



Cisco NSO Crosswork Hierarchical Controller - Function Pack

User Guide

Version 8.0.0

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. YOU MUST TAKE FULL RESPONSIBILITY FOR THE APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Copyright

© 2024 Cisco Systems, Inc. All rights reserved.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Contents

Preface.....	4
Bias-free Documentation Policy	5
Getting Started	6
Cisco Network Services Orchestrator - Overview	6
Cisco NSO CHCO FP – Overview and Architecture	8
Setting up Controllers.....	10
Setting up the CNC Controller.....	10
Setting up the NSP Controller.....	14
Service Deployments	17
Deploying L2VPN Services	18
Deploying L2VPN Services on an NSP Controller	18
Deploying L2VPN Services Using CNC Controller	24
Deploying L3VPN Services	29
Deploying L3VPN Services on an NSP Controller	29
Deploying L3VPN Services on CNC Controller.....	43
Working with Action Commands	51
Purge Action	52
Managing CHCO FP.....	52
Maintaining Logs	52
Deleting Services	52
Appendix A - Yang Models	54
hco-nm.yang.....	54
ietf-l2vpn-ntw.yang	64
ietf-l3vpn-ntw.yang	80
Augmentations Model for Kafka Settings	99

Preface

Abstract

This document serves as the documentation reference and user guide for the Cisco NSO Crosswork Hierarchical Controller - Function Pack (NSO CHCO FP).

Audience

This document describes how to configure and use CHCO FP. This document is intended for Cisco Advanced Services developers, network engineers, and system engineers who configure and deliver Cisco CHCO FP automation functionalities to Cisco customers.

Additional Documentation

This documentation requires the reader to have a good understanding of Cisco NSO and its usage as described in the Cisco NSO documentation. For documentation on additional Cisco products, see the Cisco documentation website.

Sl. No.	Documentation
1.	Cisco NSO CHCO FP Installation Guide
2.	Cisco NSO Installation Guide
3.	Cisco NSO User Guide

Bias-free Documentation Policy

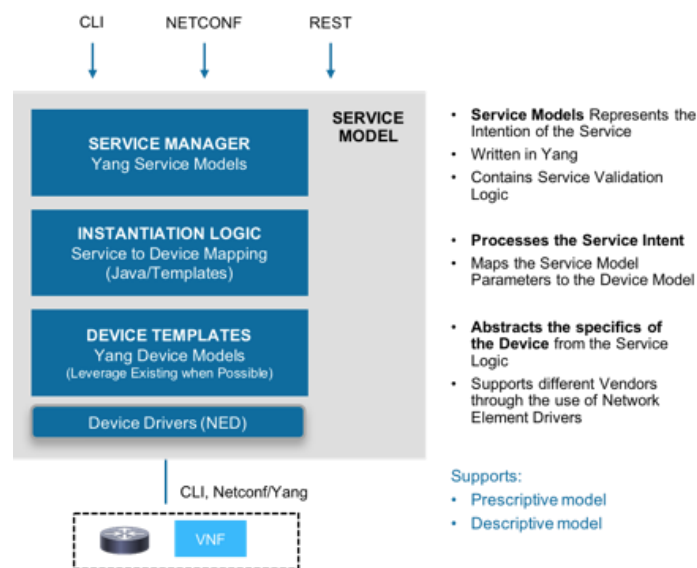
Cisco follows a bias-free documentation policy. According to this policy, Cisco treats all persons with respect—regardless of race, color, ancestry, national origin, age, sex, citizenship, veteran status, marital status, sexual orientation, physical or mental ability, religious creed, or medical condition. Language or graphic elements that offend others violate our business philosophy and our company policy.

Getting Started

This topic provides an overview of Cisco Network Services Orchestrator (NSO) and an understanding of Cisco Crosswork Hierarchical Controller Function Pack.

Cisco Network Services Orchestrator - Overview

Network engineers use NSO as a central point of management for the entire network by using a network command line interface (CLI). NSO enables service providers to dynamically adapt their orchestration solution with changes in the offered service portfolio. NSO is built on a Model-Driven Architecture which supports the dynamic addition and modification of service definitions. The service models are written in the YANG modeling language (RFC 6020).



You can also upload the configuration information to NSO by using the payloads. These payloads contain the desired service configurations in an XML format. Each service can have a single file or multiple files. Deploy the configuration information by using payloads through the NSO CLI or by invoking the Northbound interface to create and modify service configurations. This documentation uses both network CLI and sample XML payloads.

