedSVMs>
```

Activate a Registered Endpoint Protection Solution

`POST /api/2.0/endpointsecurity/activation/{vendorID}/{altitude}`

URI Parameters:

vendorID (required)	VMware-assigned ID for the vendor.

altitude	VMware-assigned number to uniquely identify a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.

Description:
Activate an endpoint protection solution that has been registered and located. Specify the following parameter in the request body.

Name	Comments
svmMoid	Managed object ID of the virtual machine of the activated endpoint protection solution.

Request:
Body: application/xml

```
<ActivationInfo>
 <moid>svmMoid</moid>
</ActivationInfo>
```

Working with Solution Activation Status

`GET /api/2.0/endpointsecurity/activation/{vendorID}/{altitude}/{moid}`

URI Parameters:

moid (required)	Managed object reference of a VM.
vendorID (required)	VMware-assigned ID for the vendor.
altitude	VMware-assigned number to uniquely identify a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.

Description:
Retrieve the endpoint protection solution activation status, either true (activated) or false (not activated).

`DELETE /api/2.0/endpointsecurity/activation/{vendorID}/{altitude}/{moid}`

URI Parameters:

moid (required)	Managed object reference of a VM.
vendorID (required)	VMware-assigned ID for the vendor.
altitude	VMware-assigned number to uniquely identify a solution. Describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.

Description:

Deactivate an endpoint protection solution on a host.

Working with Distributed Firewall

Default Firewall Configuration

`GET /api/4.0/firewall/globalroot-0/defaultconfig`

Description:

Retrieve the default firewall configuration.

The output of this method can be used to restore the firewall config back to default. For example, to replace the layer 2 or layer 3 default section, use the relevant default section from the `GET /api/4.0/firewall/globalroot-0/defaultconfig` response body to create the request body of `PUT /api/4.0/firewall/globalroot-0/config/layer2sections|layer3sections/{sectionId}`.

Method history:

Release	Modification
6.3.0	Method introduced.

Distributed Firewall Rules Configuration

The following table lists the elements that can be used in firewall rules.

Element	Keyword for API	Used in
All Edges	ALL_EDGES	appliedTo
application	Application	service
application group	ApplicationGroup	service
cluster	compute resource	ClusterComputeResource appliedTo
datacenter	Datacenter	source/destination appliedTo
distributed firewall	DISTRIBUTED_FIREWALL	appliedTo
distributed virtual port group	DistributedVirtualPortgroup	source/destination appliedTo
Edge ID	Edge	appliedTo
global root	GlobalRoot	source/destination
host	HostSystem	appliedTo
IP set	IPSet	source/destination
IPv4 addresses	Ipv4Address	source/destination
IPv6 addresses	Ipv6Address	source/destination
logical switch	VirtualWire	source/destination appliedTo
MAC address set	MACSet	source/destination

network	Network	for legacy portgroups, network can be used in source or destination instead of appliedTo
profile	ALL_PROFILE_BINDINGS	
resource pool	ResourcePool	source/destination
security group	SecurityGroup	source/destination
virtual app	VirtualApp	source/destination
virtual machine	VirtualMachine	source/destination appliedTo
vNIC	Vnic	source/destination appliedTo

`GET /api/4.0/firewall/globalroot-0/config`

Query Parameters:

`ruleType (optional)`	**ruleType** can be *LAYER3*, *LAYER2*, *L3REDIRECT*. **ruleType** is mandatory if other query parameters are sent. **Note:** Filtering is not supported for layer 2 rules, so specifying *LAYER2* will return all rule types.
`source (optional)`	**source** can contain IPv4/v6 address or vm-id.
`destination (optional)`	**destination** can contain IPv4/v6 address or vm-id.
`ruleId (optional)`	filter by **ruleId**
`comment (optional)`	**comment** can contain any portion of the comment entered for the rules. Search is case insensitive.
`name (optional)`	**name** can contain any portion of the rule name entered for the rules. Search is case insensitive.
`siProfile (optional)`	**siProfile** can contain any portion of the service profile name associated with L3 redirect rule. Search is case insensitive.
`edgeId (optional)`	Filter for rules applicable to the Edge specified by **edgeId**.
`action (optional)`	Filter for specific **action** (*allow*, *deny*)

Description:

Retrieve distributed firewall rule configuration.

If no query parameters are used, all rule configuration is retrieved. Use the query parameters to filter the rule configuration information.

Responses:

Status Code: 200

Body: application/xml

```
<firewallConfiguration timestamp="1360144793284">
 <contextId>globalroot-0</contextId>
 <layer3Sections>
   <section generationNumber="1360144793284" id="2" name="defaultSectionLayer3"
timestamp="1360144793284">
     <rule disabled="false" id="2" logged="false">
       <name>Default Rule</name>
       <action>DENY</action>
```

```
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>2</sectionId>
    </rule>
  </section>
 </layer3Sections>
 <layer2Sections>
  <section generationNumber="1360144793284" id="1" name="defaultSectionLayer2"
timestamp="1360144793284">
    <rule disabled="false" id="1" logged="false">
      <name>Default Rule</name>
      <action>ALLOW</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1</sectionId>
    </rule>
  </section>
 </layer2Sections>
</firewallConfiguration>
```

PUT /api/4.0/firewall/globalroot-0/config

Description:

Update the complete firewall configuration in all sections.

- Retrieve the configuration with GET /api/4.0/firewall/globalroot-0/config.
- Retrieve the Etag value from the response headers.
- Extract and modify the configuration from the response body as needed.
- Set the If-Match header to the Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

When updating the firewall configuration:

- IDs for new objects (rule/section) should be removed or set to zero.
- If new entities (sections/rules) have been sent in the request, the response will contain the system-generated IDs, which are assigned to these new entities.
- **appliedTo** can be any valid firewall rule element.
- **action** can be *ALLOW*, *BLOCK*, or *REJECT*. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

Request:

Body: application/xml

```
<firewallConfiguration timestamp="1359979620727">
 <contextId>globalroot-0</contextId>
 <layer3Sections>
   <section generationNumber="1359979620727" id="2" name="defaultSectionLayer3"
timestamp="1359979620727">
     <rule disabled="false" logged="true">
       <name>okn-1</name>
       <action>ALLOW</action>
       <sources excluded="false">
         <source>
           <value>datacenter-57</value>
           <type>Datacenter</type>
         </source>
         <source>
           <value>domain-c62</value>
           <type>ClusterComputeResource</type>
         </source>
         <source>
           <value>10.112.1.1</value>
           <type>Ipv4Address</type>
         </source>
       </sources>
       <services>
         <service>
           <destinationPort>80</destinationPort>
           <protocol>6</protocol>
           <subProtocol>6</subProtocol>
         </service>
         <service>
           <value>application-161</value>
           <type>Application</type>
         </service>
       </services>
       <appliedToList>
         <appliedTo>
           <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
           <type>Vnic</type>
         </appliedTo>
         <appliedTo>
           <value>vm-126</value>
           <type>VirtualMachine</type>
         </appliedTo>
       </appliedToList>
     </rule>
     <rule disabled="true" logged="true">
       <name>Matru-1</name>
       <action>ALLOW</action>
       <sectionId>2</sectionId>
     </rule>
     <rule disabled="true" logged="true">
       <name>Matru-2</name>
       <action>ALLOW</action>
       <sectionId>2</sectionId>
     </rule>
     <rule disabled="true" logged="true">
       <name>Matru-3</name>
       <action>ALLOW</action>
       <sectionId>2</sectionId>
```

```
    </rule>
    <rule disabled="true" id="2" logged="false">
      <name>Default Rule</name>
      <action>DENY</action>
      <sectionId>2</sectionId>
    </rule>
  </section>
 </layer3Sections>
 <layer2Sections>
  <section generationNumber="1359979620727" id="1" name="defaultSectionLayer2"
timestamp="1359979620727">
    <rule disabled="false" id="1" logged="false">
      <name>Default Rule</name>
      <action>ALLOW</action>
      <sectionId>1</sectionId>
    </rule>
  </section>
 </layer2Sections>
</firewallConfiguration>
```

`DELETE /api/4.0/firewall/globalroot-0/config`

Description:
Restores default configuration, which means one defaultLayer3 section with three default allow rules and one defaultLayer2Section with one default allow rule.

Working With Layer 3 Sections in Distributed Firewall

You can use sections in the firewall table to group logical rules based on AppliedTo or for a tenant use case. A firewall section is the smallest unit of configuration which can be updated independently. Section types are as follows:

- Layer3Section contains layer3 rules
- Layer2Section contains layer2 rules
- Layer3RedirectSection contains traffic redirect rules.

When Distributed Firewall is used with Service Composer, firewall sections created by Service Composer contain an additional attribute in the XML called managedBy. You should not modify Service Composer firewall sections using Distributed Firewall REST APIs.

`GET /api/4.0/firewall/globalroot-0/config/layer3sections`

Query Parameters:

name (required)	Name of the section to retrieve.

Description:
Retrieve rules from the layer 3 section specified by section **name**.

Responses:
Status Code: 200
Body: application/xml

```
<section generationNumber="1360149234572" id="4" name="TestSection" timestamp="1360149234572">
 <rule disabled="false" id="16" logged="true">
   <name>okn-2</name>
```

```
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>vm1 - Network adapter 1</name>
      <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
      <type>Vnic</type>
      <isValid>true</isValid>
    </appliedTo>
    <appliedTo>
      <name>Small XP-2</name>
      <value>vm-126</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
  <sources excluded="false">
    <source>
      <name>5.1 ESX</name>
      <value>datacenter-57</value>
      <type>Datacenter</type>
      <isValid>true</isValid>
    </source>
    <source>
      <name>5.1</name>
      <value>domain-c62</value>
      <type>ClusterComputeResource</type>
      <isValid>true</isValid>
    </source>
    <source>
      <value>10.112.1.1</value>
      <type>Ipv4Address</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <services>
    <service>
      <destinationPort>80</destinationPort>
      <protocol>6</protocol>
      <subProtocol>6</subProtocol>
    </service>
    <service>
      <name>VMware-VDM2.x-Ephemeral</name>
      <value>application-161</value>
      <isValid>true</isValid>
    </service>
  </services>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
</rule>
<rule disabled="true" id="15" logged="true">
  <name>Matru-3</name>
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
```

```
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
 </rule>
 <rule disabled="true" id="14" logged="true">
  <name>test-3</name>
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
 </rule>
 <rule disabled="true" id="13" logged="true">
  <name>test-2</name>
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
 </rule>
 <rule disabled="true" id="12" logged="false">
  <name>test-1</name>
  <action>DENY</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
 </rule>
</section>
```

`POST /api/4.0/firewall/globalroot-0/config/layer3sections`

Query Parameters:

`operation` (optional)	**operation** can be *insert_after*, *insert_before*, *insert_top*, or *insert_before_default*.
`anchorId` (optional)	Specify the section ID to use for reference with *insert_before* or *insert_after* operations.

Description:

Create a layer 3 distributed firewall section.

By default, the section is created at the top of the firewall table. You can specify a location for the section with the **operation** and **anchorId** query parameters.

Request:
Body: application/xml

```
<section name="TestSection">
 <rule disabled="false" logged="true">
   <name>okn-2</name>
   <action>ALLOW</action>
   <appliedToList>
     <appliedTo>
       <name>vm1 - Network adapter 1</name>
       <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
       <type>Vnic</type>
       <isValid>true</isValid>
     </appliedTo>
     <appliedTo>
       <name>Small XP-2</name>
       <value>vm-126</value>
       <type>VirtualMachine</type>
       <isValid>true</isValid>
     </appliedTo>
   </appliedToList>
   <sources excluded="false">
     <source>
       <name>5.1 ESX</name>
       <value>datacenter-57</value>
       <type>Datacenter</type>
       <isValid>true</isValid>
     </source>
     <source>
       <name>5.1</name>
       <value>domain-c62</value>
       <type>ClusterComputeResource</type>
       <isValid>true</isValid>
     </source>
     <source>
       <value>10.112.1.1</value>
       <type>Ipv4Address</type>
       <isValid>true</isValid>
     </source>
   </sources>
   <services>
     <service>
       <destinationPort>80</destinationPort>
       <protocol>6</protocol>
       <subProtocol>6</subProtocol>
     </service>
     <service>
       <name>VMware-VDM2.x-Ephemeral</name>
       <value>application-161</value>
       <isValid>true</isValid>
     </service>
   </services>
 </rule>
 <rule disabled="true" logged="true">
   <name>Matru-3</name>
```

```
  <action>ALLOW</action>
 </rule>
 <rule disabled="true" logged="true">
  <name>test-3</name>
  <action>ALLOW</action>
 </rule>
 <rule disabled="true" logged="true">
  <name>test-2</name>
  <action>ALLOW</action>
 </rule>
 <rule disabled="true" logged="false">
  <name>test-1</name>
  <action>DENY</action>
 </rule>
</section>
```

Working With a Specific Layer 3 Distributed Firewall Section

`GET /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}`

URI Parameters:

sectionId (required)	The ID of the section to modify.

Description:
Retrieve information about the specified layer 3 section.

Responses:
Status Code: 200
Body: application/xml

```
<section generationNumber="1360149234572" id="4" name="TestSection" timestamp="1360149234572">
 <rule disabled="false" id="16" logged="true">
  <name>okn-2</name>
  <action>ALLOW</action>
  <appliedToList>
   <appliedTo>
    <name>vm1 - Network adapter 1</name>
    <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
    <type>Vnic</type>
    <isValid>true</isValid>
   </appliedTo>
   <appliedTo>
    <name>Small XP-2</name>
    <value>vm-126</value>
    <type>VirtualMachine</type>
    <isValid>true</isValid>
   </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
  <sources excluded="false">
   <source>
    <name>5.1 ESX</name>
    <value>datacenter-57</value>
```

```
      <type>Datacenter</type>
      <isValid>true</isValid>
    </source>
    <source>
      <name>5.1</name>
      <value>domain-c62</value>
      <type>ClusterComputeResource</type>
      <isValid>true</isValid>
    </source>
    <source>
      <value>10.112.1.1</value>
      <type>Ipv4Address</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <services>
    <service>
      <destinationPort>80</destinationPort>
      <protocol>6</protocol>
      <subProtocol>6</subProtocol>
    </service>
    <service>
      <name>VMware-VDM2.x-Ephemeral</name>
      <value>application-161</value>
      <isValid>true</isValid>
    </service>
  </services>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
</rule>
<rule disabled="true" id="15" logged="true">
  <name>Matru-3</name>
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
</rule>
<rule disabled="true" id="14" logged="true">
  <name>test-3</name>
  <action>ALLOW</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>4</sectionId>
</rule>
```

```
  <rule disabled="true" id="13" logged="true">
    <name>test-2</name>
    <action>ALLOW</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
  </rule>
  <rule disabled="true" id="12" logged="false">
    <name>test-1</name>
    <action>DENY</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
  </rule>
</section>
```

`PUT /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:

Update the specified layer 3 section in distributed firewall.

- Retrieve the configuration for the specified section.
- Retrieve the Etag value from the response headers.
- Extract and modify the configuration from the response body as needed.
- Set the If-Match header to the Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

When updating the firewall configuration:

- IDs for new objects (rule/section) should be removed or set to zero.
- If new entities (sections/rules) have been sent in the request, the response will contain the system-generated IDs, which are assigned to these new entities.
- **appliedTo** can be any valid firewall rule element.
- **action** can be *ALLOW*, *BLOCK*, or *REJECT*. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

When Distributed Firewall is used with Service Composer, firewall sections created by Service Composer contain an additional attribute in the XML called managedBy. You should not modify Service Composer firewall sections using Distributed Firewall REST APIs. If you do, you must synchronize firewall rules from Service Composer using the `GET /api/2.0/services/policy/serviceprovider/firewall` API.

Request:
Body: application/xml

```
<section generationNumber="1336034461743" id="4" name="TestSectionRenamed" timestamp="1360149234572">
 <rule disabled="false" id="16" logged="false">
   <name>okn-2</name>
   <action>ALLOW</action>
   <appliedToList>
     <appliedTo>
       <name>vm1 - Network adapter 1</name>
       <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
       <type>Vnic</type>
       <isValid>true</isValid>
     </appliedTo>
     <appliedTo>
       <name>Small XP-2</name>
       <value>vm-126</value>
       <type>VirtualMachine</type>
       <isValid>true</isValid>
     </appliedTo>
   </appliedToList>
   <sectionId>4</sectionId>
   <sources excluded="false">
     <source>
       <name>5.1 ESX</name>
       <value>datacenter-57</value>
       <type>Datacenter</type>
       <isValid>true</isValid>
     </source>
     <source>
       <name>5.1</name>
       <value>domain-c62</value>
       <type>ClusterComputeResource</type>
       <isValid>true</isValid>
     </source>
     <source>
       <value>10.112.1.1</value>
       <type>Ipv4Address</type>
       <isValid>true</isValid>
     </source>
   </sources>
   <services>
     <service>
       <destinationPort>80</destinationPort>
       <protocol>6</protocol>
       <subProtocol>6</subProtocol>
     </service>
     <service>
       <name>VMware-VDM2.x-Ephemeral</name>
       <value>application-161</value>
       <isValid>true</isValid>
     </service>
   </services>
 </rule>
 <rule disabled="true" id="15" logged="true">
   <name>Matru-3</name>
   <action>DENY</action>
   <sectionId>4</sectionId>
```

```
 </rule>
 <rule disabled="true" id="14" logged="true">
   <name>test-3</name>
   <action>ALLOW</action>
   <sectionId>4</sectionId>
 </rule>
 <rule disabled="true" id="13" logged="true">
   <name>test-2</name>
   <action>ALLOW</action>
   <sectionId>4</sectionId>
 </rule>
 <rule disabled="true" id="12" logged="false">
   <name>test-1</name>
   <action>DENY</action>
   <sectionId>4</sectionId>
 </rule>
</section>
```

`POST /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Query Parameters:

`action` (required)	Set **action** to *revise* to change the position of the firewall rule section.
`operation` (optional)	**operation** can be *insert_after*, *insert_before*, *insert_top*, or *insert_before_default*.
`anchorId` (optional)	Specify the section ID to use for reference with *insert_before* or *insert_after* operations.

Description:
Move the specified layer 3 section.

Use the **action**, **operation**, and optionally **achorId** query parameters to specify the destination for the section.

```
POST /api/4.0/firewall/globalroot-0/config/layer3sections/1007
?action=revise&operation=insert_before&anchorId=1006
```

```
If-Match: 1477989118875
```

```
<section id="1007" name="Web Section" generationNumber="1477989118875" timestamp="1477989118875"
type="LAYER3">
 ***
</section>
```

Request:
Body: application/xml

```
<section>
 <name></name>
 <action></action>
 <appliedToList>
   <appliedTo>
```

```
    <name></name>
    <value></value>
    <type></type>
    <isValid></isValid>
   </appliedTo>
  </appliedToList>
  <sectionId></sectionId>
</section>
```

`DELETE /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:

Delete the specified layer 3 distributed firewall section.

If the default layer 3 firewall section is selected, the request is rejected. See `GET /api/4.0/firewall/globalroot-0/defaultconfig` for information on resetting the default firewall section.

Method history:

Release	Modification
6.3.0	Method updated. When deleting the default firewall rule section, the method previously removed all rules except for the default rule. The method now returns status 400 and the message Cannot delete default section <sectionId>.

Working With Distributed Firewall Rules in a Layer 3 Section

`POST /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}/rules`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:

Add rules to the specified layer 2 section in distributed firewall.

You add firewall rules at the global scope. You can then narrow down the scope (datacenter, cluster, distributed virtual port group, network, virtual machine, vNIC, or logical switch) at which you want to apply the rule. Firewall allows you to add multiple objects at the source and destination levels for each rule, which helps reduce the total number of firewall rules to be added. To add a identity based firewall rule, first create a security group based on Directory Group objects. Then create a firewall rule with the security group as the source or destination. Rules that direct traffic to a third part service are referred to as layer3 redirect rules, and are displayed in the layer3 redirect tab.

When Distributed Firewall is used with Service Composer, firewall rules created by Service Composer contain an additional attribute in the XML called managedBy.

Follow this procedure to add a rule:

- Retrieve the configuration for the specified section.
- Retrieve the Etag value from the response headers. **Note**: Each section contains its own Etag, generationNumber, and timestamp. When adding a new rule, you must use the Etag value of the firewall section to which you wish to

- add the rule.
- Extract and modify the configuration from the response body as needed.
- Set the If-Match header to the section Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

When updating the firewall configuration:
- IDs for new rules should be removed or set to zero.
- If new rules have been sent in the request, the response will contain the system-generated IDs, which are assigned to these new entities.
- **appliedTo** can be any valid firewall rule element.
- **action** can be *ALLOW*, *BLOCK*, or *REJECT*. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

Request:
Body: application/xml

```
<rule disabled="false" logged="false">
 <name>AddRuleTest</name>
 <action>allow</action>
 <notes></notes>
 <appliedToList>
   <appliedTo>
     <value>datacenter-26</value>
     <type>Datacenter</type>
   </appliedTo>
 </appliedToList>
 <sectionId>2</sectionId>
 <sources excluded="true">
   <source>
     <value>datacenter-26</value>
     <type>Datacenter</type>
   </source>
 </sources>
 <services>
   <service>
     <value>application-216</value>
   </service>
 </services>
</rule>
```

Working with a Specific Rule in a Specific Layer 3 Section

`GET /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}/rules/{ruleId}`

URI Parameters:

`ruleId` (required)	The ID of the rule beeing read, updated or deleted
`sectionId` (required)	The ID of the section to modify.

Description:
Retrieve information about the specified distributed firewall rule.

`PUT /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}/rules/{ruleId}`

URI Parameters:

`ruleId` (required)	The ID of the rule beeing read, updated or deleted
`sectionId` (required)	The ID of the section to modify.

Description:
Update a distributed firewall rule in a layer 3 section.

- Retrieve the configuration for the section that contains the rule you want to modify.
- Retrieve the Etag value from the response headers. **Note**: This is the Etag value of the firewall section to which you want to add the rule. If you are keeping this rule in the same section, you must keep the same Etag number.
- Extract and modify the rule configuration from the response body as needed.
- Set the If-Match header to the section Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

Request:
Body: application/xml

```
<rule disabled="enabled" id="23" logged="true">
 <name>AddRuleTestUpdated</name>
 <action>allow</action>
 <notes></notes>
 <appliedToList>
   <appliedTo>
     <value>datacenter-26</value>
     <type>Datacenter</type>
   </appliedTo>
 </appliedToList>
 <sectionId>2</sectionId>
 <sources excluded="true">
   <source>
     <value>datacenter-26</value>
     <type>Datacenter</type>
   </source>
 </sources>
 <services>
   <service>
     <value>application-216</value>
   </service>
 </services>
</rule>
```

`DELETE /api/4.0/firewall/globalroot-0/config/layer3sections/{sectionId}/rules/{ruleId}`

URI Parameters:

`ruleId` (required)	The ID of the rule beeing read, updated or deleted

sectionId (required)	The ID of the section to modify.

Description:
Delete the specified distributed firewall rule.

Working With Layer 2 Sections in Distributed Firewall

You can use sections in the firewall table to group logical rules based on AppliedTo or for a tenant use case. A firewall section is the smallest unit of configuration which can be updated independently. Section types are as follows:

- Layer3Section contains layer3 rules
- Layer2Section contains layer2 rules
- Layer3RedirectSection contains traffic redirect rules.

When Distributed Firewall is used with Service Composer, firewall sections created by Service Composer contain an additional attribute in the XML called managedBy. You should not modify Service Composer firewall sections using Distributed Firewall REST APIs.

`GET /api/4.0/firewall/globalroot-0/config/layer2sections`

Query Parameters:

name (optional)	Name of the Section to read

Description:
Retrieve rules from the layer 2 section specified by section **name**.

`POST /api/4.0/firewall/globalroot-0/config/layer2sections`

Query Parameters:

operation (optional)	**operation** can be *insert_after*, *insert_before*, *insert_top*, or *insert_before_default*.
anchorId (optional)	Specify the section ID to use for reference with *insert_before* or *insert_after* operations.

Description:
Create a layer 2 distributed firewall section.

By default, the section is created at the top of the firewall table. You can specify a location for the section with the **operation** and **anchorId** query parameters.

Request:
Body: application/xml

```
<section managedBy="" name="" type="">
 <rule disabled="" logged="">
   <name></name>
   <action></action>
   <appliedToList>
     <appliedTo>
       <name></name>
```

```
      <value></value>
      <type></type>
      <isValid></isValid>
    </appliedTo>
  </appliedToList>
  <sources excluded="">
    <source>
      <name></name>
      <value></value>
      <type></type>
      <isValid></isValid>
    </source>
  </sources>
  <destinations excluded="">
    <destination>
      <name></name>
      <value></value>
      <type></type>
      <isValid></isValid>
    </destination>
  </destinations>
  <services>
    <service>
      <destinationPort></destinationPort>
      <protocol></protocol>
      <subProtocol></subProtocol>
    </service>
  </services>
 </rule>
</section>
```

Working With a Specific Layer 2 Distributed Firewall Section

`GET /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:
Retrieve information about the specified layer 2 section.

`PUT /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:
Update the specified layer 2 section in distributed firewall.

- Retrieve the configuration for the specified section.
- Retrieve the Etag value from the response headers.
- Extract and modify the configuration from the response body as needed.
- Set the If-Match header to the Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

When updating the firewall configuration:

- IDs for new objects (rule/section) should be removed or set to zero.
- If new entities (sections/rules) have been sent in the request, the response will contain the system-generated IDs, which are assigned to these new entities.
- **appliedTo** can be any valid firewall rule element.
- **action** can be *ALLOW*, *BLOCK*, or *REJECT*. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

When Distributed Firewall is used with Service Composer, firewall sections created by Service Composer contain an additional attribute in the XML called managedBy. You should not modify Service Composer firewall sections using Distributed Firewall REST APIs. If you do, you must synchronize firewall rules from Service Composer using the `GET /api/2.0/services/policy/serviceprovider/firewall` API.

Request:

Body: application/xml

```
<section generationNumber="" id="" name="" timestamp="">
 <rule disabled="" id="" logged="">
   <name></name>
   <action></action>
   <appliedToList>
     <appliedTo>
       <name></name>
       <value></value>
       <type></type>
       <isValid></isValid>
     </appliedTo>
   </appliedToList>
   <sectionId></sectionId>
   <sources excluded="">
     <source>
       <name></name>
       <value></value>
       <type></type>
       <isValid></isValid>
     </source>
   </sources>
   <services>
     <service>
       <destinationPort></destinationPort>
       <protocol></protocol>
       <subProtocol></subProtocol>
     </service>
   </services>
 </rule>
</section>
```

`POST /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}`

URI Parameters:

sectionId (required)	The ID of the section to modify.

Query Parameters:

action (required)	Set **action** to *revise* to change the position of the firewall rule section.
operation (optional)	**operation** can be *insert_after*, *insert_before*, *insert_top*, or *insert_before_default*.
anchorId (optional)	Specify the section ID to use for reference with *insert_before* or *insert_after* operations.

Description:
Move the specified layer 2 section.

Use the **action**, **operation**, and optionally **achorId** query parameters to specify the destination for the section.

```
POST /api/4.0/firewall/globalroot-0/config/layer2sections/1009
?action=revise&operation=insert_before&anchorId=1008
```

```
If-Match: 1478307787160
```

```
<section id="1009" name="Test Section" generationNumber="1478307787160" timestamp="1478307787160"
type="LAYER2">
 ***
</section>
```

Request:
Body: application/xml

```
<section>
 <name></name>
 <action></action>
 <appliedToList>
   <appliedTo>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </appliedTo>
 </appliedToList>
 <sectionId></sectionId>
</section>
```

DELETE /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}

URI Parameters:

sectionId (required)	The ID of the section to modify.

Description:
Delete the specified layer 2 section and its contents.

If the default layer 2 firewall section is selected, the request is rejected. See `GET /api/4.0/firewall/globalroot-0/defaultconfig` for information on resetting the default firewall section.

Method history:

Release	Modification
6.3.0	Method updated. When deleting the default firewall rule section, the method previously removed all rules except for the default rule. The method now returns status 400 and the message Cannot delete default section <sectionId>.

Working With Distributed Firewall Rules in a Layer 2 Section

`POST /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}/rules`

URI Parameters:

`sectionId` (required)	The ID of the section to modify.

Description:

Add rules to the specified layer 2 section in distributed firewall.

You add firewall rules at the global scope. You can then narrow down the scope (datacenter, cluster, distributed virtual port group, network, virtual machine, vNIC, or logical switch) at which you want to apply the rule. Firewall allows you to add multiple objects at the source and destination levels for each rule, which helps reduce the total number of firewall rules to be added. To add a identity based firewall rule, first create a security group based on Directory Group objects. Then create a firewall rule with the security group as the source or destination. Rules that direct traffic to a third part service are referred to as layer3 redirect rules, and are displayed in the layer3 redirect tab.

When Distributed Firewall is used with Service Composer, firewall rules created by Service Composer contain an additional attribute in the XML called managedBy.

Follow this procedure to add a rule:
- Retrieve the configuration for the specified section.
- Retrieve the Etag value from the response headers. **Note**: Each section contains its own Etag, generationNumber, and timestamp. When adding a new rule, you must use the Etag value of the firewall section to which you wish to add the rule.
- Extract and modify the configuration from the response body as needed.
- Set the If-Match header to the section Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

When updating the firewall configuration:
- IDs for new rules should be removed or set to zero.
- If new rules have been sent in the request, the response will contain the system-generated IDs, which are assigned to these new entities.
- **appliedTo** can be any valid firewall rule element.
- **action** can be *ALLOW*, *BLOCK*, or *REJECT*. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

Request:

Body: application/xml

```
<rule disabled="" logged="">
 <name></name>
 <action></action>
 <notes></notes>
 <appliedToList>
   <appliedTo>
     <value></value>
     <type></type>
   </appliedTo>
 </appliedToList>
 <sources excluded="">
   <source>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </source>
 </sources>
 <destinations excluded="">
   <destination>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </destination>
 </destinations>
 <services>
   <service>
     <value></value>
   </service>
 </services>
</rule>
```

Working With a Specific Rule in a Specific Layer 2 Section

`GET /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}/rules/{ruleId}`

URI Parameters:

ruleId (required)	The ID of the rule.
sectionId (required)	The ID of the section to modify.

Description:
Retrieve the configuration of the specified rule.

`PUT /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}/rules/{ruleId}`

URI Parameters:

ruleId (required)	The ID of the rule.
sectionId (required)	The ID of the section to modify.

Description:

Update a distributed firewall rule in a layer 2 section.

- Retrieve the configuration for the section that contains the rule you want to modify.
- Retrieve the Etag value from the response headers. **Note**: This is the Etag value of the firewall section to which you want to add the rule. If you are keeping this rule in the same section, you must keep the same Etag number.
- Extract and modify the rule configuration from the response body as needed.
- Set the If-Match header to the section Etag value, and submit the request.

Not all fields are required while sending the request. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.

Request:

Body: application/xml

```
<rule disabled="" id="" logged="">
 <name></name>
 <action></action>
 <notes></notes>
 <sources excluded="">
   <source>
     <value></value>
     <type></type>
     <isValid></isValid>
   </source>
 </sources>
 <destinations excluded="">
   <destination>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </destination>
 </destinations>
 <services>
   <service>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </service>
 </services>
 <appliedToList>
   <appliedTo>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </appliedTo>
 </appliedToList>
</rule>
```

`DELETE /api/4.0/firewall/globalroot-0/config/layer2sections/{sectionId}/rules/{ruleId}`

URI Parameters:

`ruleId` (required)	The ID of the rule.
`sectionId` (required)	The ID of the section to modify.

Description:
Delete the specified distributed firewall rule.

Layer 3 Redirect Sections and Rules

`POST /api/4.0/firewall/globalroot-0/config/layer3redirectsections`

Description:
Add L3 redirect section

Request:
Body: application/xml

```
<section>
 <name></name>
 <action></action>
 <appliedToList>
   <appliedTo>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </appliedTo>
 </appliedToList>
 <sectionId></sectionId>
</section>
```

Layer 3 Redirect Section

`GET /api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}`

URI Parameters:

section (required)	Specify section by ID or name

Description:
Get L3 redirect section configuration

`PUT /api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}`

URI Parameters:

section (required)	Specify section by ID or name

Description:

Modify layer 3 redirect section. You will need to get the Etag value out of the GET first. Then pass the modified version of the whole redirect section configuration in the GET body.

Request:
Body: application/xml

```
<section generationNumber="" id="" name="" timestamp="">
 <rule disabled="" id="" logged="">
   <name></name>
   <action></action>
   <appliedToList>
     <appliedTo>
       <name></name>
       <value></value>
       <type></type>
       <isValid></isValid>
     </appliedTo>
   </appliedToList>
   <sectionId></sectionId>
 </rule>
</section>
```

`DELETE /api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}`

URI Parameters:

section (required)	Specify section by ID or name

Description:
Delete specified L3 redirect section

Working with Layer 3 Redirect Rules for a Specific Section

`POST /api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}/rules`

URI Parameters:

section (required)	Specify section by ID or name

Description:
Add L3 redirect rule

Request:
Body: application/xml

```
<section generationNumber="" id="" name="" timestamp="">
 <name></name>
 <action></action>
 <appliedToList>
   <appliedTo>
```

```
    <name></name>
    <value></value>
    <type></type>
    <isValid></isValid>
   </appliedTo>
 </appliedToList>
 <sectionId></sectionId>
</section>
```

Working With a Specific Layer 3 Redirect Rule for a Specific Section

GET
/api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}/rules/{ruleID}

URI Parameters:

ruleID (required)	Specified redirect rule
section (required)	Specify section by ID or name

Description:
Get L3 redirect rule

PUT
/api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}/rules/{ruleID}

URI Parameters:

ruleID (required)	Specified redirect rule
section (required)	Specify section by ID or name

Description:
Modify L3 redirect rule. You will need Etag value from the response header of GET call. Then, pass Etag value as the if-match header in PUT call

Request:
Body: application/xml

```
<rule disabled="" id="" logged="">
 <name></name>
 <action></action>
 <appliedToList>
   <appliedTo>
     <name></name>
     <value></value>
     <type></type>
     <isValid></isValid>
   </appliedTo>
 </appliedToList>
</rule>
```

```
DELETE
/api/4.0/firewall/globalroot-0/config/layer3redirectsections/{section}/rules/{ruleID}
```

URI Parameters:

ruleID (required)	Specified redirect rule
section (required)	Specify section by ID or name

Description:
Delete specified L3 redirect rule

Service Insertion Profiles and Layer 3 Redirect Rules

```
GET /api/4.0/firewall/globalroot-0/config/layer3redirect/profiles
```

Description:
Retrieve the Service Insertion profiles that can be applied to layer3 redirect rules.

Enable Distributed Firewall After Upgrade

After upgrading NSX Manager, controllers, and network virtualization components, check the status of distributed firewall. If it is ready to enable, you can enable distributed firewall.

State	Description
backwardCompatible	This is the default state after an upgrade from vCloud Networking and Security to NSX, which means that vShield App is being used for protection instead of distributed firewall.
backwardCompatibleReadyForSwitch	Once the clusters are prepared with NSX binaries, this state is enabled. You can enable distributed firewall only after firewall is in this state.
switchingToForward	This is an intermediate state when you change firewall to distributed firewall.
forward	This is the default state for green field deployments or after you have switched from vShield App to distributed firewall.
switchFailed	This state is unlikely, but may be present if NSX Manager failed to switch to distributed firewall.

