Networks and Bridges APIs

• Bridge APIs, on page 1
• Network Creation APIs, on page 5

Bridge APIs

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

Table 1: Bridge APIs

<table>
<thead>
<tr>
<th>Action</th>
<th>Method</th>
<th>Payload Required</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>To create a bridge</td>
<td>POST</td>
<td>Yes</td>
<td>/api/config/bridges</td>
</tr>
<tr>
<td>To verify a bridge configuration</td>
<td>GET</td>
<td>No</td>
<td>/api/config/bridges?deep</td>
</tr>
<tr>
<td>To get specific IP/DHCP info for all bridges</td>
<td>GET</td>
<td>No</td>
<td>/api/operational/bridge-settings/ip dhcp configuration</td>
</tr>
<tr>
<td>To get specific IP/DHCP info for specific bridge</td>
<td>GET</td>
<td>No</td>
<td>/api/operational/bridge-settings/ip dhcp configuration/br_name/ip dhcp configuration</td>
</tr>
<tr>
<td>To modify a bridge, and attach a port to the bridge</td>
<td>PUT</td>
<td>Yes</td>
<td>/api/config/bridges/bridge/&lt;bridge name&gt;</td>
</tr>
<tr>
<td>To delete a bridge</td>
<td>DELETE</td>
<td>No</td>
<td>/api/config/bridges/bridge/&lt;bridge name&gt;</td>
</tr>
</tbody>
</table>

Example for Bridge Payload

<bridge>
  <name>sc-br</name>
  <port>
    <name>eth3</name>
</port>
Table 2: Bridge Payload Description

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

Example: POST Bridge Creation API

curl -k -v -u admin:admin -H Content-Type:application/vnd.yang.data+xml -X POST https://209.165.201.1/api/config/bridges -d "<bridge><name>sc-br</name><port><name>eth3</name></port><dhcp/><dhcp-ipv6/></bridge>". 

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

curl -k -v -u admin:admin -H Content-Type: application/vnd.yang.data+xml -X GET "https://209.165.201.1/api/config/bridges?deep"

Networks and Bridges APIs

Example: GET Bridge Configuration API
Example: DELETE Bridge API

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

Example: GET IPv4 address for all bridges

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

Example: GET dhcp enabled under wan-br


Example: DELETE Bridge API
Network Creation APIs

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Method</th>
<th>Payload Required</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>To create a network</td>
<td>POST</td>
<td>Yes</td>
<td>/api/config/networks</td>
</tr>
<tr>
<td>To verify network configuration details</td>
<td>GET</td>
<td>No</td>
<td>/api/config/networks?deep</td>
</tr>
<tr>
<td>To modify a network</td>
<td>PUT</td>
<td>Yes</td>
<td>/api/config/networks/network/&lt;network name&gt;</td>
</tr>
<tr>
<td>To delete a network</td>
<td>DELETE</td>
<td>No</td>
<td>/api/config/networks/network/&lt;network name&gt;</td>
</tr>
</tbody>
</table>

Example for Network Creation Payload

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

Table 4: Network Creation Payload Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Mandatory/Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>network name</td>
<td>String</td>
<td>Name of the network.</td>
<td>Yes</td>
</tr>
<tr>
<td>bridge</td>
<td>String</td>
<td>Name of the bridge the network is attached to.</td>
<td>Yes</td>
</tr>
<tr>
<td>trunk</td>
<td>Boolean</td>
<td>Network set to trunk mode.</td>
<td>No/false</td>
</tr>
<tr>
<td>sriov</td>
<td>Boolean</td>
<td>SR-IOV supported on the network.</td>
<td>No/true</td>
</tr>
<tr>
<td>native-tagged</td>
<td>Boolean</td>
<td>Specifies if the network is tagged or not.</td>
<td>No</td>
</tr>
<tr>
<td>native-vlan</td>
<td>Integer</td>
<td>Specifies a native VLAN. It sets the native characteristics when the interface is in <strong>trunk</strong> mode. If you do not configure a native VLAN, the default VLAN 1 is used as the native VLAN.</td>
<td>No</td>
</tr>
<tr>
<td>vlan</td>
<td>Integer</td>
<td>Specifies the VLAN number. If the <strong>trunk</strong> parameter is configured as true, this parameter specifies a set of VLAN numbers and ranges. If <strong>trunk</strong> parameter is false, access mode is true, then this parameter can have only one VLAN number.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Example: POST Network API**

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

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