From a Linux console, invoke the Network Configuration Protocol NETCONF NBI command to push or load the XML payload into NSO.

```
# netconf-console --port=[port_number] --host=[host_IP_address] -u [username] -p [password] --edit-config payload.xml
```

Example

```
# netconf-console --port=830 --host=127.0.0.1 -u admin -p Fr3eB!rd$ --edit-config payload.xml
```

Run the following command to get the configuration details from the configuration database in the NSO. For more information about NETCONF, see *Cisco Networking Services Configuration Guide*.

```
# netconf-console --port=[port_number]--host=[host_IP_address] -u  
[host_username] -p [host_password] --get-config
```

Example

```
# netconf-console --port=830 --host=127.0.0.1 -u admin -p Fr3eB!rd$ --get-  
config
```

NETCONF supports the following operations:

```
<get-config>  
<edit-config> (operation="create")  
<edit-config> (operation="replace")  
<edit-config> (operation="merge")  
<edit-config> (operation="delete")
```

The typical workflow when using the network CLI in NSO is as follows:

- All changes are initially made to a (logical) copy of the NSO database of configurations.
- You can view and verify the changes prior to committing them.
- The changes are committed, meaning that the changes are copied to the NSO database and pushed out to the network. Changes that violate integrity constraints or network policies are not committed. The changes to the devices are done in a holistic distributed atomic transaction, across all devices in parallel.

Changes either succeed and remain committed or fail and are rolled back as a whole, returning the entire network to the uncommitted state.

Cisco NSO CHCO FP – Overview and Architecture

Cisco Crosswork Hierarchical Controller is a solution that enables communication service providers (CSPs) to efficiently build and operate transport networks to deliver strong customer experience.

Cisco converges the IP and optical layers so routers can natively connect to a standardized pluggable coherent optics. CHCO is the ideal solution to improve the economics to operate such a network and to simplify the operations of the multi-layer and multi-vendor infrastructure. This solution also empowers operations with speed to orchestrate services and automate assurance across converged networks.

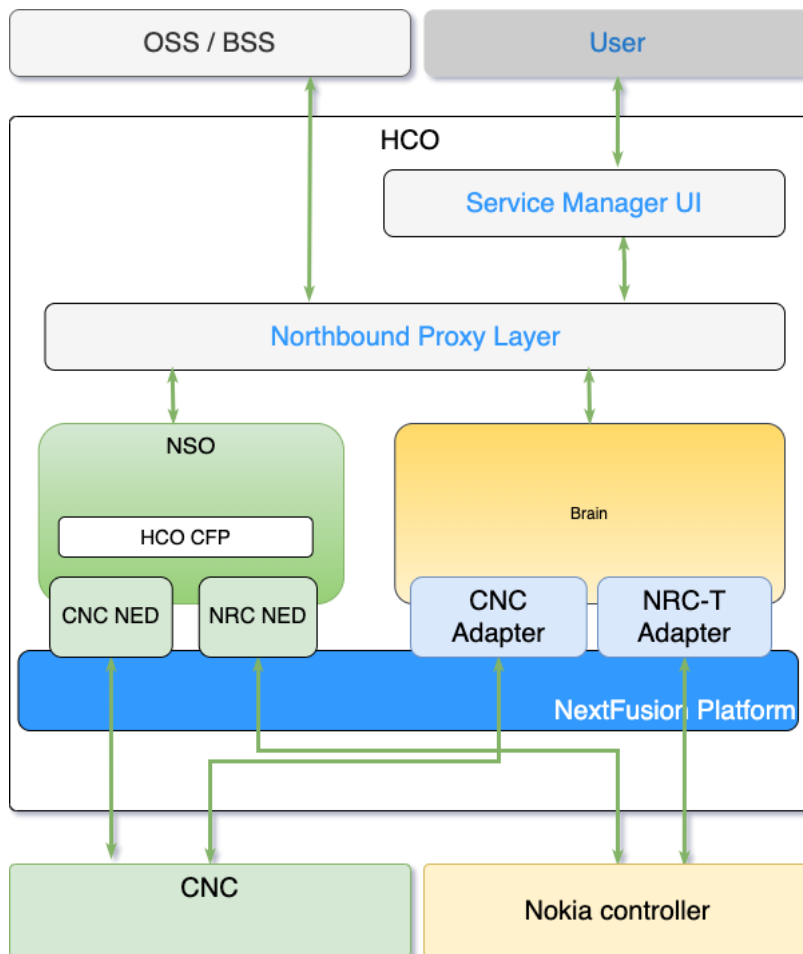
Crosswork Hierarchical Controller (CHCO) unifies complete network visibility from the service layer to the underlying topology, service discovery, and service assurance to control the network seamlessly in a multi-layer and multi-vendor environment. CHCO discovers the network into its own hierarchical and structured model. The network inventory is pulled from multiple domain controllers to construct a consistent multi-vendor view. The service health is determined and optimized by using real time and historical data, and by guided actions to provision services and automate assurance. This helps CHCO to correlate failures in different layers, optimize services, and intelligently provide assurance.

CHCO Function Pack integrates Cisco NSO with a controller to deploy services on the controller. This integration is with either a Nokia Service Provider (NSP) controller or a Cisco Crosswork Network Controller (CNC). NSO communicates with the controller through a Representational State Transfer Configuration Protocol (RESTCONF) and the Network Element Drivers (NED) for the controller.

The function pack uses Layer 2 Network Model (L2NM) and Layer 3 Network Model (L3NM) IETF standard data models as the NBI service models. The data models are same as the controller data models and have the following specifications. For more information about these models, see the IETF documentation.