```
GET /api/4.0/firewall/globalroot-0/state
```

Description:
Retrieve current state of firewall functioning after NSX upgrade.

`PUT /api/4.0/firewall/globalroot-0/state`

Description:
Enable distributed firewall.

Working with Distributed Firewall Status

Retrieve status of last publish action for each cluster in the NSX environment.

The status output displays a generation number (**generationNumber**) for each rule set, which can be used to verify whether a change in rule sets has propagated to a host. In 6.2.4, a generation number for objects (**generationNumberObjects**) has been added to the status API. This allows you to verify whether a change in objects consumed in firewall rules has propagated to a host. Note that the object generation number may change frequently and will always be equal to or greater than the ruleset generation number.

Starting in NSX 6.2.4, clusters (and hosts inside the cluster) are no longer included in the firewall status output if distributed firewall is disabled at the cluster level, or if the cluster is not prepared (NSX VIBs are not installed). In earlier versions of NSX these clusters and hosts are included in the output. However, because they are not configured for firewall, after a firewall rule publish their status is *inprogress*.

`GET /api/4.0/firewall/globalroot-0/status`

Description:
Get firewall configuration status

Method history:

Release	Modification
6.2.4	Method updated. Parameter **generationNumberObjects** added. Clusters not configured for firewall are excluded from the status output.

Responses:
Status Code: 200
Body: application/xml

```
<firewallStatus>
 <startTime>1478235234617</startTime>
 <status>published</status>
 <generationNumber>1478235234617</generationNumber>
 <generationNumberObjects>1478235234617</generationNumberObjects>
 <clusterList>
   <clusterStatus>
     <clusterId>domain-c33</clusterId>
     <status>published</status>
     <generationNumber>1478235234617</generationNumber>
     <generationNumberObjects>1478235234617</generationNumberObjects>
     <hostStatusList>
       <hostStatus>
         <hostId>host-32</hostId>
         <hostName>esx-02a.corp.local</hostName>
         <status>published</status>
         <errorCode>0</errorCode>
```

```
        <startTime>1478235235421</startTime>
        <endTime>1478235235429</endTime>
        <generationNumber>1478235234617</generationNumber>
        <clusterId>domain-c33</clusterId>
        <generationNumberObjects>1478235234617</generationNumberObjects>
      </hostStatus>
      <hostStatus>
        <hostId>host-28</hostId>
        <hostName>esx-01a.corp.local</hostName>
        <status>published</status>
        <errorCode>0</errorCode>
        <startTime>1478235235421</startTime>
        <endTime>1478235235431</endTime>
        <generationNumber>1478235234617</generationNumber>
        <clusterId>domain-c33</clusterId>
        <generationNumberObjects>1478235234617</generationNumberObjects>
      </hostStatus>
    </hostStatusList>
  </clusterStatus>
  <clusterStatus>
    <clusterId>domain-c41</clusterId>
    <status>published</status>
    <generationNumber>1478235234617</generationNumber>
    <generationNumberObjects>1478235234617</generationNumberObjects>
    <hostStatusList>
      <hostStatus>
        <hostId>host-202</hostId>
        <hostName>esxmgt-01a.corp.local</hostName>
        <status>published</status>
        <errorCode>0</errorCode>
        <startTime>1478235235436</startTime>
        <endTime>1478235235442</endTime>
        <generationNumber>1478235234617</generationNumber>
        <clusterId>domain-c41</clusterId>
        <generationNumberObjects>1478235234617</generationNumberObjects>
      </hostStatus>
      <hostStatus>
        <hostId>host-203</hostId>
        <hostName>esxmgt-02a.corp.local</hostName>
        <status>published</status>
        <errorCode>0</errorCode>
        <startTime>1478235235436</startTime>
        <endTime>1478235235444</endTime>
        <generationNumber>1478235234617</generationNumber>
        <clusterId>domain-c41</clusterId>
        <generationNumberObjects>1478235234617</generationNumberObjects>
      </hostStatus>
    </hostStatusList>
  </clusterStatus>
 </clusterList>
</firewallStatus>
```

Working with a Specific Layer 3 Section Status

`GET /api/4.0/firewall/globalroot-0/status/layer3sections/{sectionID}`

URI Parameters:

`sectionID` (required)	Section ID

Description:
Retrieve status of the last publish action for the specified layer 3 section.

Method history:

Release	Modification
6.2.4	Method updated. Parameter **generationNumberObjects** added. Clusters not configured for firewall are excluded from the status output.

Working with a Specific Layer 2 Section Status

`GET /api/4.0/firewall/globalroot-0/status/layer2sections/{sectionID}`

URI Parameters:

`sectionID` (required)	Section ID

Description:
Retrieve status of the last publish action for the specified layer 2 section.

Method history:

Release	Modification
6.2.4	Method updated. Parameter **generationNumberObjects** added. Clusters not configured for firewall are excluded from the status output.

Import and Export Firewall Configurations

`GET /api/4.0/firewall/globalroot-0/drafts`

Description:
Displays the draft IDs of all saved configurations.

`POST /api/4.0/firewall/globalroot-0/drafts`

Description:
Save a firewall configuration.

Request:
Body: application/xml

```
<firewallDraft name="">
 <description></description>
 <preserve></preserve>
 <mode></mode>
 <config>
   <contextId></contextId>
   <layer3Sections>
     <section name="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer3Sections>
   <layer2Sections>
     <section name="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer2Sections>
 </config>
</firewallDraft>
```

Working With a Specific Saved Firewall Configuration

`GET /api/4.0/firewall/globalroot-0/drafts/{draftID}`

URI Parameters:

`draftID` (required)	Specified draft ID. Use `GET /4.0/firewall/globalroot-0/drafts` to retrieve all drafts.

Description:
Get a saved firewall configuration.

`PUT /api/4.0/firewall/globalroot-0/drafts/{draftID}`

URI Parameters:

`draftID` (required)	Specified draft ID. Use `GET /4.0/firewall/globalroot-0/drafts` to retrieve all drafts.

Description:

Update a saved firewall configuration.

Request:
Body: application/xml

```
<firewallDraft name="">
 <description></description>
 <preserve></preserve>
 <mode></mode>
 <config>
   <contextId></contextId>
   <layer3Sections>
     <section name="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer3Sections>
   <layer2Sections>
     <section name="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer2Sections>
 </config>
</firewallDraft>
```

`DELETE /api/4.0/firewall/globalroot-0/drafts/{draftID}`

URI Parameters:

draftID (required)	Specified draft ID. Use `GET /4.0/firewall/globalroot-0/drafts` to retrieve all drafts.

Description:
Delete a configuration.

Export a Firewall Configuration

`GET /api/4.0/firewall/globalroot-0/drafts/{draftID}/action/export`

URI Parameters:

draftID (required)	Specified draft ID. Use `GET /4.0/firewall/globalroot-0/drafts` to retrieve all drafts.

Query Parameters:

`getLatestForUniversal` (optional)	Set to *true* to export the latest universal draft from a secondary NSX manager.

Description:
Export a configuration.

Import a Firewall Configuration

`POST /api/4.0/firewall/globalroot-0/drafts/{draftID}/action/import`

URI Parameters:

`draftID` (required)	Specified draft ID. Use `GET /4.0/firewall/globalroot-0/drafts` to retrieve all drafts.

Description:
Import a configuration.

Request:
Body: application/xml

```
<firewallDraft id="" name="" timestamp="">
 <description></description>
 <preserve></preserve>
 <user></user>
 <mode></mode>
 <config timestamp="">
   <contextId></contextId>
   <layer3Sections>
     <section name="" timestamp="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer3Sections>
   <layer2Sections>
     <section name="" timestamp="">
       <rule disabled="true|false" id="" logged="true|false">
         <name></name>
         <action></action>
         <precedence></precedence>
       </rule>
     </section>
   </layer2Sections>
   <generationNumber></generationNumber>
 </config>
</firewallDraft>
```

Working with Distributed Firewall Session Timers

You can configure session timers (session timeouts) for TCP, UDP, and ICMP. There is a default configuration, which applies to all VMs protected by Distributed Firewall. You can modify the session timers values of the default configuration, but not the **appliedTo** values.

You can add additional session timer configurations with different **appliedTo** configurations.

Parameter	Description	Comments
appliedTo > value	The ID of the object on which to apply the timeout settings.	Required. For example VM ID *vm-216*.
appliedTo > type	The type of object on which to apply the timeout settings.	Required. Can be *VirtualMachine* or *Vnic*
generationNumber	Generation number for the configuration.	When updating session timers, you must ensure the latest generation number is included in the request body.
tcpFirstPacket	The timeout value for the connection after the first packet has been sent. This will be the initial timeout for the connection once a SYN has been sent and the flow is created.	Valid timer values: *10-4320000* seconds. Default is *120*.
tcpOpen	The timeout value for the connection after a second packet has been transferred.	Valid timer values: *10-4320000* seconds. Default is *30*.
tcpEstablished	The timeout value for the connection once the connection has become fully established.	Valid timer values: *120-4320000* seconds. Default is *43200*.
tcpClosing	The timeout value for the connection after the first FIN has been sent.	Valid timer values: *10-4320000* seconds. Default is *120*.
tcpFinWait	The timeout value for the connection after both FINs have been exchanged and the connection is closed.	Valid timer values: *10-4320000* seconds. Default is *45*.
tcpClosed	The timeout value for the connection after one endpoint sends an RST.	Valid timer values: *10-4320000* seconds. Default is *20*.
udpFirstPacket	The timeout value for the connection after the first packet. This will be the initial timeout for the new UDP flow.	Valid timer values: *10-4320000* seconds. Default is *60*.
udpSingle	The timeout value for the connection if the source host sends more than one packet but the destination host has never sent one back.	Valid timer values: *10-4320000* seconds. Default is *30*.
udpMultiple	The timeout value for the connection if both hosts have sent packets.	Valid timer values: *10-4320000* seconds. Default is *60*.
icmpFirstPacket	The timeout value for the connection after the first packet. This will be the initial timeout for the new ICMP flow.	Valid timer values: *10-4320000* seconds. Default is *20*.

icmpErrorReply	The timeout value for the connection after an ICMP error came back in response to an ICMP packet.	Valid timer values: *10-4320000* seconds. Default is *10*.

`GET /api/4.0/firewall/globalroot-0/timeouts`

Description:
Retrieve Distributed Firewall session timer configuration.

Method history:

Release	Modification
6.3.0	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<firewallTimeoutConfigurations>
 <firewallTimeoutConfiguration id="1001">
   <name>Default Session Timers</name>
   <description>Default Session Timers</description>
   <appliedToList>
     <appliedTo>
       <name>DISTRIBUTED_FIREWALL</name>
       <value>DISTRIBUTED_FIREWALL</value>
       <type>DISTRIBUTED_FIREWALL</type>
       <isValid>true</isValid>
     </appliedTo>
   </appliedToList>
   <generationNumber>1489650711521</generationNumber>
   <isDefault>true</isDefault>
   <tcpFirstPacket>120</tcpFirstPacket>
   <tcpOpen>30</tcpOpen>
   <tcpEstablished>43200</tcpEstablished>
   <tcpClosing>120</tcpClosing>
   <tcpFinWait>45</tcpFinWait>
   <tcpClosed>20</tcpClosed>
   <udpFirstPacket>60</udpFirstPacket>
   <udpSingle>30</udpSingle>
   <udpMultiple>60</udpMultiple>
   <icmpFirstPacket>20</icmpFirstPacket>
   <icmpErrorReply>10</icmpErrorReply>
 </firewallTimeoutConfiguration>
</firewallTimeoutConfigurations>
```

`POST /api/4.0/firewall/globalroot-0/timeouts`

Description:
Create a Distributed Firewall session timer configuration.

Method history:

Release	Modification

6.3.0	Method introduced.

Request:
Body: application/xml

```
<firewallTimeoutConfiguration>
 <name>new VM timeout</name>
 <appliedToList>
   <appliedTo>
     <value>vm-217</value>
     <type>VirtualMachine</type>
   </appliedTo>
 </appliedToList>
 <isDefault>false</isDefault>
 <tcpFirstPacket>180</tcpFirstPacket>
 <tcpOpen>30</tcpOpen>
 <tcpEstablished>43200</tcpEstablished>
 <tcpClosing>180</tcpClosing>
 <tcpFinWait>45</tcpFinWait>
 <tcpClosed>40</tcpClosed>
 <udpFirstPacket>60</udpFirstPacket>
 <udpSingle>30</udpSingle>
 <udpMultiple>60</udpMultiple>
 <icmpFirstPacket>30</icmpFirstPacket>
 <icmpErrorReply>15</icmpErrorReply>
</firewallTimeoutConfiguration>
```

Working With a Specific Distributed Firewall Session Timer Configuration

`GET /api/4.0/firewall/globalroot-0/timeouts/{configId}`

URI Parameters:

`configId` (required)	Session timer configuration ID (**firewallTimeoutConfiguration** id). For example, *1004*.

Description:
Retrieve the specified Distributed Firewall session timer configuration.

Method history:

Release	Modification
6.3.0	Method introduced.

`PUT /api/4.0/firewall/globalroot-0/timeouts/{configId}`

URI Parameters:

`configId` (required)	Session timer configuration ID (**firewallTimeoutConfiguration** id). For example, *1004*.

Description:

Update the specified Distributed Firewall session timer configuration.

Method history:

Release	Modification
6.3.0	Method introduced.

Request:
Body: application/xml

```
<firewallTimeoutConfiguration id="1004">
 <name>new VM timeout</name>
 <appliedToList>
   <appliedTo>
     <value>vm-217</value>
     <type>VirtualMachine</type>
     <isValid>true</isValid>
   </appliedTo>
   <appliedTo>
     <value>vm-218</value>
     <type>VirtualMachine</type>
     <isValid>true</isValid>
   </appliedTo>
 </appliedToList>
 <generationNumber>1490768692562</generationNumber>
 <isDefault>false</isDefault>
 <tcpFirstPacket>180</tcpFirstPacket>
 <tcpOpen>30</tcpOpen>
 <tcpEstablished>43200</tcpEstablished>
 <tcpClosing>180</tcpClosing>
 <tcpFinWait>45</tcpFinWait>
 <tcpClosed>40</tcpClosed>
 <udpFirstPacket>60</udpFirstPacket>
 <udpSingle>30</udpSingle>
 <udpMultiple>60</udpMultiple>
 <icmpFirstPacket>30</icmpFirstPacket>
 <icmpErrorReply>15</icmpErrorReply>
</firewallTimeoutConfiguration>
```

`DELETE /api/4.0/firewall/globalroot-0/timeouts/{configId}`

URI Parameters:

`configId` (required)	Session timer configuration ID (**firewallTimeoutConfiguration** id). For example, *1004*.

Description:
Delete the specified Distributed Firewall session timer configuration.

Method history:

Release	Modification
6.3.0	Method introduced.

Working With Distributed Firewall Thresholds

Configure memory, CPU, and connections per second (CPS) thresholds for distributed firewall.

The firewall module generates system events when the memory and CPU usage crosses these thresholds.

`GET /api/4.0/firewall/stats/eventthresholds`

Description:
Retrieve threshold configuration for distributed firewall.

Responses:
Status Code: 200
Body: application/xml

```
<eventThresholds>
 <cpu>
   <percentValue>80</percentValue>
 </cpu>
 <memory>
   <percentValue>90</percentValue>
 </memory>
 <connectionsPerSecond>
   <value>250000</value>
 </connectionsPerSecond>
</eventThresholds>
```

`PUT /api/4.0/firewall/stats/eventthresholds`

Description:
Update threshold configuration for distributed firewall.

Request:
Body: application/xml

```
<eventThresholds>
 <cpu>
   <percentValue>80</percentValue>
 </cpu>
 <memory>
   <percentValue>90</percentValue>
 </memory>
 <connectionsPerSecond>
   <value>250000</value>
 </connectionsPerSecond>
</eventThresholds>
```

Working with the Distributed Firewall Global Configuration

You can use the following parameters to improve firewall performancer:

- **layer3RuleOptimize** and **layer2RuleOptimize** to turn on/off rule optimization.
- **tcpStrictOption** determines whether or not to drop an established TCP connection when the firewall does not see the initial three-way handshake. If set to true, the connection will be dropped.
- **autoDraftDisabled** improves performances when making large numbers of changes to firewall rules.

You can disable the auto draft feature by setting **autoDraftDisabled** to true. Distributed Firewall saves up to 100 configurations, including manually saved drafts (**preserve** parameter can be set to true or false) and auto saved drafts (**preserve** parameter is set to false). Once 100 configurations are saved, older drafts with the **preserve** parameter set to false will be deleted in order to save new configurations. You might want to disable the auto drafts feature before making large numbers of changes to the firewall rules, to improve performance, and to prevent previously saved drafts from being overwritten.

Note: The **autoDraftDisabled** parameter does not appear in a GET of the global configuration.

`GET /api/4.0/firewall/config/globalconfiguration`

Description:
Retrieve performance configuration for distributed firewall.

Responses:
Status Code: 200
Body: application/xml

```
<globalConfiguration>
 <layer3RuleOptimize>false</layer3RuleOptimize>
 <layer2RuleOptimize>true</layer2RuleOptimize>
 <tcpStrictOption>false</tcpStrictOption>
</globalConfiguration>
```

`PUT /api/4.0/firewall/config/globalconfiguration`

Description:
Update the distributed firewall performance configuration.

Method history:

Release	Modification
6.2.3	Method updated. **autoDraftDisabled** parameter added.

Request:
Body: application/xml

```
<globalConfiguration>
 <layer3RuleOptimize>false</layer3RuleOptimize>
 <layer2RuleOptimize>true</layer2RuleOptimize>
 <tcpStrictOption>false</tcpStrictOption>
 <autoDraftDisabled>true</autoDraftDisabled>
</globalConfiguration>
```

Synchronize Firewall

Synchronize hosts and clusters with the last good configuration in NSX Manager database.

`POST /api/4.0/firewall/forceSync/{ID}`

URI Parameters:

`ID` (required)	Specified host or cluster to synchronize

Description:
Force sync host or cluster.

Enable Firewall

Enable or disable firewall components on a cluster.

`POST /api/4.0/firewall/{domainID}/enable/{truefalse}`

URI Parameters:

`domainID` (required)	Specified cluster
`truefalse` (required)	Set parameter to true/false to enable/disable

Description:
Enable or disable firewall components on a cluster

Working with IPFIX

Configuring IPFIX exports specific flows directly from Distributed Firewall to a flow collector.

`GET /api/4.0/firewall/{contextId}/config/ipfix`

URI Parameters:

`contextId`	Specified context

Description:
Query IPFIX configuration.

`PUT /api/4.0/firewall/{contextId}/config/ipfix`

URI Parameters:

`contextId`	Specified context

Description:
Configure IPFIX.

Request:
Body: application/xml

```
<ipfixConfiguration>
 <contextId></contextId>
 <ipfixEnabled></ipfixEnabled>
 <observationDomainId></observationDomainId>
 <flowTimeout></flowTimeout>
 <collector>
   <ip></ip>
   <port></port>
 </collector>
</ipfixConfiguration>
```

`DELETE /api/4.0/firewall/{contextId}/config/ipfix`

URI Parameters:

`contextId`	Specified context

Description:
Deleting IPFIX configuration resets the config to default values

Working With SpoofGuard

After synchronizing with the vCenter Server, NSX Manager collects the IP addresses of all vCenter guest virtual machines. If a virtual machine has been compromised, the IP address can be spoofed and malicious transmissions can bypass firewall policies.

You create a SpoofGuard policy for specific networks that allows you to authorize the reported IP addresses and alter them if necessary to prevent spoofing. SpoofGuard inherently trusts the MAC addresses of virtual machines collected from the VMX files and vSphere SDK. Operating separately from Firewall rules, you can use SpoofGuard to block traffic determined to be spoofed.

Working with SpoofGuard Policies

You can create a SpoofGuard policy to specify the operation mode for specific networks. The system generated policy applies to port groups and logical switches not covered by existing SpoofGuard policies.

The operationMode for a SpoofGuard policy can be set to one of the following:
- **TOFU** - Automatically trust IP assignments on their first use
- **MANUAL** - Manually inspect and approve all IP assignments before first use
- **DISABLE** - Disable the SpoofGuard policy

`GET /api/4.0/services/spoofguard/policies`

Description:
Retrieve information about all SpoofGuard policies.

`POST /api/4.0/services/spoofguard/policies`

Description:
Create a SpoofGuard policy to specify the operation mode for networks.

Request:
Body: application/xml

```
<spoofguardPolicy>
 <name></name>
 <description></description>
 <operationMode></operationMode>
 <enforcementPoint>
   <id></id>
   <name></name>
   <type></type>
 </enforcementPoint>
 <allowLocalIPs></allowLocalIPs>
</spoofguardPolicy>
```

Working With a Specific SpoofGuard Policy

`GET /api/4.0/services/spoofguard/policies/{policyID}`

URI Parameters:

policyID	SpoofGuard policy ID.

Description:
Retrieve information about the specified SpoofGuard policy.

`PUT /api/4.0/services/spoofguard/policies/{policyID}`

URI Parameters:

policyID	SpoofGuard policy ID.

Description:
Modify the specified SpoofGuard policy.

Request:
Body: application/xml

```
<spoofguardPolicy>
 <policyId></policyId>
 <name></name>
 <description></description>
 <operationMode></operationMode>
 <enforcementPoint>
   <id></id>
   <name></name>
   <type></type>
 </enforcementPoint>
 <allowLocalIPs></allowLocalIPs>
</spoofguardPolicy>
```

`DELETE /api/4.0/services/spoofguard/policies/{policyID}`

URI Parameters:

policyID	SpoofGuard policy ID.

Description:
Delete the specified SpoofGuard policy.

Perform SpoofGuard Operations on IP Addresses in a Specific Policy

`GET /api/4.0/services/spoofguard/{policyID}`

URI Parameters:

policyID	SpoofGuard policy ID.

Query Parameters:

`list` (optional)	Specify one of the following states: *ACTIVE*, *INACTIVE*, *PUBLISHED*, *UNPUBLISHED*, *REVIEW_PENDING*, *DUPLICATE*.

Description:
Retrieve IP addresses for the specified state.

`POST /api/4.0/services/spoofguard/{policyID}`

URI Parameters:

`policyID`	SpoofGuard policy ID.

Query Parameters:

`vnicId` (optional)	Perform the specified action on IP addresses for the specified vNIC ID.
`action` (required)	Set to *approve* along with specified IP addresses in body to approve them, or set to *publish* to publish approved IP addresses.

Description:
Approve or publish IP addresses.

Request:
Body: application/xml

```
<spoofguardList>
 <spoofguard>
   <id></id>
   <vnicUuid></vnicUuid>
   <approvedIpAddress>
     <ipAddress></ipAddress>
   </approvedIpAddress>
   <approvedMacAddress></approvedMacAddress>
   <approvedBy></approvedBy>
   <approvedOn></approvedOn>
   <publishedIpAddress>
     <ipAddress></ipAddress>
   </publishedIpAddress>
   <publishedMacAddress></publishedMacAddress>
   <publishedBy></publishedBy>
   <publishedOn></publishedOn>
 </spoofguard>
</spoofguardList>
```

Working with Flow Monitoring

Working With Flow Monitoring Statistics

`GET /api/2.1/app/flow/flowstats`

Query Parameters:

`contextId`	vCenter MOB ID of the datacenter, portgroup, vm, or UUID of the vNIC for which traffic flow is to be retrieved.
`flowType`	Type of flow to be retrieved. Possible values are: • *TCP_UDP* • *LAYER2* • *LAYER3*
`startTime`	Flows with start time greater than specified time are retrieved.
`endTime`	Flows with start time less than specified time are retrieved.
`startIndex (optional)`	The starting point for returning results.
`pageSize (optional)`	The number of results to return. Range is 1-1024.

Description:
Retrieve flow statistics for a datacenter, port group, VM, or vNIC.

Response values for flow statistics:
- **blocked** - indicates whether traffic is blocked:
  - 0 - flow allowed
  - 1 - flow blocked
  - 2 - flow blocked by SpoofGuard
- **protocol** - protocol in flow:
  - 0 - TCP
  - 1 - UDP
  - 2 - ICMP
- **direction** - direction of flow:
  - 0 - to virtual machine
  - 1 - from virtual machine
- **controlDirection** - control direction for dynamic TCP traffic:
  - 0 - source -> destination
  - 1 - destination -> source

Responses:
Status Code: 200
Body: application/xml

```
<FlowStatsPage>
 <pagingInfo>
   <contextId>datacenter-2538</contextId>
   <flowType>TCP_UDP</flowType>
   <startTime>1327405883000</startTime>
```

```
    <endTime>1327482600000</endTime>
    <totalCount>817</totalCount>
    <startIndex>0</startIndex>
    <pageSize>2</pageSize>
  </pagingInfo>
  <flowStatsTcpUdp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>1449</sessions>
    <sourcePackets>1449</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>227493</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.174</sourceIp>
    <destinationIp>255.255.255.255</destinationIp>
    <destinationPort>17500</destinationPort>
    <controlProtocol></controlProtocol>
    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpUdp>
  <flowStatsTcpUdp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>69</sessions>
    <sourcePackets>69</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>17832</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.13</sourceIp>
    <destinationIp>10.112.199.255</destinationIp>
    <destinationPort>138</destinationPort>
    <controlProtocol></controlProtocol>
    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpUdp>
</FlowStatsPage>
```

Working With Flow Monitoring Meta-Data

`GET /api/2.1/app/flow/flowstats/info`

Description:

Retrieve flow statistics meta-data.

This method retrieves the following information for each flow type:
- minimum start time
- maximum end time
- total flow count

Responses:

Status Code: 200

Body: application/xml

```
<FlowStatsInfo>
 <flowStatsInfoTcpUdp>
   <minimumStartTime>1327405883000</minimumStartTime>
   <maximumEndTime>1327482600000</maximumEndTime>
   <totalCount>817</totalCount>
 </flowStatsInfoTcpUdp>
 <flowStatsInfoLayer3>
   <minimumStartTime>1327405883000</minimumStartTime>
   <maximumEndTime>1327482600000</maximumEndTime>
   <totalCount>21</totalCount>
 </flowStatsInfoLayer3>
 <flowStatsInfoLayer2>
 <minimumStartTime>1327405883000</minimumStartTime>
 <maximumEndTime>1327482600000</maximumEndTime>
 <totalCount>531</totalCount>
 </flowStatsInfoLayer2>
</FlowStatsInfo>
```

Working With Flow Monitoring Configuration

Flow records generated on all hosts are sent to NSX Manager, which consumes the records and displays aggregated information. Hosts can generate large numbers of flow records. You can configure flow monitoring to exclude certain records from collection. The flow configuration applies to all hosts.
- **collectFlows** - if true, flow collection is enabled.
- **ignoreBlockedFlows** - if true, ignore blocked flows.
- **ignoreLayer2Flows** - if true, ignore layer 2 flows.
- **sourceIPs** - source IPs to exclude. For example: 10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1/24.
- **sourceContainer** - source containers to exclude. Containers can contain VM, vNic, IP Set, MAC Set.
- **destinationIPs** - destination IPs to exclude.
- **destinationContainer** - destination containers to exclude. Containers can contain VM, vNic, IP Set, MAC Set.
- **destinationPorts** - destination ports to exclude.
- **serviceContainers** - service containers to exclude. Container can contain application or application group.

Flow exclusion happens at the host. The following flows are discarded by default:
- Broadcast IP (255.255.255.255)
- Local multicast group (224.0.0.0/24)
- Broadcast MAC address (FF:FF:FF:FF:FF:FF)

`GET /api/2.1/app/flow/config`

Description:

Retrieve flow monitoring configuration.

Responses:

Status Code: 200
Body: application/xml

```
<FlowConfiguration>
 <collectFlows>true</collectFlows>
 <ignoreBlockedFlows>false</ignoreBlockedFlows>
 <ignoreLayer2Flows>false</ignoreLayer2Flows>
 <sourceIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1/24</sourceIPs>
 <sourceContainer>
   <name>vm1 - Network adapter 1</name>
   <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
   <type>Vnic</type>
 </sourceContainer>
 <sourceContainer>
   <name>Large XP-1</name>
   <id>vm-126</id>
   <type>VirtualMachine</type>
 </sourceContainer>
 <destinationIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1/24</destinationIPs>
 <destinationContainer>
   <name>vm2 - Network adapter 2</name>
   <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
   <type>Vnic</type>
 </destinationContainer>
 <destinationContainer>
   <name>Small XP-2</name>
   <id>vm-226</id>
   <type>VirtualMachine</type>
 </destinationContainer>
 <destinationPorts>22, 40-50, 60</destinationPorts>
 <service>
   <name>VMware-VDM2.x-Ephemeral</name>
   <id>application-161</id>
 </service>
</FlowConfiguration>
```

PUT /api/2.1/app/flow/config

Description:
Update flow monitoring configuration.

Request:
Body: application/xml

```
<FlowConfiguration>
 <collectFlows>true</collectFlows>
 <ignoreBlockedFlows>false</ignoreBlockedFlows>
 <ignoreLayer2Flows>false</ignoreLayer2Flows>
 <sourceIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1/24</sourceIPs>
 <sourceContainer>
   <name>vm1 - Network adapter 1</name>
   <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
   <type>Vnic</type>
 </sourceContainer>
 <sourceContainer>
```

```
    <name>Large XP-1</name>
    <id>vm-126</id>
    <type>VirtualMachine</type>
  </sourceContainer>
  <destinationIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1/24</destinationIPs>
  <destinationContainer>
    <name>vm2 - Network adapter 2</name>
    <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
    <type>Vnic</type>
  </destinationContainer>
  <destinationContainer>
    <name>Small XP-2</name>
    <id>vm-226</id>
    <type>VirtualMachine</type>
  </destinationContainer>
  <destinationPorts>22, 40-50, 60</destinationPorts>
  <service>
    <name>VMware-VDM2.x-Ephemeral</name>
    <id>application-161</id>
  </service>
</FlowConfiguration>
```

Working with Flow Configuration for a Specific Context

`DELETE /api/2.1/app/flow/{contextId}`

URI Parameters:

`contextId`	Context ID.

Description:
Delete flow records for the specified context.

Exclude Virtual Machines from Firewall Protection

`GET /api/2.1/app/excludelist`

Description:
Retrieve the set of VMs in the exclusion list.

Working with the Exclusion List

`PUT /api/2.1/app/excludelist/{memberID}`

URI Parameters:

`memberID`	vc-moref-id of a virtual machine.

Description:
Add a vm to the exclusion list.

`DELETE /api/2.1/app/excludelist/{memberID}`

URI Parameters:

`memberID`	vc-moref-id of a virtual machine.

Description:
Delete a vm from exclusion list.

Working with NSX Edge

There are two types of NSX Edge: Edge services gateway and logical (distributed) router.

Edge Services Gateway

The services gateway gives you access to all NSX Edge services such as firewall, NAT, DHCP, VPN, load balancing, and high availability. You can install multiple Edge services gateway virtual appliances in a datacenter. Each Edge service gateway virtual appliance can have a total of ten uplink and internal network interfaces.

The internal interfaces connect to secured port groups and act as the gateway for all protected virtual machines in the port group. The subnet assigned to the internal interface can be a publicly routed IP space or a NATed/routed RFC 1918 private space. Firewall rules and other NSX Edge services are enforced on traffic between network interfaces.

Uplink interfaces of NSX Edge connect to uplink port groups that have access to a shared corporate network or a service that provides access layer networking. Multiple external IP addresses can be configured for load balancer, site-to-site VPN, and NAT services.

Logical (Distributed) Router

The logical router provides East-West distributed routing with tenant IP address space and data path isolation. Virtual machines or workloads that reside on the same host on different subnets can communicate with one another without having to traverse a traditional routing interface.

A logical router can have up to 9 uplink interfaces and up to 990 internal interfaces.

```
GET /api/4.0/edges
```

Query Parameters:

`datacenter (optional)`	Retrieve Edges by datacenter MoId.
`tenant (optional)`	Retrieve Edges on specified tenant (by tenant ID).
`pg (optional)`	Retrieve Edges with one interface on specified port group (by port group MoId).

Description:

Retrieve a list of all NSX Edges in your inventory. You can use the query parameters to filter results.

```
POST /api/4.0/edges
```

Query Parameters:

`isUniversal (optional)`	Set to "true" when creating a universal logical router. Note the **type** in the request body must be *distributedRouter*.

Description:

You can install NSX Edge as a services gateway or as a logical router.

The **type** parameter determines which type of NSX Edge is deployed: *distributedRouter* or *gatewayServices*. If no type is specified, the type is *gatewayServices*.

Other parameters for this method will differ depending on which type of NSX Edge you are deploying. See the examples and parameter tables for more information.

NSX Edge: Service Gateway

The NSX Edge installation API copies the NSX Edge OVF from the Edge Manager to the specified datastore and deploys an NSX Edge on the given datacenter. After the NSX Edge is installed, the virtual machine powers on and initializes according to the given network configuration. If an appliance is added, it is deployed with the specified configuration.

Installing an NSX Edge instance adds a virtual machine to the vCenter Server inventory, you must specify an IP address for the management interface, and you may name the NSX Edge instance.

The configuration you specify when you install an NSX Edge is stored in the database. If an appliance is added, the configuration is applied to it and it is deployed.

NOTE: Do not use hidden/system resource pool IDs as they are not supported on the UI.

Request Body to Create Edge Services Gateway

```
<edge>
 <datacenterMoid>datacenter-2</datacenterMoid>
 <name>org1-edge</name>
 <description>Description for the edge gateway</description>
 <tenant>org1</tenant>
 <fqdn>org1edge1</fqdn>
 <vseLogLevel>info</vseLogLevel>
 <enableAesni>false</enableAesni>
 <enableFips>true</enableFips>
 <appliances>
   <applianceSize>compact</applianceSize>
   <enableCoreDump>true</enableCoreDump>
   <appliance>
     <resourcePoolId>resgroup-53</resourcePoolId>
     <datastoreId>datastore-29</datastoreId>
     <hostId>host-28</hostId>
     <vmFolderId>group-v38</vmFolderId>
     <customField>
       <key>system.service.vmware.vsla.main01</key>
       <value>string</value>
     </customField>
     <cpuReservation>
       <limit>2399</limit>
       <reservation>500</reservation>
       <shares>500</shares>
     </cpuReservation>
     <memoryReservation>
       <limit>5000</limit>
       <reservation>500</reservation>
       <shares>20480</shares>
     </memoryReservation>
   </appliance>
 </appliances>
 <vnics>
   <vnic>
     <index>0</index>
     <name>internal0</name>
     <type>internal</type>
     <portgroupId>dvportgroup-114</portgroupId>
     <addressGroups>
       <addressGroup>
         <primaryAddress>192.168.3.1</primaryAddress>
         <secondaryAddresses>
           <ipAddress>192.168.3.2</ipAddress>
           <ipAddress>192.168.3.3</ipAddress>
         </secondaryAddresses>
         <subnetMask>255.255.255.0</subnetMask>
```

```
      </addressGroup>
      <addressGroup>
        <primaryAddress>192.168.4.1</primaryAddress>
        <secondaryAddresses>
          <ipAddress>192.168.4.2</ipAddress>
          <ipAddress>192.168.4.3</ipAddress>
        </secondaryAddresses>
        <subnetPrefixLength>24</subnetPrefixLength>
      </addressGroup>
      <addressGroup>
        <primaryAddress>ffff::1</primaryAddress>
        <secondaryAddresses>
          <ipAddress>ffff::2</ipAddress>
        </secondaryAddresses>
        <subnetPrefixLength>64</subnetPrefixLength>
      </addressGroup>
    </addressGroups>
    <macAddress>
      <edgeVmHaIndex>0</edgeVmHaIndex>
      <value>00:50:56:01:03:23</value>
    </macAddress>
    <fenceParameter>
      <key>ethernet0.filter1.param1</key>
      <value>1</value>
    </fenceParameter>
    <mtu>1500</mtu>
    <enableProxyArp>false</enableProxyArp>
    <enableSendRedirects>true</enableSendRedirects>
    <isConnected>true</isConnected>
    <inShapingPolicy>
      <averageBandwidth>200000000</averageBandwidth>
      <peakBandwidth>200000000</peakBandwidth>
      <burstSize>0</burstSize>
      <enabled>true</enabled>
      <inherited>false</inherited>
    </inShapingPolicy>
    <outShapingPolicy>
      <averageBandwidth>400000000</averageBandwidth>
      <peakBandwidth>400000000</peakBandwidth>
      <burstSize>0</burstSize>
      <enabled>true</enabled>
      <inherited>false</inherited>
    </outShapingPolicy>
  </vnic>
</vnics>
<cliSettings>
  <userName>test</userName>
  <password>test123!</password>
  <remoteAccess>false</remoteAccess>
</cliSettings>
<autoConfiguration>
  <enabled>true</enabled>
  <rulePriority>high</rulePriority>
</autoConfiguration>
<dnsClient>
  <primaryDns>10.117.0.1</primaryDns>
  <secondaryDns>10.117.0.2</secondaryDns>
  <domainName>vmware.com</domainName>
  <domainName>foo.com</domainName>
</dnsClient>
<queryDaemon>
  <enabled>true</enabled>
```

```
   <port>5666</port>
 </queryDaemon>
</edge>
```

NSX Edge: Logical (Distributed) Router

Before installing a logical router, you must prepare the hosts on the appropriate clusters.

The user specified configuration is stored in the database and Edge identifier is returned to the user. This identifier must be used for future configurations on the given Edge. If any appliance(s) are specified and at least one connected interface/vnic is specified, then the appliance(s) are deployed and configuration is applied to them.

It is not possible to set the true property upon creation of a distributed logicalrouter Edge and a subsequent API call is required to enable ECMP.

DHCP relay settings are not able to be used when creating a distributed logical router Edge and a subsequent API call is required to configure DHCP relay properties.

Request Body to Create Logical (Distributed) Router

```
<edge>
 <datacenterMoid>datacenter-2</datacenterMoid>
 <type>distributedRouter</type>
 <appliances>
   <appliance>
   <resourcePoolId>resgroup-20</resourcePoolId>
   <datastoreId>datastore-23</datastoreId>
   </appliance>
 </appliances>
 <mgmtInterface>
   <connectedToId>dvportgroup-38</connectedToId>
     <addressGroups>
       <addressGroup>
         <primaryAddress>10.112.196.165</primaryAddress>
         <subnetMask>255.255.252.0</subnetMask>
       </addressGroup>
      </addressGroups>
 </mgmtInterface>
 <interfaces>
   <interface>
     <type>uplink</type>
     <mtu>1500</mtu>
     <isConnected>true</isConnected>
     <addressGroups>
       <addressGroup>
         <primaryAddress>192.168.10.1</primaryAddress>
         <subnetMask>255.255.255.0</subnetMask>
       </addressGroup>
     </addressGroups>
     <connectedToId>dvportgroup-39</connectedToId>
   </interface>
   <interface>
     <type>internal</type>
     <mtu>1500</mtu>
     <isConnected>true</isConnected>
     <addressGroups>
       <addressGroup>
         <primaryAddress>192.168.20.1</primaryAddress>
         <subnetMask>255.255.255.0</subnetMask>
       </addressGroup>
```

```
    </addressGroups>
    <connectedToId>dvportgroup-40</connectedToId>
  </interface>
 </interfaces>
</edge>
```

Request and Response Body Parameters for NSX Edge

General Request Body Parameters: Edge Services Gateway and Logical (Distributed) Router

Parameter	Description	Comments
datacenterMoid	Specify vCenter Managed Object Identifier of data center on which edge has to be deployed	Required.
type	Specify which kind of NSX Edge to deploy. Choice of *distributedRouter* or *gatewayServices*.	Optional. Default is *gatewayServices*.
name	Specify a name for the new NSX Edge.	Optional. Default is *NSX-<edgeId>*. Used as a VM name on vCenter appended by *-<haIndex>*.
description	NSX Edge description.	Optional.
tenant	Specify the tenant. Used for syslog messages.	Optional.
fqdn	Fully Qualified Domain Name for the edge.	Optional. Default is *NSX-<edgeId>* Used to set hostname on the VM. Appended by *-<haIndex>*
vseLogLevel	Defines the log level for log messages captured in the log files.	Optional. Choice of: *emergency*, *alert*, *critical*, *error*, *warning*, *notice*, *debug*. Default is *info*.
enableAesni	Enable support for Advanced Encryption Standard New Instructions on the Edge.	Optional. True/False. Default is *true*.
enableCoreDump	Deploys a new NSX Edge for debug/core-dump purpose.	Optional. Default is false. Enabling core-dump will deploy an extra disk for core-dump files.

Appliances Configuration: Edge Services Gateway and Logical (Distributed) Router

Parameter	Description	Comments
applianceSize	Edge form factor, it determines the NSX Edge size and capability.	Required. Choice of: *compact*, *large*, *quadlarge*, *xlarge*. Default is *compact*.
deployAppliances	Determine whether to deploy appliances.	Default is *true*.

appliance	Appliance configuration details.	Required. Can configure a maximum of two appliances. Until one appliance is configured and NSX Edge VM is deployed successfully, none of the configured features will serve the network.
resourcePoolId	Details of resource pool on which to deploy NSX Edge.	Required. Can be resource pool ID, e.g. *resgroup-15* or cluster ID, e.g. *domain-c41*.
datastoreId	Details of datastore on which to deploy NSX Edge.	Required.
hostId	ID of the host on which to deploy the NSX Edge.	Optional.
vmFolderId	The folder in which to save the NSX Edge.	Optional.
customField	Custom key-value attributes.	Optional. Use custom attributes to associate user-specific meta-information with VMs and managed hosts, stored on vCenter Server.
customField > key	Meta information Key.	Required if customField is specified.
customField > value	Meta information Value.	Required if customField is specified.
cpuReservation > limit	Maximum CPU capacity the NSX Edge can use, specified in MHz.	Optional. -1 (unlimited), any positive integer
cpuReservation > reservation	CPU capacity reserved for NSX Edge in MHz.	Optional.
cpuReservation > shares	Higher value implies NSX Edge has priority when accessing resources.	Optional.
memoryReservation > limit	Maximum memory the NSX Edge can use, specified in MB.	Optional. -1 (unlimited), any positive integer
memoryReservation > reservation	Memory capacity reserved for NSX Edge in MB.	Optional.
memoryReservation > shares	Higher value implies NSX Edge has priority when accessing resources.	Optional.
cliSettings > userName	User name.	Required. length 1-33.
cliSettings > password	Password.	Required. The password must be at least 12 characters long. Must contain at-least 1 uppercase, 1 lowercase, 1 special character and 1 digit. In addition, a character cannot be repeated 3 or more times consectively.
cliSettings > remoteAccess	Enables or disables remote access through SSH.	Required. Relevant firewall rules to allow traffic on port 22 must be opened by user/client

autoConfiguration > enabled	Enable/Disable status of autoConfiguration	Optional. True/False. Default is *true*. If autoConfiguration is enabled, firewall rules are automatically created to allow control traffic. Rules to allow data traffic are not created. For example, if you are using IPsec VPN, and **autoConfiguration** is *true*, firewall rules will automatically be configured to allow IKE traffic. However, you will need to add additional rules to allow the data traffic for the IPsec tunnel. If HA is enabled, firewall rules are always created, even if **autoConfiguration** is *false*, otherwise both HA appliances will become active.
autoConfiguration > rulePriority	Defines the priority of system-defined rules over user-defined rules.	Optional. High, Low. Default is *high*.
queryDaemon > enabled	Configure the communication between server load balancer and NSX Edge VM.	Default is *false*.
queryDaemon > port	Defines the port through which the communication happens.	Integer 1-65535. Default is *5666*.

DNS Client: Edge Services Gateway and Logical (Distributed) Router

Parameter	Description	Comments
dnsClient	Configures the DNS settings of the Edge Services Gateway.	Optional. If the primary/secondary are specified and the DNS service is not specified, the primary/secondary will be used as the default of the DNS service.
primaryDns	Primary DNS IP	
secondaryDns	Secondary DNS IP	
domainName	Domain Name of Edge	
domainName	Secondary Domain Name of Edge	

vNIC Parameters: Edge Services Gateway Only

Parameter	Description	Comments
vnic	Configure interface (vNic).	Required. Until one connected vNic is configured, none of the configured features will serve the network.
index	Index of vNic to be configured. Value varies from 0-9. 4094 sub-interfaces can be configured in trunk mode.	Required.
name	Name of the vNic.	Optional. System provides default names: vnic0...vnic9.
label	Label for the vNic.	Optional. System provides default labels: vNic_0...vNic_9.

type	Type of interface connected to vNic.	Optional. Choice of: *Uplink*, *Internal*, *TRUNK*. Default is *Internal*. *TRUNK* should be specified when sub-interfaces are configured.
portgroupId	Connect NSX Edge to the network through this port group.	Required. Choice of: *portgroupId* or *virtualWireId*. *portgroupId* needs to be defined if *isConnected=true*
addressGroup	Address Group assigned to vNic.	Required. More than one addressGroup/subnets can be assigned to the vNic.
primaryAddress	Primary Address of Edge Interface.	Required. IPv4 and IPv6 addresses are supported.
secondaryAddresses > ipAddress	IP assigned to interface.	Optional. One or more **ipAddress** parameters are allowed, to enable assigning multiple IP addresses to a vNic, for example, for load balancing, NAT, VPN. At least one is required if **secondaryAddresses** is specified.
subnetMask or **subnetPrefixLength**	Subnet mask or prefix value.	Required. Either **subnetMask** or **subnetPrefixLength** should be provided. When both are provided then **subnetprefixLength** is ignored.
macAddress	Option to manually specify the MAC address.	Optional. Managed by vCenter if not provided.
macAddress > edgeVmHaIndex	HA index of the Edge VM.	Required. 0 or 1.
macAddress > value	Value of the MAC address.	Optional. Ensure that MAC addresses provided are unique within the given layer 2 domain.
vnic > mtu	The maximum transmission value for the data packets.	Optional. Default is *1500*.
enableProxyArp	Enables proxy ARP. Do not use this flag unless you want NSX Edge to proxy ARP for all configured subnets.	Optional. True/False. Default is *false*.
enableSendRedirects	Enables ICMP redirect.	Optional. True/False. Default is *true*.
isConnected	Sets if the interface is connected to the port group network.	Optional. True/False. Default is *false*. **portgroupId** needs to be defined if *isConnected=true*.
inShapingPolicy	Configure Incoming Traffic.	Optional.
outShapingPolicy	Configure Outgoing Traffic.	Optional.
averageBandwidth (inShapingPolicy or outShapingPolicy)	Sets average bandwidth for traffic.	Optional.
peakBandwidth (inShapingPolicy or outShapingPolicy)	Sets peak bandwidth for traffic.	Required.
burstSize (inShapingPolicy or outShapingPolicy)	Sets the burst size of the interface.	Required.

enabled (inShapingPolicy or outShapingPolicy)	Enable/disable status of this traffic policy.	Required.
inherited (inShapingPolicy or outShapingPolicy)	Determine whether properties should be inherited to the vNic from the port group.	Required.

HA (Management) Interfaces and Interfaces Configuration: Logical (Distributed) Router Only

Parameter	Description	Comments
mgmtInterface	High availability interface configuration. Interface index 0 is assigned.	Required.
interface	Interface configuration. 1-9 are reserved for uplinks, 10-999 are used for internal interfaces.	Optional. Can be added after logical router creation.
connectedToId (mgmtInterface or interface)	Managed Object ID of logical switch or port group.	For example, *virtualwire-1* or *dvportgroup-50*. Logical router interfaces do not support legacy port groups.
name (mgmtInterface or interface)	Name assigned to interface.	Optional.
addressGroup (mgmtInterface or interface)	Address Group assigned to interface.	Required. Only one **addressGroup** can be configured on each logical router **mgmtInterface** or **interface**.
primaryAddress (mgmtInterface or interface)	Primary Address of interface.	Required. Secondary Addresses are not supported on logical routers. Address must be IPv4.
subnetMask or **subnetPrefixLength** (mgmtInterface or interface)	Subnet mask or prefix value.	Required. Either **subnetMask** or **subnetPrefixLength** should be provided. When both are provided then **subnetprefixLength** is ignored.
mtu (mgmtInterface or interface)	The maximum transmission value for the data packets.	Optional. Default is 1500.
type	Type of interface.	Required. Choice of *uplink* or *internal*.

Request:
Body: application/xml

```
<edge>
 <datacenterMoid>datacenter-2</datacenterMoid>
 <name>org1-edge</name>
 <description>Description for the edge gateway</description>
 <tenant>org1</tenant>
 <fqdn>org1edge1</fqdn>
 <vseLogLevel>info</vseLogLevel>
 <enableAesni>false</enableAesni>
 <enableFips>true</enableFips>
 <appliances>
   <applianceSize>compact</applianceSize>
```

```
  <enableCoreDump>true</enableCoreDump>
  <appliance>
    <resourcePoolId>resgroup-53</resourcePoolId>
    <datastoreId>datastore-29</datastoreId>
    <hostId>host-28</hostId>
    <vmFolderId>group-v38</vmFolderId>
    <customField>
      <key>system.service.vmware.vsla.main01</key>
      <value>string</value>
    </customField>
    <cpuReservation>
      <limit>2399</limit>
      <reservation>500</reservation>
      <shares>500</shares>
    </cpuReservation>
    <memoryReservation>
      <limit>5000</limit>
      <reservation>500</reservation>
      <shares>20480</shares>
    </memoryReservation>
  </appliance>
</appliances>
<vnics>
  <vnic>
    <index>0</index>
    <name>internal0</name>
    <type>internal</type>
    <portgroupId>dvportgroup-114</portgroupId>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.3.1</primaryAddress>
        <secondaryAddresses>
          <ipAddress>192.168.3.2</ipAddress>
          <ipAddress>192.168.3.3</ipAddress>
        </secondaryAddresses>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
      <addressGroup>
        <primaryAddress>192.168.4.1</primaryAddress>
        <secondaryAddresses>
          <ipAddress>192.168.4.2</ipAddress>
          <ipAddress>192.168.4.3</ipAddress>
        </secondaryAddresses>
        <subnetPrefixLength>24</subnetPrefixLength>
      </addressGroup>
      <addressGroup>
        <primaryAddress>ffff::1</primaryAddress>
        <secondaryAddresses>
          <ipAddress>ffff::2</ipAddress>
        </secondaryAddresses>
        <subnetPrefixLength>64</subnetPrefixLength>
      </addressGroup>
    </addressGroups>
    <macAddress>
      <edgeVmHaIndex>0</edgeVmHaIndex>
      <value>00:50:56:01:03:23</value>
    </macAddress>
    <fenceParameter>
      <key>ethernet0.filter1.param1</key>
      <value>1</value>
    </fenceParameter>
    <mtu>1500</mtu>
```

```
      <enableProxyArp>false</enableProxyArp>
      <enableSendRedirects>true</enableSendRedirects>
      <isConnected>true</isConnected>
      <inShapingPolicy>
        <averageBandwidth>200000000</averageBandwidth>
        <peakBandwidth>200000000</peakBandwidth>
        <burstSize>0</burstSize>
        <enabled>true</enabled>
        <inherited>false</inherited>
      </inShapingPolicy>
      <outShapingPolicy>
        <averageBandwidth>400000000</averageBandwidth>
        <peakBandwidth>400000000</peakBandwidth>
        <burstSize>0</burstSize>
        <enabled>true</enabled>
        <inherited>false</inherited>
      </outShapingPolicy>
    </vnic>
  </vnics>
  <cliSettings>
    <userName>test</userName>
    <password>test123!</password>
    <remoteAccess>false</remoteAccess>
  </cliSettings>
  <autoConfiguration>
    <enabled>true</enabled>
    <rulePriority>high</rulePriority>
  </autoConfiguration>
  <dnsClient>
    <primaryDns>10.117.0.1</primaryDns>
    <secondaryDns>10.117.0.2</secondaryDns>
    <domainName>vmware.com</domainName>
    <domainName>foo.com</domainName>
  </dnsClient>
  <queryDaemon>
    <enabled>true</enabled>
    <port>5666</port>
  </queryDaemon>
</edge>
```

Working With a Specific NSX Edge

`GET /api/4.0/edges/{edgeId}`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`isUniversal` (optional)	Filter output to display only universal logical routers.

Description:

Retrieve information about the specified NSX Edge.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState**, **configuredResourcePool**, **configuredDataStore**, **configuredHost**, **configuredVmFolder** parameters added.

`PUT /api/4.0/edges/{edgeId}`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the NSX Edge configuration.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState** parameter added.
6.3.0	Method updated. **dnatMatchSourceAddress**, **snatMatchDestinationAddress**, **dnatMatchSourcePort**, **snatMatchDestinationPort** parameters added. **protocol**, **originalPort**, and **translatedPort** now supported in SNAT rules.

Request:
Body: application/xml

```
<edge>
 <id></id>
 <description></description>
 <datacenterMoid></datacenterMoid>
 <name></name>
 <type></type>
 <fqdn></fqdn>
 <enableAesni></enableAesni>
 <enableFips></enableFips>
 <vseLogLevel></vseLogLevel>
 <vnics>
   <vnic>
     <index></index>
     <name></name>
     <type></type>
     <portgroupId></portgroupId>
     <addressGroups>
       <addressGroup>
         <primaryAddress></primaryAddress>
         <secondaryAddresses>
           <ipAddress></ipAddress>
         </secondaryAddresses>
         <subnetMask></subnetMask>
       </addressGroup>
     </addressGroups>
```

```
    <mtu></mtu>
    <enableProxyArp></enableProxyArp>
    <enableSendRedirects></enableSendRedirects>
    <isConnected></isConnected>
    <inShapingPolicy>
      <averageBandwidth></averageBandwidth>
      <peakBandwidth></peakBandwidth>
      <burstSize></burstSize>
      <enabled></enabled>
      <inherited></inherited>
    </inShapingPolicy>
    <outShapingPolicy>
      <averageBandwidth></averageBandwidth>
      <peakBandwidth></peakBandwidth>
      <burstSize></burstSize>
      <enabled></enabled>
      <inherited></inherited>
    </outShapingPolicy>
  </vnic>
</vnics>
<appliances>
  <applianceSize></applianceSize>
  <appliance>
    <haAdminState></haAdminState>
    <resourcePoolId></resourcePoolId>
    <datastoreId></datastoreId>
    <vmFolderId></vmFolderId>
  </appliance>
</appliances>
<cliSettings>
  <remoteAccess></remoteAccess>
  <userName></userName>
</cliSettings>
<features>
  <firewall>
    <defaultPolicy>
      <action></action>
      <loggingEnabled></loggingEnabled>
    </defaultPolicy>
    <firewallRules>
      <firewallRule>
        <id></id>
        <ruleTag></ruleTag>
        <name></name>
        <ruleType></ruleType>
        <source>
          <exclude></exclude>
          <groupingObjectId></groupingObjectId>
        </source>
        <destination></destination>
        <application>
          <applicationId></applicationId>
        </application>
        <action></action>
        <enabled></enabled>
        <loggingEnabled></loggingEnabled>
        <matchTranslated></matchTranslated>
      </firewallRule>
    </firewallRules>
  </firewall>
  <routing>
    <staticRouting>
```

```
      <defaultRoute>
        <vnic></vnic>
        <gatewayAddress></gatewayAddress>
        <description></description>
      </defaultRoute>
      <staticRoutes>
        <route>
          <vnic></vnic>
          <network></network>
          <nextHop></nextHop>
          <type></type>
        </route>
      </staticRoutes>
    </staticRouting>
    <ospf>
      <enabled></enabled>
    </ospf>
  </routing>
  <highAvailability>
    <enabled></enabled>
    <declareDeadTime></declareDeadTime>
    <logging>
      <enable></enable>
      <logLevel></logLevel>
    </logging>
  </highAvailability>
  <syslog>
    <protocol></protocol>
    <serverAddresses>
      <ipAddress></ipAddress>
    </serverAddresses>
  </syslog>
  <ipsec>
    <enabled></enabled>
    <logging>
      <enable></enable>
      <logLevel></logLevel>
    </logging>
    <sites>
      <site>
        <enabled></enabled>
        <name></name>
        <localId></localId>
        <localIp></localIp>
        <peerId></peerId>
        <encryptionAlgorithm></encryptionAlgorithm>
        <mtu></mtu>
        <enablePfs></enablePfs>
        <dhGroup></dhGroup>
        <localSubnets>
          <subnet></subnet>
        </localSubnets>
        <peerSubnets>
          <subnet></subnet>
        </peerSubnets>
        <psk></psk>
        <authenticationMode></authenticationMode>
      </site>
    </sites>
    <global>
      <caCertificates></caCertificates>
      <crlCertificates></crlCertificates>
```

```
    </global>
  </ipsec>
  <dhcp>
    <enabled></enabled>
    <staticBindings>
      <staticBinding>
        <autoConfigureDNS></autoConfigureDNS>
        <bindingId></bindingId>
        <vmId></vmId>
        <vnicId></vnicId>
        <hostname></hostname>
        <ipAddress></ipAddress>
        <defaultGateway></defaultGateway>
        <leaseTime></leaseTime>
      </staticBinding>
    </staticBindings>
    <ipPools>
      <ipPool>
        <autoConfigureDNS></autoConfigureDNS>
        <poolId></poolId>
        <ipRange></ipRange>
        <defaultGateway></defaultGateway>
        <leaseTime></leaseTime>
      </ipPool>
    </ipPools>
    <logging>
      <enable></enable>
      <logLevel></logLevel>
    </logging>
  </dhcp>
  <nat>
    <natRules>
      <natRule>
        <ruleId></ruleId>
        <ruleTag></ruleTag>
        <ruleType></ruleType>
        <action>dnat</action>
        <vnic></vnic>
        <originalAddress></originalAddress>
        <translatedAddress></translatedAddress>
        <dnatMatchSourceAddress></dnatMatchSourceAddress>
        <loggingEnabled></loggingEnabled>
        <enabled></enabled>
        <protocol></protocol>
        <originalPort></originalPort>
        <translatedPort></translatedPort>
        <dnatMatchSourcePort></dnatMatchSourcePort>
      </natRule>
    </natRules>
  </nat>
 </features>
 <autoConfiguration>
  <enabled></enabled>
  <rulePriority></rulePriority>
 </autoConfiguration>
</edge>
```

POST /api/4.0/edges/{edgeId}

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

action (required)	Options include: • *forcesync* - Resync of the edge • *redeploy* - Redeply the edge • *upgrade* - Upgrade the edge to a newer version

Description:
Manage NSX Edge.

`DELETE /api/4.0/edges/{edgeId}`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete specified NSX Edge configuration. Associated appliances are also deleted.

Working with DNS Client Configuration

`PUT /api/4.0/edges/{edgeId}/dnsclient`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update Edge DNS settings.

Request:
Body: application/xml

```
<dnsClient>
 <primaryDns></primaryDns>
 <secondaryDns></secondaryDns>
 <domainName></domainName>
</dnsClient>
```

Working with AESNI

`POST /api/4.0/edges/{edgeId}/aesni`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enable (required)	

Description:
Modify AESNI setting.

Working With Core Dumps

Enabling core-dump feature results in deployment of built-in extra disk to save core-dump files. Disabling detaches the disk.

`POST /api/4.0/edges/{edgeId}/coredump`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify core dump setting.

Working with FIPS on NSX Edge

`POST /api/4.0/edges/{edgeId}/fips`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enable (required)	Choice of *true* or *false*. Changing the FIPS mode will reboot the NSX Edge appliance.

Description:
Modify FIPS setting.

Working With NSX Edge Logs

`POST /api/4.0/edges/{edgeId}/logging`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`level`	Logging level.

Description:
Modify log setting.

Working With NSX Edge Summary

`GET /api/4.0/edges/{edgeId}/summary`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve details about the specified NSX Edge.

Method history:

Release	Modification
6.3.0	Method updated. **enableFips** parameter added to **appliancesSummary**.

Responses:
Status Code: 200
Body: application/xml

```
<edgeSummary>
 <objectId>edge-3</objectId>
 <objectTypeName>Edge</objectTypeName>
 <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
 <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
 <revision>9</revision>
 <type>
   <typeName>Edge</typeName>
 </type>
 <name>Perimeter-Gateway-01</name>
 <clientHandle></clientHandle>
 <extendedAttributes></extendedAttributes>
 <isUniversal>false</isUniversal>
 <universalRevision>0</universalRevision>
 <id>edge-3</id>
 <state>deployed</state>
 <edgeType>gatewayServices</edgeType>
 <datacenterMoid>datacenter-21</datacenterMoid>
 <datacenterName>Datacenter Site A</datacenterName>
 <tenantId>default</tenantId>
 <apiVersion>4.0</apiVersion>
 <recentJobInfo>
   <jobId>jobdata-35884</jobId>
   <status>SUCCESS</status>
 </recentJobInfo>
 <edgeStatus>GREEN</edgeStatus>
 <numberOfConnectedVnics>2</numberOfConnectedVnics>
```

```
<appliancesSummary>
  <vmVersion>6.2.4</vmVersion>
  <vmBuildInfo>6.2.4-4259031</vmBuildInfo>
  <applianceSize>compact</applianceSize>
  <fqdn>Perimeter-Gateway-01</fqdn>
  <numberOfDeployedVms>1</numberOfDeployedVms>
  <activeVseHaIndex>0</activeVseHaIndex>
  <vmMoidOfActiveVse>vm-391</vmMoidOfActiveVse>
  <vmNameOfActiveVse>Perimeter-Gateway-01-0</vmNameOfActiveVse>
  <hostMoidOfActiveVse>host-203</hostMoidOfActiveVse>
  <hostNameOfActiveVse>esxmgt-02a.corp.local</hostNameOfActiveVse>
  <resourcePoolMoidOfActiveVse>resgroup-42</resourcePoolMoidOfActiveVse>
  <resourcePoolNameOfActiveVse>Resources</resourcePoolNameOfActiveVse>
  <dataStoreMoidOfActiveVse>datastore-29</dataStoreMoidOfActiveVse>
  <dataStoreNameOfActiveVse>ds-site-a-nfs01</dataStoreNameOfActiveVse>
  <statusFromVseUpdatedOn>1487375637539</statusFromVseUpdatedOn>
  <communicationChannel>msgbus</communicationChannel>
  <deployAppliances>true</deployAppliances>
  <enableFips>false</enableFips>
</appliancesSummary>
<featureCapabilities>
  <timestamp>1487375669338</timestamp>
  <featureCapability>
    <service>dhcp</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_POOL_AND_BINDINGS</key>
      <value>2048</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>syslog</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_SERVER_IPS</key>
      <value>2</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>bridging</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
```

```
    <configurationLimit>
      <key>MAX_BRIDGES</key>
      <value>500</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>nat</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_RULES</key>
      <value>2048</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>l2vpn</service>
    <isSupported>false</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>false</isLicensed>
    </permission>
  </featureCapability>
  <featureCapability>
    <service>ipsec</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_TUNNELS</key>
      <value>64</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_TUNNELS_COMPACT</key>
      <value>512</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_TUNNELS_LARGE</key>
      <value>1600</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_TUNNELS_QUADLARGE</key>
      <value>4096</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_TUNNELS_XLARGE</key>
      <value>6000</value>
    </configurationLimit>
  </featureCapability>
```

```
<featureCapability>
  <service>systemControl</service>
  <isSupported>true</isSupported>
  <permission>
    <accessPermission>
      <viewPermission>true</viewPermission>
      <managePermission>true</managePermission>
    </accessPermission>
    <isLicensed>true</isLicensed>
  </permission>
</featureCapability>
<featureCapability>
  <service>gslb</service>
  <isSupported>true</isSupported>
  <permission>
    <accessPermission>
      <viewPermission>true</viewPermission>
      <managePermission>true</managePermission>
    </accessPermission>
    <isLicensed>true</isLicensed>
  </permission>
  <configurationLimit>
    <key>MAX_GSLB_SITES</key>
    <value>10</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_GSLB_IPS</key>
    <value>32</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_GSLB_POOLs</key>
    <value>32</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_MEMBERS_PER_POOL</key>
    <value>10</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_GSLB_MONITORS</key>
    <value>128</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_MONITOR_INSTANCES</key>
    <value>320</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>edge</service>
  <isSupported>true</isSupported>
  <permission>
    <accessPermission>
      <viewPermission>true</viewPermission>
      <managePermission>true</managePermission>
    </accessPermission>
    <isLicensed>true</isLicensed>
  </permission>
  <configurationLimit>
    <key>MAX_APPLIANCES</key>
    <value>2</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
```

```
    <service>firewall</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_RULES</key>
      <value>2048</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>sslvpn</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_SSLVPN_IPPOOLS</key>
      <value>4</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_PRIVATE_NETWORK</key>
      <value>16</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_USERS</key>
      <value>1024</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_AUTH_SERVERS</key>
      <value>4</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_INSTALLATION_PACKAGES</key>
      <value>2</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_WEB_RESOURCE</key>
      <value>16</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_SSLVPN_SCRIPTS</key>
      <value>4</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>highAvailability</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
```

```
    </permission>
    <configurationLimit>
      <key>MAX_MANAGEMENT_IPS</key>
      <value>2</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>dns</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_SERVER_IPS</key>
      <value>2</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_CACHE_SIZE</key>
      <value>8192</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_VIEWS</key>
      <value>100</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_ZONES_PER_VIEW</key>
      <value>1000</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_RECORDS_PER_ZONE</key>
      <value>1000</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_VALUES_PER_RECORD</key>
      <value>100</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>loadBalancer</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_MEMBERS_IN_POOL</key>
      <value>32</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_MONITOR_INSTANCES</key>
      <value>320</value>
    </configurationLimit>
    <configurationLimit>
      <key>MAX_POOLS</key>
      <value>64</value>
```

```
    </configurationLimit>
    <configurationLimit>
      <key>MAX_VIRTUAL_SERVERS</key>
      <value>64</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>routing</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_ROUTES</key>
      <value>2048</value>
    </configurationLimit>
  </featureCapability>
  <featureCapability>
    <service>vnics</service>
    <isSupported>true</isSupported>
    <permission>
      <accessPermission>
        <viewPermission>true</viewPermission>
        <managePermission>true</managePermission>
      </accessPermission>
      <isLicensed>true</isLicensed>
    </permission>
    <configurationLimit>
      <key>MAX_SUB_INTERFACES</key>
      <value>200</value>
    </configurationLimit>
  </featureCapability>
 </featureCapabilities>
 <hypervisorAssist>false</hypervisorAssist>
 <allowedActions>
  <string>Change Log Level</string>
  <string>Add Edge Appliance</string>
  <string>Delete Edge Appliance</string>
  <string>Edit Edge Appliance</string>
  <string>Edit CLI Credentials</string>
  <string>Change edge appliance size</string>
  <string>Force Sync</string>
  <string>Redeploy Edge</string>
  <string>Change Edge Appliance Core Dump Configuration</string>
  <string>Enable hypervisorAssist</string>
  <string>Edit Highavailability</string>
  <string>Edit Dns</string>
  <string>Edit Syslog</string>
  <string>Edit Automatic Rule Generation Settings</string>
  <string>Disable SSH</string>
  <string>Download Edge TechSupport Logs</string>
 </allowedActions>
 <localEgressEnabled>false</localEgressEnabled>
</edgeSummary>
```

Working With NSX Edge Status

`GET /api/4.0/edges/{edgeId}/status`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

getlatest (optional)	If *true*: retrieve the status live from NSX Edge. If *false*: retrieve the latest available status from database.
detailed (optional)	If *true*: retrieve detailed status per feature. If *false*: retrieve aggregated summary of status per feature.
preRulesStatus (optional)	If *true*: retreive detailed output for pre rules in addition to the regular output.

Description:
Retrieve the status of the specified Edge.

The **edgeStatus** has the following possible states:
- *GREEN*: Health checks are successful, status is good.
- *YELLOW*: Intermittent health check failure. If health check fails for five consecutive times for all appliances, status will turn *RED*.
- *GREY*: unknown status.
- *RED*: None of the appliances for this NSX Edge are in a serving state.

Responses:
Status Code: 200
Body: application/xml

```
<edgeStatus>
 <timestamp>1343739873000</timestamp>
 <systemStatus>good</systemStatus>
 <activeVseHaIndex>0</activeVseHaIndex>
 <edgeStatus>GREEN</edgeStatus>
 <publishStatus>APPLIED</publishStatus>
 <version>8</version>
 <edgeVmStatus>
   <edgeVmStatus>
     <edgeVMStatus>GREEN</edgeVMStatus>
     <haState>active</haState>
     <index>0</index>
     <id>vm-358</id>
     <name>test2-0</name>
   </edgeVmStatus>
   <edgeVmStatus>
     <edgeVMStatus>GREEN</edgeVMStatus>
     <haState>active</haState>
     <index>1</index>
     <id>vm-362</id>
     <name>test2-1</name>
   </edgeVmStatus>
```

```
 </edgeVmStatus>
 <featureStatuses>
   <featureStatus>
     <service>loadBalancer</service>
     <configured>false</configured>
     <serverStatus>down</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>dhcp</service>
     <configured>true</configured>
     <publishStatus>Applied</publishStatus>
     <serverStatus>up</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>sslvpn</service>
     <configured>false</configured>
     <serverStatus>down</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>syslog</service>
     <configured>false</configured>
     <serverStatus>up</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>nat</service>
     <configured>false</configured>
   </featureStatus>
   <featureStatus>
     <service>dns</service>
     <configured>false</configured>
     <serverStatus>down</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>ipsec</service>
     <configured>false</configured>
     <serverStatus>down</serverStatus>
   </featureStatus>
   <featureStatus>
     <service>firewall</service>
     <configured>true</configured>
     <publishStatus>Applied</publishStatus>
   </featureStatus>
   <featureStatus>
     <service>staticRouting</service>
     <configured>false</configured>
   </featureStatus>
   <featureStatus>
     <service>highAvailability</service>
     <configured>true</configured>
     <publishStatus>Applied</publishStatus>
     <serverStatus>up</serverStatus>
   </featureStatus>
 </featureStatuses>
</edgeStatus>
```

Working with NSX Edge Tech Support Logs

`GET /api/4.0/edges/{edgeId}/techsupportlogs`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the tech support logs for Edge.

Working with NSX Edge CLI Settings

`PUT /api/4.0/edges/{edgeId}/clisettings`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify CLI credentials and enable/disable SSH for Edge.

Request:
Body: application/xml

```
<cliSettings>
 <userName></userName>
 <password></password>
 <remoteAccess></remoteAccess>
 <passwordExpiry></passwordExpiry>
 <sshLoginBannerText></sshLoginBannerText>
</cliSettings>
```

Working with NSX Edge Remote Access

`POST /api/4.0/edges/{edgeId}/cliremoteaccess`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enable (required)	

Description:
Change CLI remote access

Working with NSX Edge System Control Configuration

`GET /api/4.0/edges/{edgeId}/systemcontrol/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all NSX Edge system control configuration.

If no system control parameters are configured, the response is empty.

`PUT /api/4.0/edges/{edgeId}/systemcontrol/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the NSX Edge system control (sysctl) configuration.

Request:
Body: application/xml

```
<systemControl>
 <property></property>
</systemControl>
```

`DELETE /api/4.0/edges/{edgeId}/systemcontrol/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

rebootNow (required)	You must specify the **rebootNow** query parameter to delete the system control configuration. The NSX Edge appliace is rebooted.

Description:
Delete all NSX Edge system control configuration.

Deleting the system control configuration requires a reboot of the NSX Edge appliance.

Working With NSX Edge Firewall Configuration

Configures firewall for an Edge and stores the specified configuration in database. If any appliances are associated with this Edge, applies the configuration to them. While using this API, you should send the globalConfig, defaultPolicy and the rules. If either of them are not sent, the previous config if any on those fields will be removed and will be changed to the system defaults.

ruleId uniquely identifies a rule and should be specified only for rules that are being updated. If **ruleTag** is specified, the rules on Edge are configured using this user input. Otherwise, Edge is configured using **ruleIds** generated by NSX Manager.

Parameter	Comments
tcpPickOngoingConnections	Boolean, optional, default: *false*.
tcpAllowOutOfWindowPackets	Boolean, optional, default: *false*.
tcpSendResetForClosedVsePorts	Boolean, optional, default: *true*.
dropInvalidTraffic	Boolean, optional, default: *true*.
logInvalidTraffic	Boolean, optional, default: *false*.
tcpTimeoutOpen	Integer, optional, default: *30*.
tcpTimeoutEstablished	Integer, optional, default: *21600*.
tcpTimeoutClose	Integer, optional, default: *30*.
udpTimeout	Integer, optional, default: *60*.
icmpTimeout	Integer, optional, default: *10*.
icmp6Timeout	Integer, optional, default: *10*.
ipGenericTimeout	Integer, optional, default: *120*.
enableSynFloodProtection	Protect against SYN flood attacks by detecting bogus TCP connections and terminating them without consuming firewall state tracking resources. Boolean, optional, default: *false*.
logIcmpErrors	Boolean, optional, default *false*.
dropIcmpReplays	Boolean, optional, default *false*.
defaultPolicy	Optional. Default is *deny*.
action	String, values: *accept*, *deny*.
loggingEnabled	Boolean, optional, default: *false*.
firewallRules	Optional.
action	String. Valid values: *accept*, *deny*.
source	Optional. Default is *any*.
destination	Optional. Default is *any*.
exclude (source or destination)	Boolean. Exclude the specified source or destination.
ipAddress (source or destination)	List of string. Can specify single IP address, range of IP address, or in CIDR format. Can define multiple.
groupingObjectId (source or destination)	List of string, Id of cluster, datacenter, distributedPortGroup, legacyPortGroup, VirtualMachine, vApp, resourcePool, logicalSwitch, IPSet, securityGroup. Can defined multiple.
vnicGroupId (source or destination)	List of string. Possible values are *vnic-index-[1-9]*, *vse*, *external* or *internal*. Can define multiple.
application	optional. When absent its treated as *any*.

applicationId	List of string. Id of service or serviceGroup groupingObject.
service	List.
protocol	String.
port	List of string.
sourcePort	List of string.
icmpType	String.
name	String.
description	String.
enabled	Boolean, optional. Default *true*.
loggingEnabled	Boolean, optional. Default *false*.
matchTranslated	Boolean.
direction	String, optional. Possible values *in* or *out*. When absent its treated as *any*.
ruleTag	Long, optional. This can be used to specify user controlled **ruleId**. If not specified, NSX Manager will generate **ruleId**. Valid values: *1-65536*.

`GET /api/4.0/edges/{edgeId}/firewall/config`

URI Parameters:

`edgeId` `(required)`	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the NSX Edge firewall configuration.

Method history:

Release	Modification
6.2.3	Method updated. **enableSynFloodProtection** parameter added.
6.3.0	Method updated. **logIcmpErrors** and **dropIcmpReplays** parameters added.

`PUT /api/4.0/edges/{edgeId}/firewall/config`

URI Parameters:

`edgeId` `(required)`	Specify the ID of the edge in *edgeId*.

Description:
Configure NSX Edge firewall.

Method history:

Release	Modification
6.2.3	Method updated. **enableSynFloodProtection** parameter added. Default value of **tcpTimeoutEstablished** increased from 3600 to 21600 seconds (6 hours).

6.3.0	Method updated. **logIcmpErrors** and **dropIcmpReplays** parameters added.

Request:
Body: application/xml

```
<firewall>
 <defaultPolicy>
   <action>deny</action>
   <loggingEnabled>false</loggingEnabled>
 </defaultPolicy>
 <globalConfig>
   <tcpPickOngoingConnections>false</tcpPickOngoingConnections>
   <tcpAllowOutOfWindowPackets>false</tcpAllowOutOfWindowPackets>
   <tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts>
   <dropInvalidTraffic>true</dropInvalidTraffic>
   <logInvalidTraffic>false</logInvalidTraffic>
   <tcpTimeoutOpen>30</tcpTimeoutOpen>
   <tcpTimeoutEstablished>21600</tcpTimeoutEstablished>
   <tcpTimeoutClose>30</tcpTimeoutClose>
   <udpTimeout>60</udpTimeout>
   <icmpTimeout>10</icmpTimeout>
   <icmp6Timeout>10</icmp6Timeout>
   <ipGenericTimeout>120</ipGenericTimeout>
   <enableSynFloodProtection>false</enableSynFloodProtection>
   <logIcmpErrors>false</logIcmpErrors>
   <dropIcmpReplays>false</dropIcmpReplays>
 </globalConfig>
 <firewallRules>
   <firewallRule>
     <ruleTag>1</ruleTag>
     <name>rule1</name>
     <source>
       <vnicGroupId>vnic-index-5</vnicGroupId>
       <groupingObjectId>ipset-128</groupingObjectId>
       <ipAddress>1.1.1.1</ipAddress>
     </source>
     <destination>
       <groupingObjectId>ipset-126</groupingObjectId>
       <vnicGroupId>vnic-index-5</vnicGroupId>
       <groupingObjectId>ipset-128</groupingObjectId>
       <ipAddress>192.168.10.0/24</ipAddress>
     </destination>
     <application>
       <applicationId>application-155</applicationId>
       <service>
         <protocol>tcp</protocol>
         <port>80</port>
         <sourcePort>1500</sourcePort>
       </service>
     </application>
     <matchTranslated>true</matchTranslated>
     <direction>in</direction>
     <action>accept</action>
     <enabled>true</enabled>
     <loggingEnabled>true</loggingEnabled>
     <description>comments</description>
   </firewallRule>
 </firewallRules>
```

```
</firewall>
```

`DELETE /api/4.0/edges/{edgeId}/firewall/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete NSX Edge firewall configuration.

Working With Firewall Rules

`POST /api/4.0/edges/{edgeId}/firewall/config/rules`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

aboveRuleId	rule Id.

Description:
Add one or more rules. You can add a rule above a specific rule using the query parameter, indicating the desired ruleID.

Request:
Body: application/xml

```
<firewallRules>
 <firewallRule>
   <ruleTag></ruleTag>
   <name></name>
   <source>
     <ipAddress></ipAddress>
     <groupingObjectId></groupingObjectId>
     <vnicGroupId></vnicGroupId>
   </source>
   <destination>
     <ipAddress></ipAddress>
     <groupingObjectId></groupingObjectId>
     <vnicGroupId></vnicGroupId>
   </destination>
   <application>
     <applicationId></applicationId>
     <service>
       <protocol></protocol>
       <port></port>
       <sourcePort></sourcePort>
     </service>
   </application>
```

```
   <matchTranslated></matchTranslated>
   <direction></direction>
   <action></action>
   <enabled></enabled>
   <loggingEnabled></loggingEnabled>
   <description></description>
 </firewallRule>
</firewallRules>
```

Working With a Specific Firewall Rule

`GET /api/4.0/edges/{edgeId}/firewall/config/rules/{ruleId}`

URI Parameters:

ruleId (required)	Rule ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve specific rule.

`PUT /api/4.0/edges/{edgeId}/firewall/config/rules/{ruleId}`

URI Parameters:

ruleId (required)	Rule ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify a specific firewall rule.

Request:
Body: application/xml

```
<firewallRule>
 <ruleTag></ruleTag>
 <name></name>
 <source>
   <vnicGroupId></vnicGroupId>
   <groupingObjectId></groupingObjectId>
   <ipAddress></ipAddress>
 </source>
 <destination>
   <groupingObjectId></groupingObjectId>
   <vnicGroupId></vnicGroupId>
   <ipAddress></ipAddress>
 </destination>
 <application>
   <applicationId></applicationId>
   <service>
     <protocol></protocol>
```

```
    <port></port>
    <sourcePort></sourcePort>
   </service>
  </application>
  <matchTranslated></matchTranslated>
  <direction></direction>
  <action></action>
  <enabled></enabled>
  <loggingEnabled></loggingEnabled>
  <description></description>
</firewallRule>
```

`DELETE /api/4.0/edges/{edgeId}/firewall/config/rules/{ruleId}`

URI Parameters:

ruleId (required)	Rule ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete firewall rule

Working With the NSX Edge Global Firewall Configuration

`GET /api/4.0/edges/{edgeId}/firewall/config/global`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the firewall default policy for an Edge.

Method history:

Release	Modification
6.2.3	Method updated. **enableSynFloodProtection** parameter added.
6.3.0	Method updated. **logIcmpErrors** and **dropIcmpReplays** parameters added.

`PUT /api/4.0/edges/{edgeId}/firewall/config/global`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure firewall global config for an Edge.

Method history:

Release	Modification
6.2.3	Method updated. **enableSynFloodProtection** parameter added. Default value of **tcpTimeoutEstablished** increased from 3600 to 21600 seconds (6 hours).
6.3.0	Method updated. **logIcmpErrors** and **dropIcmpReplays** parameters added.

Request:
Body: application/xml

```
<globalConfig>
 <tcpPickOngoingConnections></tcpPickOngoingConnections>
 <tcpAllowOutOfWindowPackets></tcpAllowOutOfWindowPackets>
 <tcpSendResetForClosedVsePorts></tcpSendResetForClosedVsePorts>
 <dropInvalidTraffic></dropInvalidTraffic>
 <logInvalidTraffic></logInvalidTraffic>
 <tcpTimeoutOpen></tcpTimeoutOpen>
 <tcpTimeoutEstablished></tcpTimeoutEstablished>
 <tcpTimeoutClose></tcpTimeoutClose>
 <udpTimeout></udpTimeout>
 <icmpTimeout></icmpTimeout>
 <icmp6Timeout></icmp6Timeout>
 <ipGenericTimeout></ipGenericTimeout>
 <enableSynFloodProtection></enableSynFloodProtection>
 <logIcmpErrors></logIcmpErrors>
 <dropIcmpReplays></dropIcmpReplays>
</globalConfig>
```

Working With the Default Firewall Policy for an Edge

`GET /api/4.0/edges/{edgeId}/firewall/config/defaultpolicy`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve default firewall policy

`PUT /api/4.0/edges/{edgeId}/firewall/config/defaultpolicy`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure default firewall policy

Request:
Body: application/xml

```
<firewallDefaultPolicy>
 <action></action>
 <loggingEnabled></loggingEnabled>
</firewallDefaultPolicy>
```

Working With NSX Edge Firewall Statistics

`GET /api/4.0/edges/{edgeId}/firewall/statistics/firewall`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`interval`	60 min by default, can be given as 1 -60 min, oneDay | oneWeek | oneMonth | oneYear.

Description:
Retrieve number of ongoing connections for the firewall configuration.

Working with Statistics for a Specific Firewall Rule

`GET /api/4.0/edges/{edgeId}/firewall/statistics/{ruleId}`

URI Parameters:

`ruleId` (required)	Specified rule.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve stats for firewall rule.

Working With NAT Configuration

NSX Edge provides network address translation (NAT) service to protect the IP addresses of internal (private) networks from the public network.

You can configure NAT rules to provide access to services running on privately addressed virtual machines. There are two types of NAT rules that can be configured: SNAT and DNAT.

For the data path to work, you need to add firewall rules to allow the required traffic for IP addresses and port per the NAT rules.

NAT Parameter Table

Parameter	Description	Other information
enabled	Enable rule.	Boolean. Optional. Default is *true*.
loggingEnabled	Enable logging.	Boolean. Optional. Default is *false*.
ruleTag	Rule tag.	This can be used to specify user-controlled **ruleId**. If not specified, NSX Manager will generate **ruleId**. Optional. Must be between 65537-131072.
ruleId	Identifier for the rule.	Read-only. Long.
ruleType	Rule type.	Read-only. Values: *user*, *internal_high*.
action	Type of NAT.	Valid values: *snat* or *dnat*.
vnic	Interface on which the translating is applied.	String. Optional.
originalAddress	Original address or address range. This is the source address for SNAT rules, and the destination address for DNAT rules.	String. Specify *any*, an IP address (e.g. *192.168.10.10*), an IP range (e.g. *192.168.10.10-192.168.10.19*), or a subnet in CIDR notation (e.g. *192.168.10.1/24*). Default is *any*.
translatedAddress	Translated address or address range. For SNAT rules, this address must be configured on the NSX Edge.	String. Specify *any*, an IP address (e.g. *192.168.10.10*), an IP range (e.g. *192.168.10.10-192.168.10.19*), or a subnet in CIDR notation (e.g. *192.168.10.1/24*). Default is *any*.
dnatMatchSourceAddress	Source address to match in DNAT rules.	String. Specify *any*, an IP address (e.g. *192.168.10.10*), an IP range (e.g. *192.168.10.10-192.168.10.19*), or a subnet in CIDR notation (e.g. *192.168.10.1/24*). Default is *any*. Not valid for SNAT rules.
snatMatchDestinationAddress	Destination address to match in SNAT rules.	String. Specify *any*, an IP address (e.g. *192.168.10.10*), an IP range (e.g. *192.168.10.10-192.168.10.19*), or a subnet in CIDR notation (e.g. *192.168.10.1/24*). Default is *any*. Not valid for DNAT rules.
protocol	Protocol.	String. Optional. Default is *any*.
icmpType	ICMP type.	String. Only supported when protocol is *icmp*. Default is *any*.
originalPort	Original port. This is the source port for SNAT rules, and the destination port for DNAT rules.	String. Optional. Specify *any*, a port (e.g. 1234) or port range (1234-1239). Default is *any*.
translatedPort	Translated port.	String. Optional. Specify *any*, a port (e.g. 1234) or port range (1234-1239). Default is *any*.

dnatMatchSourcePort	Source port in DNAT rules.	String. Optional. Specify *any*, a port (e.g. 1234) or port range (1234-1239). Default is *any*. Not valid for SNAT rules.
snatMatchDestinationPort	Destination port in SNAT rules.	String. Optional. Specify *any*, a port (e.g. 1234) or port range (1234-1239). Default is *any*. Not valid for DNAT rules.

`GET /api/4.0/edges/{edgeId}/nat/config`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve SNAT and DNAT rules for the specified NSX Edge.

Method history:

Release	Modification
6.3.0	Method updated. **dnatMatchSourceAddress**, **snatMatchDestinationAddress**, **dnatMatchSourcePort**, **snatMatchDestinationPort** parameters added. **protocol**, **originalPort**, and **translatedPort** now supported in SNAT rules.