- RFC 9291: A YANG Network Data Model for Layer 2 VPNs.
- RFC 9182: A YANG Network Data Model for Layer 3 VPNs
- RFC 9181: A Common YANG Data Model for Layer 2 and Layer 3 VPNs

The following diagram explains the architecture and process flow of CHCO FP.



Service requests are submitted to NSO as service through the L2VPN and/or L3VPN data models. Each service deployment must specify a controller to which to push the service to. You can also define a global default controller. The FP interacts with the chosen controller NED in NSO to realize the services on the controller.

Each service deployment can specify a controller to which to push the service to. You can also define a global default controller.

The service submission to NSO and passing the processed request to the controller is asynchronous. The transaction is successful if the controller accepts the service. If the controller encounters any issues, the transaction aborts and the service commit on NSO fails. The L2NM and the L3NM service plans show the deployment status of the services.

For NSP controller, the Kafka monitor watches the service deployment and can be set up to push Kafka notifications. The plan remains in the **not reached** state until the Kafka monitor sends a notification to the NSP controller. The plan displays a **ready reached** state when the controller receives the deployment notification. Similarly, for CNC controller, the RESTCONF notification indicates the service deployment and the service plan displays the **ready reached** state once the controller receives the deployment notification.

Setting up Controllers

CHCO FP provides a single pane to configure different controllers, such as Nokia NSP controller and Cisco CNC controller. CHCO FP can be configured to use either the NSP controller or the CNC controller. Once the controller type is configured, set up a default controller at the global level or choose to use a specific controller at the service level to push the configuration to. Make sure the controller is configured before proceeding to use CHCO FP. For more information on configuring the Nokia NSP controller and Cisco CNC controller, see *Nokia* documentation and *Cisco CNC* documentation respectively.

Onboard CNC Controller To NSO

To set up the CNC controller:

1. Onboard the controller as a device to the NSO device tree.

NSO uses the **cisco-cnc_rc-gen-1.0** NED to onboard the CNC controller to the NSO device tree.

The following are sample payloads with sample NED settings and configuration. To onboard the controller, load merge the payload for the controller to NSO. For detailed information about NED settings for your environment, see the **/nso-6.1.4-hco-2.0.0/function-packages/<NED_file>/README** file located in the CHCO FP installation directory.

Sample authentication to authenticate onboarding a controller

The following is a sample authentication payload to authenticate the controller onboarding process. The auth group name **cnc-auth** is used to authenticate, the remote username and password..

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <authgroups>
      <group>
        <name>cnc_auth</name>
        <default-map>
          <remote-name>$name</remote-name>
          <remote-password>$password</remote-password>
        </default-map>
      </group>
```

```

    </authgroups>
  </devices>
</config>

```

Sample payload to onboard CNC controller

The following is a sample payload to onboard the CNC controller **CNC**. Make sure to modify the controller IP address and the authentication information according to your device.

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <device>
      <name>CNC</name>
      <address>$IP_address</address>
      <port>30603</port>
      <authgroup>cnc_auth</authgroup>
      <device-type>
        <generic>
          <ned-id xmlns:cisco-cnc_rc-gen-1.0="http://tail-f.com/ns/ned-id/cisco-cnc_rc-gen-1.0">cisco-cnc_rc-gen-1.0:cisco-cnc_rc-gen-1.0</ned-id>
        </generic>
      </device-type>
      <ned-settings>
        <cisco-cnc_rc xmlns="http://tailf.com/ned/cisco-cnc_rc/meta">
          <connection>
            <ssl>
              <accept-any>true</accept-any>
            </ssl>
          </connection>
          <restconf>
            <url-base>/crosswork/proxy/nso/restconf</url-base>
            <cache>
              <model>enabled</model>
              <capability>enabled</capability>
            </cache>
            <notif>
              <automatic-stream-discovery>enabled</automatic-stream-
discovery>
              <preferred-encoding>json</preferred-encoding>
            </notif>
          </restconf>
        </cisco-cnc_rc>
      </ned-settings>
    </device>
  </devices>
</config>

```

```

    </cisco-cnc_rc>
</ned-settings>
<state>
  <admin-state>unlocked</admin-state>
</state>
<notifications>
  <subscription>
    <name>service-state-changes</name>
    <stream>service-state-changes</stream>
    <local-user>admin</local-user>
    <store-in-cdb>true</store-in-cdb>
  </subscription>
</notifications>
</device>
</devices>
</config>

```

2. Verify the controller is onboarded as a device to the NSO device tree. The output displays the controller name and the corresponding NED.

```
admin@ncs% run show devices list
```

3. Connect the hierarchical controller to authenticate the controller and perform a sync-from to sync information from the controller to NSO. The device and NSO must be in-sync to deploy the L2VPN/L3VPN services.

```
admin@ncs% request devices device <controller> connect
admin@ncs% request devices device <controller> sync-from
```

Example

```
admin@ncs% request devices device cnc connect
result true
info (admin) Connected to cnc - 10.10.0.1:30603

admin@ncs% request devices device cnc sync-from
result true
```

4. Set up a default controller to push the service configuration. A controller name can either be specified in the service or it can be configured as a default controller at the global level to use in both L2VPN and L3VPN services. If configured as a default controller, it is not required to specify the controller in the service payload. The controller specified in a service overrides the default controller.

Example

```

admin@ncs% set hco-nm-settings controller-type CNC
[ok]
admin@ncs% set hco-nm-settings default-hco-controller <cnc-device-name>
[ok]
admin@ncs% commit
Commit complete.
[ok]
admin@ncs% show hco-nm-settings | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <hco-nm-settings xmlns="http://cisco.com/ns/nso/fp/cisco-hco-nm">
    <controller-type>CNC</controller-type>
    <default-hco-controller>cnc</default-hco-controller>
  </hco-nm-settings>
</config>

```

5. Update the **ncs.conf** file on NSO to add **<keepalive-timeout>** parameter and its settings as follows and restart NSO.

```

<webui>
  <enabled>true</enabled>
  <transport>
    <tcp>
      <enabled>true</enabled>
      <ip>0.0.0.0</ip>
      <port>8080</port>
      <keepalive-timeout>600</keepalive-timeout>
      <keepalive>true</keepalive>
    </tcp>
    <ssl>
      <enabled>false</enabled>
      <ip>0.0.0.0</ip>
      <port>8888</port>
      <keepalive-timeout>600</keepalive-timeout>
      <keepalive>true</keepalive>
      <key-file>${NCS_DIR}/etc/ncs/ssl/cert/host.key</key-
file>
      <cert-
file>${NCS_DIR}/etc/ncs/ssl/cert/host.cert</cert-file>
      <!-- <ha-primary-listen> -->
      <!-- <ip>0.0.0.0</ip> -->
      <!-- <port>28888</port> -->

```

```

        <!-- </ha-primary-listen> -->
    </ssl>
</transport>

```

Onboarding the NSP Controller to NSO

Do the following to set up the NSP controller.

1. Onboard the controller as a device to the NSO device tree. NSO uses the **nokia-nsp_rc-gen-1.0** NED to onboard the NSP controller to the NSO device tree.

The following is a sample payload with sample NED settings and configuration.

To onboard the controller, load merge the payload to NSO. Make sure to modify the controller IP address and the authentication information according to your device.

For detailed information about NED settings for your environment, see the **/nso-6.1.4-hco-2.0.0/function-packages/<NED_file>/README** file located in the CHCO FP installation directory.

Sample authentication to authenticate onboarding a controller

The following is a sample authentication payload to authenticate the controller onboarding process.

The auth group name **nsp-auth** is used to authenticate, the remote username and password.

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <authgroups>
      <group>
        <name>nsp-auth</name>
        <default-map>
          <remote-name>user name</remote-name>
          <remote-password>password</remote-password>
        </default-map>
      </group>
    </authgroups>
  </devices>
</config>

```

Sample payload to onboard NSP controller

```

<devices xmlns="http://tail-f.com/ns/ncs">
  <device>
    <name>nsp-0</name> <!-- set the desired name -->

```

```

<address>10.10.0.1</address> <!-- nsp controller IP address -->
<port>7888</port> <!-- nsp controller restconf port -->
<authgroup>nsp-auth</authgroup> <!-- authgroup to authenticate to NSP-->
<device-type>
  <generic>
    <ned-id xmlns:nokia-nsp_rc-gen-1.0="http://tail-f.com/ns/ned-
id/nokia-nsp_rc-gen-1.0">nokia-nsp_rc-gen-1.0:nokia-nsp_rc-gen-1.0</ned-id>
    </generic>
  </device-type>
<state>
  <admin-state>unlocked</admin-state>
</state>
<ned-settings> <!-- for more on the ned settings please refer to ned
readme -->
  <nokia-nsp_rc xmlns="http://tailf.com/ned/nokia-nsp_rc/meta">
    <connection>
      <authentication>
        <method>basic</method>
      </authentication>
      <ssl>
        <accept-any>true</accept-any>
      </ssl>
    </connection>
    <live-status>
      <time-to-live>30</time-to-live>
    </live-status>
    <restconf>
      <profile>nokia-nsp</profile>
    </restconf>
  </nokia-nsp_rc>
</ned-settings>
</device>
</devices>

```

2. Verify the controller is onboarded as a device to the NSO device tree. The output displays the controller name and the corresponding NED.

```
admin@ncs% run show devices list
```

3. Connect the hierarchical controller to authenticate the controller and perform a sync-from to sync information from the controller to NSO. The device and NSO must be in-sync to deploy the L2VPN/L3VPN services.