Responses:
Status Code: 200
Body: application/xml

```
<nat>
 <natRules>
   <natRule>
     <ruleTag>196609</ruleTag>
     <ruleId>196609</ruleId>
     <action>dnat</action>
     <vnic>0</vnic>
     <originalAddress>10.112.196.116</originalAddress>
     <translatedAddress>172.16.1.10</translatedAddress>
     <loggingEnabled>true</loggingEnabled>
     <enabled>true</enabled>
     <description>my comments</description>
     <protocol>tcp</protocol>
     <translatedPort>3389</translatedPort>
     <originalPort>3389</originalPort>
     <ruleType>user</ruleType>
   </natRule>
   <natRule>
     <ruleTag>196609</ruleTag>
     <ruleId>196609</ruleId>
     <action>snat</action>
     <vnic>1</vnic>
     <originalAddress>172.16.1.10</originalAddress>
     <translatedAddress>10.112.196.116</translatedAddress>
     <loggingEnabled>false</loggingEnabled>
     <enabled>true</enabled>
```

```
    <description>no comments</description>
    <protocol>any</protocol>
    <originalPort>any</originalPort>
    <translatedPort>any</translatedPort>
    <ruleType>user</ruleType>
   </natRule>
 </natRules>
</nat>
```

PUT /api/4.0/edges/{edgeId}/nat/config

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Configure SNAT and DNAT rules for an Edge.

If you use this method to add new NAT rules, you must include all existing rules in the request body. Any rules that are omitted will be deleted.

Method history:

Release	Modification
6.2.3	Method updated. **vnic** parameter is now optional. The **originalAddress** for DNAT rules is no longer required to be one of the IPs on the NSX Edge vNics.
6.3.0	Method updated. **dnatMatchSourceAddress**, **snatMatchDestinationAddress**, **dnatMatchSourcePort**, **snatMatchDestinationPort** parameters added. **protocol**, **originalPort**, and **translatedPort** now supported in SNAT rules.

Request:

Body: application/xml

```
<nat>
 <natRules>
   <natRule>
     <ruleTag>65537</ruleTag>
     <action>dnat</action>
     <vnic>0</vnic>
     <originalAddress>10.112.196.116</originalAddress>
     <translatedAddress>172.16.1.10</translatedAddress>
     <dnatMatchSourceAddress>any</dnatMatchSourceAddress>
     <loggingEnabled>true</loggingEnabled>
     <enabled>true</enabled>
     <description>my comments</description>
     <protocol>tcp</protocol>
     <originalPort>3389</originalPort>
     <translatedPort>3389</translatedPort>
     <dnatMatchSourcePort>any</dnatMatchSourcePort>
   </natRule>
   <natRule>
```

```
    <ruleTag>65538</ruleTag>
    <action>snat</action>
    <vnic>1</vnic>
    <originalAddress>172.16.1.10</originalAddress>
    <translatedAddress>10.112.196.116</translatedAddress>
    <snatMatchDestinationAddress>any</snatMatchDestinationAddress>
    <loggingEnabled>false</loggingEnabled>
    <enabled>true</enabled>
    <description>no comments</description>
    <snatMatchDestinationPort>any</snatMatchDestinationPort>
   </natRule>
 </natRules>
</nat>
```

`DELETE /api/4.0/edges/{edgeId}/nat/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all NAT rules for the specified NSX Edge. The auto plumbed rules continue to exist.

Working With NAT Rules

`POST /api/4.0/edges/{edgeId}/nat/config/rules`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

aboveRuleId (optional)	Specified rule ID. If no NAT rules exist, you can specify rule ID *0*.

Description:
Add a NAT rule above a specific rule in the NAT rules table (using **aboveRuleId** query parameter) or append NAT rules to the bottom.

Method history:

Release	Modification
6.2.3	Method updated. **vnic** parameter is now optional. The **originalAddress** for DNAT rules is no longer required to be one of the IPs on the NSX Edge vNics.
6.3.0	Method updated. **dnatMatchSourceAddress**, **snatMatchDestinationAddress**, **dnatMatchSourcePort**, **snatMatchDestinationPort** parameters added. **protocol**, **originalPort**, and **translatedPort** now supported in SNAT rules.

Request:
Body: application/xml

```
<natRule>
 <action>dnat</action>
 <vnic>0</vnic>
 <originalAddress>10.112.196.116</originalAddress>
 <translatedAddress>172.16.1.10</translatedAddress>
 <dnatMatchSourceAddress>any</dnatMatchSourceAddress>
 <loggingEnabled>true</loggingEnabled>
 <enabled>true</enabled>
 <description>my comments</description>
 <protocol>tcp</protocol>
 <translatedPort>3389</translatedPort>
 <originalPort>3389</originalPort>
 <dnatMatchSourcePort>any</dnatMatchSourcePort>
</natRule>
```

Working With a Specific NAT Rule

`PUT /api/4.0/edges/{edgeId}/nat/config/rules/{ruleID}`

URI Parameters:

`ruleID` (required)	Specified rule ID.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the specified NAT rule.

Method history:

Release	Modification
6.2.3	Method updated. **vnic** parameter is now optional. The **originalAddress** for DNAT rules is no longer required to be one of the IPs on the NSX Edge vNics.
6.3.0	Method updated. **dnatMatchSourceAddress**, **snatMatchDestinationAddress**, **dnatMatchSourcePort**, **snatMatchDestinationPort** parameters added. **protocol**, **originalPort**, and **translatedPort** now supported in SNAT rules.

Request:
Body: application/xml

```
<natRule>
 <action>dnat</action>
 <vnic>0</vnic>
```

```
 <originalAddress>10.112.196.116</originalAddress>
 <translatedAddress>172.16.1.10</translatedAddress>
 <dnatMatchSourceAddress>any</dnatMatchSourceAddress>
 <loggingEnabled>true</loggingEnabled>
 <enabled>true</enabled>
 <description>my comments</description>
 <protocol>tcp</protocol>
 <translatedPort>3389</translatedPort>
 <originalPort>3389</originalPort>
 <dnatMatchSourcePort>any</dnatMatchSourcePort>
</natRule>
```

`DELETE /api/4.0/edges/{edgeId}/nat/config/rules/{ruleID}`

URI Parameters:

ruleID (required)	Specified rule ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified NAT rule.

Working with the NSX Edge Routing Configuration

You can specify static and dynamic routing for each NSX Edge.

Dynamic routing provides the necessary forwarding information between layer 2 broadcast domains, thereby allowing you to decrease layer 2 broadcast domains and improve network efficiency and scale. NSX extends this intelligence to where the workloads reside for doing East-West routing. This allows more direct virtual machine to virtual machine communication without the costly or timely need to extend hops. At the same time, NSX also provides North-South connectivity, thereby enabling tenants to access public networks.

Global Routing Configuration

Parameter	Description	Comments
routerId	The first uplink IP address of the NSX Edge that pushes routes to the kernel for dynamic routing	Optional. RouterId is set only when configuring the dynamic routing protocols OSPF and BGP.
ecmp	Enables equal-cost multi-path routing (ECMP)	Optional. Boolean. By default, ecmp is set to false.
logging	Logging configuration.	Optional.
logging > enable	Enable/disable status of logging.	Optional. Default is *false*.
logging > logLevel	Sets the log level.	Default is *info*. Valid values: *emergency*, *alert*, *critical*, *error*, *warning*, *notice*, *info*, *debug*.
ipPrefix	Details for one IP prefix.	Optional. Required only if you define redistribution rules in dynamic routing protocols like ospf, bgp.

ipPrefix > name	The name of the IP prefix.	All defined IP prefixes must have unique names.
ipPrefix > ipAddress	IP addresses for the IP prefix.	Optional. String.

Default Route Configuration

Parameter	Description	Comments
description	A description for the default route.	
type	Specifies whether the static route was created by the system as an auto-generated route or the default route (internal); or whether it is a local (user) route.	
mtu	The maximum transmission value for the data packets	Default is 1500. The MTU value cannot be higher than the MTU value set on the NSX Edge interface. By default, mtu is the MTU value of the interface on which the route is configured.
vnic	Interface on which the default route is added.	
gatewayAddress	Default gateway IP used for routing.	
adminDistance	Admin distance. Used to determine which routing protocol to use if two protocols provide route information for the same destination.	Optional. Default value is *1*.

Static Route Configuration

Parameter	Description	Comments
vnic	Interface on which the route is added.	
description	A description for the static route.	
mtu	The maximum transmission value for the data packet.	Default is 1500. By default, mtu is the MTU value of the interface on which the route is configured.
network	The network in CIDR notation.	
nextHop	Next hop IP address.	The router must be able to directly reach the next hop. When ECMP is enabled, multiple next hops can be displayed.
adminDistance	Admin distance. Used to determine which routing protocol to use if two protocols provide route information for the same destination.	Optional. Default value is *1*.

type	Specifies whether the static route was created by the system as an auto-generated route or the default route (internal); or whether it is a local (user) route.	

OSPF Configuration

Parameter	Description	Comments
enabled	OSPF enabled status.	When not specified, it will be treated as false, When false, it will delete the existing config.
gracefulRestart	For packet forwarding to be uninterrupted during restart of services.	Optional.
defaultOriginate	Allows the Edge Services Gateway to advertise itself as a default gateway to its peers.	Optional. Default is *false*. Not allowed on a logical distributed router.
forwardingAddress	The IP address of one of the uplink interfaces.	Logical (distributed) router only.
protocolAddress	An IP address on the same subnet as the forwarding address.	Logical (distributed) router only.
areaId	The area ID. The NSX Edge supports an area ID in the form of a decimal number. Valid values are 0-4294967295.	Required. The value for areaId must be a unique number.
translateType7ToType5	Configure whether this NSX Edge should be used for translating Type 7 to Type 5 LSAs for this OSPF area. If not set, the router with highest router ID is used for translating.	Valid values: *true* or *false*. Optional, default is *false*. Only configurable for OSFP areas of type NSSA.
type	Gives whether the type is *normal* or *nssa*.	Optional. Default type is normal. NSSAs (the not-so-stubby areas feature described in RFC 3101) prevents the flooding of AS-external link-state advertisements (LSAs). They rely on default routing to external destinations. Therefore, NSSAs are placed at the edge of an OSPF routing domain. NSSA can import external routes into the OSPF routing domain, thereby providing transit service to small routing domains that are not part of the OSPF routing domain.
authentication > type	Authentication type.	Choice of *none*, *password*, or *md5*. If authentication information isn't provided, type is *none*. Type *password*: a password is included in the transmitted packet. Type *md5*: an MD5 checksum is included in the transmitted packet.

authentication > value	The password or MD5 key, respectively	
vnic	The interface that is mapped to OSPF Area	Required. The interface specifies the external network that both NSX Edges are connected to.
areaId	An area ID. Can be in the form of an IP address or decimal number.	Required.
helloInterval	The default interval between hello packets that are sent on the interface	Optional. By default, set to 10 seconds with valid values 1-255.
deadInterval	The default interval during which at least one hello packet must be received from a neighbor before the router declares that neighbor down.	Optional. By default, set to 40 seconds. Valid values are 1-65535.
priority	The default priority of the interface. The interface with the highest priority is the designated router.	Optional. By default, set to 128 with valid values 0-255.
cost	The default overhead required to send packets across that interface	Optional. Integer. The cost of an interface is inversely proportional to the bandwidth of that interface. The larger the bandwidth, the smaller the cost. Valid values are 1-65535.
mtuIgnore	Ignore interface MTU setting	*true* or *false*.

BGP Configuration

Parameter	Description	Comments
enabled	BGP routing enable/disable status.	Optional. By default, enabled is set to false.
gracefulRestart	For packet forwarding to be uninterrupted during restart of services.	Optional.
defaultOriginate	Allows the Edge Services Gateway to advertise itself as a default gateway to its peers.	Optional. Default is *false*. Not allowed on a logical distributed router.
localAS	The 2 byte local Autonomous System number that is assigned to the NSX Edge. The path of autonomous systems that a route traverses is used as one metric when selecting the best path to a destination.	Integer. A value (a globally unique number between 1-65535) for the local AS. This local AS is advertised when the NSX Edge peers with routers in other autonomous systems. Either **localAS** or **localASNumber** is required.
localASNumber	The 2 or 4 byte local Autonomous System number that is assigned to the NSX Edge. The path of autonomous systems that a route traverses is used as one metric when selecting the best path to a destination.	Integer. A value (a globally unique number between 1-4294967295) for the Local AS. This local AS is advertised when the NSX Edge peers with routers in other autonomous systems. Can be in plain or dotted format (e.g. 2 byte: 65001 or 0.65001, 4 byte: 65545 or 1.9). Either **localAS** or **localASNumber** is required.

bgpNeighbour > ipAddress	The IP address of the on-premises border device.	Required. String. IPv4 only. IPv6 is not supported. Should not be the same as any of the NSX Edge interfaces's IPs, forwardingAddress, protocolAddress.
bgpNeighbour > forwardingAddress	The IP address of one of the uplink interfaces.	Logical (distributed) router only.
bgpNeighbour > protocolAddress	An IP address on the same subnet as the forwarding address.	Logical (distributed) router only.
bgpNeighbour > remoteAS	The 2 byte remote Autonomous System number that is assigned to the the border device you are creating the connection for.	Integer. A value (a globally unique number between 1-65535) for the remote AS. Either **remoteAS** or **remoteASNumber** is required.
bgpNeighbour > remoteASNumber	The 2 or 4 byte remote Autonomous System number that is assigned to the border device you are creating the connection for.	Integer. A value (a globally unique number between 1-4294967295) for the remote AS. Can be in plain or dotted format (e.g. 2 byte: 65001 or 0.65001, 4 byte: 65545 or 1.9). Either **remoteAS** or **remoteASNumber** is required.
bgpNeighbour > weight	Weight for the neighbor connection	Optional. Integer. By default, weight is set to 60.
bgpNeighbour > holdDownTimer	Interval for the hold down timer	Optional. Integer. The NSX Edge uses the standard, default values for the keep alive timer (60 seconds) and the hold down timer. The default value for the hold down timer is 3x keepalive or 180 seconds. Once peering between two neighbors is achieved, the NSX Edge starts a hold down timer. Every keep alive message it receives from the neighbor resets the hold down timer to 0. When the NSX Edge fails to receive three consecutive keep alive messages, so that the hold down timer reaches 180 seconds, the NSX Edge considers the neighbor down and deletes the routes from this neighbor.
bgpNeighbour > keepAliveTimer	Interval for the keep alive timer.	Optional. Integer. Default is *60*. Valid values are 1-65534.
bgpNeighbour > password	The authentication password.	Optional. String. Each segment sent on the connection between the neighbors is verified. MD5 authentication must be configured with the same password on both BGP neighbors, otherwise, the connection between them will not be made.
bgpFilter > direction	Indicate whether you are filtering traffic to or from the neighbor	Optional. Choice of *in* or *out*.
bgpFilter > action	Permit or deny traffic.	Optional. Choice of *permit* or *deny*.
bgpFilter > network	The network that you want to filter to or from the neighbor.	CIDR format. IPv4 only. IPv6 is not supported.

bgpFilter > ipPrefixGe	The IP prefixes that are to be filtered. Filter prefixes greater than or equal to this value.	Optional. Integer. Specify valid IPv4 prefixes.
bgpFilter > ipPrefixLe	The IP prefixes that are to be filtered. Filter prefixes less than or equal to this value.	Optional. Integer. Specify valid IPv4 prefixes.

Note: New parameters **localASNumber** and **remoteASNumber** have been added in NSX 6.3.0 to allow configuration of 4 byte AS numbers. The previous parameters, **localAS** and **remoteAS** are still valid.

When you configure BGP, you must provide a local AS number parameter (**localAS** or **localASNumber**) and a remote AS number parameter (**remoteAS** or **remoteASNumber**). If you provide both forms of either local or remote AS number (for example, **localAS** and **localASNumber**), the values must be the same.

You can use any combination of remote and local AS parameters, as long as the values are valid. For example, **localAS** of *65501* and **remoteASNumber** of *70000*.

If you configure a 2 byte number, both forms of the AS number parameters are returned with a GET request (for example, **localAS** and **localASNumber**). If you configure a 4 byte number, only the 4 byte parameter is returned (**localASNumber**).

If both parameters are present (for example **localAS** and **localASNumber**), and you update one parameter (**localAS**) subsequent GET requests will show both parameters updated.

Route Redistribution Configuration for OSPF or BGP

Parameter	Description	Comments
enabled	Enabled status of route redistribution for the parent protocol (OSFP or BGP).	Optional. Default is *false*.
rule	Route redistribution rule.	
id	The ID for the rule.	Required. Number.
prefixName	The name for the IP prefix to add for route redistribution	Optional. String. Default is *any*. **prefixName** is set using **routingGlobalConfig > ipPrefixes**. By default, the value of prefixName is set to *any*.
from > ospf	Whether OSPF is a learner protocol (it learns routes from other protocols).	Optional. By default set to false for ospf.
from > bgp	Whether BGP is a learner protocol (it learns routes from other protocols).	Optional. By default set to false for bgp.
from > static	Whether routes can be learned from static networks.	Optional. By default set to false for static.
from > connected	Whether routes can be learned from connected networks.	Optional. By default set to false for connected.
action	Whether to permit or deny redistribution from the selected types of networks.	Required. Choice of *deny* or *permit*.

```
GET /api/4.0/edges/{edgeId}/routing/config
```

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve routes.

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameter **translateType7ToType5** added to OSPF section. Parameters **localASNumber** and **remoteASNumber** added to BGP section.

`PUT /api/4.0/edges/{edgeId}/routing/config`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure NSX Edge global routing configuration, static routing, and dynamic routing (OSPF and BGP).

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameter **translateType7ToType5** added to OSPF section. Parameters **localASNumber** and **remoteASNumber** added to BGP section.

Request:
Body: application/xml

```
<routing>
 <routingGlobalConfig>
   <routerId>1.1.1.1</routerId>
   <logging>
     <enable>false</enable>
     <logLevel>info</logLevel>
   </logging>
   <ipPrefixes>
     <ipPrefix>
       <name>a</name>
       <ipAddress>192.168.10.0/24</ipAddress>
     </ipPrefix>
   </ipPrefixes>
 </routingGlobalConfig>
 <staticRouting>
   <staticRoutes>
     <route>
       <description>route1</description>
       <vnic>0</vnic>
```

```
      <network>3.1.1.0/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>
    </route>
  </staticRoutes>
  <defaultRoute>
  <description>defaultRoute</description>
  <vnic>0</vnic>
  <gatewayAddress>172.16.1.12</gatewayAddress>
  <mtu>1500</mtu>
  </defaultRoute>
</staticRouting>
<ospf>
  <enabled>true</enabled>
  <forwardingAddress>192.168.10.2</forwardingAddress>
  <protocolAddress>192.168.10.3</protocolAddress>
  <ospfAreas>
    <ospfArea>
      <areaId>100</areaId>
      <translateType7ToType5>true</translateType7ToType5>
      <type>normal</type>
      <authentication>
        <type>password</type>
        <value>vmware123</value>
      </authentication>
    </ospfArea>
  </ospfAreas>
  <ospfInterfaces>
    <ospfInterface>
      <vnic>0</vnic>
      <areaId>100</areaId>
      <helloInterval>10</helloInterval>
      <deadInterval>40</deadInterval>
      <priority>128</priority>
      <cost>10</cost>
      <mtuIgnore>false</mtuIgnore>
    </ospfInterface>
  </ospfInterfaces>
  <redistribution>
    <enabled>true</enabled>
    <rules>
      <rule>
        <prefixName>a</prefixName>
        <from>
          <ospf>false</ospf>
          <bgp>false</bgp>
          <static>false</static>
          <connected>true</connected>
        </from>
        <action>deny</action>
      </rule>
      <rule>
        <prefixName>b</prefixName>
        <from>
          <ospf>false</ospf>
          <bgp>true</bgp>
          <static>false</static>
          <connected>false</connected>
        </from>
        <action>permit</action>
      </rule>
    </rules>
```

```
    </redistribution>
  </ospf>
  <bgp>
    <enabled>true</enabled>
    <localAS>65535</localAS>
    <localASNumber>65535</localASNumber>
    <bgpNeighbours>
      <bgpNeighbour>
        <ipAddress>192.168.10.10</ipAddress>
        <forwardingAddress>192.168.1.10</forwardingAddress>
        <protocolAddress>192.168.1.11</protocolAddress>
        <remoteAS>65500</remoteAS>
        <remoteASNumber>65500</remoteASNumber>
        <weight>60</weight>
        <holdDownTimer>180</holdDownTimer>
        <keepAliveTimer>60</keepAliveTimer>
        <password>vmware123</password>
        <bgpFilters>
          <bgpFilter>
            <direction>in</direction>
            <action>permit</action>
            <network>10.0.0.0/8</network>
            <ipPrefixGe>17</ipPrefixGe>
            <ipPrefixLe>32</ipPrefixLe>
          </bgpFilter>
        </bgpFilters>
      </bgpNeighbour>
    </bgpNeighbours>
    <redistribution>
      <enabled>true</enabled>
      <rules>
        <rule>
          <from>
            <ospf>true</ospf>
            <bgp>false</bgp>
            <static>true</static>
            <connected>false</connected>
          </from>
          <action>deny</action>
        </rule>
      </rules>
    </redistribution>
  </bgp>
</routing>
```

DELETE /api/4.0/edges/{edgeId}/routing/config

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Delete the routing config stored in the NSX Manager database and the default routes from the specified NSX Edge appliance.

Working with the NSX Edge Global Configuration

`GET /api/4.0/edges/{edgeId}/routing/config/global`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve routing info from NSX Manager database (default route settings, static route configurations).

`PUT /api/4.0/edges/{edgeId}/routing/config/global`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure global route.

Request:
Body: application/xml

```
<routingGlobalConfig>
 <routerId></routerId>
 <ecmp></ecmp>
 <logging>
   <enable></enable>
   <logLevel></logLevel>
 </logging>
 <ipPrefixes>
   <ipPrefix>
     <name></name>
     <ipAddress></ipAddress>
   </ipPrefix>
 </ipPrefixes>
</routingGlobalConfig>
```

Working with Static and Default Routes

`GET /api/4.0/edges/{edgeId}/routing/config/static`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Read static and default routes.

`PUT /api/4.0/edges/{edgeId}/routing/config/static`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure static and default routes.

Request:
Body: application/xml

```
<staticRouting>
 <staticRoutes>
   <route>
     <description></description>
     <vnic></vnic>
     <network></network>
     <nextHop></nextHop>
     <mtu></mtu>
   </route>
 </staticRoutes>
 <defaultRoute>
   <description></description>
   <vnic></vnic>
   <gatewayAddress></gatewayAddress>
   <mtu></mtu>
 </defaultRoute>
</staticRouting>
```

`DELETE /api/4.0/edges/{edgeId}/routing/config/static`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete both static and default routing config stored in the NSX Manager database.

Working With OSPF Routing for NSX Edge

NSX Edge supports OSPF, an interior gateway protocol that routes IP packets only within a single routing domain. It gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer, which makes routing decisions based on the destination IP address found in IP packets.

OSPF routing policies provide a dynamic process of traffic load balancing between routes of equal cost. An OSPF network is divided into routing areas to optimize traffic. An area is a logical collection of OSPF networks, routers, and links that have the same area identification.

Areas are identified by an Area ID.

`GET /api/4.0/edges/{edgeId}/routing/config/ospf`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve OSPF configuration.

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameter **translateType7ToType5** added to OSPF section.

`PUT /api/4.0/edges/{edgeId}/routing/config/ospf`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure OSPF.

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameter **translateType7ToType5** added to OSPF section.

Request:
Body: application/xml

```
<ospf>
 <enabled>true</enabled>
 <ospfAreas>
   <ospfArea>
     <areaId>100</areaId>
     <translateType7ToType5></translateType7ToType5>
     <type>normal</type>
     <authentication>
       <type>password</type>
       <value>vmware123</value>
     </authentication>
   </ospfArea>
 </ospfAreas>
 <ospfInterfaces>
   <ospfInterface>
     <vnic>0</vnic>
     <areaId>100</areaId>
     <helloInterval>10</helloInterval>
     <deadInterval>40</deadInterval>
```

```
    <priority>128</priority>
    <cost>10</cost>
  </ospfInterface>
 </ospfInterfaces>
 <redistribution>
  <enabled>true</enabled>
  <rules>
   <rule>
    <prefixName>a</prefixName>
    <from>
     <isis>true</isis>
     <ospf>false</ospf>
     <bgp>false</bgp>
     <static>false</static>
     <connected>true</connected>
    </from>
    <action>deny</action>
   </rule>
   <rule>
    <prefixName>b</prefixName>
    <from>
     <isis>false</isis>
     <ospf>false</ospf>
     <bgp>true</bgp>
     <static>false</static>
     <connected>false</connected>
    </from>
    <action>permit</action>
   </rule>
  </rules>
 </redistribution>
</ospf>
```

`DELETE /api/4.0/edges/{edgeId}/routing/config/ospf`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete OSPF routing.

Working with BGP Routes for NSX Edge

Border Gateway Protocol (BGP) makes core routing decisions. It includes a table of IP networks or prefixes which designate network reachability among autonomous systems. An underlying connection between two BGP speakers is established before any routing information is exchanged. Keep alive messages are sent out by the BGP speakers in order to keep this relationship alive. Once the connection is established, the BGP speakers exchange routes and synchronize their tables.

`GET /api/4.0/edges/{edgeId}/routing/config/bgp`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve BGP configuration. responses: 200: body: application/xml: example: | true 65535 192.168.1.10 65500 60 180 60 vmware123 in permit 10.0.0.0/8 17 32 out deny 20.0.0.0/26 true 1 a true false true false deny 0 false false false true permit

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameters **localASNumber** and **remoteASNumber** added to BGP section.

`PUT /api/4.0/edges/{edgeId}/routing/config/bgp`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure BGP.

Method history:

Release	Modification
6.2.3	Method updated. **isis** configuration section removed.
6.3.0	Method updated. Parameter **defaultOriginate** removed for logical router NSX Edges. Parameters **localASNumber** and **remoteASNumber** added to BGP section.

Request:
Body: application/xml

```
<bgp>
 <enabled>true</enabled>
 <localAS>65534</localAS>
 <localASNumber>65534</localASNumber>
 <bgpNeighbours>
   <bgpNeighbour>
     <ipAddress>192.168.1.10</ipAddress>
     <remoteAS>65500</remoteAS>
     <remoteASNumber>65500</remoteASNumber>
     <weight>60</weight>
     <holdDownTimer>180</holdDownTimer>
     <keepAliveTimer>60</keepAliveTimer>
     <password>vmware123</password>
     <bgpFilters>
       <bgpFilter>
         <direction>in</direction>
         <action>permit</action>
         <network>10.0.0.0/8</network>
         <ipPrefixGe>17</ipPrefixGe>
         <ipPrefixLe>32</ipPrefixLe>
```

```
      </bgpFilter>
    </bgpFilters>
  </bgpNeighbour>
 </bgpNeighbours>
 <redistribution>
  <enabled>true</enabled>
  <rules>
    <rule>
      <prefixName>a</prefixName>
      <from>
        <ospf>false</ospf>
        <bgp>false</bgp>
        <static>false</static>
        <connected>true</connected>
      </from>
      <action>permit</action>
    </rule>
  </rules>
 </redistribution>
</bgp>
```

`DELETE /api/4.0/edges/{edgeId}/routing/config/bgp`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete BGP Routing

Working With Layer 2 Bridging

`GET /api/4.0/edges/{edgeId}/bridging/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve bridge configuration.

`PUT /api/4.0/edges/{edgeId}/bridging/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure a bridge.

Request:

Body: application/xml

```
<bridges>
 <version></version>
 <enabled></enabled>
 <bridge>
   <name></name>
   <virtualWire></virtualWire>
   <dvportGroup></dvportGroup>
 </bridge>
</bridges>
```

`DELETE /api/4.0/edges/{edgeId}/bridging/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete bridges.

Working With NSX Edge Load Balancer

The NSX Edge load balancer enables network traffic to follow multiple paths to a specific destination. It distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. Load balancing thus helps in achieving optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload. NSX Edge provides load balancing up to Layer 7.

You map an external, or public, IP address to a set of internal servers for load balancing. The load balancer accepts TCP, HTTP, or HTTPS requests on the external IP address and decides which internal server to use. Port 8090 is the default listening port for TCP, port 80 is the default port for HTTP, and port 443 is the default port for HTTPs.

`GET /api/4.0/edges/{edgeId}/loadbalancer/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Get load balancer configuration.

`PUT /api/4.0/edges/{edgeId}/loadbalancer/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure load balancer.

The input contains five parts: application profile, virtual server, pool, monitor and application rule.

For the data path to work, you need to add firewall rules to allow required traffic as per the load balancer configuration.

General Load Balancer Parameters

Parameter	Description	Comments
logging	Load balancer logging setting.	Optional.
enable	Whether logging is enabled.	Optional. Options are *True* or *False*. Default is *False*.
logLevel	Logging level.	Optional. Options are: *EMERGENCY*, *ALERT*, *CRITICAL*, *ERROR*, *WARNING*, *NOTICE*, *INFO*, and *DEBUG*. Default is *INFO*.
accelerationEnabled	Whether **accelerationEnabled** is enabled.	Optional. Options are *True* or *False*. Default is *False*.
enabled	Whether load balancer is enabled.	Optional. Options are *True* or *False*. Default is *True*.

Parameter Table for Monitors

Parameter	Description	Comments
monitor	Monitor list.	Optional.
monitorId	Generated monitor identifier.	Optional. Required if it is used in a pool.
name	Name of the monitor.	Required.
type	Monitor type.	Required. Options are : *HTTP*, *HTTPS*, *TCP*, *ICMP*, *UDP*.
interval	Interval in seconds in which a server is to be tested.	Optional. Default is *5*.
timeout	Timeout value is the maximum time in seconds within which a response from the server must be received.	Optional. Default is *15*.
maxRetries	Maximum number of times the server is tested before it is declared DOWN.	Optional. Default is *3*.
method	Method to send the health check request to the server.	Optional. Options are: *OPTIONS*, *GET*, *HEAD*, *POST*, *PUT*, *DELETE*, *TRACE*, *CONNECT*. Default is *GET* for HTTP monitor.
url	URL to *GET* or *POST*.	Optional. Default is *"/"* for HTTP monitor.
expected	Expected string.	Optional. Default is "HTTP/1" for HTTP/HTTPS protocol.
send	String to be sent to the backend server after a connection is established.	Optional. URL encoded HTTP POST data for HTTP/HTTPS protocol.
receive	String to be received from the backend server for HTTP/HTTPS protocol.	Optional.
extension	Advanced monitor configuration.	Optional.

Parameter Table for Virtual Servers

Parameter	Description	Comments

virtualServer	Virtual server list.	Optional. 0-64 **virtualServer** items can be added
name	Name of the virtual server.	Required. Unique virtualServer name per NSX Edge.
description	Description of the virtual server.	Optional.
enabled	Whether the virtual server is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *True*.
ipAddress	IP address that the load balancer is listening on.	Required. A valid Edge vNic IP address (IPv4 or IPv6).
protocol	Virtual server protocol.	Required. Options are: *HTTP*, *HTTPS*, *TCP*, *UDP*.
port	Port number or port range.	Required. Port number such as *80*, port range such as *80,443* or *1234-1238*, or a combination such as *443,6000-7000*. Valid range: 1-65535.
connectionLimit	Maximum concurrent connections.	Optional. Long. Default is *0*.
connectionRateLimit	Maximum incoming new connection requests per second.	Optional. Long. Default is *null*.
defaultPoolId	Default pool ID.	Optional.
applicationProfileId	Application profile ID.	Optional.
accelerationEnabled	Use the faster L4 load balancer engine rather than L7 load balancer engine.	Optional. Boolean. Options are *True* or *False*. Default is *False*. If a virtual server configuration such as application rules, HTTP type, or cookie persistence, is using the L7 load balancer engine, then the L7 load balancer engine is used, even if **accelerationEnabled** is not set to true.
applicationRuleId	Application rule ID list.	Optional.

Parameter Table for Pools

Parameter	Description	Comments
pool	Pool list.	Optional.
poolId	Generated pool identifier.	Optional. Required if you specify pool object.
name	Name of the pool.	Required.
description	Description of the pool.	Optional.
algorithm	Pool member balancing algorithm.	Optional. Options are: *round-robin*, *ip-hash*, *uri*, *leastconn*, *url*, *httpheader*. Default is *round-robin*.
algorithmParameters	Algorithm parameters for *httpheader*, *url*.	Optional. Required for *url*,*httpheader* algorithm.
transparent	Whether client IP addresses are visible to the backend servers.	Optional. Options are *True* or *False*. Default is *False*.
monitorId	Monitor identifier list.	Optional. Only one monitor is supported.
member	Pool member list.	Optional.

memberId	Generated member identifier.	Optional. Required if you specify member object.
name	Member name.	Optional. Required when it is used in ACL rule.
ipAddress	Member IP address (IPv4/IPv6).	Optional. Required if **groupingObjectId** is null.
groupingObjectId	Member grouping object identifier.	Optional. Required if **ipAddress** is null.
groupingObjectName	Member grouping object name.	Optional.
weight	Member weight.	Optional. Default is *1*.
monitorPort	Monitor port.	Optional. Long. Either **monitorPort** or **port** must be configured.
port	Member port.	Optional. Long. Either **monitorPort** or **port** must be configured.
maxConn	Maximum number of concurrent connections a member can handle.	Optional. Default is *0* which means unlimited.
minConn	Minimum number of concurrent connections a member can handle.	Optional. Default is *0* which means unlimited.
condition	Whether the member is enabled or disabled.	Optional. Options are: *enabled* or *disabled*. Default is *enabled*.

Parameter Table for Application Profiles

Parameter	Description	Comments
applicationProfile	Application profile list.	Optional.
applicationProfileId	Generated application profile identifier.	Optional. Required if it is used in virtual server.
name	Name of application profile.	Required.
persistence	Persistence setting.	Optional.
method	Persistent method.	Required. Options are: *cookie*, *ssl_sessionid*, *sourceip*, *msrdp*.
cookieName	Cookie name.	Optional.
cookieMode	Cookie mode.	Optional. Options are: *insert*, *prefix*, *app*.
expire	Expire time.	Optional.
insertXForwardedFor	Whether **insertXForwardedFor** is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *False*.
sslPassthrough	Whether **sslPassthrough** is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *False*.
httpRedirect	HTTP redirect setting.	Optional.
to	HTTP redirect to.	Required. Required if **httpRedirect** is specified.
serverSslEnabled	Whether **serverSsl** offloading is enabled.	Optional. Boolean. Options are *True* or *False*.
serverSsl	Server SSL setting.	Optional.

ciphers	Cipher suites.	Optional. Options are: *DEFAULT* *ECDHE-RSA-AES128-GCM-SHA256*, *ECDHE-RSA-AES256-GCM-SHA384*, *ECDHE-RSA-AES256-SHA*, *ECDHE-ECDSA-AES256-SHA*, *ECDH-ECDSA-AES256-SHA*, *ECDH-RSA-AES256-SHA*, *AES256-SHA AES128-SHA*, *DES-CBC3-SHA*. Default is *DEFAULT*.
serviceCertificate	Service certificate identifier list.	Optional. Only one certificate is supported.
caCertificate	CA identifier list.	Optional. Required if **serverAuth** is required.
crlCertificate	CRL identifier list.	Optional.
serverAuth	Whether peer certificate should be verified.	Optional. Options are *Required* or *Ignore*. Default is *Ignore*.
clientSsl	Client SSL setting.	Optional.
ciphers	Cipher suites.	Optional. Options are: *DEFAULT* *ECDHE-RSA-AES128-GCM-SHA256*, *ECDHE-RSA-AES256-GCM-SHA384*, *ECDHE-RSA-AES256-SHA*, *ECDHE-ECDSA-AES256-SHA*, *ECDH-ECDSA-AES256-SHA*, *ECDH-RSA-AES256-SHA*, *AES256-SHA AES128-SHA*, *DES-CBC3-SHA*. Default is *DEFAULT*.
serviceCertificate	Service certificate identifier list.	Required. Only one certificate is supported.
caCertificate	CA identifier list.	Optional.
crlCertificate	CRL identifier list.	Optional.
clientAuth	Whether peer certificate should be verified.	Optional. Options are *Required* or *Ignore*. Default is *Ignore*.

Parameter Table for Application Rules

Parameter	Description	Comments
applicationRule	Application rule list.	Optional.
applicationRuleId	Generated application rule identifier.	Optional.
name	Name of application rule.	Required.
script	Application rule script.	Required.

For the data path to work, you need to add firewall rules to allow required traffic as per the load balancer configuration.

The following configuration example is displaying the HTTP/HTTPS Redirection, SSL Offloading, Content Switching, HTTP HealthMonitor parameters:

```
<loadBalancer>
 <enabled>true</enabled>
 <accelerationEnabled>true</accelerationEnabled>
 <logging>
   <enable>true</enable>
   <logLevel>debug</logLevel>
 </logging>
 <applicationRule>
```

```
   <applicationRuleId>applicationRule-1</applicationRuleId>
   <name>traffic_ctrl_rule</name>
   <script>
    &nbspacl; srv1_full srv_conn(pool - http / m1)gt 50
    &nbspacl; srv2_full srv_conn(pool - http / m2)gt 50 use_backend pool - backup if srv1_full or
srv2_full
   </script>
 </applicationRule>
 <applicationRule>
   <applicationRuleId>applicationRule-2</applicationRuleId>
   <name>redirection_rule</name>
   <script>
    &nbspacl; google_page url_beg / google redirect location https : //www.google.com/ if
google_page</script>
   </applicationRule>
   <applicationRule>
     <applicationRuleId>applicationRule-3</applicationRuleId>
     <name>l7_rule</name>
     <script>acl backup_page url_beg / backup use_backend pool - backup if backup_page</script>
 </applicationRule>
 <virtualServer>
   <virtualServerId>virtualServer-1</virtualServerId>
   <name>http_redirection_vip</name>
   <description>http redirection virtualServer</description>
   <enabled>true</enabled>
   <ipAddress>10.117.35.171</ipAddress>
   <protocol>http</protocol>
   <port>80</port>
   <connectionLimit>123</connectionLimit>
   <connectionRateLimit>123</connectionRateLimit>
   <applicationProfileId>applicationProfile-1</applicationProfileId>
   <enableServiceInsertion>false</enableServiceInsertion>
   <accelerationEnabled>true</accelerationEnabled>
 </virtualServer>
 <virtualServer>
   <virtualServerId>virtualServer-2</virtualServerId>
   <name>https_vip</name>
   <description>https virtualServer</description>
   <enabled>true</enabled>
   <ipAddress>10.117.35.171</ipAddress>
   <protocol>https</protocol>
   <port>443</port>
   <connectionLimit>123</connectionLimit>
   <connectionRateLimit>123</connectionRateLimit>
   <defaultPoolId>pool-1</defaultPoolId>
   <applicationProfileId>applicationProfile-2</applicationProfileId>
   <applicationRuleId>applicationRule-1</applicationRuleId>
   <applicationRuleId>applicationRule-2</applicationRuleId>
   <applicationRuleId>applicationRule-3</applicationRuleId>
   <enableServiceInsertion>false</enableServiceInsertion>
   <accelerationEnabled>true</accelerationEnabled>
 </virtualServer>
 <applicationProfile>
   <applicationProfileId>applicationProfile-1</applicationProfileId>
   <name>https_redirection_application_profile</name>
   <insertXForwardedFor>false</insertXForwardedFor>
   <sslPassthrough>false</sslPassthrough>
   <httpRedirect>
     <to>https://10.117.35.171</to>
   </httpRedirect>
 </applicationProfile>
 <applicationProfile>
```

```
  <applicationProfileId>applicationProfile-2</applicationProfileId>
  <name>ssl_offloading_application_profile</name>
  <insertXForwardedFor>false</insertXForwardedFor>
  <serverSslEnabled>true</serverSslEnabled>
  <sslPassthrough>false</sslPassthrough>
  <clientSsl>
    <clientAuth>ignore</clientAuth>
    <ciphers>AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH</ciphers>
    <serviceCertificate>certificate-4</serviceCertificate>
    <caCertificate>certificate-3</caCertificate>
    <crlCertificate>crl-1</crlCertificate>
  </clientSsl>
  <serverSsl>
    <ciphers>AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH</ciphers>
    <serviceCertificate>certificate-4</serviceCertificate>
    <caCertificate>certificate-3</caCertificate>
    <crlCertificate>crl-1</crlCertificate>
  </serverSsl>
</applicationProfile>
<pool>
  <poolId>pool-1</poolId>
  <name>pool-http</name>
  <description>pool-http</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-1</monitorId>
  <member>
    <memberId>member-1</memberId>
    <ipAddress>192.168.101.101</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m1</name>
  </member>
  <member>
    <memberId>member-2</memberId>
    <ipAddress>192.168.101.102</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m2</name>
  </member>
</pool>
<pool>
  <poolId>pool-2</poolId>
  <name>pool-backup</name>
  <description>pool backup</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-1</monitorId>
  <member>
    <memberId>member-3</memberId>
    <ipAddress>192.168.102.101</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.102.102</ipAddress>
```

```
    <weight>1</weight>
    <port>80</port>
    <name>m4</name>
  </member>
 </pool>
 <monitor>
  <monitorId>monitor-1</monitorId>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor</name>
 </monitor>
```

Request:
Body: application/xml

```
<loadBalancer>
 <enabled>true</enabled>
 <enableServiceInsertion>false</enableServiceInsertion>
 <accelerationEnabled>true</accelerationEnabled>
 <logging>
  <enable>true</enable>
  <logLevel>debug</logLevel>
 </logging>
 <virtualServer>
  <virtualServerId>virtualServer-1</virtualServerId>
  <name>http_vip</name>
  <description>http virtualServer</description>
  <enabled>true</enabled>
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>http</protocol>
  <port>80</port>
  <connectionLimit>123</connectionLimit>
  <connectionRateLimit>123</connectionRateLimit>
  <applicationProfileId>applicationProfile-1</applicationProfileId>
  <defaultPoolId>pool-1</defaultPoolId>
  <enableServiceInsertion>false</enableServiceInsertion>
  <accelerationEnabled>true</accelerationEnabled>
  <vendorProfile>
   <vendorTemplateId>577</vendorTemplateId>
   <vendorTemplateName>F5</vendorTemplateName>
   <profileAttributes>
    <attribute>
     <key>abcd</key>
     <name>abcd</name>
     <value>1234</value>
    </attribute>
   </profileAttributes>
  </vendorProfile>
 </virtualServer>
 <virtualServer>
  <virtualServerId>virtualServer-2</virtualServerId>
  <name>https_vip</name>
  <description>https virtualServer</description>
  <enabled>true</enabled>
```

```
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>https</protocol>
  <port>443</port>
  <connectionLimit>123</connectionLimit>
  <connectionRateLimit>123</connectionRateLimit>
  <applicationProfileId>applicationProfile-2</applicationProfileId>
  <defaultPoolId>pool-2</defaultPoolId>
  <enableServiceInsertion>false</enableServiceInsertion>
  <accelerationEnabled>false</accelerationEnabled>
</virtualServer>
<virtualServer>
  <virtualServerId>virtualServer-3</virtualServerId>
  <name>tcp_transparent_vip</name>
  <description>tcp virtualServer</description>
  <enabled>true</enabled>
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>tcp</protocol>
  <port>1234</port>
  <connectionLimit>123</connectionLimit>
  <applicationProfileId>applicationProfile-3</applicationProfileId>
  <defaultPoolId>pool-3</defaultPoolId>
  <enableServiceInsertion>false</enableServiceInsertion>
  <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
<virtualServer>
  <virtualServerId>virtualServer-4</virtualServerId>
  <name>tcp_snat_vip</name>
  <description>tcp snat virtualServer</description>
  <enabled>true</enabled>
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>tcp</protocol>
  <port>1235</port>
  <connectionLimit>123</connectionLimit>
  <applicationProfileId>applicationProfile-3</applicationProfileId>
  <defaultPoolId>pool-4</defaultPoolId>
  <enableServiceInsertion>false</enableServiceInsertion>
  <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
<applicationProfile>
  <applicationProfileId>applicationProfile-1</applicationProfileId>
  <name>http_application_profile</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>true</sslPassthrough>
  <persistence>
    <method>cookie</method>
    <cookieName>JSESSIONID</cookieName>
    <cookieMode>insert</cookieMode>
  </persistence>
</applicationProfile>
<applicationProfile>
  <applicationProfileId>applicationProfile-2</applicationProfileId>
  <name>https_application_profile</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>true</sslPassthrough>
  <persistence>
    <method>ssl_sessionid</method>
  </persistence>
</applicationProfile>
<applicationProfile>
  <applicationProfileId>applicationProfile-3</applicationProfileId>
  <name>tcp_application_profile</name>
  <insertXForwardedFor>false</insertXForwardedFor>
```

```
  <sslPassthrough>true</sslPassthrough>
</applicationProfile>
<pool>
  <poolId>pool-1</poolId>
  <name>pool-http</name>
  <description>pool-http</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-1</monitorId>
  <member>
    <memberId>member-1</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <groupingObjectId>vm-24</groupingObjectId>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m1</name>
  </member>
  <member>
    <memberId>member-2</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m2</name>
    <condition>enabled</condition>
  </member>
</pool>
<pool>
  <poolId>pool-2</poolId>
  <name>pool-https</name>
  <description>pool-https</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-2</monitorId>
  <member>
    <memberId>member-3</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m4</name>
  </member>
</pool>
<pool>
  <poolId>pool-3</poolId>
  <name>pool-tcp</name>
  <description>pool-tcp</description>
  <transparent>true</transparent>
  <algorithm>round-robin</algorithm>
```

```
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-5</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m5</name>
    <monitorPort>80</monitorPort>
  </member>
  <member>
    <memberId>member-6</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m6</name>
    <monitorPort>80</monitorPort>
  </member>
</pool>
<pool>
  <poolId>pool-4</poolId>
  <name>pool-tcp-snat</name>
  <description>pool-tcp-snat</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-7</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m7</name>
    <monitorPort>80</monitorPort>
  </member>
  <member>
    <memberId>member-8</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m8</name>
    <monitorPort>80</monitorPort>
  </member>
</pool>
<monitor>
  <monitorId>monitor-1</monitorId>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor</name>
  <expected>HTTP/1</expected>
  <send>hello</send>
  <receive>ok</receive>
```

```
    <extension>no-bodymax-age=3hcontent-type=Application/xml</extension>
  </monitor>
  <monitor>
    <monitorId>monitor-2</monitorId>
    <type>https</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <method>GET</method>
    <url>/</url>
    <name>https-monitor</name>
  </monitor>
  <monitor>
    <monitorId>monitor-3</monitorId>
    <type>tcp</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <name>tcp-monitor</name>
  </monitor>
</loadBalancer>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete load balancer configuration.

Working with Application Profiles

You create an application profile to define the behavior of a particular type of network traffic. After configuring a profile, you associate the profile with a virtual server. The virtual server then processes traffic according to the values specified in the profile. Using profiles enhances your control over managing network traffic, and makes traffic-management tasks easier and more efficient.

See *Working With NSX Edge Load Balancer* for **applicationProfiles** parameter information.

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all application profiles on the specified Edge.

Responses:
Status Code: 200
Body: application/xml

```
<loadBalancer>
 <applicationProfile>
   <applicationProfileId>applicationProfile-2</applicationProfileId>
   <name>HTTPS-Application-Profile</name>
   <insertXForwardedFor>true</insertXForwardedFor>
   <sslPassthrough>false</sslPassthrough>
   <template>HTTPS</template>
   <serverSslEnabled>false</serverSslEnabled>
 </applicationProfile>
 <applicationProfile>
   <applicationProfileId>applicationProfile-3</applicationProfileId>
   <persistence>
     <method>cookie</method>
     <cookieName>JSESSIONID</cookieName>
     <cookieMode>insert</cookieMode>
   </persistence>
   <name>HTTP-Application-Profile</name>
   <insertXForwardedFor>true</insertXForwardedFor>
   <sslPassthrough>false</sslPassthrough>
   <template>HTTP</template>
   <serverSslEnabled>false</serverSslEnabled>
 </applicationProfile>
 <applicationProfile>
   <applicationProfileId>applicationProfile-4</applicationProfileId>
   <persistence>
     <method>sourceip</method>
   </persistence>
   <name>TCP-Application-Profile</name>
   <insertXForwardedFor>false</insertXForwardedFor>
   <sslPassthrough>false</sslPassthrough>
   <template>TCP</template>
   <serverSslEnabled>false</serverSslEnabled>
 </applicationProfile>
</loadBalancer>
```

`POST /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add an application profile.

Request:
Body: application/xml

```
<applicationProfile>
 <name>http_application_profile_2</name>
 <insertXForwardedFor>true</insertXForwardedFor>
 <sslPassthrough>true</sslPassthrough>
 <persistence>
   <method>cookie</method>
   <cookieName>JSESSIONID</cookieName>
   <cookieMode>insert</cookieMode>
 </persistence>
```

```
</applicationProfile>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all application profiles on the specified Edge.

Working With a Specific Application Profile

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles/{appProfileID}`

URI Parameters:

appProfileID (required)	Specified application profile.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve an application profile.

Responses:
Status Code: 200
Body: application/xml

```
<applicationProfile>
 <applicationProfileId>applicationProfile-2</applicationProfileId>
 <name>HTTPS-Application-Profile</name>
 <insertXForwardedFor>true</insertXForwardedFor>
 <sslPassthrough>false</sslPassthrough>
 <template>HTTPS</template>
 <serverSslEnabled>false</serverSslEnabled>
</applicationProfile>
```

`PUT /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles/{appProfileID}`

URI Parameters:

appProfileID (required)	Specified application profile.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify an application profile.

Request:

Body: application/xml

```
<applicationProfile>
 <name>http_application_profile_2</name>
 <insertXForwardedFor>true</insertXForwardedFor>
 <sslPassthrough>true</sslPassthrough>
 <persistence>
   <method>cookie</method>
   <cookieName>JSESSIONID</cookieName>
   <cookieMode>insert</cookieMode>
 </persistence>
</applicationProfile>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/applicationprofiles/{appProfileID}`

URI Parameters:

appProfileID (required)	Specified application profile.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete an application profile.

Working With Application Rules

You can write an application rule to directly manipulate and manage IP application traffic.

See *Working With NSX Edge Load Balancer* for **applicationRule** parameter information.

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all application rules.

Responses:
Status Code: 200
Body: application/xml

```
<applicationRule>
 <name>redirection_rule</name>
 <script>
   acl vmware_page url_beg / vmware redirect location https : //www.vmware.com/ if
   vmware_page
 </script>
</applicationRule>
```

`POST /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add an application rule.

Request:
Body: application/xml

```
<applicationRule>
 <name>redirection_rule</name>
 <script>
   acl vmware_page url_beg / vmware redirect location https : //www.vmware.com/ if
   vmware_page
 </script>
</applicationRule>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all application rules.

Working with a Specific Application Rule

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules/{appruleID}`

URI Parameters:

appruleID (required)	Specified application rule.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve an application rule.

Responses:
Status Code: 200
Body: application/xml

```
<applicationRule>
 <name>redirection_rule</name>
 <script>
```

```
   acl vmware_page url_beg / vmware redirect location https : //www.vmware.com/ if
   vmware_page
 </script>
</applicationRule>
```

`PUT /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules/{appruleID}`

URI Parameters:

appruleID (required)	Specified application rule.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify an application rule.

Request:
Body: application/xml

```
<applicationRule>
 <name>redirection_rule</name>
 <script>
   acl vmware_page url_beg / vmware redirect location https : //www.vmware.com/ if
   vmware_page
 </script>
</applicationRule>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/applicationrules/{appruleID}`

URI Parameters:

appruleID (required)	Specified application rule.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete an application rule.

Working With Load Balancer Monitors

You create a service monitor to define health check parameters for a particular type of network traffic. When you associate a service monitor with a pool, the pool members are monitored according to the service monitor parameters.

See *Working With NSX Edge Load Balancer* for **monitor** parameter information.

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/monitors`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve all load balancer monitors.

Responses:
Status Code: 200
Body: application/xml

```
<loadBalancer>
 <monitor>
   <monitorId>monitor-1</monitorId>
   <type>http</type>
   <interval>5</interval>
   <timeout>15</timeout>
   <maxRetries>3</maxRetries>
   <method>GET</method>
   <url>/</url>
   <name>http-monitor</name>
 </monitor>
 <monitor>
   <monitorId>monitor-2</monitorId>
   <type>https</type>
   <interval>5</interval>
   <timeout>15</timeout>
   <maxRetries>3</maxRetries>
   <method>GET</method>
   <url>/</url>
   <name>https-monitor</name>
 </monitor>
 <monitor>
   <monitorId>monitor-3</monitorId>
   <type>tcp</type>
   <interval>5</interval>
   <timeout>15</timeout>
   <maxRetries>3</maxRetries>
   <name>tcp-monitor</name>
 </monitor>
</loadBalancer>
```

`POST /api/4.0/edges/{edgeId}/loadbalancer/config/monitors`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add a load balancer monitor.

Request:
Body: application/xml

```
<monitor>
 <type>http</type>
 <interval>5</interval>
 <timeout>15</timeout>
 <maxRetries>3</maxRetries>
 <method>GET</method>
```

```
 <url>/</url>
 <name>http-monitor-2</name>
</monitor>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/monitors`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all load balancer monitors.

Working With a Specific Load Balancer Monitor

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/monitors/{monitorID}`

URI Parameters:

monitorID (required)	Specified monitor.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve a load balancer monitor.

Responses:
Status Code: 200
Body: application/xml

```
<monitor>
 <type>http</type>
 <interval>5</interval>
 <timeout>15</timeout>
 <maxRetries>3</maxRetries>
 <method>GET</method>
 <url>/</url>
 <name>http-monitor-2</name>
</monitor>
```

`PUT /api/4.0/edges/{edgeId}/loadbalancer/config/monitors/{monitorID}`

URI Parameters:

monitorID (required)	Specified monitor.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify a load balancer monitor.

Request:
Body: application/xml

```
<monitor>
 <type>http</type>
 <interval>15</interval>
 <timeout>25</timeout>
 <maxRetries>3</maxRetries>
 <method>GET</method>
 <url>/</url>
 <name>http-monitor-2</name>
</monitor>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/monitors/{monitorID}`

URI Parameters:

monitorID (required)	Specified monitor.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete a load balancer monitor.

Working With Virtual Servers

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/virtualservers`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all virtual servers.

`POST /api/4.0/edges/{edgeId}/loadbalancer/config/virtualservers`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add a virtual server.

You can add an NSX Edge internal or uplink interface as a virtual server.

See *Working With NSX Edge Load Balancer* for **virtualServer** parameter information.

Request:

Body: application/xml

```
<virtualServer>
 <name>http_vip_2</name>
 <description>http virtualServer 2</description>
 <enabled>true</enabled>
 <ipAddress>10.117.35.172</ipAddress>
 <protocol>http</protocol>
 <port>443,6000-7000</port>
 <connectionLimit>123</connectionLimit>
 <connectionRateLimit>123</connectionRateLimit>
 <applicationProfileId>applicationProfile-1</applicationProfileId>
 <defaultPoolId>pool-1</defaultPoolId>
 <enableServiceInsertion>false</enableServiceInsertion>
 <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
```

DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/virtualservers

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all virtual servers.

Specified virtual server.

GET /api/4.0/edges/{edgeId}/loadbalancer/config/virtualservers/{virtualserverID}

URI Parameters:

virtualserverID (required)	Specified virtual server ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve details for the specified virtual server.

DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/virtualservers/{virtualserverID}

URI Parameters:

virtualserverID (required)	Specified virtual server ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified virtual server.

Working with Server Pools

You can add a server pool to manage and share backend servers flexibly and efficiently. A pool manages load balancer distribution methods and has a service monitor attached to it for health check parameters.

See *Working With NSX Edge Load Balancer* for **pools** parameter information.

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/pools`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Get all server pools on the specified NSX Edge.

Responses:
Status Code: 200
Body: application/xml

```
<loadBalancer>
 <pool>
   <type>slb</type>
   <poolId>pool-1</poolId>
   <name>pool-http</name>
   <description>pool-http</description>
   <algorithm>round-robin</algorithm>
   <transparent>true</transparent>
   <monitorId>monitor-1</monitorId>
   <member>
     <memberId>member-1</memberId>
     <ipAddress>192.168.101.201</ipAddress>
     <weight>1</weight>
     <port>80</port>
     <maxConn>100</maxConn>
     <minConn>10</minConn>
     <condition>enabled</condition>
     <name>m1</name>
   </member>
   <member>
     <memberId>member-2</memberId>
     <ipAddress>192.168.101.202</ipAddress>
     <weight>1</weight>
     <port>80</port>
     <maxConn>100</maxConn>
     <minConn>10</minConn>
     <condition>enabled</condition>
     <name>m2</name>
   </member>
 </pool>
 <pool>
   <type>slb</type>
   <poolId>pool-2</poolId>
   <name>pool-https</name>
   <description>pool-https</description>
   <algorithm>round-robin</algorithm>
   <transparent>false</transparent>
   <monitorId>monitor-2</monitorId>
   <member>
     <memberId>member-11</memberId>
```

```
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m4</name>
  </member>
</pool>
<pool>
  <type>slb</type>
  <poolId>pool-3</poolId>
  <name>pool-tcp</name>
  <description>pool-tcp</description>
  <algorithm>round-robin</algorithm>
  <transparent>true</transparent>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-5</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m5</name>
  </member>
  <member>
    <memberId>member-6</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m6</name>
  </member>
</pool>
<pool>
  <type>slb</type>
  <poolId>pool-4</poolId>
  <name>pool-tcp-snat</name>
  <description>pool-tcp-snat</description>
  <algorithm>round-robin</algorithm>
  <transparent>false</transparent>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-7</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
```

```
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m7</name>
  </member>
  <member>
    <memberId>member-8</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m8</name>
  </member>
 </pool>
</loadBalancer>
```

POST /api/4.0/edges/{edgeId}/loadbalancer/config/pools

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add a load balancer server pool to the Edge.

Method history:

Release	Modification
6.3.0	Method updated. Member **condition** can be set to *drain*.

Request:
Body: application/xml

```
<pool>
 <name>pool-tcp-snat-2</name>
 <description>pool-tcp-snat-2</description>
 <transparent>false</transparent>
 <algorithm>round-robin</algorithm>
 <monitorId>monitor-3</monitorId>
 <member>
   <ipAddress>192.168.101.201</ipAddress>
   <weight>1</weight>
   <port>80</port>
   <minConn>10</minConn>
   <maxConn>100</maxConn>
   <name>m5</name>
   <monitorPort>80</monitorPort>
 </member>
 <member>
   <ipAddress>192.168.101.202</ipAddress>
```

```
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m6</name>
    <monitorPort>80</monitorPort>
  </member>
</pool>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/pools`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all server pools configured on the specified NSX Edge.

Working With a Specific Server Pool

`GET /api/4.0/edges/{edgeId}/loadbalancer/config/pools/{poolID}`

URI Parameters:

`poolID` (required)	Specified pool ID.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve information about the specified server pool.

`PUT /api/4.0/edges/{edgeId}/loadbalancer/config/pools/{poolID}`

URI Parameters:

`poolID` (required)	Specified pool ID.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the specified server pool.

Method history:

Release	Modification
6.3.0	Method updated. Member **condition** can be set to *drain*.

Request:
Body: application/xml

```
<pool>
 <name>pool-tcp-snat-2</name>
 <description>pool-tcp-snat-3</description>
 <transparent>false</transparent>
 <algorithm>round-robin</algorithm>
 <monitorId>monitor-3</monitorId>
 <member>
   <ipAddress>192.168.101.201</ipAddress>
   <weight>1</weight>
   <port>1234</port>
   <minConn>10</minConn>
   <maxConn>100</maxConn>
   <name>m5</name>
   <condition>enabled\disabled</condition>
   <monitorPort>80</monitorPort>
 </member>
 <member>
   <ipAddress>192.168.101.202</ipAddress>
   <weight>1</weight>
   <port>1234</port>
   <minConn>10</minConn>
   <maxConn>100</maxConn>
   <name>m6</name>
   <condition>enabled\disabled</condition>
   <monitorPort>80</monitorPort>
 </member>
</pool>
```

`DELETE /api/4.0/edges/{edgeId}/loadbalancer/config/pools/{poolID}`

URI Parameters:

poolID (required)	Specified pool ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified server pool.

Working With a Specific Load Balancer Member

`POST /api/4.0/edges/{edgeId}/loadbalancer/config/members/{memberID}`

URI Parameters:

memberID (required)	Member ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enable (required)	Set to *true* to enable member, *false* to disable member.

Description:
Update enabled status of the specified member.

Working With Load Balancer Statistics

Retrieves load balancer statistics.

Load Balancer Statistics Parameters

Parameter	Description
virtualServer	Virtual server list.
virtualServerId	Virtual server identifier.
name	Name of the virtual server.
description	Description of virtual server.
ipAddress	IP address that the load balancer is listening on.
status	Virtual server status.
bytesIn	Number of bytes in.
bytesOut	Number of bytes out.
curSessions	Number of current sessions.
httpReqTotal	Total number of HTTP requests received.
httpReqRate	HTTP requests per second over last elapsed second.
httpReqRateMax	Maximum number of HTTP requests per second observed.
maxSession	Number of maximum sessions.
rate	Number of sessions per second over last elapsed second.
rateLimit	Configured limit on new sessions per second.
rateMax	Maximum number of new sessions per second.
totalSession	Total number of sessions.
pool	Pool list.
poolId	Generated pool identifier.
name	Name of the pool.
description	Description of the pool.
status	Pool status.
bytesIn	Number of bytes in.
bytesOut	Number of bytes out.
curSessions	Number of current sessions.
httpReqTotal	Total number of HTTP requests received.
httpReqRate	HTTP requests per second over last elapsed second.
httpReqRateMax	Maximum number of HTTP requests per second observed.
maxSession	Number of maximum sessions.
rate	Number of sessions per second over last elapsed second.

rateLimit	Configured limit on new sessions per second.
rateMax	Maximum number of new sessions per second.
totalSession	Total number of sessions.
member	Pool member list.
memberId	Generated member identifier.
name	Member name.
ipAddress	Member IP address.
groupingObjectId	Member grouping object identifier.
status	Member status.
bytesIn	Number of bytes in.
bytesOut	Number of bytes out.
curSessions	Number of current sessions.
httpReqTotal	Total number of HTTP requests received.
httpReqRate	HTTP requests per second over last elapsed second.
httpReqRateMax	Maximum number of HTTP requests per second observed.
maxSession	Number of maximum sessions.
rate	Number of sessions per second over last elapsed second.
rateLimit	Configured limit on new sessions per second.
rateMax	Maximum number of new sessions per second.
totalSession	Total number of sessions.
timestamp	Timestamp to fetch load balancer statistics.
serverStatus	Load balancer server status.

`GET /api/4.0/edges/{edgeId}/loadbalancer/statistics`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve load balancer statistics.

Responses:
Status Code: 200
Body: application/xml

```
<loadBalancerStatusAndStats>
 <timeStamp>1359722922</timeStamp>
 <pool>
   <poolId>pool-1</poolId>
   <name>pool-http</name>
   <member>
     <memberId>member-1</memberId>
     <name>m1</name>
     <ipAddress>192.168.101.201</ipAddress>
     <status>UP</status>
```

```
    <bytesIn>70771</bytesIn>
    <bytesOut>74619</bytesOut>
    <curSessions>0</curSessions>
    <maxSessions>1</maxSessions>
    <rate>0</rate>
    <rateMax>17</rateMax>
    <totalSessions>142</totalSessions>
  </member>
  <member>
    <memberId>member-2</memberId>
    <name>m2</name>
    <ipAddress>192.168.101.202</ipAddress>
    <status>UP</status>
    <bytesIn>70823</bytesIn>
    <bytesOut>70605</bytesOut>
    <curSessions>0</curSessions>
    <maxSessions>1</maxSessions>
    <rate>0</rate>
    <rateMax>17</rateMax>
    <totalSessions>141</totalSessions>
  </member>
  <status>UP</status>
  <bytesIn>141594</bytesIn>
  <bytesOut>145224</bytesOut>
  <curSessions>0</curSessions>
  <maxSessions>2</maxSessions>
  <rate>0</rate>
  <rateMax>34</rateMax>
  <totalSessions>283</totalSessions>
</pool>
<virtualServer>
  <virtualServerId>virtualServer-9</virtualServerId>
  <name>http_vip</name>
  <ipAddress>10.117.35.172</ipAddress>
  <status>OPEN</status>
  <bytesIn>141594</bytesIn>
  <bytesOut>145224</bytesOut>
  <curSessions>1</curSessions>
  <httpReqTotal>283</httpReqTotal>
  <httpReqRate>0</httpReqRate>
  <httpReqRateMax>34</httpReqRateMax>
  <maxSessions>2</maxSessions>
  <rate>0</rate>
  <rateLimit>0</rateLimit>
  <rateMax>2</rateMax>
  <totalSessions>13</totalSessions>
</virtualServer>
<globalSite>
  <name>BJ site</name>
  <globalSiteId>site-3</globalSiteId>
  <msgSent>3</msgSent>
  <msgRecv>747</msgRecv>
  <msgRate>0</msgRate>
  <dnsReq>0</dnsReq>
  <dnsResolved>0</dnsResolved>
</globalSite>
<globalIp>
  <fqdn>www.company.com</fqdn>
  <globalIpId>gip-3</globalIpId>
  <dnsReq>0</dnsReq>
  <dnsResolved>0</dnsResolved>
  <dnsMiss>0</dnsMiss>
```

```
  </globalIp>
  <globalPool>
    <name>www-primary</name>
    <poolId>pool-1</poolId>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
    <dnsMiss>0</dnsMiss>
    <member>
      <name>10.117.7.110</name>
      <memberId>member-3</memberId>
      <status>up</status>
      <dnsHit>0</dnsHit>
      <cpuUsage>3</cpuUsage>
      <memUsage>91</memUsage>
      <sessions>0</sessions>
      <curConn>14</curConn>
      <sessLimit>0</sessLimit>
      <sessRate>0</sessRate>
      <totalThroughput>0</totalThroughput>
      <packagesPerSec>0</packagesPerSec>
    </member>
  </globalPool>
  <globalPool>
    <name>www-primary</name>
    <poolId>pool-1</poolId>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
    <dnsMiss>0</dnsMiss>
    <member>
      <name>10.117.7.110</name>
      <memberId>member-3</memberId>
      <status>up</status>
      <dnsHit>0</dnsHit>
      <cpuUsage>3</cpuUsage>
      <memUsage>91</memUsage>
      <sessions>0</sessions>
      <curConn>14</curConn>
      <sessLimit>0</sessLimit>
      <sessRate>0</sessRate>
      <totalThroughput>0</totalThroughput>
      <packagesPerSec>0</packagesPerSec>
    </member>
  </globalPool>
</loadBalancerStatusAndStats>
```

Working With Load Balancer Acceleration

`POST /api/4.0/edges/{edgeId}/loadbalancer/acceleration`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`enable` (required)	Set to *true* to enable or *false* to disable load balancer acceleration mode.

Description:
Configure load balancer acceleration mode.

Working with NSX Edge DNS Server Configuration

You can configure external DNS servers to which NSX Edge can relay name resolution requests from clients. NSX Edge will relay client application requests to the DNS servers to fully resolve a network name and cache the response from the servers.

Configure DNS updates the DNS server configuration. The DNS server list allows two addresses – primary and secondary. The default cache size is 16 MB where the minimum can be 1 MB, and the maximum 8196 MB. The default listeners is any, which means listen on all VSE interfaces. If provided, the listener's IP address must be assigned to an internal interface. Logging is disabled by default.

`GET /api/4.0/edges/{edgeId}/dns/config`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve DNS configuration.

`PUT /api/4.0/edges/{edgeId}/dns/config`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure DNS servers.

Request:
Body: application/xml

```
<dns>
 <version>2</version>
 <enabled>true</enabled>
 <cacheSize>128</cacheSize>
 <listeners>
   <ipAddress>192.168.100.1</ipAddress>
   <ipAddress>192.168.100.2</ipAddress>
 </listeners>
 <dnsViews>
   <dnsView>
     <viewId>view-0</viewId>
     <name>vsm-default-view</name>
     <enabled>true</enabled>
     <viewMatch>
       <ipAddress>any</ipAddress>
```

```
      <vnic>any</vnic>
    </viewMatch>
    <recursion>false</recursion>
    <forwarders>
      <ipAddress>10.117.0.1</ipAddress>
    </forwarders>
  </dnsView>
 </dnsViews>
 <logging>
  <enable>true</enable>
  <logLevel>info</logLevel>
 </logging>
</dns>
```

`DELETE /api/4.0/edges/{edgeId}/dns/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete DNS configuration

Get DNS server statistics

`GET /api/4.0/edges/{edgeId}/dns/statistics`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Get DNS server statistics

DNS Server Statistics Parameters

Parameter Name	Parameter Information
requests > total	Indicates all of the incoming requests to the DNS server, including DNS query and other types of requests such as transfers, and updates.
requests > queries	Indicates all of the DNS queries the server received.
requests > total	Indicates all of the responses the server returned to requests. This might be different from the requests.total because some requests might be rejected. total = success + nxrrset + servFail + formErr + nxdomain + others.
responses > success	Indicates all of the successful DNS responses.
responses > nxrrset	Indicates the count of no existent resource record.
responses > servFail	Indicates the count of the SERVFAIL responses.
responses > formErr	Indicates the count of the format error responses.
responses > nxdomain	Indicates the count of no-suhc-domain answer
responses > others	Indicates the count of other types of responses.

Responses:
Status Code: 200
Body: application/xml

```
<dns>
 <stats>
   <timeStamp>2011-10-10 12:12:12</timeStamp>
   <requests>
     <total>120000</total>
     <queries>110000</queries>
   </requests>
   <responses>
     <total>108000</total>
     <success>105000</success>
     <nxrrset>1000</nxrrset>
     <servFail>400</servFail>
     <formErr>300</formErr>
     <nxdomain>1000</nxdomain>
     <others>300</others>
   </responses>
   <cachedDBRRSet>15000</cachedDBRRSet>
 </stats>
</dns>
```

Configure DHCP for NSX Edge

NSX Edge provides DHCP service to bind assigned IP addresses to MAC addresses, helping to prevent MAC spoofing attacks. All virtual machines protected by a NSX Edge can obtain IP addresses dynamically from the NSX Edge DHCP service.

NSX Edge supports IP address pooling and one-to-one static IP address allocation based on the vCenter managed object ID (vmId) and interface ID (interfaceId) of the requesting client.

If either bindings or pools are not included in the PUT call, existing bindings or pools are deleted.

If the NSX Edge autoConfiguration flag and autoConfigureDNS is true, and the primaryNameServer or secondaryNameServer parameters are not specified, NSX Manager applies the DNS settings to the DHCP configuration.

NSX Edge DHCP service adheres to the following rules:
- Listens on the NSX Edge internal interface (non-uplink interface) for DHCP discovery.
- As stated above, vmId specifies the vc-moref-id of the virtual machine, and vnicId specifies the index of the vNic for the requesting client. The hostname is an identification of the binding being created. This hostName is not pushed as the specified host name of the virtual machine.
- By default, all clients use the IP address of the internal interface of the NSX Edge as the default gateway address. To override it, specify **defaultGateway** per binding or per pool. The client's broadcast and subnetMask values are from the internal interface for the container network.
- **leaseTime** can be infinite, or a number of seconds. If not specified, the default lease time is 1 day.
- Logging is disabled by default.
- Setting the parameter **enable** to *true* starts the DHCP service while setting **enable** to *false* stops the service.
- Both **staticBinding** and **ipPools** must be part of the PUT request body. If either bindings or pools are not included in the PUT call, existing bindings or pools are deleted.

DHCP Configuration Parameters

Parameter Name	Parameter Information
enabled	Default is true.
staticBinding	Assign an IP address via DHCP statically rather than dynamically. You can either specify **macAddress** directly, or specify **vmId** and **vnicId**. In case both are specified, only **macAddress** will be used; **vmId** and **vnicId** will be ignored.
staticBinding > macAddress	Optional.
staticBinding > vmId	Optional. The VM must be connected to the specified **vnicId**.
staticBinding > vnicId	Optional. Possible values 0 to 9.
staticBinding > hostname	Optional. Disallow duplicate.
staticBinding > ipAddress	The IP can either belong to a a subnet of one of Edge's vNics or it can be any valid IP address, but the IP must not overlap with any primary/secondary IP addresses associated with any of Edge's vNICs. If the IP does not belong to any Edge vNic subnets, you must ensure that the default gateway and subnetMask are configured via this API call.
ipPool > ipRange	Required. The IP range can either fall entirely within one of the Edge vNIC subnets, or it can be a valid IP range outside any Edge subnets. The IP range, however, cannot contain an IP that is defined as a vNic primary secondary IP. If the range does not fall entirely within one of the Edge vNIC subnets, you must provide correct **subnetMask** and **defaultGateway**.
defaultGateway (staticBinding and ipPool)	Optional. If the ipRange (for ipPool) or assigned IP (for staticBinding) falls entirely within one of the Edge vNIC subnets, **defaultGateway** is set to the primary IP of the vNIC configured with the matching subnet. Otherwise, you must provide the correct gateway IP. If an IP is not provided, the client host may not get default gateway IP from the DHCP server.
subnetMask (staticBinding and ipPool)	Optional. If not specified, and the the ipRange (for ipPool) or assigned IP (for staticBinding) belongs to an Edge vNic subnet, it is defaulted to the subnet mask of this vNic subnet. Otherwise, it is defaulted to a minimum subnet mask which is figured out with the IP range itself, e.g. the mask of range 192.168.5.2-192.168.5.20 is 255.255.255.224. You can edit this range, if required.
domainName (staticBinding and ipPool)	Optional.
primaryNameServer **secondaryNameServer** (staticBinding and ipPool)	Optional. If **autoConfigureDNS** is *true*, the DNS primary/secondary IPs will be generated from DNS service (if configured).
leaseTime (staticBinding and ipPool)	Optional. In seconds, default is *86400*. Valid **leaseTime** is a valid number or *infinite*.
autoConfigureDns (staticBinding and ipPool)	Optional. Default is *true*.
nextServer (staticBinding and ipPool)	Global TFTP server setting. If an IP pool or static binding has a TFTP server configured via **option66** or **option150**, that server will be used instead.

dhcpOptions (staticBinding and ipPool)	Optional.
dhcpOptions > option121 (staticBinding and ipPool)	Add a static route.
dhcpOptions > option121 > destinationSubnet (staticBinding and ipPool)	Destination network, for example 1.1.1.4/30.
dhcpOptions > option121 > router (staticBinding and ipPool)	Router IP address.
dhcpOptions > option66 (staticBinding and ipPool)	Hostname or IP address of a single TFTP server for this IP pool.
dhcpOptions > option67 (staticBinding and ipPool)	Filename to be downloaded from TFTP server.
dhcpOptions > option150 (staticBinding and ipPool)	IP address of TFTP server.
dhcpOptions > option150 > server (staticBinding and ipPool)	Use to specify more than one TFTP server by IP address for this IP Pool.
dhcpOptions > option26 (staticBinding and ipPool)	MTU.
dhcpOptions > other (staticBinding and ipPool)	Add DHCP options other than 26, 66, 67, 121, 150.
dhcpOptions > other > code (staticBinding and ipPool)	Use the DHCP option number only. For example, to specify dhcp option 80, enter *80*.
dhcpOptions > other > value (staticBinding and ipPool)	The DHCP option value, in hex. For example, *2F766172*.
logging	Optional. Logging is disabled by default.
logging > enable	Optional, default is *false*.
logging > logLevel	Optional, default is *info*.

```
GET /api/4.0/edges/{edgeId}/dhcp/config
```

URI Parameters:

`edgeId` `(required)`	Specify the ID of the edge in *edgeId*.

Description:
Get DHCP configuration.

```
PUT /api/4.0/edges/{edgeId}/dhcp/config
```

URI Parameters:

`edgeId` `(required)`	Specify the ID of the edge in *edgeId*.

Description:
Configure DHCP service.

Release	Modification
6.2.3	Method updated. DHCP options added.

Request:

Body: application/xml

```
<dhcp>
 <enabled>true</enabled>
 <staticBindings>
   <staticBinding>
     <macAddress>12:34:56:78:90:AB</macAddress>
     <vmId>vm-111</vmId>
     <vnicId>1</vnicId>
     <hostname>abcd</hostname>
     <ipAddress>192.168.4.2</ipAddress>
     <subnetMask>255.255.255.0</subnetMask>
     <defaultGateway>192.168.4.1</defaultGateway>
     <domainName>eng.vmware.com</domainName>
     <primaryNameServer>192.168.4.1</primaryNameServer>
     <secondaryNameServer>4.2.2.4</secondaryNameServer>
     <leaseTime>infinite</leaseTime>
     <autoConfigureDNS>true</autoConfigureDNS>
   </staticBinding>
 </staticBindings>
 <ipPools>
   <ipPool>
     <ipRange>192.168.4.192-192.168.4.220</ipRange>
     <defaultGateway>192.168.4.1</defaultGateway>
     <subnetMask>255.255.255.0</subnetMask>
     <domainName>eng.vmware.com</domainName>
     <primaryNameServer>192.168.4.1</primaryNameServer>
     <secondaryNameServer>4.2.2.4</secondaryNameServer>
     <leaseTime>3600</leaseTime>
     <autoConfigureDNS>true</autoConfigureDNS>
     <nextServer>11.11.18.105</nextServer>
     <dhcpOptions>
       <option121>
         <staticRoute>
           <destinationSubnet>1.1.1.4/30</destinationSubnet>
           <router>10.10.10.254</router>
         </staticRoute>
         <staticRoute>
           <destinationSubnet>2.2.2.4/30</destinationSubnet>
           <router>10.10.10.210</router>
         </staticRoute>
       </option121>
       <option66>boot.tftp.org</option66>
       <option67>/opt/tftpServer</option67>
       <option150>
         <server>10.10.10.1</server>
         <server>100.100.100.1</server>
       </option150>
       <option26>2048</option26>
       <other>
         <code>80</code>
         <value>2F766172</value>
       </other>
       <other>
         <code>85</code>
         <value>01010101</value>
       </other>
     </dhcpOptions>
   </ipPool>
 </ipPools>
```

```
 <logging>
   <enable>false</enable>
   <logLevel>info</logLevel>
 </logging>
</dhcp>
```

`DELETE /api/4.0/edges/{edgeId}/dhcp/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the DHCP configuration, restoring it to factory default.

Working with DHCP IP Pools

`POST /api/4.0/edges/{edgeId}/dhcp/config/ippools`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add an IP pool to the DHCP configuration. Returns a pool ID within a Location HTTP header.

Method history:

Release	Modification
6.2.3	Method updated. DHCP options added.

Request:
Body: application/xml

```
<ipPool>
 <ipRange>192.168.5.2-192.168.5.20</ipRange>
 <defaultGateway>192.168.5.1</defaultGateway>
 <domainName>eng.vmware.com</domainName>
 <primaryNameServer>1.2.3.4</primaryNameServer>
 <secondaryNameServer>4.3.2.1</secondaryNameServer>
 <leaseTime>3600</leaseTime>
 <autoConfigureDNS>true</autoConfigureDNS>
 <nextServer>11.11.18.105</nextServer>
 <dhcpOptions>
   <option121>
     <staticRoute>
       <destinationSubnet>1.1.1.4/30</destinationSubnet>
       <router>10.10.10.254</router>
     </staticRoute>
     <staticRoute>
       <destinationSubnet>2.2.2.4/30</destinationSubnet>
```

```
      <router>10.10.10.210</router>
    </staticRoute>
  </option121>
  <option66>boot.tftp.org</option66>
  <option67>/opt/tftpServer</option67>
  <option150>
    <server>10.10.10.1</server>
    <server>100.100.100.1</server>
  </option150>
  <option26>2048</option26>
  <other>
    <code>80</code>
    <value>2F766172</value>
  </other>
  <other>
    <code>85</code>
    <value>01010101</value>
  </other>
 </dhcpOptions>
</ipPool>
```

Working with a Specific DHCP IP Pool

`DELETE /api/4.0/edges/{edgeId}/dhcp/config/ippools/{poolID}`

URI Parameters:

poolID (required)	Specified DHCP IP pool
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete a pool specified by pool ID

Working With DHCP Static Bindings

`POST /api/4.0/edges/{edgeId}/dhcp/config/bindings`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Append a static-binding to DHCP config. A static-binding ID is returned within a Location HTTP header.

Method history:

Release	Modification
6.2.3	Method updated. DHCP options added.