```
admin@ncs% request devices device <controller> connect
admin@ncs% request devices device <controller> sync-from
```

Example

```
admin@ncs% request devices device nsp-0 connect
result true
info (admin) Connected to nsp-0 - 10.10.0.1:7888

admin@ncs% request devices device nsp-0 sync-from
result true
```

4. Perform the mandatory NED settings as follows for the controller. The controller nsp-0 is chosen here for demonstration. These settings are required to be able to make modifications at the device level for existing services and also to delete services.

```
admin@ncs> show configuration devices device nsp-0 ned-settings nokia-nsp_rc
restconf config custom-set-endpoints

custom-set-endpoints PATCH /[^:]*:?l2vpn-ntw/vpn-services/vpn-service[^/]+;
custom-set-endpoints PATCH /[^:]*:?l3vpn-ntw/vpn-services/vpn-service[^/]+;

custom-set-endpoints PATCH /[^:]*:l2vpn-ntw/vpn-services/vpn-
service[^/]+/vpn-nodes/vpn-node[^/]+/signaling-option;

custom-set-endpoints PATCH /[^:]*:l3vpn-ntw/vpn-services/vpn-
service[^/]+/vpn-nodes/vpn-node[^/]+;

custom-set-endpoints DELETE /[^:]*:?l2vpn-ntw/vpn-services/vpn-
service[^/]+/underlay-transport;

custom-set-endpoints DELETE /[^:]*:?l2vpn-ntw/vpn-services/vpn-
service[^/]+/vpn-nodes/vpn-node[^/]+/signaling-option;

custom-set-endpoints DELETE /[^:]*:?l3vpn-ntw/vpn-services/vpn-
service[^/]+/vpn-nodes/vpn-node[^/]+/vpn-network-accesses/vpn-network-
access[^/]+/connection/encapsulation/dot1q;

custom-set-endpoints DELETE /[^:]*:?l3vpn-ntw/vpn-services/vpn-
service[^/]+/vpn-nodes/vpn-node[^/]+/vpn-network-accesses/vpn-network-
access[^/]+/connection/encapsulation/priority-tagged;

admin@ncs> show configuration devices device <controller> ned-settings
nokia-nsp_rc restconf profile
profile nokia-nsp;
```

5. Set up a default controller to push the service configuration. A controller name can either be specified in the service or it can be configured as a default controller at the global level to use in both L2VPN and L3VPN services. If configured as a default controller, it is not required to specify the controller in the service payload. The controller specified in a service overrides the default controller.

Example


```
admin@ncs% set hco-nm-settings controller-type NSP

admin@ncs% set hco-nm-settings default-hco-controller nsp-0
[ok]
[edit]
admin@ncs% commit
Commit complete.
[ok]
[edit]

admin@ncs% show hco-nm-settings default-hco-controller | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <hco-nm-settings xmlns="http://cisco.com/ns/nso/ncs/hco-nm">
    <controller-type>NSP</controller-type>
    <default-hco-controller>nsp-0</default-hco-controller>
  </hco-nm-settings>
</config>
[ok]
```

Service Deployments

The following topics discuss how to create and deploy L2VPN-NM and L3VPN-NM services and push the configuration information to the NSP/CNC controller, which further configures the Nokia/CNC devices respectively.

You must select a controller type to use. For example, select to use a CNC or an NSP controller type. Once you select the controller type to use, you can create multiple controllers of the type to push the services to. A service payload contains device identification information to push the service configuration to. A service can be pushed to only one controller at a time. You can, however, edit the service to modify the controller information and commit again. A single controller can accept multiple services. The service plan displays the status of the deployment.

Provide the controller name either in the service or set it up as a default controller at the global level to use in both the L2VPN and L3VPN services. If set up as a default controller, it is not required to specify the controller in the service payload. For more information on setting up a controller, see [Setting up Controllers](#).

The Kafka settings for NSP controller and the RESTCONF notifications for the CNC controller notify the service creation and thereafter any changes to the service.

Deploying L2VPN Services

This section discusses how to deploy L2VPN service and pushing the configurations to a default controller or to a specific controller mentioned in the service payload.

The following sample payloads demonstrate the service deployment for both NSP and CNC controllers.

Deploying L2VPN Services on an NSP Controller

Deploying L2VPN service on a default nsp controller

Create the L2VPN-NM service and push the configuration to a default NSP controller.

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <l2vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw">
    <vpn-services>
      <vpn-service>
        <vpn-id>PW01-tldp-interAS-A-55000-1130</vpn-id>
        <vpn-name>PW01-tldp-interAS-A-55000-1130</vpn-name>
        <vpn-description>demo</vpn-description>
        <customer-name>1</customer-name>
        <vpn-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:vpws</vpn-type>
        <signaling-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ldp-signaling</signaling-type>
        <underlay-transport>
          <protocol xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ldp</protocol>
          <protocol xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:rsvp-te</protocol>
        </underlay-transport>
        <status>
          <admin-status>
            <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:admin-up</status>
          </admin-status>
        </status>
        <vpn-nodes>
          <vpn-node>
            <vpn-node-id>PW01-tldp-interAS-A-55000-1130_R8</vpn-node-id>
            <description>ietf_l2vpn</description>
            <ne-id>10.10.10.8</ne-id>
            <status>
```

```

    <admin-status>
      <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:admin-up</status>
    </admin-status>
  </status>
  <signaling-option>
    <signaling-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ldp-
signaling</signaling-type>
    <ldp-or-l2tp>
      <t-ldp-pw-type>vpws-type</t-ldp-pw-type>
      <pw-type>ethernet</pw-type>
      <ac-pw-list>
        <peer-addr>10.10.10.9</peer-addr>
        <vc-id>1130</vc-id>
        <pw-priority>1</pw-priority>
      </ac-pw-list>
    </ldp-or-l2tp>
  </signaling-option>
  <vpn-network-accesses>
    <vpn-network-access>
      <id>1</id>
      <interface-id>Port 1/1/1</interface-id>
      <connection>
        <encapsulation>
          <encap-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-
common:dot1q</encap-type>
          <dot1q>
            <cvlan-id>1130</cvlan-id>
          </dot1q>
        </encapsulation>
      </connection>
      <service>
        <mtu>1492</mtu>
      </service>
    </vpn-network-access>
  </vpn-network-accesses>
</vpn-node>
<vpn-node>