Request:
Body: application/xml

```
<staticBinding>
 <vmId></vmId>
 <vnicId></vnicId>
 <hostname></hostname>
 <ipAddress></ipAddress>
 <defaultGateway></defaultGateway>
 <domainName></domainName>
 <primaryNameServer></primaryNameServer>
 <secondaryNameServer></secondaryNameServer>
 <leaseTime></leaseTime>
 <autoConfigureDNS></autoConfigureDNS>
 <nextServer>11.11.18.105</nextServer>
 <dhcpOptions>
   <option121>
     <staticRoute>
       <destinationSubnet>1.1.1.4/30</destinationSubnet>
       <router>10.10.10.254</router>
     </staticRoute>
     <staticRoute>
       <destinationSubnet>2.2.2.4/30</destinationSubnet>
       <router>10.10.10.210</router>
     </staticRoute>
   </option121>
   <option66>boot.tftp.org</option66>
   <option67>/opt/tftpServer</option67>
   <option150>
     <server>10.10.10.1</server>
     <server>100.100.100.1</server>
   </option150>
   <option26>2048</option26>
   <other>
     <code>80</code>
     <value>2F766172</value>
   </other>
   <other>
     <code>85</code>
     <value>01010101</value>
   </other>
 </dhcpOptions>
</staticBinding>
```

Working with a Specific DHCP Static Binding

`DELETE /api/4.0/edges/{edgeId}/dhcp/config/bindings/{bindingID}`

URI Parameters:

`bindingID` (required)	Specified static-binding ID
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:

Delete the static-binding by ID

Working With DHCP Relays

Dynamic Host Configuration Protocol (DHCP) relay enables you to leverage your existing DHCP infrastructure from within NSX without any interruption to the IP address management in your environment. DHCP messages are relayed from virtual machine(s) to the designated DHCP server(s) in the physical world. This enables IP addresses within NSX to continue to be in synch with IP addresses in other environments.

DHCP configuration is applied on the logical router port and can list several DHCP servers. Requests are sent to all listed servers. While relaying the DHCP request from the client, the relay adds a Gateway IP Address to the request. The external DHCP server uses this gateway address to match a pool and allocate an IP address for the request. The gateway address must belong to a subnet of the NSX port on which the relay is running.

You can specify a different DHCP server for each logical switch and can configure multiple DHCP servers on each logical router to provide support for multiple IP domains.

NOTE DHCP relay does not support overlapping IP address space (option 82).

DHCP Relay and DHCP service cannot run on a port/vNic at the same time. If a relay agent is configured on a port, a DHCP pool cannot be configured on the subnet(s) of this port.

Parameter	Description	Comments
relay	You can configure ipPool, static-binding and relay at the same time if	
there is not any overlap on vnic.		
relayServer	There must be at least one external server.	Required.
groupingObjectId	A list of dhcp server IP addresses.	

```
There can be multiple sever group objects, the maximum groupObject is 4 the maxium number of server IP
addresses is 16 |
```

ipAddress |Supports both IP address and FQDN| **fqdn** |Specify the IP of the fqdn, and add a Firewall rule to allow the response from the server represented by the fqdn such as: src - the IP; dest - any; service - udp:67:any.| **relayAgents** |There must be at least one relay agent.|Required. **vnicIndex** | No default. Specify the vNic that proxy the dhcp request. | Required. **giAddress> | Defaults to the vNic primary address. Only one giAddress allowed. | Optional.

`GET /api/4.0/edges/{edgeId}/dhcp/config/relay`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Query DHCP relay

Responses:
Status Code: 200
Body: application/xml

```
<relay>
 <relayServer>
   <groupingObjectId>IPset1</groupingObjectId>
   <groupingObjectId>IPset2</groupingObjectId>
 </relayServer>
 <relayAgents>
   <relayAgent>
     <vnicIndex>1</vnicIndex>
     <giAddress>
       192.168.1.254</giAddress>
   </relayAgent>
   <relayAgent>
     <vnicIndex>3</vnicIndex>
     <giAddress>192.168.3.254</giAddress>
   </relayAgent>
 </relayAgents>
</relay>
```

PUT /api/4.0/edges/{edgeId}/dhcp/config/relay

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure DHCP relay

Request:
Body: application/xml

```
<relay>
 <relayServer>
   <groupingObjectId>IPset1</groupingObjectId>
   <groupingObjectId>IPset2</groupingObjectId>
   <ipAddress>10.117.35.202</ipAddress>
   <fqdn>www.dhcpserver</fqdn>
 </relayServer>
 <relayAgents>
   <relayAgent>
     <giAddress>192.168.1.254</giAddress>
   </relayAgent>
   <relayAgent>
     <vnicIndex>3</vnicIndex>
     <giAddress>192.168.3.254</giAddress>
   </relayAgent>
 </relayAgents>
</relay>
```

DELETE /api/4.0/edges/{edgeId}/dhcp/config/relay

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Delete DHCP relay configuration

Working With DHCP Leases

`GET /api/4.0/edges/{edgeId}/dhcp/leaseInfo`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Get DHCP lease information.

Working with NSX Edge High Availability

High Availability (HA) ensures that a NSX Edge appliance is always available on your virtualized network. You can enable HA either when installing NSX Edge or on an installed NSX Edge instance.

If a single appliance is associated with NSX Edge, the appliance configuration is cloned for the standby appliance. If two appliances are associated with NSX Edge and one of them is deployed, this REST call deploys the remaining appliance and push HA configuration to both.

HA relies on an internal interface. If an internal interface does not exist, this call will not deploy the secondary appliance, or push HA config to appliance. The enabling of HA will be done once an available internal interface is added. If the PUT call includes an empty xml or enabled=false, it acts as a DELETE call.

`GET /api/4.0/edges/{edgeId}/highavailability/config`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Get high availability configuration.

Responses:
Status Code: 200
Body: application/xml

```
<highAvailability>
 <vnic>1</vnic>
 <ipAddresses>
   <ipAddress>192.168.10.1/30</ipAddress>
   <ipAddress>192.168.10.2/30</ipAddress>
 </ipAddresses>
 <declareDeadTime>6</declareDeadTime>
</highAvailability>
```

`PUT /api/4.0/edges/{edgeId}/highavailability/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Configure high availability.| **ipAddress** - Optional. A pair of ipAddresses with /30 subnet mandatory, one for each appliance. If provided, they must NOT overlap with any subnet defined on the Edge vNics. If not specified, a pair of IPs will be picked up from reserved subnet, 169.254.0.0/16. **declareDeadTime** Optional. The default is 6 seconds. **enabled** - Optional. The default is set to true. The enabled flag will cause the HA appliance to be deployed or destroyed.

Request:

Body: application/xml

```
<highAvailability>
 <ipAddresses>
   <ipAddress>192.168.10.1/30</ipAddress>
   <ipAddress>192.168.10.2/30</ipAddress>
 </ipAddresses>
 <declareDeadTime>6</declareDeadTime>
 <enabled>true</enabled>
</highAvailability>
```

`DELETE /api/4.0/edges/{edgeId}/highavailability/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

NSX Manager deletes the standby appliance and removes the HA config from the active appliance.You can also delete the HA configuration by using a PUT call with empty xml or with false.

Working With Remote Syslog Server on NSX Edge

You can configure one or two remote syslog servers. Edge events and logs related to firewall events that flow from Edge appliances are sent to the syslog servers

`GET /api/4.0/edges/{edgeId}/syslog/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve syslog servers information.

`PUT /api/4.0/edges/{edgeId}/syslog/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Configure syslog servers.

Request:
Body: application/xml

```
<syslog>
 <protocol>udp</protocol>
 <serverAddresses>
   <ipAddress>1.1.1.1</ipAddress>
   <ipAddress>1.1.1.2</ipAddress>
 </serverAddresses>
</syslog>
```

`DELETE /api/4.0/edges/{edgeId}/syslog/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete syslog servers.

Working With SSL VPN

With SSL VPN-Plus, remote users can connect securely to private networks behind a NSX Edge gateway. Remote users can access servers and applications in the private networks.

`GET /api/4.0/edges/{edgeId}/sslvpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve SSL VPN details.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the entire SSL VPN configuration to the specified NSX Edge in a single call.

`POST /api/4.0/edges/{edgeId}/sslvpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enableService (required)	Set to *true* to enable, *false* to disable.

Description:
Enable or disable SSL VPN on the NSX Edge appliance.

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the SSL VPN configurations on the Edge.

Working With SSL VPN Server

`GET /api/4.0/edges/{edgeId}/sslvpn/config/server`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve server settings.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/server`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update server settings.

Request:
Body: application/xml

```
<serverSettings>
 <serverAddresses>
   <ipAddress>10.112.243.109</ipAddress>
 </serverAddresses>
 <port>443</port>
 <certificateId>certificate-1</certificateId>
 <cipherList>
   <cipher>AES128-SHA</cipher>
   <cipher>AES256-SHA</cipher>
 </cipherList>
</serverSettings>
```

Working With Private Networks

You can use a private network to expose to remote users over SSL VPN tunnel.

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all private network profiles in the SSL VPN instance.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update all private network configs of NSX Edge with the given list of private network configs. If the config is present, it is updated; otherwise, a new private network config is created. Existing configs not included in the call body are deleted.

`POST /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure a private network.

Request:
Body: application/xml

```
<privateNetwork>
 <description></description>
 <network></network>
 <sendOverTunnel>
   <ports></ports>
   <optimize></optimize>
 </sendOverTunnel>
 <enabled></enabled>
</privateNetwork>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all private networks from the SSL VPN instance.

Working With a Specific Private Network

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks/{networkID}`

URI Parameters:

networkID (required)	Specified private network
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the specified private network in the SSL VPN service.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks/{networkID}`

URI Parameters:

networkID (required)	Specified private network
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the specified private network in the SSL VPN service.

Request:
Body: application/xml

```
<privateNetwork>
 <description></description>
 <network></network>
 <sendOverTunnel>
   <ports></ports>
   <optimize></optimize>
 </sendOverTunnel>
 <enabled></enabled>
</privateNetwork>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/privatenetworks/{networkID}`

URI Parameters:

networkID (required)	Specified private network
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Delete private network

Working With IP Pools for SSL VPN

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all IP pools configured on SSL VPN.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update all IP pools with the given list of pools. If the pool is present, it is updated; otherwise, a new pool is created. Existing pools not in the body are deleted.

Request:
Body: application/xml

```
<ipAddressPool>
 <description></description>
 <ipRange></ipRange>
 <netmask></netmask>
 <gateway></gateway>
 <primaryDns></primaryDns>
 <secondaryDns></secondaryDns>
 <dnsSuffix></dnsSuffix>
 <winsServer></winsServer>
 <enabled></enabled>
</ipAddressPool>
```

`POST /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Create an IP pool.

Request:

Body: application/xml

```
<ipAddressPool>
 <description>description</description>
 <ipRange>10.112.243.11-10.112.243.57</ipRange>
 <netmask>255.0.0.0</netmask>
 <gateway>192.168.1.1</gateway>
 <primaryDns>192.168.10.1</primaryDns>
 <secondaryDns>4.2.2.2</secondaryDns>
 <dnsSuffix></dnsSuffix>
 <winsServer>10.112.243.201</winsServer>
 <enabled>true</enabled>
</ipAddressPool>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all IP pools configured on SSL VPN

Working With a Specific IP Pool for SSL VPN

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools/{ippoolID}`

URI Parameters:

ippoolID (required)	Specified IP pool ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve details of specified IP pool.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools/{ippoolID}`

URI Parameters:

ippoolID (required)	Specified IP pool ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update specified IP pool.

Request:
Body: application/xml

```
<ipAddressPool>
 <description>description</description>
 <ipRange>10.112.243.11-10.112.243.57</ipRange>
 <netmask>255.0.0.0</netmask>
 <gateway>192.168.1.1</gateway>
 <primaryDns>192.168.10.1</primaryDns>
 <secondaryDns>4.2.2.2</secondaryDns>
 <dnsSuffix></dnsSuffix>
 <winsServer>10.112.243.201</winsServer>
 <enabled>true</enabled>
</ipAddressPool>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/ippools/{ippoolID}`

URI Parameters:

ippoolID (required)	Specified IP pool ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified IP pool.

Working With Network Extension Client Parameters

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/clientconfig`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve client configuration.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/clientconfig`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Set advanced parameters for full access client configurations, such as whether client should auto-reconnect in case of network failures or network unavailability, or whether the client should be uninstalled after logout.

Request:
Body: application/xml

```
<clientConfiguration>
 <autoReconnect>true</autoReconnect>
 <fullTunnel>
```

```
    <excludeLocalSubnets>false</excludeLocalSubnets>
    <gatewayIp>10.112.243.11</gatewayIp>
  </fullTunnel>
  <upgradeNotification>false</upgradeNotification>
</clientConfiguration>
```

Working With SSL VPN Client Installation Packages

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve information about all installation packages.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update all installation packages with the given list. If the package is present, it is updated; otherwise a new installation package is created. Existing packages not included in the body are deleted.

`POST /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Creates setup executables (installers) for full access network clients. These setup binaries are later downloaded by remote clients and installed on their systems. The primary parameters needed to configure this setup are hostname of the gateway, and its port and a profile name which is shown to the user to identify this connection. The Administrator can also set other parameters such as whether to automatically start the application on windows login, or hide the system tray icon.

Request:
Body: application/xml

```
<clientInstallPackage>
 <profileName></profileName>
 <gatewayList>
   <gateway>
     <hostName></hostName>
     <port></port>
   </gateway>
 </gatewayList>
```

```
  <startClientOnLogon></startClientOnLogon>
  <hideSystrayIcon></hideSystrayIcon>
  <rememberPassword></rememberPassword>
  <silentModeOperation></silentModeOperation>
  <silentModeInstallation></silentModeInstallation>
  <hideNetworkAdaptor></hideNetworkAdaptor>
  <createDesktopIcon></createDesktopIcon>
  <enforceServerSecurityCertValidation></enforceServerSecurityCertValidation>
  <createLinuxClient></createLinuxClient>
  <createMacClient></createMacClient>
  <description></description>
  <enabled></enabled>
</clientInstallPackage>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all client installation packages.

Working With a Specific SSL VPN Client Installation Package

`GET /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages/{packageID}`

URI Parameters:

packageID (required)	Specified installation package ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Get information about the specified installation package.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages/{packageID}`

URI Parameters:

packageID (required)	Specified installation package ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify the specified installation package.

Request:
Body: application/xml

```
<clientInstallPackage>
 <profileName></profileName>
 <gatewayList>
   <gateway>
     <hostName></hostName>
     <port></port>
   </gateway>
 </gatewayList>
 <startClientOnLogon></startClientOnLogon>
 <hideSystrayIcon></hideSystrayIcon>
 <rememberPassword></rememberPassword>
 <silentModeOperation></silentModeOperation>
 <silentModeInstallation></silentModeInstallation>
 <hideNetworkAdaptor></hideNetworkAdaptor>
 <createDesktopIcon></createDesktopIcon>
 <enforceServerSecurityCertValidation></enforceServerSecurityCertValidation>
 <createLinuxClient></createLinuxClient>
 <createMacClient></createMacClient>
 <description></description>
 <enabled></enabled>
</clientInstallPackage>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/client/networkextension/installpackages/{packageID}`

URI Parameters:

packageID (required)	Specified installation package ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified installation package.

Working With Portal Layout

`GET /api/4.0/edges/{edgeId}/sslvpn/config/layout`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve layout configuration.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/layout`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the portal layout.

Request:
Body: application/xml

```
<layout>
 <portalTitle></portalTitle>
 <companyName></companyName>
 <logoBackgroundColor></logoBackgroundColor>
 <titleColor></titleColor>
 <topFrameColor></topFrameColor>
 <menuBarColor></menuBarColor>
 <rowAlternativeColor></rowAlternativeColor>
 <bodyColor></bodyColor>
 <rowColor></rowColor>
</layout>
```

Working With Image Files for SSL VPN

`POST /api/4.0/edges/{edgeId}/sslvpn/config/layout/images/{imageType}`

URI Parameters:

`imageType` (required)	Type of image to upload. Choice of *portallogo*, *phatbanner*, *connecticon*, *disconnecticon*, *desktopicon*, or *erroricon*.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Upload images for use with SSL VPN portal and client.

You can upload a logo to use in the SSL VPN portal, and a banner and icons to use in the SSL VPN client.

You must upload the image files using the form-data content-type. Consult the documentation for your REST client for instructions.

Do not set other Content-type headers in your request, for example, *Content-type: application/xml*.

When you upload a file as form-data, you must provide a **key** and a **value** for the file. See the table below for the form-data **key** to use for each image type. The **value** is the path to the image file.

Image Type	form-data key	Image format requirements
portallogo	layoutFile	n/a
phatbanner	banner	bmp
connecticon	icon	ico
disconnecticon	icon	ico
erroricon	icon	ico
desktopicon	icon	ico

Example using curl

```
/usr/bin/curl -v -k -i -F layoutFile=@/tmp/portalLogo.jpg -H 'Authorization: Basic YWRtaW46ZGXXXXXXXX==' https://192.168.110.42/api/4.0/edges/edge-3/sslvpn/config/layout/images/portallogo
```

Working With Portal Users

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/auth/localserver/users`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Modify the portal user specified in the request body.

Request:
Body: application/xml

```
<user>
 <userId></userId>
 <password></password>
 <firstName></firstName>
 <lastName></lastName>
 <description></description>
 <disableUserAccount></disableUserAccount>
 <passwordNeverExpires></passwordNeverExpires>
 <allowChangePassword>
   <changePasswordOnNextLogin></changePasswordOnNextLogin>
 </allowChangePassword>
</user>
```

`POST /api/4.0/edges/{edgeId}/sslvpn/config/auth/localserver/users`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Add a new portal user.

Request:
Body: application/xml

```
<user>
 <userId></userId>
 <password></password>
 <firstName></firstName>
 <lastName></lastName>
```

```
  <description></description>
  <disableUserAccount></disableUserAccount>
  <passwordNeverExpires></passwordNeverExpires>
  <allowChangePassword>
    <changePasswordOnNextLogin></changePasswordOnNextLogin>
  </allowChangePassword>
</user>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/auth/localserver/users`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all users on the specifed SSL VPN instance

Working With a Specific Portal User

`GET /api/4.0/edges/{edgeId}/sslvpn/config/auth/localserver/users/{userID}`

URI Parameters:

userID	User ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Get information about the specified user.

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/auth/localserver/users/{userID}`

URI Parameters:

userID	User ID.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the specified user.

Working With Authentication Settings

`GET /api/4.0/edges/{edgeId}/sslvpn/config/auth/settings`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve information about authentication settings.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/auth/settings`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update authentication settings for remote users. Specify username/password authentication, active directory, ldap, radius, client certificate based authentication.

Request:
Body: application/xml

```
<authenticationConfig>
 <passwordAuthentication>
 <authenticationTimeout></authenticationTimeout>
   <primaryAuthServers>
     <com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
       <ip></ip>
       <port></port>
       <timeOut></timeOut>
       <enableSsl></enableSsl>
       <searchBase></searchBase>
       <bindDomainName></bindDomainName>
       <bindPassword></bindPassword>
       <loginAttributeName></loginAttributeName>
       <searchFilter></searchFilter>
       <enabled></enabled>
     </com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
     <com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
       <ip></ip>
       <port></port>
       <timeOut></timeOut>
       <secret></secret>
       <nasIp></nasIp>
       <retryCount></retryCount>
     </com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
     <com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
       <enabled></enabled>
       <passwordPolicy>
         <minLength></minLength>
         <maxLength></maxLength>
         <minAlphabets></minAlphabets>
         <minDigits></minDigits>
         <minSpecialChar></minSpecialChar>
         <allowUserIdWithinPassword></allowUserIdWithinPassword>
         <passwordLifeTime></passwordLifeTime>
         <expiryNotification></expiryNotification>
       </passwordPolicy>
       <accountLockoutPolicy>
         <retryCount></retryCount>
         <retryDuration></retryDuration>
         <lockoutDuration></lockoutDuration>
       </accountLockoutPolicy>
     </com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
```

```
    <com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
      <timeOut></timeOut>
      <sourceIp></sourceIp>
    </com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
  </primaryAuthServers>
  <secondaryAuthServer>
    <com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
      <ip>1.1.1.1</ip>
      <port>90</port>
      <timeOut>20</timeOut>
      <enableSsl>false</enableSsl>
      <searchBase>searchbasevalue</searchBase>
      <bindDomainName>binddnvalue</bindDomainName>
      <bindPassword>password</bindPassword>
      <loginAttributeName>cain</loginAttributeName>
      <searchFilter>found</searchFilter>
      <terminateSessionOnAuthFails>false</terminateSessionOnAuthFails>
      <enabled>true</enabled>
    </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
  </secondaryAuthServer>
 </passwordAuthentication>
</authenticationConfig>
```

Working With the RSA Config File

`POST /api/4.0/edges/{edgeId}/sslvpn/config/auth/settings/rsaconfigfile`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Upload RSA config file (See "Generate the Authentication Manager Configuration File" section of the RSA Authentication Manager Administrator's guide for instructions on how to configure and download the RSA config file from RSA Authentication Manager).

SSL VPN Advanced Configuration

`GET /api/4.0/edges/{edgeId}/sslvpn/config/advancedconfig`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve SSL VPN advanced configuration.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/advancedconfig`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update SSL VPN advanced configuration.

Request:
Body: application/xml

```
<advancedConfig>
 <enableCompression></enableCompression>
 <forceVirtualKeyboard></forceVirtualKeyboard>
 <preventMultipleLogon></preventMultipleLogon>
 <randomizeVirtualkeys></randomizeVirtualkeys>
 <timeout>
   <forcedTimeout></forcedTimeout>
   <sessionIdleTimeout></sessionIdleTimeout>
 </timeout>
 <clientNotification></clientNotification>
 <enableLogging></enableLogging>
</advancedConfig>
```

Working with Logon and Logoff Scripts for SSL VPN

`GET /api/4.0/edges/{edgeId}/sslvpn/config/script`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all script configurations.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/script`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update all script configurations with the given list of configurations. If the config is present, its is updated; otherwise, a new config is created. Existing configs not included in the body are deleted.

Request:
Body: application/xml

```
<logonLogoffScript>
 <scriptFileId></scriptFileId>
 <type></type>
```

```
 <description></description>
 <enabled></enabled>
</logonLogoffScript>
```

`POST /api/4.0/edges/{edgeId}/sslvpn/config/script`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure parameters associated with the uploaded script file.

Request:
Body: application/xml

```
<logonLogoffScript>
 <scriptFileId></scriptFileId>
 <type></type>
 <description></description>
 <enabled></enabled>
</logonLogoffScript>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/script`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete all script configurations

Working With Uploaded Script Files

`GET /api/4.0/edges/{edgeId}/sslvpn/config/script/{fileID}`

URI Parameters:

fileID (required)	Specified script file.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve parameters associated with the specified script file.

`PUT /api/4.0/edges/{edgeId}/sslvpn/config/script/{fileID}`

URI Parameters:

fileID (required)	Specified script file.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update parameters associated with the specified script file.

Request:
Body: application/xml

```
<logonLogoffScript>
 <scriptFileId></scriptFileId>
 <type></type>
 <description></description>
 <enabled></enabled>
</logonLogoffScript>
```

`DELETE /api/4.0/edges/{edgeId}/sslvpn/config/script/{fileID}`

URI Parameters:

fileID (required)	Specified script file.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete script parameters.

Uploading Script Files for SSL VPN

`POST /api/4.0/edges/{edgeId}/sslvpn/config/script/file/`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
You can add multiple login or logoff scripts. For example, you can bind a login script for starting Internet Explorer with gmail.com. When the remote user logs in to the SSL client, Internet Explorer opens up gmail.com. This method returns a *scriptFileId* which can be used to update parameters associated with the script file.

You must upload the script files using the form-data content-type. Consult the documentation for your REST client for instructions.

Do not set other Content-type headers in your request, for example, *Content-type: application/xml*.

When you upload a file as form-data, you must provide a **key** and a **value** for the file. The **key** is *file*, and the **value** is the location of the script file.

Example using curl

```
/usr/bin/curl -v -k -i -F file=@/tmp/script.sh -H 'Authorization: Basic YWRtaW46ZGXXXXXXXX==' 
https://192.168.110.42/api/4.0/edges/edge-3/sslvpn/config/script/file/
```

Working with SSL VPN Users

`PUT /api/4.0/edges/{edgeId}/sslvpn/auth/localusers/users`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update all users with the given list of users. If the user is present, it is updated. Otherwise, and new user is created. Existing users not included in the body are deleted.

Request:
Body: application/xml

```
<users>
 <user>
   <userId></userId>
   <password></password>
   <firstName></firstName>
   <lastName></lastName>
   <description></description>
   <disableUserAccount></disableUserAccount>
   <passwordNeverExpires></passwordNeverExpires>
   <allowChangePassword>
     <changePasswordOnNextLogin></changePasswordOnNextLogin>
   </allowChangePassword>
 </user>
</users>
```

Working With Active Client Sessions

`GET /api/4.0/edges/{edgeId}/sslvpn/activesessions`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve a list of active clients for the SSL VPN session.

Working With a Specific Active Client Session

`DELETE /api/4.0/edges/{edgeId}/sslvpn/activesessions/{sessionID}`

URI Parameters:

`sessionID` (required)	Specified client session.
`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Disconnect an active client.

Working With SSL VPN Dashboard Statistics

`GET /api/4.0/edges/{edgeId}/statistics/dashboard/sslvpn`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`interval`	Specify a range; can be *1 - 60* minutes, or *oneDay*, *oneWeek*, *oneMonth*, or *oneYear*. Default is *60* minutes.

Description:
Retrieve SSL VPN statistics on the specified NSX Edge.

Working With Tunnel Traffic Dashboard Statistics

`GET /api/4.0/edges/{edgeId}/statistics/dashboard/ipsec`

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

`interval`	Specify a range; can be *1 - 60* minutes, or *oneDay*, *oneWeek*, *oneMonth*, or *oneYear*. Default is *60* minutes.

Description:
Retrieve tunnel traffic statistics for specified time interval.

Working With Interface Dashboard Statistics

```
GET /api/4.0/edges/{edgeId}/statistics/dashboard/interface
```

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

interval (required)	Specify a start and end time range in seconds.

Description:

Retrieves dashboard statistics between the specified start and end times. When start and end time are not specified, all statistics since the Edge deployed are displayed. When no end time is specified, the current Edge Manager time is set as endTime. Each record has the stats of 5 minutes granularity.

Working With Interface Statistics

```
GET /api/4.0/edges/{edgeId}/statistics/interfaces
```

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve interface statistics.

Working With Uplink Interface Statistics

```
GET /api/4.0/edges/{edgeId}/statistics/interfaces/uplink
```

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve uplink interface statistics.

Working With Internal Interface Statistics

```
GET /api/4.0/edges/{edgeId}/statistics/interfaces/internal
```

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Retrieve internal interface statistics.

Working with L2 VPN

L2 VPN allows you to configure a tunnel between two sites. VMs can move between the sites and stay on the same subnet, enabling you to extend your datacenter. An NSX Edge at one site can provide all services to VMs on the other site.

`GET /api/4.0/edges/{edgeId}/l2vpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the current L2VPN configuration for NSX Edge.

Responses:
Status Code: 200
Body: application/xml

```
<l2Vpn>
 <version>4</version>
 <enabled>true</enabled>
 <logging>
   <enable>false</enable>
   <logLevel>info</logLevel>
 </logging>
 <l2VpnSites>
   <l2VpnSite>
     <client>
       <configuration>
         <serverAddress>192.168.15.23</serverAddress>
         <serverPort>443</serverPort>
         <caCertificate>certificate-4</caCertificate>
         <vnic>10</vnic>
         <egressOptimization>
           <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
         </egressOptimization>
         <encryptionAlgorithm>AES128-SHA</encryptionAlgorithm>
       </configuration>
       <l2VpnUser>
         <userId>apple</userId>
       </l2VpnUser>
       <proxySetting>
         <type>https</type>
         <address>10.112.243.202</address>
         <port>443</port>
         <userName>root</userName>
       </proxySetting>
     </client>
   </l2VpnSite>
 </l2VpnSites>
</l2Vpn>
```

```
PUT /api/4.0/edges/{edgeId}/l2vpn/config
```

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Configure L2VPN for server or client.

You first enable the L2 VPN service on the NSX Edge instance and then configure a server and a client.

L2 VPN Parameters

Parameter	Description	Comments
enabled	Whether L2 VPN is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *True*.
logging	L2 VPN logging setting.	Optional. Disable by default.
logging > enable	Whether logging is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *False*.
logging > logLevel	Logging level.	Optional. Options are: EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFO, and DEBUG. Default is *INFO*.
listenerIp	IP of external interface on which L2VPN service listens to.	Required.
listenerPort	Port on which L2VPN service listens to.	Optional. Default is 443.
encryptionAlgorithm	Encryption algorithm for communication between the server and the client.	Mandatory. Supported ciphers are *RC4-MD5*, *AES128-SHA*, *AES256-SHA*, *DES-CBC3-SHA*, *AES128-GCM-SHA256*, and *NULL-MD5*.
serverCertificate	Select the certificate to be bound to L2 VPN server.	Optional. If not specified server will use its default (self-signed) certificate.

Peer Site Parameters

Parameter	Description	Comments
peerSites	To connect multiple sites to the L2 VPN server.	Required. Minimum one peer site must be configured to enable L2 VPN server service.
name	Unique name for the site getting configured.	Required.
description	Description about the site.	Optional.
l2VpnUser	Every peer site must have a user configuration.	Required.
l2VpnUser > userId	L2 VPN user ID.	Required.
l2VpnUser > password	Password for L2 VPN user.	Required.
vnics	List of vNICs to be stretched over the tunnel.	Required.

vnics > index	Select the virtual machine NIC to bind to the IP address.	Required.
egressOptimization > gatewayIpAddress	The gateway IP addresses for which the traffic should be locally routed or for which traffic is to be blocked over the tunnel.	Optional.
enabled	Whether the peer site is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *True*.

Example to configure L2 VPN for Client

```
<l2Vpn>
 <enabled>true</enabled>
 <logging>
   <enable>false</enable>
   <logLevel>info</logLevel>
 </logging>
 <l2VpnSites>
   <l2VpnSite>
     <client>
       <configuration>
         <serverAddress>192.168.15.23</serverAddress>
         <serverPort>443</serverPort>
         <vnic>10</vnic>
         <encryptionAlgorithm>AES128-SHA</encryptionAlgorithm>
         <caCertificate>certificate-4</caCertificate>
         <egressOptimization>
           <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
         </egressOptimization>
       </configuration>
       <proxySetting>
         <type>https</type>
         <address>10.112.243.202</address>
         <port>443</port>
         <userName>root</userName>
         <password>java123</password>
       </proxySetting>
       <l2VpnUser>
         <userId>apple</userId>
         <password>apple</password>
       </l2VpnUser>
     </client>
   </l2VpnSite>
 </l2VpnSites>
</l2Vpn>
```

Example to configure L2 VPN for Server

```
<l2Vpn>
 <enabled>true</enabled>
 <logging>
   <enable>false</enable>
   <logLevel>info</logLevel>
 </logging>
 <l2VpnSites>
   <l2VpnSite>
     <server>
       <configuration>
         <listenerIp>192.168.15.65</listenerIp>
```

```
        <listenerPort>443</listenerPort>
        <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
        <peerSites>
          <peerSite>
            <name>PeerSite1</name>
            <description>description</description>
            <l2VpnUser>
              <userId>apple</userId>
              <password>apple</password>
            </l2VpnUser>
            <vnics>
              <index>10</index>
            </vnics>
            <egressOptimization>
              <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
            </egressOptimization>
            <enabled>true</enabled>
          </peerSite>
        </peerSites>
      </configuration>
    </server>
  </l2VpnSite>
 </l2VpnSites>
</l2Vpn>
```

Request:
Body: application/xml

```
<l2Vpn>
 <enabled>true</enabled>
 <logging>
   <enable>false</enable>
   <logLevel>info</logLevel>
 </logging>
 <l2VpnSites>
   <l2VpnSite>
     <server>
       <configuration>
         <listenerIp>192.168.15.65</listenerIp>
         <listenerPort>443</listenerPort>
         <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
         <peerSites>
           <peerSite>
             <name>PeerSite1</name>
             <description>description</description>
             <l2VpnUser>
               <userId>apple</userId>
               <password>apple</password>
             </l2VpnUser>
             <vnics>
               <index>10</index>
             </vnics>
             <egressOptimization>
               <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
             </egressOptimization>
             <enabled>true</enabled>
           </peerSite>
         </peerSites>
```

```
      </configuration>
    </server>
  </l2VpnSite>
 </l2VpnSites>
</l2Vpn>
```

`POST /api/4.0/edges/{edgeId}/l2vpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

enableService (required)	Enable (*true*) or disable (*false*) L2 VPN.

Description:
Enable or disable L2 VPN service.

`DELETE /api/4.0/edges/{edgeId}/l2vpn/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the L2 VPN configuration.

Working With L2 VPN Statistics

`GET /api/4.0/edges/{edgeId}/l2vpn/config/statistics`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve L2 VPN statistics, which has information such as tunnel status, sent bytes, received bytes for the specified Edge.

Responses:
Status Code: 200
Body: application/xml

```
<l2vpnStatusAndStats>
 <timeStamp>1403285853</timeStamp>
 <siteStats>
   <l2vpnStats>
     <name>site-1</name>
     <tunnelStatus>up</tunnelStatus>
     <establishedDate>1403285827</establishedDate>
```

```
    <txBytesFromLocalSubnet>478</txBytesFromLocalSubnet>
    <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
    <rxBytesOnLocalSubnet>42</rxBytesOnLocalSubnet>
  </l2vpnStats>
  <l2vpnStats>
    <name>site-2</name>
    <tunnelStatus>up</tunnelStatus>
    <establishedDate>1403285829</establishedDate>
    <txBytesFromLocalSubnet>408</txBytesFromLocalSubnet>
    <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
    <rxBytesOnLocalSubnet>450</rxBytesOnLocalSubnet>
  </l2vpnStats>
 </siteStats>
</l2vpnStatusAndStats>
```

Working With IPsec VPN

NSX Edge supports site-to-site IPsec VPN between an NSX Edge instance and remote sites. NSX Edge supports certificate authentication, preshared key mode, IP unicast traffic, and no dynamic routing protocol between the NSX Edge instance and remote VPN routers. Behind each remote VPN router, you can configure multiple subnets to connect to the internal network behind an NSX Edge through IPsec tunnels. These subnets and the internal network behind a NSX Edge must have address ranges that do not overlap.

You can deploy an NSX Edge agent behind a NAT device. In this deployment, the NAT device translates the VPN address of an NSX Edge instance to a publicly accessible address facing the Internet. Remote VPN routers use this public address to access the NSX Edge instance.

You can place remote VPN routers behind a NAT device as well. You must provide the VPN native address and the VPN Gateway ID to set up the tunnel. On both ends, static one-to-one NAT is required for the VPN address.

You can have a maximum of 64 tunnels across a maximum of 10 sites.

IPsec VPN Parameters

Parameter	Description	Comments
logging	IPsec VPN logging setting.	Optional. Disable by default.
logging > logLevel	Logging level.	Optional. Options are: EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFO, and DEBUG. Default is *INFO*.
logging > enable	Whether logging is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *False*.
psk	Indicates that the secret key shared between NSX Edge and the peer site is to be used for authentication.	Optional. Required only when peerIp is specified as *Any* in site configuration.
encryptionAlgorithm	Encryption algorithm for communication.	Mandatory. Supported ciphers are *AES*, *AES256*, *Triple DES*, and *AES-GCM*.
serviceCertificate	Select the certificate to be bound to IPsec VPN server.	Optional. Required when *x.509* certificate mode is selected.
caCertificate	List of CA certificates.	Optional.
crlCertificate	List of CRL certificates.	Optional.

enablePfs	Perfect Forward Secrecy (PFS) ensures that each new cryptographic key is unrelated to any previous key.	Optional. Boolean. Options are *True* or *False*. Default is *True*.
authenticationMode	Select authentication mode as *psk* or *x.509*.	Required.
site	To connect multiple sites to the IPsec VPN server.	Required. Minimum one site must be configured to enable IPsec VPN server service.
site > enabled	Whether site is enabled.	Optional. Boolean. Options are *True* or *False*. Default is *True*.
site > name	Unique name for the site getting configured.	Optional.
site > description	Description about the site.	Optional.
localId	Type the IP address of the NSX Edge instance.	
localIp	Type the IP address of the local endpoint.	
localSubnets	Type the subnets to share between the sites.	
peerId	Type the peer ID to uniquely identify the peer site. This should be a Distinguishing Name (DN) if authentication mode is *x.509*.	
peerIp > index	Select the virtual machine NIC to bind to the IP address. This can be a IPv4 address such as 11.0.0.3.	
egressOptimization	The gateway IP addresses for which the traffic should be locally routed or for which traffic is to be blocked over the tunnel.	Optional.
dhGroup	In Diffie-Hellman (DH) Group, select the cryptography scheme that will allow the peer site and the NSX Edge to establish a shared secret over an insecure communications channel.	Optional. *dh2* is selected by default.
extension	Default value is *securelocaltrafficbyip=192.168.11.1*. To disable this extension, replace with securelocaltrafficbyip=0.	

```
GET /api/4.0/edges/{edgeId}/ipsec/config
```

URI Parameters:

`edgeId` (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve IPsec configuration.

Responses:
Status Code: 200
Body: application/xml

```
<ipsec>
 <enabled>true</enabled>
 <logging>
   <logLevel>debug</logLevel>
   <enable>true</enable>
 </logging>
 <global>
   <psk>hello123</psk>
   <serviceCertificate>certificate-4</serviceCertificate>
   <caCertificates>
     <caCertificate>certificate-3</caCertificate>
   </caCertificates>
   <crlCertificates>
     <crlCertificate>crl-1</crlCertificate>
   </crlCertificates>
 </global>
 <sites>
   <site>
     <enabled>true</enabled>
     <name>VPN to edge-pa-1</name>
     <description>psk VPN to edge-pa-1 192.168.11.0/24 == 192.168.1.0/24</description>
     <localId>11.0.0.11</localId>
     <localIp>11.0.0.11</localIp>
     <peerId>11.0.0.1</peerId>
     <peerIp>any</peerIp>
     <encryptionAlgorithm>aes256</encryptionAlgorithm>
     <authenticationMode>psk</authenticationMode>
     <enablePfs>true</enablePfs>
     <dhGroup>dh2</dhGroup>
     <localSubnets>
       <subnet>192.168.11.0/24</subnet>
     </localSubnets>
     <peerSubnets>
       <subnet>192.168.1.0/24</subnet>
     </peerSubnets>
   </site>
   <site>
     <name>VPN to edge-right</name>
     <description>certificate VPN to edge-right 192.168.22.0/24 == 192.168.2.0/24</description>
     <localId>11.0.0.12</localId>
     <localIp>11.0.0.12</localIp>
     <peerId>C=CN, ST=BJ, L=BJ, O=VMware, OU=DEV, CN=Right</peerId>
     <peerIp>11.0.0.2</peerIp>
     <encryptionAlgorithm>aes256</encryptionAlgorithm>
     <authenticationMode>x.509</authenticationMode>
     <enablePfs>true</enablePfs>
     <dhGroup>dh2</dhGroup>
     <localSubnets>
       <subnet>192.168.22.0/24</subnet>
     </localSubnets>
     <peerSubnets>
       <subnet>192.168.2.0/24</subnet>
     </peerSubnets>
   </site>
 </sites>
</ipsec>
```

PUT /api/4.0/edges/{edgeId}/ipsec/config

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update IPsec VPN configuration.

Request:
Body: application/xml

```
<ipsec>
 <enabled>true</enabled>
   <logging>
     <logLevel>debug</logLevel>
     <enable>true</enable>
   </logging>
 <global>
   <psk>hello123</psk>
   <serviceCertificate>certificate-4</serviceCertificate>
   <caCertificates>
     <caCertificate>certificate-3</caCertificate>
   </caCertificates>
   <crlCertificates>
     <crlCertificate>crl-1</crlCertificate>
   </crlCertificates>
 </global>
 <sites>
   <site>
     <enabled>true</enabled>
     <name>VPN to edge-pa-1</name>
     <description>psk VPN to edge-pa-1 192.168.11.0/24 == 192.168.1.0/24</description>
     <localId>11.0.0.11</localId>
     <localIp>11.0.0.11</localIp>
     <peerId>11.0.0.1</peerId>
     <peerIp>any</peerIp>
     <encryptionAlgorithm>aes256</encryptionAlgorithm>
     <authenticationMode>psk</authenticationMode>
     <enablePfs>true</enablePfs>
     <dhGroup>dh2</dhGroup>
     <localSubnets>
       <subnet>192.168.11.0/24</subnet>
     </localSubnets>
     <peerSubnets>
       <subnet>192.168.1.0/24</subnet>
     </peerSubnets>
   </site>
   <site>
     <name>VPN to edge-right</name>
     <description>certificate VPN to edge-right 192.168.22.0/24 == 192.168.2.0/24</description>
     <localId>11.0.0.12</localId>
     <localIp>11.0.0.12</localIp>
     <peerId>C=CN, ST=BJ, L=BJ, O=VMware, OU=DEV, CN=Right</peerId>
     <peerIp>11.0.0.2</peerIp>
     <encryptionAlgorithm>aes256</encryptionAlgorithm>
     <authenticationMode>x.509</authenticationMode>
     <enablePfs>true</enablePfs>
     <dhGroup>dh2</dhGroup>
     <localSubnets>
       <subnet>192.168.22.0/24</subnet>
```

```
    </localSubnets>
    <peerSubnets>
      <subnet>192.168.2.0/24</subnet>
    </peerSubnets>
    <extension>securelocaltrafficbyip=192.168.11.1</extension>
  </site>
 </sites>
</ipsec>
```

`DELETE /api/4.0/edges/{edgeId}/ipsec/config`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the IPsec configuration.

Working With IPsec Statistics

`GET /api/4.0/edges/{edgeId}/ipsec/statistics`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve IPsec statistics.

Responses:
Status Code: 200
Body: application/xml

```
<ipsecStatusAndStats>
 <siteStatistics>
  <ikeStatus>
   <channelStatus>up</channelStatus>
   <channelState>STATE_MAIN_I4 (ISAKMP SA established)</channelState>
   <lastInformationalMessage></lastInformationalMessage>
   <localIpAddress>10.0.0.12</localIpAddress>
   <peerId>11.0.0.12</peerId>
   <peerIpAddress>10.0.0.2</peerIpAddress>
  </ikeStatus>
  <tunnelStats>
   <tunnelStatus>up</tunnelStatus>
   <tunnelState>STATE_QUICK_I2 (sent QI2, IPsec SA established)</tunnelState>
   <lastInformationalMessage></lastInformationalMessage>
   <localSubnet>192.168.2.0/24</localSubnet>
   <peerSubnet>192.168.22.0/24</peerSubnet>
  </tunnelStats>
 </siteStatistics>
 <siteStatistics>
  <ikeStatus>
```

```
    <channelStatus>up</channelStatus>
    <channelState>STATE_MAIN_I4 (ISAKMP SA established)</channelState>
    <lastInformationalMessage></lastInformationalMessage>
    <localIpAddress>10.0.0.11</localIpAddress>
    <peerId>11.0.0.11</peerId>
    <peerIpAddress>10.0.0.1</peerIpAddress>
   </ikeStatus>
   <tunnelStats>
    <tunnelStatus>up</tunnelStatus>
    <tunnelState>STATE_QUICK_I2 (sent QI2, IPsec SA established)</tunnelState>
    <lastInformationalMessage></lastInformationalMessage>
    <localSubnet>192.168.1.0/24</localSubnet>
    <peerSubnet>192.168.11.0/24</peerSubnet>
   </tunnelStats>
  </siteStatistics>
  <timeStamp>1325766138</timeStamp>
</ipsecStatusAndStats>
```

Automatic Configuration of Firewall Rules

If autoConfiguration is enabled, firewall rules are automatically created to allow control traffic. Rules to allow data traffic are not created. For example, if you are using IPsec VPN, and **autoConfiguration** is *true*, firewall rules will automatically be configured to allow IKE traffic. However, you will need to add additional rules to allow the data traffic for the IPsec tunnel. If HA is enabled, firewall rules are always created, even if **autoConfiguration** is *false*, otherwise both HA appliances will become active.

`GET /api/4.0/edges/{edgeId}/autoconfiguration`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the auto configuration settings for the NSX Edge.

`PUT /api/4.0/edges/{edgeId}/autoconfiguration`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the auto configuration settings for the NSX Edge.

Request:
Body: application/xml

```
<autoConfiguration>
 <enabled></enabled>
 <rulePriority></rulePriority>
</autoConfiguration>
```

Working With NSX Edge Appliance Configuration

See *Working with NSX Edge* for additional parameters used to configure appliances.

When you create an NSX Edge, you define parameters that determine how the appliance is deployed, including resourcePoolId, dataStoreId, hostId, and VmFolderId. After the appliance is deployed, these deployment details may change, and the appliance parameters are updated to reflect the current, live location.

You can view the originally configured paramters by using the configuredResourcePool, configuredDataStore, configuredHost, and configuredVmFolder parameters.

You can trigger a high availability failover on the active NSX Edge appliance by changing the haAdminState value to *down* as part of appliance configuration for an NSX Edge. The haAdminState parameter determines whether or not an NSX Edge appliance is participating in high availability. Both appliances in an NSX Edge high availability configuration normally have an haAdminState of *up*. When you set the haAdminState of the active appliance to be *down*, it stops participating in high availability, and informs the standby appliance of its status. The standby appliance becomes active immediately.

Parameter	Description	Comments
highAvailabilityIndex	Index number of the appliance	Read only.
haAdminState	Indicates whether appliance is participating in high availability.	If the active appliance **haAdminState** is set to *down*, it stops participating in HA, and informs the standby appliance of its status. The standby appliance becomes active immediately.
configuredResourcePool > id	ID of resource pool on which NSX Edge was originally deployed.	Read only.
configuredResourcePool > name	Name of resource pool on which NSX Edge was originally deployed.	Read only.
configuredResourcePool > isValid	True if resource pool on which NSX Edge was originally deployed currently exists.	Read only. *true* or *false*.
configuredDataStore > id	ID of data store on which NSX Edge was originally deployed.	Read only.
configuredDataStore > name	Name of data store on which NSX Edge was originally deployed.	Read only.
configuredDataStore > isValid	True if resource pool on which NSX Edge was originally deployed currently exists.	Read only. *true* or *false*.
configuredHost > id	ID of host on which NSX Edge was originally deployed.	Read only.
configuredHost > name	Name of host on which NSX Edge was originally deployed.	Read only.
configuredHost > isValid	True if resource pool on which NSX Edge was originally deployed currently exists.	Read only. *true* or *false*.
configuredVmFolder > id	ID of folder in which NSX Edge was originally deployed.	Read only.
configuredVmFolder > name	Name of folder in which NSX Edge was originally deployed.	Read only.

configuredVmFolder > isValid	True if resource pool on which NSX Edge was originally deployed currently exists.	Read only. *true* or *false*.

`GET /api/4.0/edges/{edgeId}/appliances`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve appliance configuration.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState**, **configuredResourcePool**, **configuredDataStore**, **configuredHost**, **configuredVmFolder** parameters added.

Responses:
Status Code: 200
Body: application/xml

```
<appliances>
 <applianceSize>compact</applianceSize>
 <appliance>
   <highAvailabilityIndex>0</highAvailabilityIndex>
   <haAdminState>up</haAdminState>
   <vcUuid>502e2dd9-3df7-4820-6925-29832a1c0b79</vcUuid>
   <vmId>vm-417</vmId>
   <haAdminState>up</haAdminState>
   <resourcePoolId>domain-c41</resourcePoolId>
   <resourcePoolName>Management &amp; Edge Cluster</resourcePoolName>
   <datastoreId>datastore-29</datastoreId>
   <datastoreName>ds-site-a-nfs01</datastoreName>
   <hostId>host-202</hostId>
   <hostName>esxmgt-01a.corp.local</hostName>
   <vmFolderId>group-v242</vmFolderId>
   <vmFolderName>NSX Edges</vmFolderName>
   <vmHostname>Perimeter-Gateway-02-0</vmHostname>
   <vmName>Perimeter-Gateway-02-0</vmName>
   <deployed>true</deployed>
   <cpuReservation>
     <reservation>1000</reservation>
   </cpuReservation>
   <memoryReservation>
     <reservation>512</reservation>
   </memoryReservation>
   <edgeId>edge-5</edgeId>
   <configuredResourcePool>
     <id>domain-c41</id>
     <name>Management &amp; Edge Cluster</name>
     <isValid>true</isValid>
   </configuredResourcePool>
   <configuredDataStore>
     <id>datastore-29</id>
```

```
      <name>ds-site-a-nfs01</name>
      <isValid>true</isValid>
    </configuredDataStore>
    <configuredHost>
      <id>host-202</id>
      <name>esxmgt-01a.corp.local</name>
      <isValid>true</isValid>
    </configuredHost>
    <configuredVmFolder>
      <id>group-v242</id>
      <name>NSX Edges</name>
      <isValid>true</isValid>
    </configuredVmFolder>
  </appliance>
  <appliance>
    <highAvailabilityIndex>1</highAvailabilityIndex>
    <haAdminState>up</haAdminState>
    <vcUuid>502e3ebf-02cb-dcd2-9701-91db1e0e3bd8</vcUuid>
    <vmId>vm-429</vmId>
    <haAdminState>up</haAdminState>
    <resourcePoolId>domain-c41</resourcePoolId>
    <resourcePoolName>Management &amp;amp; Edge Cluster</resourcePoolName>
    <datastoreId>datastore-29</datastoreId>
    <datastoreName>ds-site-a-nfs01</datastoreName>
    <hostId>host-202</hostId>
    <hostName>esxmgt-01a.corp.local</hostName>
    <vmFolderId>group-v242</vmFolderId>
    <vmFolderName>NSX Edges</vmFolderName>
    <vmHostname>Perimeter-Gateway-02-1</vmHostname>
    <vmName>Perimeter-Gateway-02-1</vmName>
    <deployed>true</deployed>
    <edgeId>edge-5</edgeId>
    <configuredResourcePool>
      <id>domain-c41</id>
      <name>Management &amp;amp; Edge Cluster</name>
      <isValid>true</isValid>
    </configuredResourcePool>
    <configuredDataStore>
      <id>datastore-29</id>
      <name>ds-site-a-nfs01</name>
      <isValid>true</isValid>
    </configuredDataStore>
    <configuredHost>
      <id>host-202</id>
      <name>esxmgt-01a.corp.local</name>
      <isValid>true</isValid>
    </configuredHost>
    <configuredVmFolder>
      <id>group-v242</id>
      <name>NSX Edges</name>
      <isValid>true</isValid>
    </configuredVmFolder>
  </appliance>
  <deployAppliances>true</deployAppliances>
</appliances>
```

`PUT /api/4.0/edges/{edgeId}/appliances`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
You can retrieve the configuration of both appliances by using the GET call and replace the size, resource pool, datastore, and custom parameters of the appliances by using a PUT call. If there were two appliances earlier and you PUT only one appliance, the other appliance is deleted.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState** parameter added.

`POST /api/4.0/edges/{edgeId}/appliances`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

size (required)	set to compact|large|xlarge

Description:
Change the size of both appliances.

Working With NSX Edge Appliance Configuration by Index

`GET /api/4.0/edges/{edgeId}/appliances/{haIndex}`

URI Parameters:

haIndex (required)	Specified appliance HA index
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the configuration of the specified appliance.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState**, **configuredResourcePool**, **configuredDataStore**, **configuredHost**, **configuredVmFolder** parameters added.

`PUT /api/4.0/edges/{edgeId}/appliances/{haIndex}`

URI Parameters:

haIndex (required)	Specified appliance HA index
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update the configuration of the specified appliance.

Method history:

Release	Modification
6.2.3	Method updated. **haAdminState** parameter added.

Request:
Body: application/xml

```
<appliance>
 <haAdminState>up</haAdminState>
 <resourcePoolId>domain-c41</resourcePoolId>
 <datastoreId>datastore-29</datastoreId>
 <hostId>host-203</hostId>
 <vmFolderId>group-v242</vmFolderId>
 <cpuReservation>
   <limit>-1</limit>
   <reservation>1000</reservation>
 </cpuReservation>
 <memoryReservation>
   <limit>-1</limit>
   <reservation>512</reservation>
 </memoryReservation>
 <edgeId>edge-3</edgeId>
</appliance>
```

`POST /api/4.0/edges/{edgeId}/appliances/{haIndex}`

URI Parameters:

`haIndex` `(required)`	Specified appliance HA index
`edgeId` `(required)`	Specify the ID of the edge in *edgeId*.

Query Parameters:

`action` `(optional)`	Used to send CLI Commands to the Edge Gw. Use action=execute to send the command

Description:
Used to send CLI Commands to the Edge Gw. To use CLI commands you also need to add an additional Accept Header with type text\plain, as well as the query parameter action=execute

Request:
Body: application/xml

```
<cliCmd>
 <cmdStr>show ip ospf neighbours</cmdStr>
</cliCmd>
```

`DELETE /api/4.0/edges/{edgeId}/appliances/{haIndex}`

URI Parameters:

haIndex (required)	Specified appliance HA index
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete the appliance

Working With Edge Services Gateway Interfaces

See *Working with NSX Edge* for descriptions of parameters used to configure Edge Service Gateway interfaces.

`GET /api/4.0/edges/{edgeId}/vnics`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all interfaces for the specified Edge Services Gateway.

`POST /api/4.0/edges/{edgeId}/vnics`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

action (required)	Set to *patch*.

Description:
Add an interface or sub interface.

Request:
Body: application/xml

```
<vnics>
 <vnic>
   <name></name>
   <addressGroups>
     <addressGroup>
       <primaryAddress></primaryAddress>
       <secondaryAddresses>
         <ipAddress></ipAddress>
       </secondaryAddresses>
       <subnetMask></subnetMask>
     </addressGroup>
   </addressGroups>
   <mtu></mtu>
```

```
    <type></type>
    <index></index>
    <portgroupId></portgroupId>
    <portgroupName></portgroupName>
    <macAddress>
      <edgeVmHaIndex></edgeVmHaIndex>
      <value></value>
    </macAddress>
    <fenceParameter>
      <key></key>
      <value></value>
    </fenceParameter>
    <enableProxyArp></enableProxyArp>
    <enableSendRedirects></enableSendRedirects>
    <enableBridgeMode></enableBridgeMode>
    <isConnected></isConnected>
    <inShapingPolicy>
      <averageBandwidth></averageBandwidth>
      <peakBandwidth></peakBandwidth>
      <burstSize></burstSize>
      <enabled></enabled>
      <inherited></inherited>
    </inShapingPolicy>
    <outShapingPolicy>
      <averageBandwidth></averageBandwidth>
      <peakBandwidth></peakBandwidth>
      <burstSize></burstSize>
      <enabled></enabled>
      <inherited></inherited>
    </outShapingPolicy>
  </vnic>
</vnics>
```

Working With a Specific Edge Services Gateway Interface

See *Working with NSX Edge* for descriptions of parameters used to configure Edge Service Gateway interfaces.

`GET /api/4.0/edges/{edgeId}/vnics/{index}`

URI Parameters:

index (required)	Specified interface
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the specified interface.

`PUT /api/4.0/edges/{edgeId}/vnics/{index}`

URI Parameters:

index (required)	Specified interface
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:

Update the specified interface.

Request:
Body: application/xml

```
<vnics>
 <vnic>
   <name></name>
   <addressGroups>
     <addressGroup>
       <primaryAddress></primaryAddress>
       <secondaryAddresses>
         <ipAddress></ipAddress>
       </secondaryAddresses>
       <subnetMask></subnetMask>
     </addressGroup>
   </addressGroups>
   <mtu></mtu>
   <type></type>
   <index></index>
   <portgroupId></portgroupId>
   <portgroupName></portgroupName>
   <macAddress>
     <edgeVmHaIndex></edgeVmHaIndex>
     <value></value>
   </macAddress>
   <fenceParameter>
     <key></key>
     <value></value>
   </fenceParameter>
   <enableProxyArp></enableProxyArp>
   <enableSendRedirects></enableSendRedirects>
   <enableBridgeMode></enableBridgeMode>
   <isConnected></isConnected>
   <inShapingPolicy>
     <averageBandwidth></averageBandwidth>
     <peakBandwidth></peakBandwidth>
     <burstSize></burstSize>
     <enabled></enabled>
     <inherited></inherited>
   </inShapingPolicy>
   <outShapingPolicy>
     <averageBandwidth></averageBandwidth>
     <peakBandwidth></peakBandwidth>
     <burstSize></burstSize>
     <enabled></enabled>
     <inherited></inherited>
   </outShapingPolicy>
 </vnic>
</vnics>
```

`DELETE /api/4.0/edges/{edgeId}/vnics/{index}`

URI Parameters:

index (required)	Specified interface
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete interface

Working with Logical Router HA (Management) Interface

`GET /api/4.0/edges/{edgeId}/mgmtinterface`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve the management interface configuration for the logical router.

`PUT /api/4.0/edges/{edgeId}/mgmtinterface`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Configure high availability (management) interface for logical (distributed) router. See *Working with NSX Edge* for descriptions of parameters used to configure the logical router HA interface.

Request:
Body: application/xml

```
<mgmtInterface>
 <addressGroups>
   <addressGroup>
     <primaryAddress></primaryAddress>
     <subnetMask></subnetMask>
   </addressGroup>
 </addressGroups>
 <mtu></mtu>
 <connectedToId></connectedToId>
</mgmtInterface>
```

Working With Logical Router Interfaces

Configure interfaces for logical (distributed) router. See *Working with NSX Edge* for descriptions of parameters used to configure the logical router interfaces.

`GET /api/4.0/edges/{edgeId}/interfaces`

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve all interfaces on the logical router.

POST /api/4.0/edges/{edgeId}/interfaces

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

action (required)	Set to *patch*.

Description:
Add interfaces for a logical router.

Request:
Body: application/xml

```
<interfaces>
 <interface>
   <name></name>
   <addressGroups>
     <addressGroup>
       <primaryAddress></primaryAddress>
       <subnetMask></subnetMask>
     </addressGroup>
   </addressGroups>
   <mtu></mtu>
   <type></type>
   <isConnected></isConnected>
   <connectedToId></connectedToId>
 </interface>
</interfaces>
```

DELETE /api/4.0/edges/{edgeId}/interfaces

URI Parameters:

edgeId (required)	Specify the ID of the edge in *edgeId*.

Query Parameters:

index	Specify index of interface to delete (e.g. ?index=&index;=). If no indices specified, all interfaces are deleted

Description:
Delete all interfaces on the logical router.

Working With a Specific Logical Router Interface

`GET /api/4.0/edges/{edgeId}/interfaces/{index}`

URI Parameters:

index (required)	Specified router interface.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Retrieve information about the specified logical router interface.

`PUT /api/4.0/edges/{edgeId}/interfaces/{index}`

URI Parameters:

index (required)	Specified router interface.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Update interface configuration for the specified logical router interface.

`DELETE /api/4.0/edges/{edgeId}/interfaces/{index}`

URI Parameters:

index (required)	Specified router interface.
edgeId (required)	Specify the ID of the edge in *edgeId*.

Description:
Delete interface configuration and reset to factory default.

Configuring Edge Services in Async Mode

You can configure NSX Edge to work in async mode. In the async mode, accepted commands return an Accepted status and a taskId. To know the status of the task, you can check the status of that taskId. The advantage of the async mode is that APIs are returned very fast and actions like vm deployment, reboots, publish to NSX Edge appliance, are done behind the scene under the taskId . To configure async mode, ?async=true at the end of any 4.0 service configuration URL for POST, PUT, and DELETE calls. Without async mode, the location header in HTTP response has the resource ID whereas in async mode, location header has the job ID.

The job status response includes the job status (*SUCCESS*, *FAILED*, *QUEUED*, *RUNNING*, *ROLLBACK*), URI of the resource, and ID of the resource.

`GET /api/4.0/edges/jobs`

Query Parameters:

status (optional)	**status** can be *all* or *active*.

Description:

Retrieve NSX Edge job status.

Responses:
Status Code: 200
Body: application/xml

```
<edgeJobs>
 <edgeJob>
   <jobId>jobdata-917</jobId>
   <status>COMPLETED</status>
   <result>
     <key>edgeId</key>
     <value>edge-4</value>
   </result>
 </edgeJob>
 <edgeJob>
   <jobId>jobdata-915</jobId>
   <status>COMPLETED</status>
   <result>
     <key>edgeId</key>
     <value>edge-4</value>
   </result>
 </edgeJob>
</edgeJobs>
```

Working With a Specific Edge Job Status

`GET /api/4.0/edges/jobs/{jobId}`

URI Parameters:

jobId (required)	Job ID

Description:
Retrieve job status for the specified job.

Responses:
Status Code: 200
Body: application/xml

```
<edgeJob>
 <jobId>jobdata-2128</jobId>
 <message>Deploying vShield Edge Virtual Machine TestEdge11-0</message>
 <status>RUNNING</status>
 <result>
   <key>ResultURI</key>
   <value>/api/4.0/edges/edge-4</value>
 </result>
 <result>
   <key>edgeId</key>
   <value>edge-4</value>
```

```
 </result>
</edgeJob>
```

Working with NSX Edge Configuration Publishing

Working With NSX Edge Tuning Configuration

Starting in 6.2.3 you can configure default values for NSX Edge configuration parameters, including publishing and health check timeouts, and CPU and memory reservation, which are applicable to all NSX Edges. The values for the tuning configuration parameters have been set to sensible defaults and may not require any changes. However, based on datacenter capacity and requirements, you can change the default CPU and memory resource reservation percentages using this API. This percentage is applied across all Edge VM Sizes {COMPACT, LARGE, QUADLARGE, XLARGE}. The default values are:

- 100% for CPU reservation
- 100% for Memory reservation
- 1000 MHz per CPU

Name	Comments
lockUpdatesOnEdge	Default value is false. Serialize specific Edge operations related to DHCP and vnic configuration to avoid concurrency errors when too many configuration change requests arrive at the same time.
aggregatePublishing	Default value is true (enabled). Speed up configuration change publishing to the NSX Edge by aggregating over the configuration versions.
edgeVMHealthCheckIntervalInMin	Default value for time interval between NSX Edge VM's health check is 0, where NSX Manager manages the interval based on the number of NSX Edge VM's. A positive integer value overrides the default behavior.
healthCheckCommandTimeoutInMs	Default timeout value for health check command is 120000.
maxParallelVixCallsForHealthCheck	The maximum concurrent health check calls that can be made for NSX Edge VM's based on VIX communication channel is 25.
publishingTimeoutInMs	The timeout value to publish a configuration change on NSX Edge appliance. Default is 1200000 (20 minutes).
edgeVCpuReservationPercentage	Integer value [0-100], specifying the CPU reservation percentage which will be applied to the NSX Edge appliance. To disable this resource reservation, enter 0.
edgeMemoryReservationPercentage	integer value [0-100], specifying the memory reservation percentage which will be applied to the NSX Edge appliance. To disable this resource reservation, enter 0.
megaHertzPerVCpu	integer value specifying the megahertz per each vCPU (1000, 1500, 2000)

```
GET /api/4.0/edgePublish/tuningConfiguration
```

Description:
Retrieve the NSX Edge tuning configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<tuningConfiguration>
 <lockUpdatesOnEdge>false</lockUpdatesOnEdge>
 <aggregatePublishing>true</aggregatePublishing>
 <edgeVMHealthCheckIntervalInMin>0</edgeVMHealthCheckIntervalInMin>
 <healthCheckCommandTimeoutInMs>120000</healthCheckCommandTimeoutInMs>
 <maxParallelVixCallsForHealthCheck>25</maxParallelVixCallsForHealthCheck>
 <publishingTimeoutInMs>1200000</publishingTimeoutInMs>
 <edgeVCpuReservationPercentage>100</edgeVCpuReservationPercentage>
 <edgeMemoryReservationPercentage>100</edgeMemoryReservationPercentage>
 <megaHertzPerVCpu>1000</megaHertzPerVCpu>
</tuningConfiguration>
```

PUT /api/4.0/edgePublish/tuningConfiguration

Description:
Update the NSX Edge tuning configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<tuningConfiguration>
 <lockUpdatesOnEdge>false</lockUpdatesOnEdge>
 <aggregatePublishing>true</aggregatePublishing>
 <edgeVMHealthCheckIntervalInMin>0</edgeVMHealthCheckIntervalInMin>
 <healthCheckCommandTimeoutInMs>120000</healthCheckCommandTimeoutInMs>
 <maxParallelVixCallsForHealthCheck>25</maxParallelVixCallsForHealthCheck>
 <publishingTimeoutInMs>1200000</publishingTimeoutInMs>
 <edgeVCpuReservationPercentage>0</edgeVCpuReservationPercentage>
 <edgeMemoryReservationPercentage>0</edgeMemoryReservationPercentage>
 <megaHertzPerVCpu>1000</megaHertzPerVCpu>
</tuningConfiguration>
```

Working with Certificates

NSX Edge supports self-signed certificates, certificates signed by a Certification Authority (CA), and certificates generated and signed by a CA.

Working with Certificates and Certificate Chains

`POST /api/2.0/services/truststore/certificate`

Query Parameters:

`csrId` (required)	Specify the ID of a CSR.

Description:
Import a certificate or a certificate chain against a certificate signing request.

Request:
Body: application/xml

```
<trustObject>
 <pemEncoding></pemEncoding>
</trustObject>
```

Working With Certificates on a Specific Scope

`GET /api/2.0/services/truststore/certificate/scope/{scopeId}`

URI Parameters:

`scopeId` (required)	Scope ID. Specify the ID of an NSX Edge, e.g. *edge-5*, or *globalroot-0*.

Description:
Retrieve all certificates on the specified scope.

Responses:
Status Code: 200
Body: application/xml

```
<csrs>
 <csr></csr>
 <csr></csr>
</csrs>
```

Working With Self-Signed Certificates

`POST /api/2.0/services/truststore/certificate/{scopeId}`

URI Parameters:

`scopeId` (required)	Scope ID. Specify the ID of an NSX Edge, e.g. *edge-5*, or *globalroot-0*.

Description:
Create a single certificate

You can create a certificate for a specific NSX Edge, or if you specify a scope of *globalroot-0* you can create a global certificate in NSX Manager which is available to all NSX Edges.

Request:
Body: application/xml

```
<trustObject>
 <pemEncoding></pemEncoding>
 <privateKey></privateKey>
 <passphrase></passphrase>
</trustObject>
```

Working With a Specific Certificate

`GET /api/2.0/services/truststore/certificate/{certificateId}`

URI Parameters:

`certificateId` (required)	Certificate ID

Description:
Retrieve the certificate object specified by ID. If the ID specifies a chain, multiple certificate objects are retrieved.

`DELETE /api/2.0/services/truststore/certificate/{certificateId}`

URI Parameters:

`certificateId` (required)	Certificate ID

Description:
Delete the specified certificate.

Working with Certificate Signing Requests (CSRs)

`POST /api/2.0/services/truststore/csr/{scopeId}`

URI Parameters:

scopeId (required)	Specified scope ID

Description:
Create a certificate signing request (CSR).

Request:
Body: application/xml

```
<csr>
 <subject>
   <attribute>
     <key>CN</key>
     <value>VSM</value>
   </attribute>
   <attribute>
     <key>O</key>
     <value>VMware</value>
   </attribute>
   <attribute>
     <key>OU</key>
     <value>IN</value>
   </attribute>
   <attribute>
     <key>C</key>
     <value>IN</value>
   </attribute>
 </subject>
 <algorithm>RSA</algorithm>
 <keySize>1024</keySize>
</csr>
```

Working With Self-Signed Certificate for CSR

`GET /api/2.0/services/truststore/csr/{csrId}`

URI Parameters:

csrId (required)	CSR ID

Description:
Retrieve the specified certificate signing request (CSR).

`PUT /api/2.0/services/truststore/csr/{csrId}`

URI Parameters:

csrId (required)	CSR ID

Query Parameters:

noOfDays (required)	Number of days the certificate is valid.

Description:
Create a self-signed certificate for CSR.

Working With Certificate Signing Requests on a Specific Scope

`GET /api/2.0/services/truststore/csr/scope/{scopeId}`

URI Parameters:

scopeId (required)	Specified scope.

Description:
Retrieve certificate signing requests (CSR) on the specified scope.

Working With Certificate Revocation Lists on a Specific Scope

`POST /api/2.0/services/truststore/crl/{scopeId}`

URI Parameters:

scopeId (required)	Specified scope.

Description:
Create a certificate revocation list (CRL) on the specified scope.

Request:
Body: application/xml

```
<trustObject>
 <pemEncoding></pemEncoding>
</trustObject>
```

Working with CRL Certificates in a Specific Scope

`GET /api/2.0/services/truststore/crl/scope/{scopeId}`

URI Parameters:

scopeId (required)	Specified scope

Description:
Retrieve all certificates for the specified scope.

Working with a Specific CRL Certificate

`GET /api/2.0/services/truststore/crl/{crlId}`

URI Parameters:

crlId (required)	CRL ID

Description:
Retrieve certificate object for the specified certificate revocation list (CRL).

`DELETE /api/2.0/services/truststore/crl/{crlId}`

URI Parameters:

crlId (required)	CRL ID

Description:
Delete the specified certificate revocation list (CRL).

Working with Service Composer

Service Composer helps you provision and assign network and security services to applications in a virtual infrastructure. You map these services to a security group, and the services are applied to the virtual machines in the security group.

Security Groups

You begin by creating a security group to define assets that you want to protect. Security groups may be static (including specific virtual machines) or dynamic where membership may be defined in one or more of the following ways:

- vCenter containers (clusters, port groups, or datacenters).
- Security tags, IPset, MACset, or even other security groups. For example, you may include a criteria to add all members tagged with the specified security tag (such as AntiVirus.virusFound) to the security group.
- Directory Groups (if NSX Manager is registered with Active Directory).
- Regular expressions such as virtual machines with name *VM1*.

Note that security group membership changes constantly. For example, a virtual machine tagged with the AntiVirus.virusFound tag is moved into the Quarantine security group. When the virus is cleaned and this tag is removed from the virtual machine, it again moves out of the Quarantine security group.

Security Policies

A security policy is a collection of the following service configurations.

Service	Description	Applies to
Distributed Firewall rules **category**: *firewall*	Rules that define the traffic to be allowed to, from, or within the security group.	vNIC
Guest Introspection service **category**: *endpoint*	Third party solution provider services such as anti-virus or vulnerability management services.	virtual machines
Network Introspection services (NetX or Network Extensibility) **category**: *traffic_steering*	Services that monitor your network such as IPS.	virtual machines

Applying Security Policies to Security Groups

You apply a security policy (say SP1) to a security group (say SG1). The services configured for SP1 are applied to all virtual machines that are members of SG1. If a virtual machine belongs to more than one security group, the services that are applied to the virtual machine depends on the precedence of the security policy mapped to the security groups. Service Composer profiles can be exported and imported as backups or for use in other environments. This approach to managing network and security services helps you with actionable and repeatable security policy management.

Service Composer Parameters

The following parameters are related to Service Composer, security policies, and security groups.

Common Parameters

- **actionType** - Defines the type of action belonging to a given executionOrderCategory
- **executionOrderCategory** - Category to which the action belongs to (endpoint, firewall or traffic_steering)
- **isActive** - In a security policy hierarchy, an action within a policy may or may not be active based on the precedence of the policy or usage of isActionEnforced flag in that hierarchy

- **isActionEnforced** - Enforces an action of a parent policy on its child policies for a given actionType and executionOrderCategory. Note that in a policy hierarchy, for a given actionType and executionOrderCategory, there can be only one action which can be marked as enforced.
- **isEnabled** - Indicates whether an action is enabled
- **secondarySecurityGroup** - Applicable for actions which need secondary security groups, say a source-destination firewall rule
- **securityPolicy** - Parent policy in an action

Output-only Parameters

- **executionOrder** - Defines the sequence in which actions belonging to an executionOrderCategory are executed. Note that this is not an input parameter and its value is implied by the index in the list.

Firewall Category Parameters

- **action** - Allow or block the traffic
- **applications** - Applications / application groups on which the rules are to be applied
- **direction** - Direction of traffic towards primary security group. Possible values: inbound, outbound, intra
- **logged** - Flag to enable logging of the traffic that is hit by this rule
- **outsideSecondaryContainer** - Flag to specify outside i.e. outside securitygroup-3

Endpoint Category Parameters

- **serviceId** - ID of the service (as registered with the service insertion module). If this tag is null, the functionality type (as defined in actionType tag) is not applied which will also result in blocking the actions (of given functionality type) that are inherited from the parent security policy. This is true if there is no action of enforce type.
- **invalidServiceId** - Flag to indicate that the service that was referenced in this rule is deleted, which make the rule ineffective (or deviate from the original intent that existed while configuring the rule). You must either modify this rule by adding correct Service or delete this rule.
- **serviceName** -Name of the service
- **serviceProfile** - Profile to be referenced in Endpoint rule.
- **invalidServiceProfile** - Flag to indicate that the service profile that was referenced in this rule is deleted, which makes the rule ineffective (or deviate from the original intent that existed while configuring the rule). You must either modify this rule by adding correct Service Profile or delete this rule.

The following parameters are deprecated:
- **vendorTemplateId**
- **invalidVendorTemplateId**
- **vendorTemplateName**

Traffic Steering/NetX Category Parameters

- **redirect** - Flag to indicate whether to redirect the traffic or not
- **serviceProfile** - Service profile for which redirection is being configured
- **logged** - Flag to enable logging of the traffic that is hit by this rule

---

Working with Security Policies

A security policy is a set of Endpoint, firewall, and network introspection services that can be applied to a security group.

See *Working with Security Groups* for more information about managing security groups.

```
POST /api/2.0/services/policy/securitypolicy
```

Description:
Create a security policy.

When creating a security policy, a parent security policy can be specified if required. The security policy inherits services from the parent security policy. Security group bindings and actions can also be specified while creating the policy. Note that execution order of actions in a category is implied by their order in the list. The response of the call has Location header populated with the URI using which the created object can be fetched.

Ensure that:

- the required VMware built in services (such as Distributed Firewall and Endpoint) are installed. See *NSX Installation Guide*.
- the required partner services have been registered with NSX Manager.
- the required security groups have been created.

Request:

Body: application/xml

```
<securityPolicy>
 <name>name</name>
 <description>decription</description>
 <precedence></precedence>
 <parent>
   <objectId></objectId>
 </parent>
 <securityGroupBinding>
   <objectId></objectId>
 </securityGroupBinding>
 <securityGroupBinding>
   <objectId></objectId>
 </securityGroupBinding>
 <actionsByCategory>
   <category>firewall</category>
   <action class="firewallSecurityAction">
     <name>name</name>
     <description>description</description>
     <category></category>
     <actionType></actionType>
     <isActionEnforced></isActionEnforced>
     <isActive></isActive>
     <isEnabled></isEnabled>
     <secondarySecurityGroup>
       <objectId></objectId>
     </secondarySecurityGroup>
     <secondarySecurityGroup>
       <objectId></objectId>
     </secondarySecurityGroup>
     <applications>
       <application>
         <objectId></objectId>
       </application>
       <applicationGroup>
         <objectId></objectId>
       </applicationGroup>
     </applications>
     <logged></logged>
     <action></action>
     <direction></direction>
     <outsideSecondaryContainer></outsideSecondaryContainer>
   </action>
   <action>
     ***
   </action>
 </actionsByCategory>
```

```
<actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <serviceId></serviceId>
    <serviceProfile>
      <objectId>serviceprofile-1</objectId>
      ***
    </serviceProfile>
    <invalidServiceProfile>false</invalidServiceProfile>
  </action>
</actionsByCategory>
<actionsByCategory>
  <category>traffic_steering</category>
  <action class="trafficSteeringSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <logged></logged>
    <redirect></redirect>
    <serviceProfile>
      <objectId></objectId>
    </serviceProfile>
  </action>
</actionsByCategory>
</securityPolicy>
```

Working With a Specific Security Policy

`GET /api/2.0/services/policy/securitypolicy/{ID}`

URI Parameters:

ID (required)	Security policy, for example, *policy-5*.

Description:
Retrieve security policy information. To view all security policies, specify *all* as the security policy ID.

Responses:
Status Code: 200
Body: application/xml

```
<securityPolicy>
 <objectId>policy-5</objectId>
```

```
<objectTypeName>Policy</objectTypeName>
<vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
<nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
<revision>10</revision>
<type>
  <typeName>Policy</typeName>
</type>
<name>Test Security Policy</name>
<description></description>
<scope>
  <id>globalroot-0</id>
  <objectTypeName>GlobalRoot</objectTypeName>
  <name>Global</name>
</scope>
<clientHandle></clientHandle>
<extendedAttributes></extendedAttributes>
<isUniversal>false</isUniversal>
<universalRevision>0</universalRevision>
<inheritanceAllowed>false</inheritanceAllowed>
<precedence>4300</precedence>
<securityGroupBinding>
  <objectId>securitygroup-10</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
  <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>Local_Web_Tier</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <isUniversal>false</isUniversal>
  <universalRevision>0</universalRevision>
</securityGroupBinding>
<actionsByCategory>
  <category>firewall</category>
  <action class="firewallSecurityAction">
    <objectId>firewallpolicyaction-1</objectId>
    <objectTypeName>FirewallPolicyAction</objectTypeName>
    <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
    <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
    <revision>7</revision>
    <type>
      <typeName>FirewallPolicyAction</typeName>
    </type>
    <name>allow to DB_SG</name>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
```

```
    <category>firewall</category>
    <executionOrder>1</executionOrder>
    <isEnabled>true</isEnabled>
    <isActionEnforced>false</isActionEnforced>
    <secondarySecurityGroup>
      <objectId>securitygroup-12</objectId>
      <objectTypeName>SecurityGroup</objectTypeName>
      <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
      <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
      <revision>2</revision>
      <type>
        <typeName>SecurityGroup</typeName>
      </type>
      <name>Local_DB_Tier</name>
      <description></description>
      <scope>
        <id>globalroot-0</id>
        <objectTypeName>GlobalRoot</objectTypeName>
        <name>Global</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
    </secondarySecurityGroup>
    <invalidSecondaryContainers>false</invalidSecondaryContainers>
    <invalidApplications>false</invalidApplications>
    <logged>true</logged>
    <action>allow</action>
    <direction>outbound</direction>
    <outsideSecondaryContainer>false</outsideSecondaryContainer>
  </action>
 </actionsByCategory>
 <statusesByCategory>
  <category>firewall</category>
  <status>in_sync</status>
 </statusesByCategory>
</securityPolicy>
```

`PUT /api/2.0/services/policy/securitypolicy/{ID}`

URI Parameters:

ID (required)	Security policy, for example, *policy-5*.

Description:

Edit a security policy.

To update a security policy, you must first fetch it. Then edit the received XML and pass it back as the input. The specified configuration replaces the current configuration.

Security group mappings provided in the PUT call replaces the security group mappings for the security policy. To remove all mappings, delete the securityGroupBindings parameter.

You can add or update actions for the security policy by editing the actionsByCategory parameter. To remove all actions (belonging to all categories), delete the actionsByCategory parameter. To remove actions belonging to a specific category, delete the block for that category.

Request:

Body: application/xml

```
<securityPolicy>
 <securityPolicy>
   <name></name>
   <description></description>
   <precedence></precedence>
   <parent>
     <objectId></objectId>
   </parent>
   <securityGroupBinding>
     <objectId></objectId>
   </securityGroupBinding>
   <actionsByCategory>
     <category></category>
     <action class="">
       <name></name>
       <description></description>
       <category></category>
       <actionType></actionType>
       <isActionEnforced></isActionEnforced>
       <isActive></isActive>
       <isEnabled></isEnabled>
       <secondarySecurityGroup>
         <objectId></objectId>
       </secondarySecurityGroup>
       <applications>
         <application>
           <objectId></objectId>
         </application>
         <applicationGroup>
           <objectId></objectId>
         </applicationGroup>
       </applications>
       <logged></logged>
       <scope>
         <id></id>
         <name></name>
         <objectTypeName></objectTypeName>
       </scope>
     </action>
     <direction></direction>
     <outsideSecondaryContainer></outsideSecondaryContainer>
   </actionsByCategory>
 </securityPolicy>
</securityPolicy>
```

`DELETE /api/2.0/services/policy/securitypolicy/{ID}`

URI Parameters:

ID (required)	Security policy, for example, *policy-5*.

Query Parameters:

force (optional)	If set to true, the security policy is deleted even if it is in use.

Description:

Delete a security policy.

When you delete a security policy, its child security policies and all the actions in it are deleted as well.

Working With Security Group Bindings

`PUT /api/2.0/services/policy/securitypolicy/{ID}/sgbinding/{securityGroupId}`

URI Parameters:

`securityGroupId` (required)	Security group ID, for example, *securitygroup-11*.
`ID` (required)	Security policy, for example, *policy-5*.

Description:
Apply the specified security policy to the specified security group.

Working with Security Actions on a Security Policy

`GET /api/2.0/services/policy/securitypolicy/{ID}/securityactions`

URI Parameters:

`ID` (required)	Security policy, for example, *policy-5*.

Description:
Retrieve all security actions applicable on a security policy.

This list includes security actions from associated parent security policies, if any. Security actions per Execution Order Category are sorted based on the weight of security actions in descending order.

Responses:
Status Code: 200
Body: application/xml

```
<securityPolicies>
 <securityPolicy></securityPolicy>
 <securityPolicy></securityPolicy>
</securityPolicies>
```

Working with Service Composer Status

`GET /api/2.0/services/policy/securitypolicy/status`

Description:

Retrieve the consolidated status of Service Composer.