```

```

<vpn-node-id>PW01-tldp-interAS-A-55000-1130_R9</vpn-node-id>
<description>ietf_l2vpn</description>
<ne-id>10.10.10.9</ne-id>
<status>
  <admin-status>
    <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:admin-up</status>
  </admin-status>
</status>
<signaling-option>
  <signaling-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ldp-
signaling</signaling-type>
  <ldp-or-l2tp>
    <t-ldp-pw-type>vpws-type</t-ldp-pw-type>
    <pw-type>ethernet</pw-type>
    <ac-pw-list>
      <peer-addr>10.10.10.8</peer-addr>
      <vc-id>1130</vc-id>
      <pw-priority>1</pw-priority>
    </ac-pw-list>
  </ldp-or-l2tp>
</signaling-option>
<vpn-network-accesses>
  <vpn-network-access>
    <id>1</id>
    <interface-id>Port 1/1/1</interface-id>
    <connection>
      <encapsulation>
        <encap-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-
common:dot1q</encap-type>
        <dot1q>
          <cvlan-id>1130</cvlan-id>
        </dot1q>
      </encapsulation>
    </connection>
    <service>
      <mtu>1492</mtu>
    </service>
  </vpn-network-access>
</vpn-network-accesses>

```

```

        </vpn-network-access>
    </vpn-network-accesses>
</vpn-node>
</vpn-nodes>
</vpn-service>
</vpn-services>
</l2vpn-ntw>
</config>

```

Deploying L2VPN-NM service on the specified controller in the service

The following is a sample payload to create the L2VPN-NM service with hco-controller set during service creation. In this example, the service configuration is pushed to the hco-controller **nsp-0** specified in the payload.

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <l2vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw">
    <vpn-services>
      <vpn-service>
        <vpn-id>PW01-tldp-interAS-A-65000-1130</vpn-id>
        <vpn-name>PW01-tldp-interAS-A-65000-1130</vpn-name>
        <vpn-description>demo</vpn-description>
        <customer-name>1</customer-name>
        <vpn-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:vpws</vpn-type>
        <signaling-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:ldp-signaling</signaling-type>
        <underlay-transport>
          <protocol xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:bgp-lu</protocol>
        </underlay-transport>
        <status>
          <admin-status>
            <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
          </admin-status>
        </status>
        <vpn-nodes>
          <vpn-node>
            <vpn-node-id>PW01-tldp-interAS-A-65000-1130_R5</vpn-node-id>
            <description>ietf_l2vpn</description>

```

```

    <ne-id>10.10.10.5</ne-id>
    <status>
      <admin-status>
        <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
      </admin-status>
    </status>
    <signaling-option>
      <signaling-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:bgp-
lu</signaling-type>
    </signaling-option>
    <vpn-network-accesses>
      <vpn-network-access>
        <id>1</id>
        <interface-id>Port 1/1/13</interface-id>
        <connection>
          <encapsulation>
            <encap-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:dot1q</encap-
type>
              <dot1q>
                <cvlan-id>1130</cvlan-id>
              </dot1q>
            </encapsulation>
          </connection>
          <service>
            <mtu>1492</mtu>
          </service>
        </vpn-network-access>
      </vpn-network-accesses>
    </vpn-node>
  </vpn-node>
  <vpn-node-id>PW01-tldp-interAS-A-65000-1130_R6</vpn-node-id>
  <description>ietf_l2vpn</description>
  <ne-id>10.10.10.6</ne-id>
  <status>
    <admin-status>
      <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
  </status>

```

```

        </admin-status>
    </status>
    <signaling-option>
        <signaling-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:bgp-
lu</signaling-type>
    </signaling-option>
    <vpn-network-accesses>
        <vpn-network-access>
            <id>1</id>
            <interface-id>Port 1/1/3</interface-id>
            <connection>
                <encapsulation>
                    <encap-type xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:dot1q</encap-
type>
                    <dot1q>
                        <cvlan-id>1130</cvlan-id>
                    </dot1q>
                </encapsulation>
            </connection>
            <service>
                <mtu>1492</mtu>
            </service>
        </vpn-network-access>
    </vpn-network-accesses>
</vpn-node>
</vpn-nodes>
    <hco-controller xmlns="http://cisco.com/ns/nso/fp/cisco-hco-nm">nsp-
0</hco-controller>
</vpn-service>
</vpn-services>
</l2vpn-ntw>
</config>

```

6. Load merge the payload and commit the transaction to create the service. The following is an example.

```

admin@ncs% load merge l2vpn-nsp-file-name.xml
[ok]

```

```
[edit]
admin@ncs% commit
Commit complete.
[ok]
```

7. Display the service plan to view and verify the status of the service. The following is an example.

```
admin@ncs% run show l2vpn-ntw vpn-services vpn-service-plan PW01-tldp-
interAS-A-65000-1130 plan
```

TYPE	NAME	BACK		STATE	STATUS	WHEN	POST ACTION			
		TRACK	GOAL				ref	STATUS		
self	self	false	-	init			reached	2023-10-12T16:55:32	-	-
-				cisco-hco-l2vpn-nano:config-apply			reached	2023-10-12T16:55:32	-	-
-				cisco-hco-l2vpn-nano:provisioned			not-reached	-	-	-
		WAITING FOR KAFKA NOTIFICATION		ready			not-reached	-	-	-
-										

```
[ok]
```

The plan is in the **not-reached** state when it is awaiting Kafka notifications from the nsp device. The provisioned state indicates if the plan received the Kafka notifications. The plan updates to reached state upon receiving the Kafka notifications. A successful plan in the ready reached state indicates the l2vpn-service is pushed to the NSP controller.

The timeout value is set in the **kafka-notif-timeout** leaf in the Kafka monitor.

```
admin@ncs% set hco-nm-settings kafka-notif-timeout 360
```

If the Kafka notification is not received within this set timeout timeframe, the plan fails for service creation, updation, and/or deletion.

Deploying L2VPN Services Using CNC Controller

Deploying L2VPN service on a default cnc-controller

Create the L2VPN-NM service and push the configuration to a default cnc-controller.

```
<l2vpn-ntw
  xmlns="urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw">
  <vpn-services>
    <vpn-service>
      <vpn-id>l2nm-p2p-test</vpn-id>
      <vpn-type>vpn-common:vpws</vpn-type>
      <vpn-nodes>
        <vpn-node>
          <vpn-node-id>PCE1</vpn-node-id>
          <signaling-option>
```



```
<ldp-or-l2tp>
  <pw-peer-list>
    <peer-addr>50.0.252.2</peer-addr>
    <vc-id>1001</vc-id>
  </pw-peer-list>
</ldp-or-l2tp>
</signaling-option>
<vpn-network-accesses>
  <vpn-network-access>
    <id>l2vpn-p2p-ac1</id>
    <interface-id>GigabitEthernet0/0/0/1</interface-id>
    <connection>
      <encapsulation>
        <encap-type>vpn-common:dot1q</encap-type>
        <dot1q>
          <cvlan-id>601</cvlan-id>
          <tag-operations>
            <push/>
            <tag-1>123</tag-1>
          </tag-operations>
        </dot1q>
      </encapsulation>
    </connection>
    <service>
      <mtu>65</mtu>
    </service>
  </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
  <vpn-node-id>PE3</vpn-node-id>
  <signaling-option>
    <ldp-or-l2tp>
      <pw-peer-list>
        <peer-addr>50.0.252.3</peer-addr>
        <vc-id>1001</vc-id>
      </pw-peer-list>
    </ldp-or-l2tp>
```

```

    </signaling-option>
  <vpn-network-accesses>
    <vpn-network-access>
      <id>l2vpn-p2p-ac1</id>
      <interface-id>GigabitEthernet0/0/0/1</interface-id>
      <connection>
        <encapsulation>
          <encap-type>vpn-common:dot1q</encap-type>
          <dot1q>
            <cvlan-id>601</cvlan-id>
            <tag-operations>
              <push/>
              <tag-1>234</tag-1>
            </tag-operations>
          </dot1q>
        </encapsulation>
      </connection>
      <service>
        <mtu>64</mtu>
      </service>
    </vpn-network-access>
  </vpn-network-accesses>
</vpn-node>
</vpn-nodes>
</vpn-service>
</vpn-services>
</l2vpn-ntw>
</config>

```

Deploying L2VPN-NM service on a CNC controller specified in the service.

The following is a sample payload to create the L2VPN-NM service with CNC controller set during service creation. In this example, the service configuration is pushed to the cnc controller specified in the payload.