The possible return of value for status are: *in_sync*, *in_progress*, *out_of_sync*, and *pending*.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<serviceComposerStatus>
 <status>in_sync</status>
</serviceComposerStatus>
```

Working with All Service Composer Alarms

`GET /api/2.0/services/policy/securitypolicy/alarms/all`

Description:
Retrieve all system alarms that are raised at Service Composer level and policy level.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<systemAlarms>
 <systemAlarm>
   <eventId></eventId>
   <timestamp></timestamp>
   <severity></severity>
   <eventSource></eventSource>
   <eventCode></eventCode>
   <message></message>
   <module></module>
   <objectId></objectId>
   <reporterName></reporterName>
   <reporterType></reporterType>
   <sourceType></sourceType>
   <displayName></displayName>
   <eventMetadata>
     <data>
     <key></key>
```

```
    <value></value>
    </data>
    <data>
    ***
    </data>
    ***
    <data>
    ***
    </data>
  </eventMetadata>
  <resolutionAttempted></resolutionAttempted>
  <resolvable></resolvable>
  <alarmId></alarmId>
  <alarmCode></alarmCode>
  <alarmSource></alarmSource>
  <alarmBeingResolved></alarmBeingResolved>
  <alarmMetadata>
    <data>
    <key></key>
    <value></value>
    </data>
    <data>
    ***
    </data>
    ***
    <data>
    ***
    </data>
  </alarmMetadata>
 </systemAlarm>
</systemAlarms>
```

Working with Service Composer Firewall Applied To Setting

You can set the applied to setting for all firewall rules created though Service Composer to either Distributed Firewall or Policy's Security Groups. By default, the applied to is set to Distributed Firewall. When Service Composer firewall rules have an applied to setting of distributed firewall, the rules are applied to all clusters on which distributed firewall is installed. If the firewall rules are set to apply to the policy's security groups, you have more granular control over the firewall rules, but may need multiple security policies or firewall rules to get the desired result.

Applied To Values for Service Composer Firewall Rules

Value	Description
dfw_only	Firewall rules are applied to all clusters on which Distributed Firewall is installed.
policy_security_group	Firewall rules are applied to security groups on which the security policy is applied.

`GET /api/2.0/services/policy/securitypolicy/serviceprovider/firewall`

Description:
Retrieve the Service Composer firewall applied to setting.

Responses:
Status Code: 200

Body: application/xml

```
<SecurityPolicyFirewallConfig>
 <appliedTo>dfw_only</appliedTo>
</SecurityPolicyFirewallConfig>
```

`PUT /api/2.0/services/policy/securitypolicy/serviceprovider/firewall`

Description:
Update the Service Composer firewall applied to setting.

Request:
Body: application/xml

```
<SecurityPolicyFirewallConfig>
 <appliedTo>policy_security_group</appliedTo>
</SecurityPolicyFirewallConfig>
```

Working With Service Composer Configuration Import and Export

`GET /api/2.0/services/policy/securitypolicy/hierarchy`

Query Parameters:

`policyIds` (optional)	Comma separated list of security policy IDs to export. If omitted, all security policy IDs are exported.
`prefix` (optional)	A prefix to add before the names of the objects in the exported XML.

Description:
Export a Service Composer configuration (along with the security groups to which the security policies are mapped). You can save the response to a file. The saved configuration can be used as a backup for situations where you may accidentally delete a policy configuration, or it can be exported for use in another NSX Manager environment.

If a prefix is specified, it is added before the names of the security policy, security action, and security group objects in the exported XML. The prefix can thus be used to indicate the remote source from where the hierarchy was exported.

`POST /api/2.0/services/policy/securitypolicy/hierarchy`

Query Parameters:

`suffix` (optional)	A suffix to add after the names of the objects in the imported XML.

Description:
Import a security policy configuration

You can create multiple security policies and parent-child hierarchies using the data fetched through export. All objects including security policies, security groups and security actions are created on a global scope.

The policy that is being imported needs to be included in the request body.

If a suffix is specified, it is added after the names of the security policy, security action, and security group objects in the exported XML. The suffix can thus be used to differentiate locally created objects from imported ones.

The location of the newly created security policy objects (multiple locations are separated by commas) is populated in the Location header of the response.

Request:
Body: application/xml

```
<securityPolicyHierarchy>
 <name></name>
 <description></description>
 <securityPolicy></securityPolicy>
 <securityGroup></securityGroup>
</securityPolicyHierarchy>
```

Working with Virtual Machines with Security Actions Applied

`GET /api/2.0/services/policy/securityaction/{category}/virtualmachines`

URI Parameters:

`category`	Category of security action. Choice of *endpoint* (Guest Introspection), *firewall* (Distributed Firewall) or *traffic_steering* (Network Introspection/Network Extensibility).

Query Parameters:

`attributeKey`	Attribute key.
`attributeValue`	Attribute value.

Description:
Retrieve all VirtualMachine objects on which security action of a given category and attribute has been applied.

Responses:
Status Code: 200
Body: application/xml

```
<vmnodes>
 <vmnode>
   <vmId></vmId>
   <vmName></vmName>
 </vmnode>
 <vmnode>
   <vmId></vmId>
   <vmName></vmName>
```

```
 </vmnode>
</vmnodes>
```

Working With Security Actions Applicable on a Security Group

`GET /api/2.0/services/policy/securitygroup/{ID}/securityactions`

URI Parameters:

ID	Specified security group.

Description:
Retrieve all security actions applicable on a security group for all ExecutionOrderCategories. The list is sorted based on the weight of security actions in descending order. The **isActive** tag indicates if a securityaction will be applied (by the enforcement engine) on the security group.

Responses:
Status Code: 200
Body: application/xml

```
<securityActionsByCategoryMap>
 <actionsByCategory>
   <category>firewall</category>
   <action class="firewallSecurityAction">
     <objectId></objectId>
     <objectTypeName></objectTypeName>
     <vsmUuid></vsmUuid>
     <revision></revision>
     <type>
       <typeName></typeName>
     </type>
     <name>name</name>
     <description>description</description>
     <category></category>
     <executionOrder></executionOrder>
     <actionType></actionType>
     <isActionEnforced></isActionEnforced>
     <isActive></isActive>
     <isEnabled></isEnabled>
     <secondarySecurityGroup>
       <objectId></objectId>
       <objectTypeName></objectTypeName>
       <vsmUuid></vsmUuid>
       <revision></revision>
       <type>
         <typeName></typeName>
       </type>
       <name>name</name>
       <description>description</description>
       <scope>
         <id></id>
         <objectTypeName></objectTypeName>
         <name>name</name>
```

```
    <description>description</description>
  </scope>
  <extendedAttributes></extendedAttributes>
</secondarySecurityGroup>
<secondarySecurityGroup>
  ***
</secondarySecurityGroup>
***
***
<secondarySecurityGroup>
  ***
</secondarySecurityGroup>
<securityPolicy>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name>name</name>
  <description>description</description>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name>name</name>
    <description>description</description>
  </scope>
</securityPolicy>
<invalidSecondaryContainers></invalidSecondaryContainers>
<applications>
  <application>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clientHandle></clientHandle>
     <extendedAttributes></extendedAttributes>
    <inheritanceAllowed></inheritanceAllowed>
    <element>
      <applicationProtocol></applicationProtocol>
      <value></value>
    </element>
  </application>
  <application>
    ***
  </application>
  ***
  ***
</applications>
<invalidApplications>false</invalidApplications>
<logged>false</logged>
<action>block</action>
```

```
    <direction>inbound</direction>
    <outsideSecondaryContainer>true</outsideSecondaryContainer>
  </action>
  <action>
  </action>
  ***
  ***
  <action>
    ***
  </action>
</actionsByCategory>
<actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name>name</name>
    <description>description</description>
    <category></category>
    <executionOrder></executionOrder>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <securityPolicy>
      <objectId></objectId>
      <objectTypeName></objectTypeName>
      <vsmUuid></vsmUuid>
      <revision></revision>
      <type>
        <typeName></typeName>
      </type>
      <name></name>
      <description></description>
      <scope>
        <id></id>
        <objectTypeName></objectTypeName>
        <name>name</name>
        <description>description</description>
      </scope>
    </securityPolicy>
    <serviceName></serviceName>
    <serviceId></serviceId>
    <invalidServiceId></invalidServiceId>
    <ServiceProfile>
      <objectId>serviceprofile-1</objectId>
      ***
    </ServiceProfile>
    <invalidServiceProfile>false</invalidServiceProfile>
  </action>
  <action>
  </action>
  ***
  ***
  <action>
    ***
  </action>
```

```
</actionsByCategory>
<actionsByCategory>
  <category>traffic_steering</category>
  <action class="trafficSteeringSecurityAction">
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name>name</name>
    <description>description</description>
    <category></category>
    <executionOrder></executionOrder>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <securityPolicy>
      <objectId></objectId>
      <objectTypeName></objectTypeName>
      <vsmUuid></vsmUuid>
      <revision></revision>
      <type>
        <typeName></typeName>
      </type>
      <name>name</name>
      <description>description</description>
      <scope>
        <id></id>
        <objectTypeName></objectTypeName>
        <name>name</name>
        <description>description</description>
      </scope>
    </securityPolicy>
    <logged></logged>
    <serviceProfile>
      <objectId></objectId>
      <objectTypeName></objectTypeName>
      <vsmUuid></vsmUuid>
      <revision></revision>
      <type>
        <typeName></typeName>
      </type>
      <name>P</name>
      <clientHandle>
      </clientHandle>
       <extendedAttributes></extendedAttributes>
      <profileAttributes>
        <id></id>
        <revision></revision>
        <attribute>
          <id></id>
          <revision></revision>
          <key></key>
          <name></name>
          <value></value>
        </attribute>
        <attribute>
          ***
        </attribute>
```

```
      </profileAttributes>
      <service>
        <objectId></objectId>
        <objectTypeName></objectTypeName>
        <vsmUuid></vsmUuid>
        <revision></revision>
        <type>
          <typeName></typeName>
        </type>
        <name>name</name>
        <clientHandle></clientHandle>
         <extendedAttributes></extendedAttributes>
      </service>
      <category></category>
      <vendorTemplate>
        <id></id>
        <revision></revision>
        <name>name</name>
        <idFromVendor></idFromVendor>
        <vendorAttributes>
          <id></id>
          <revision></revision>
        </vendorAttributes>
      </vendorTemplate>
      <status></status>
      <vendorAttributes>
        <id></id>
        <revision></revision>
      </vendorAttributes>
      <runtime>
        <nonCompliantDvpg></nonCompliantDvpg>
        <nonCompliantVwire></nonCompliantVwire>
      </runtime>
      <serviceProfileBinding>
        <distributedVirtualPortGroups></distributedVirtualPortGroups>
        <virtualWires></virtualWires>
        <excludedVnics></excludedVnics>
        <virtualServers></virtualServers>
      </serviceProfileBinding>
    </serviceProfile>
    <redirect></redirect>
  </action>
  <action>
  </action>
  ***
  ***
  <action>
    ***
  </action>
 </actionsByCategory>
</securityActionsByCategoryMap>
```

Working with Security Actions Applicable on a Virtual Machine

`GET /api/2.0/services/policy/virtualmachine/{ID}/securityactions`

URI Parameters:

ID (required)	VM ID.

Description:
You can retrieve the security actions applicable on a virtual machine for all ExecutionOrderCategories. The list of SecurityActions per ExecutionOrderCategory is sorted based on the weight of security actions in descending order. The **isActive** tag indicates whether a security action will be applied (by the enforcement engine) on the virtual machine.

Responses:
Status Code: 200
Body: application/xml

```
<securityPolicies>
 <securityPolicy></securityPolicy>
 <securityPolicy></securityPolicy>
</securityPolicies>
```

Working with Service Composer Firewall

`GET /api/2.0/services/policy/serviceprovider/firewall`

Description:
If Service Composer goes out of sync with Distributed Firewall, you must re-synchronize Service Composer rules with firewall rules. If Service Composer stays out of sync, firewall configuration may not stay enforced as expected.

This GET method can perform the following functions, depending on the request body provided. **Note:** Some REST clients do not allow you to specify a request body with a GET request.

Check if Service Composer firewall and Distributed Firewall are in sync

Note: Deprecated. Use `GET /2.0/services/policy/securitypolicy/status` instead.
- If they are in sync, the response body does not contain any data.
- If they are out of sync, the response body contains the unix timestamp representing the time since when Service Composer firewall is out of sync.

```
<keyValues>
 <keyValue>
   <key>getServiceComposerFirewallOutOfSyncTimestamp</key>
 </keyValue>
</keyValues>
```

Synchronize Service Composer firewall with Distributed Firewall

```
<keyValues>
 <keyValue>
   <key>forceSync</key>
 </keyValue>
```

```
</keyValues>
```

Retrieve the state of the auto save draft property in Service Composer

Retrieve the state of the auto save draft property in Service Composer. Response is true or false.

```
<keyValues>
 <keyValue>
   <key>getAutoSaveDraft</key>
 </keyValue>
</keyValues>
```

Change the state of the auto save draft property in Service Composer

Note: Deprecated.

Change the state of the auto save draft property in Service Composer. Provide request body value of true or false.

```
<keyValues>
 <keyValue>
   <key>autoSaveDraft</key>
   <value>false</value>
 </keyValue>
</keyValues>
```

Method history:

Release	Modification
6.2.3	Method updated and some functions deprecated. Changing auto save draft with the **autoSaveDraft** parameter is deprecated, and will be removed in a future release. The default setting of **autoSaveDraft** is changed from *true* to *false*. Method to check if Service Composer and Distributed Firewall are in sync is deprecated, and will be removed in a future release. Use `GET /2.0/services/policy/securitypolicy/status` instead.

Request:
Body: application/xml

```
<keyValues>
 <keyValue>
   <key></key>
   <value></value>
 </keyValue>
</keyValues>
```

Working with Security Policies Mapped to a Security Group

`GET /api/2.0/services/policy/securitygroup/{ID}/securitypolicies`

URI Parameters:

ID (required)	Specified security group ID

Description:

Retrieve security policies mapped to a security group.

The list is sorted based on the precedence of security policy precedence in descending order. The security policy with the highest precedence (highest numeric value) is the first entry (index = 0) in the list.

Responses:

Status Code: 200

Body: application/xml

```
<securityPolicies>
 <securityPolicy>
   <objectId>policy-5</objectId>
   <objectTypeName>Policy</objectTypeName>
   <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
   <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
   <revision>10</revision>
   <type>
     <typeName>Policy</typeName>
   </type>
   <name>Test Security Policy</name>
   <description></description>
   <scope>
     <id>globalroot-0</id>
     <objectTypeName>GlobalRoot</objectTypeName>
     <name>Global</name>
   </scope>
   <clientHandle></clientHandle>
   <extendedAttributes></extendedAttributes>
   <isUniversal>false</isUniversal>
   <universalRevision>0</universalRevision>
   <inheritanceAllowed>false</inheritanceAllowed>
   <precedence>4300</precedence>
   <securityGroupBinding>
     <objectId>securitygroup-10</objectId>
     <objectTypeName>SecurityGroup</objectTypeName>
     <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
     <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
     <revision>2</revision>
     <type>
       <typeName>SecurityGroup</typeName>
     </type>
     <name>Local_Web_Tier</name>
     <description></description>
     <scope>
       <id>globalroot-0</id>
       <objectTypeName>GlobalRoot</objectTypeName>
       <name>Global</name>
     </scope>
```

```
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </securityGroupBinding>
  <actionsByCategory>
    <category>firewall</category>
    <action class="firewallSecurityAction">
      <objectId>firewallpolicyaction-1</objectId>
      <objectTypeName>FirewallPolicyAction</objectTypeName>
      <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
      <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
      <revision>7</revision>
      <type>
        <typeName>FirewallPolicyAction</typeName>
      </type>
      <name>allow to DB_SG</name>
      <scope>
        <id>globalroot-0</id>
        <objectTypeName>GlobalRoot</objectTypeName>
        <name>Global</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
      <category>firewall</category>
      <executionOrder>1</executionOrder>
      <isEnabled>true</isEnabled>
      <isActionEnforced>false</isActionEnforced>
      <secondarySecurityGroup>
        <objectId>securitygroup-12</objectId>
        <objectTypeName>SecurityGroup</objectTypeName>
        <vsmUuid>423A993F-BEE6-1285-58F1-54E48D508D90</vsmUuid>
        <nodeId>916287b3-761d-430b-8ab2-83878dfe3e7f</nodeId>
        <revision>2</revision>
        <type>
          <typeName>SecurityGroup</typeName>
        </type>
        <name>Local_DB_Tier</name>
        <description></description>
        <scope>
          <id>globalroot-0</id>
          <objectTypeName>GlobalRoot</objectTypeName>
          <name>Global</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
        <isUniversal>false</isUniversal>
        <universalRevision>0</universalRevision>
      </secondarySecurityGroup>
      <invalidSecondaryContainers>false</invalidSecondaryContainers>
      <invalidApplications>false</invalidApplications>
      <logged>true</logged>
      <action>allow</action>
      <direction>outbound</direction>
      <outsideSecondaryContainer>false</outsideSecondaryContainer>
    </action>
  </actionsByCategory>
  <statusesByCategory>
    <category>firewall</category>
    <status>in_sync</status>
```

```
   </statusesByCategory>
 </securityPolicy>
</securityPolicies>
```

Working with SNMP

NSX Manager receives events from other NSX components, including NSX Edge, network fabric, and hypervisors.

You can configure NSX Manager to forward SNMP traps to an SNMP Manager.

Working with SNMP Status Settings

You can configure settings for SNMP on the NSX Manager.

Parameter	Description
serviceStatus	Boolean. Set to true to enable SNMP. There must be at least one SNMP manager configured to enable SNMP.
groupNotification	Boolean. Set to true to group similar SNMP notifications. This reduces the number of notifications being sent out, which can improve SNMP manager performance when there is a high volume of SNMP notifications.

`GET /api/2.0/services/snmp/status`

Description:
Retrieve SNMP status settings.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<snmpServiceStatus>
 <serviceStatus>false</serviceStatus>
 <groupedNotification>true</groupedNotification>
</snmpServiceStatus>
```

`PUT /api/2.0/services/snmp/status`

Description:
Update SNMP status settings.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:

Body: application/xml

```
<snmpServiceStatus>
 <serviceStatus>true</serviceStatus>
 <groupedNotification>true</groupedNotification>
</snmpServiceStatus>
```

Working with SNMP Managers

GET /api/2.0/services/snmp/manager

Description:
Retrieve information about SNMP managers.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<snmpManagers>
 <snmpManager>
   <managerId>1330</managerId>
   <ip>10.10.10.10</ip>
   <port>162</port>
   <communityString>NSXManager</communityString>
   <enabled>true</enabled>
 </snmpManager>
 <snmpManager>
   <managerId>1331</managerId>
   <ip>10.10.10.11</ip>
   <port>162</port>
   <communityString>NSXManager</communityString>
   <enabled>true</enabled>
 </snmpManager>
</snmpManagers>
```

POST /api/2.0/services/snmp/manager

Description:
Add an SNMP manager.

Method history:

Release	Modification

6.2.3	Method introduced.

Request:
Body: application/xml

```
<snmpManager>
 <ip>10.10.10.10</ip>
 <port>162</port>
 <communityString>NSXManager</communityString>
 <enabled>true</enabled>
</snmpManager>
```

Working with a Specific SNMP Manager

`GET /api/2.0/services/snmp/manager/{managerId}`

URI Parameters:

`managerId`	ID of the SNMP manager.

Description:
Retrieve information about the specified SNMP manager.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<snmpManager>
 <managerId>1330</managerId>
 <ip>10.10.10.10</ip>
 <port>162</port>
 <communityString>NSXManager</communityString>
 <enabled>true</enabled>
</snmpManager>
```

`PUT /api/2.0/services/snmp/manager/{managerId}`

URI Parameters:

`managerId`	ID of the SNMP manager.

Description:

Update an SNMP manager configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<snmpManager>
 <managerId>1330</managerId>
 <ip>10.10.10.10</ip>
 <port>162</port>
 <communityString>NSXManager</communityString>
 <enabled>false</enabled>
</snmpManager>
```

`DELETE /api/2.0/services/snmp/manager/{managerId}`

URI Parameters:

managerId	ID of the SNMP manager.

Description:
Delete an SNMP manager configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Working with SNMP Traps

`GET /api/2.0/services/snmp/trap`

Description:
Retrieve information about SNMP traps.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<trapConfigs>
 <trapConfig>
   <eventId>300001</eventId>
   <oid>1.3.6.1.4.1.6876.90.1.2.10.0.1</oid>
   <componentName>ServiceComposer</componentName>
   <enabled>true</enabled>
 </trapConfig>
 <trapConfig>
   <eventId>300009</eventId>
   <oid>1.3.6.1.4.1.6876.90.1.2.10.0.10</oid>
   <componentName>ServiceComposer</componentName>
   <enabled>true</enabled>
 </trapConfig>
 ***
</trapConfigs>
```

Working with a Specific SNMP Trap

`GET /api/2.0/services/snmp/trap/{oid}`

URI Parameters:

oid	SNMP object identifier.

Description:
Retrieve information about the specified SNMP trap.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<trapConfig>
 <eventId>321100</eventId>
 <oid>1.3.6.1.4.1.6876.90.1.2.9.0.6</oid>
 <componentName>Messaging</componentName>
 <enabled>true</enabled>
</trapConfig>
```

`PUT /api/2.0/services/snmp/trap/{oid}`

URI Parameters:

oid	SNMP object identifier.

Description:

Update the specified SNMP trap.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<trapConfig>
 <oid>1.3.6.1.4.1.6876.90.1.2.3.0.1</oid>
 <enabled>false</enabled>
</trapConfig>
```

Working with the Central CLI

`POST /api/1.0/nsx/cli`

Query Parameters:

`action` (required)	Use *action=execute*.

Description:

The central command-line interface (central CLI) commands are run from the NSX Manager command line, and retrieve information from the NSX Manager and other devices. These commands can also be executed in the API.

You can insert any valid Central CLI command as the **command** parameter. For a complete list of the Central CLI commands executable through the API, please see the Central CLI chapter of the *NSX Command Line Interface Reference*.

Request:

Body: application/xml

```
<nsxcli>
 <command>show logical-switch list host host-21 vni</command>
</nsxcli>
```

Communication Status

This feature allows the user to check the connection status between the NSX Manager and hosts. A hash map is used to hold all hosts' connection status. It keeps track of the latest heartbeat from each host. When querying a host's connection status, NSX Manager will get the latest heartbeat information to compare the last heartbeat time and current time. If the duration is longer than a threshold, it returns *DOWN*, otherwise it returns *UP*. If no last heartbeat information is found and this host has not been prepared or the netcpa version on this host is lower than 6.2.0, it will return *NOT_AVAILABLE*. But if no last heartbeat information is found and the host has been prepared with netcpa version no less than 6.2.0, it will return *DOWN*. When a host has been unprepared, its heartbeat information will be removed from the NSX Manager memory.

Communication Status of a Specific Host

`GET /api/2.0/vdn/inventory/host/{hostId}/connection/status`

URI Parameters:

`hostId` (required)	ID of the host to check.

Description:
Retrieve the status of the specified host.

History:

Release	Modification
6.2.3	Method updated. Introduced **hostToControllerConnectionErrors** array. Deprecated **fullSyncCount** parameter. Parameter is still present, but always has value of -1.

Responses:
Status Code: 200
Body: application/xml

```
<hostConnStatus>
 <hostId>host-32</hostId>
 <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
 <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
 <hostToControllerConn>UP</hostToControllerConn>
 <fullSyncCount>-1</fullSyncCount>
</hostConnStatus>
```

Communication Status of a List of Hosts

`GET /api/2.0/vdn/inventory/hosts/connection/status`

Query Parameters:

hostId (required)	ID of a host to check. You can provide multiple hosts with ?hostId=*host1*&hostId;=*host2*...

Description:
Retrieve the status of a list of hosts.

Release	Modification
6.2.3	Method updated. Introduced **hostToControllerConnectionErrors** array. Deprecated **fullSyncCount** parameter. Parameter is still present, but always has value of -1.

Responses:
Status Code: 200
Body: application/xml

```
<hostConnStatusList>
 <hostConnStatuses>
   <hostConnStatus>
     <hostId>host-31</hostId>
     <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
     <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
     <hostToControllerConn>UP</hostToControllerConn>
     <fullSyncCount>-1</fullSyncCount>
   </hostConnStatus>
   <hostConnStatus>
     <hostId>host-32</hostId>
     <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
     <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
     <hostToControllerConn>DOWN</hostToControllerConn>
     <fullSyncCount>-1</fullSyncCount>
     <hostToControllerConnectionErrors>
       <hostToControllerConnectionError>
         <controllerIp>10.160.203.236</controllerIp>
         <errorCode>1255604</errorCode>
         <errorMessage>Connection Refused</errorMessage>
         </hostToControllerConnectionError>
         <hostToControllerConnectionError>
         <controllerIp>10.160.203.237</controllerIp>
         <errorCode>1255603</errorCode>
         <errorMessage>SSL Handshake Failure</errorMessage>
       </hostToControllerConnectionError>
     </hostToControllerConnectionErrors>
   </hostConnStatus>
   ***
 </hostConnStatuses>
</hostConnStatusList>
```

Working with Hardware Gateways

`GET /api/2.0/vdn/hardwaregateways`

Description:
Retrieve information about all hardware gateways.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<list>
   <hardwareGateway>
     <objectId>torgateway-1</objectId>
     <revision>0</revision>
     <name>torgateway1</name>
     <description>this is tor instance 1</description>
     <clientHandle></clientHandle>
     <isUniversal>false</isUniversal>
     <universalRevision>0</universalRevision>
     <uuid>6536bcf5-2f55-47f6-8b26-9fa632061d8c</uuid>
     <status>UP</status>
     <thumbprint>B9:0E:E9:6C:AA:7B:AD:11:64:4C:33:92:4E:0C:D8:16:10:95:02:A7</thumbprint>
     <bfdEnabled>true</bfdEnabled>
     <managementIp>10.116.255.160</managementIp>
     <bindingCount>2</bindingCount>
   </hardwareGateway>
   <hardwareGateway>
     <objectId>torgateway-2</objectId>
     <revision>0</revision>
     <name>torgateway2</name>
     <description>this is tor instance 2</description>
     <clientHandle></clientHandle>
     <isUniversal>false</isUniversal>
     <universalRevision>0</universalRevision>
     <uuid>f1e9b733-c0c3-4905-b00d-4bd6d8649f48</uuid>
     <status>UP</status>
     <thumbprint>3C:9D:C0:9B:F7:57:AF:EA:6A:9F:49:27:7B:23:25:D3:5E:0D:53:ED</thumbprint>
     <bfdEnabled>true</bfdEnabled>
     <managementIp>10.116.251.149</managementIp>
     <bindingCount>2</bindingCount>
   </hardwareGateway>
</list>
```

`POST /api/2.0/vdn/hardwaregateways`

Description:
Install a hardware gateway.

bfdEnabled is true by default.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewaySpec>
 <name></name>
 <description></description>
 <certificate></certificate>
 <bfdEnabled></bfdEnabled>
</hardwareGatewaySpec>
```

Working With a Specific Hardware Gateway

`GET /api/2.0/vdn/hardwaregateways/{hardwareGatewayId}`

URI Parameters:

`hardwareGatewayId` (required)	Object ID of the hardware gateway.

Description:
Retrieve information about the specified hardware gateway.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGateway>
 <objectId>torgateway-1</objectId>
 <revision>0</revision>
 <name>torgateway1</name>
 <description>this is tor instance 1</description>
 <clientHandle></clientHandle>
 <isUniversal>false</isUniversal>
 <universalRevision>0</universalRevision>
 <uuid>6536bcf5-2f55-47f6-8b26-9fa632061d8c</uuid>
 <status>UP</status>
 <thumbprint>B9:0E:E9:6C:AA:7B:AD:11:64:4C:33:92:4E:0C:D8:16:10:95:02:A7</thumbprint>
 <bfdEnabled>true</bfdEnabled>
 <managementIp>10.116.255.160</managementIp>
```

```
 <bindingCount>2</bindingCount>
</hardwareGateway>
```

`PUT /api/2.0/vdn/hardwaregateways/{hardwareGatewayId}`

URI Parameters:

hardwareGatewayId (required)	Object ID of the hardware gateway.

Description:
Update the specified hardware gateway.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewaySpec>
 <name></name>
 <description></description>
 <certificate></certificate>
 <bfdEnabled></bfdEnabled>
</hardwareGatewaySpec>
```

`DELETE /api/2.0/vdn/hardwaregateways/{hardwareGatewayId}`

URI Parameters:

hardwareGatewayId (required)	Object ID of the hardware gateway.

Description:
Delete the specified hardware gateway.

Method history:

Release	Modification
6.2.3	Method introduced.

Working With Switches on a Specific Hardware Gateway

`GET /api/2.0/vdn/hardwaregateways/{hardwareGatewayId}/switches`

URI Parameters:

hardwareGatewayId (required)	Object ID of the hardware gateway.

Description:
Retrieve information about switches on the specified hardware gateway.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewaySwitches>
 <hardwareGatewaySwitch>
   <switchname>1-switch-579</switchname>
   <description></description>
   <faults></faults>
 </hardwareGatewaySwitch>
 <hardwareGatewayId>torgateway-1</hardwareGatewayId>
</hardwareGatewaySwitches>
```

Working With a Specific Switch on a Specific Hardware Gateway

Working With Ports on a Specific Switch on a Specific Hardware Gateway

`GET /api/2.0/vdn/hardwaregateways/{hardwareGatewayId}/switches/{switchName}/switchports`

URI Parameters:

`switchName`	Switch Name
`hardwareGatewayId` (required)	Object ID of the hardware gateway.

Description:
Retrive information about the hardware gateway switch ports for the specified switch and hardware gateway.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewaySwitchPorts>
   <hardwareGatewaySwitchPort>
     <portname>p4</portname>
     <description></description>
     <faults></faults>
   </hardwareGatewaySwitchPort>
   <hardwareGatewaySwitchPort>
     <portname>p3</portname>
     <description></description>
     <faults></faults>
   </hardwareGatewaySwitchPort>
   <hardwareGatewaySwitchPort>
     <portname>p2</portname>
     <description></description>
     <faults></faults>
   </hardwareGatewaySwitchPort>
   <hardwareGatewaySwitchPort>
     <portname>p1</portname>
     <description></description>
     <faults></faults>
   </hardwareGatewaySwitchPort>
   <hardwareGatewaySwitch>
     <switchname>1-switch-579</switchname>
   </hardwareGatewaySwitch>
   <hardwareGatewayId>torgateway-1</hardwareGatewayId>
</hardwareGatewaySwitchPorts>
```

Working With the Hardware Gateway Replication Cluster

`GET /api/2.0/vdn/hardwaregateways/replicationcluster`

Description:
Retrieve information about the hardware gateway replication cluster.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<replicationCluster>
 <hosts>
   <basicinfo>
     <objectId>host-26</objectId>
     <objectTypeName>HostSystem</objectTypeName>
     <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
     <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
```

```
    <revision>32</revision>
    <type>
      <typeName>HostSystem</typeName>
    </type>
    <name>10.116.254.9</name>
    <scope>
      <id>domain-c24</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>ComputeCluster2-$$</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </basicinfo>
  <basicinfo>
    <objectId>host-21</objectId>
    <objectTypeName>HostSystem</objectTypeName>
    <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
    <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
    <revision>31</revision>
    <type>
      <typeName>HostSystem</typeName>
    </type>
    <name>10.116.247.220</name>
    <scope>
      <id>domain-c18</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>ComputeCluster1-$$</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </basicinfo>
  <basicinfo>
    <objectId>host-20</objectId>
    <objectTypeName>HostSystem</objectTypeName>
    <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
    <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
    <revision>33</revision>
    <type>
      <typeName>HostSystem</typeName>
    </type>
    <name>10.116.254.157</name>
    <scope>
      <id>domain-c18</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>ComputeCluster1-$$</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </basicinfo>
 </hosts>
</replicationCluster>
```

PUT /api/2.0/vdn/hardwaregateways/replicationcluster

Description:
Update the hardware gateway replication cluster.

Add or remove hosts on a replication cluster.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: appplication/xml

```
<replicationCluster>
 <hosts>
   <basicinfo>
     <objectId>host-20</objectId>
   </basicinfo>
   <basicinfo>
     <objectId>host-21</objectId>
   </basicinfo>
   <basicinfo>
     <objectId>host-26</objectId>
   </basicinfo>
 </hosts>
</replicationCluster>
```

Retrieve Information About Hardware Gateway Bindings

`GET /api/2.0/vdn/hardwaregateways/bindings`

Query Parameters:

`hardwareGatewayId` (optional)	ID of the hardware gateway.
`vni` (optional)	VNI of the logical switch.

Description:
Retrieve information about hardware gateway bindings.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewayBinding>
 <id>hardware gateway binding id</id>
 <hardwareGatewayId>hwgateway1</hardwareGatewayId>
 <vlan>201</vlan>
 <switchName>s1</switchName>
 <portname>s1</portname>
 <virtualWire>virtualwire-1</virtualWire>
 <vni>5000</vni>
</hardwareGatewayBinding>
```

`POST /api/2.0/vdn/hardwaregateways/bindings`

Description:
Create a hardware gateway binding.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewayBinding>
 <hardwareGatewayId></hardwareGatewayId>
 <vlan></vlan>
 <switchName></switchName>
 <portName></portName>
</hardwareGatewayBinding>
```

Working With a Specific Hardware Gateway Binding

`GET /api/2.0/vdn/hardwaregateways/bindings/{bindingId}`

URI Parameters:

`bindingId`	hardware gateway binding ID.

Description:
Retrieve information about the specified hardware gateway binding.

Method history:

Release	Modification
6.2.3	Method introduced.

`PUT /api/2.0/vdn/hardwaregateways/bindings/{bindingId}`

URI Parameters:

bindingId	hardware gateway binding ID.

Description:
Update the specified hardware gateway binding.

You can update the binding parameters. This API will fail if:
- the specified *hardwareGatewayId* does not exist.
- the specified logical switch (*virtualWire*) is not present or there is a software gateway on the binding.
- the new binding value is a duplicate of an existing binding.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewayBinding>
 <hardwareGatewayId>hardwaregateway1</hardwareGatewayId>
 <vlan>201</vlan>
 <switchName>s1</switchName>
 <portname>s1</portname>
 <virtualWire>virtualwire-1</virtualWire>
</hardwareGatewayBinding>
```

`DELETE /api/2.0/vdn/hardwaregateways/bindings/{bindingId}`

URI Parameters:

bindingId	hardware gateway binding ID.

Description:
Delete the specified hardware gateway binding.

Method history:

Release	Modification
6.2.3	Method introduced.

Working with Hardware Gateway Binding Statistics

`GET /api/2.0/vdn/hardwaregateways/bindings/{bindingId}/statistic`

URI Parameters:

bindingId	hardware gateway binding ID.

Description:

Retrieve statistics for the specified hardware gateway binding.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewayStats>
 <bindingId>hwgwbinding-5</bindingId>
 <timestamp>long type timestamp for this query response</timestamp>
 <packetsFromLocal>23431</packetsFromLocal>
 <bytesFromLocal>734754</bytesFromLocal>
 <packetsToLocal>2343</packetsToLocal>
 <bytesToLocal>74364</bytesToLocal>
</hardwareGatewayStats>
```

Working With Hardware Gateway Binding Objects

`POST /api/2.0/vdn/hardwaregateways/bindings/manage`

Description:
Manage hardware gateway binding objects.

Use this API to attach, detach, and update multiple bindings in a single API call. This API accepts three lists for add, update, and delete. Each list accepts a hardwareGatewayManageBindingsItem with a full description of the new binding with its objectID. This API handles a maximum of 100 HardwareGatewayManageBindingsItem objects for each of the Add/Update/Delete lists.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewayManageBindings>
 <addItems>
   <hardwareGatewayManageBindingItem>
     <hardwareGatewayId></hardwareGatewayId>
     <virtualWireId></virtualWireId>
     <switchName></switchName>
     <portname></portname>
     <vlan></vlan>
     <virtualWire></virtualWire>
```

```
    </hardwareGatewayManageBindingItem>
  </addItems>
  <updateItems>
    <hardwareGatewayManageBindingItem>
      <objectId></objectId>
      <hardwareGatewayId></hardwareGatewayId>
      <virtualWireId></virtualWireId>
      <switchName></switchName>
      <portname></portname>
      <vlan></vlan>
      <virtualWire></virtualWire>
    </hardwareGatewayManageBindingItem>
  </updateItems>
  <deleteItems>
    <hardwareGatewayManageBindingItem>
      <objectId></objectId>
    </hardwareGatewayManageBindingItem>
  </deleteItems>
</hardwareGatewayManageBindings>
```

Working With Hardware Gateway BFD (Bidirectional Forwarding Detection)

Working With Hardware Gateway BFD Configuration

`GET /api/2.0/vdn/hardwaregateways/bfd/config`

Description:
Retrieve global hardware gateway BFD configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewayBfdParams>
  <bfdEnabled>true</bfdEnabled>
  <probeInterval>100</probeInterval>
</hardwareGatewayBfdParams>
```

`PUT /api/2.0/vdn/hardwaregateways/bfd/config`

Description:

Update global hardware gateway BFD configuration.

Method history:

Release	Modification
6.2.3	Method introduced.

Request:
Body: application/xml

```
<hardwareGatewayBfdParams>
 <bfdEnabled>true</bfdEnabled>
 <probeInterval>100</probeInterval>
</hardwareGatewayBfdParams>
```

Working With Hardware Gateway BFD Tunnel Status

`GET /api/2.0/vdn/hardwaregateways/bfd/status`

Description:
Retrieve hardware gateway BFD tunnel status for all tunnel endpoints, including hosts and hardware gateways.

Method history:

Release	Modification
6.2.3	Method introduced.

Responses:
Status Code: 200
Body: application/xml

```
<hardwareGatewayBfdStatusList>
 <statuses>
   <hardwareGatewayBfdStatus>
     <probeSourceId>torgateway-2</probeSourceId>
     <bfdTunnelList>
       <bfdTunnelStatus>
         <diagnostic>Neighbor Signaled Session Down</diagnostic>
         <enabled>true</enabled>
         <forwarding>true</forwarding>
         <info></info>
         <localVtepIp>172.21.145.84</localVtepIp>
         <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
         <remoteState>UP</remoteState>
         <remoteVtepIp>172.19.152.226</remoteVtepIp>
         <state>UP</state>
       </bfdTunnelStatus>
       <bfdTunnelStatus>
         <diagnostic>Neighbor Signaled Session Down</diagnostic>
         <enabled>true</enabled>
```

```
        <forwarding>true</forwarding>
        <info></info>
        <localVtepIp>172.21.145.84</localVtepIp>
        <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
        <remoteState>UP</remoteState>
        <remoteVtepIp>172.18.171.169</remoteVtepIp>
        <state>UP</state>
      </bfdTunnelStatus>
      <bfdTunnelStatus>
        <diagnostic>Neighbor Signaled Session Down</diagnostic>
        <enabled>true</enabled>
        <forwarding>true</forwarding>
        <info></info>
        <localVtepIp>172.21.145.84</localVtepIp>
        <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
        <remoteState>UP</remoteState>
        <remoteVtepIp>172.18.171.168</remoteVtepIp>
        <state>UP</state>
      </bfdTunnelStatus>
    </bfdTunnelList>
  </hardwareGatewayBfdStatus>
  <hardwareGatewayBfdStatus>
    <probeSourceId>torgateway-1</probeSourceId>
    <bfdTunnelList>
      <bfdTunnelStatus>
        <diagnostic>Control Detection Time Expired</diagnostic>
        <enabled>true</enabled>
        <forwarding>true</forwarding>
        <info></info>
        <localVtepIp>172.21.145.85</localVtepIp>
        <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
        <remoteState>UP</remoteState>
        <remoteVtepIp>172.19.152.226</remoteVtepIp>
        <state>UP</state>
      </bfdTunnelStatus>
      <bfdTunnelStatus>
        <diagnostic>Neighbor Signaled Session Down</diagnostic>
        <enabled>true</enabled>
        <forwarding>true</forwarding>
        <info></info>
        <localVtepIp>172.21.145.85</localVtepIp>
        <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
        <remoteState>UP</remoteState>
        <remoteVtepIp>172.18.171.168</remoteVtepIp>
        <state>UP</state>
      </bfdTunnelStatus>
      <bfdTunnelStatus>
        <diagnostic>Neighbor Signaled Session Down</diagnostic>
        <enabled>true</enabled>
        <forwarding>true</forwarding>
        <info></info>
        <localVtepIp>172.21.145.85</localVtepIp>
        <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
        <remoteState>UP</remoteState>
        <remoteVtepIp>172.18.171.169</remoteVtepIp>
        <state>UP</state>
      </bfdTunnelStatus>
    </bfdTunnelList>
  </hardwareGatewayBfdStatus>
 </statuses>
</hardwareGatewayBfdStatusList>
```