```

<l2vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw">
  <vpn-services>
    <vpn-service>
      <vpn-id>l2nm-vpws</vpn-id>
      <vpn-type>vpn-common:vpws</vpn-type>
    </vpn-service>
  </vpn-services>
</l2vpn-ntw>

```

```
<vpn-node>
  <vpn-node-id>PCE1</vpn-node-id>
  <signaling-option>
    <ldp-or-l2tp>
      <pw-peer-list>
        <peer-addr>50.0.252.2</peer-addr>
        <vc-id>1001</vc-id>
      </pw-peer-list>
    </ldp-or-l2tp>
  </signaling-option>
  <vpn-network-accesses>
    <vpn-network-access>
      <id>l2vpn-p2p-ac1</id>
      <interface-id>GigabitEthernet0/0/0/1</interface-id>
      <connection>
        <encapsulation>
          <encap-type>vpn-common:dot1q</encap-type>
          <dot1q>
            <cvlan-id>601</cvlan-id>
            <tag-operations>
              <push />
              <tag-1>123</tag-1>
            </tag-operations>
          </dot1q>
        </encapsulation>
      </connection>
      <service>
        <mtu>65</mtu>
      </service>
    </vpn-network-access>
  </vpn-network-accesses>
</vpn-node>
<vpn-node>
  <vpn-node-id>PE3</vpn-node-id>
  <signaling-option>
    <ldp-or-l2tp>
      <pw-peer-list>
        <peer-addr>50.0.252.3</peer-addr>
```

```

        <vc-id>1001</vc-id>
    </pw-peer-list>
</ldp-or-l2tp>
</signaling-option>
<vpn-network-accesses>
  <vpn-network-access>
    <id>l2vpn-p2p-ac1</id>
    <interface-id>GigabitEthernet0/0/0/1</interface-id>
    <connection>
      <encapsulation>
        <encap-type>vpn-common:dot1q</encap-type>
        <dot1q>
          <cvlan-id>601</cvlan-id>
          <tag-operations>
            <push />
            <tag-1>234</tag-1>
          </tag-operations>
        </dot1q>
      </encapsulation>
    </connection>
  </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
</vpn-nodes>
  <hco-controller xmlns="http://cisco.com/ns/nso/fp/cisco-hco-nm">cnc-
0</hco-controller>
</vpn-service>
</vpn-services>
</l2vpn-ntw>

```

1. Load merge the payload and commit the transaction to create the service. The following is an example.

```

admin@ncs% load merge l2_cnc.xml
[ok]
[edit]
admin@ncs% commit

```

```
Commit complete.
```

```
[ok]
```

2. Display the service plan to view and verify the status of the service. The following is an example.

```
admin@ncs% run show l3vpn-ntw vpn-services vpn-service-plan l2nm-vpws plan |
tab
```

TYPE	NAME	BACK TRACK	GOAL	STATE	STATUS	WHEN	ref	POST ACTION STATUS
self	self	false	-	init	reached	2023-11-18T01:24:48	-	-
-	-	-	-	cisco-hco-l2vpn-nano:config-apply	reached	2023-11-18T01:24:48	-	-
-	-	-	-	cisco-hco-l2vpn-nano:provisioned	not-reached	-	-	-
WAITING FOR CNC NOTIFICATION				ready	not-reached	-	-	-
-	[ok]	-	-	-	-	-	-	-

The plan is in the **not-reached** state when it is awaiting CNC notifications from the CNC device. The provisioned state indicates if the plan received the CNC notifications and updates to **reached** state upon receiving the CNC notifications. A successful plan in the **ready reached** state indicates the l2vpn-service is pushed to the CNC controller.

Deploying L3VPN Services

This section discusses how to deploy the L3VPN service using a default controller or to a specific controller mentioned in the service payload.

1. The following sample payloads demonstrate the service deployment for both NSP and CNC controllers.

Deploying L3VPN Services on an NSP Controller

Deploying L3VPN service on a default nsp controller

Create the L3VPN-NM service and push the configuration to a default nsp controller.

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <l3vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw">
    <vpn-services>
      <vpn-service>
        <vpn-id>L3VPN02-single-instance-2210</vpn-id>
        <vpn-name>L3VPN02-single-instance-2210</vpn-name>
        <vpn-description>L3VPN02-single-instance-2210</vpn-description>
        <customer-name>1</customer-name>
      </vpn-service>
    </vpn-services>
  </l3vpn-ntw>
</config>
```

```

    <vpn-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:l3vpn</vpn-type>
    <vpn-service-topology xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:hub-spoke</vpn-
service-topology>
    <underlay-transport>
    <protocol xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:sr-mpls</protocol>
    </underlay-transport>
    <vpn-nodes>
    <vpn-node>
    <vpn-node-id>L3VPN02-single-instance-2210-1</vpn-node-id>
    <description>L3VPN02-single-instance-2210</description>
    <ne-id>10.10.10.1</ne-id>
    <router-id>10.10.10.1</router-id>
    <active-vpn-instance-profiles>
    <vpn-instance-profile>
    <profile-id>profile1</profile-id>
    <rd>0:65000:2210</rd>
    <address-family>
    <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
    <vpn-targets>
    <vpn-policies>
    <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
    <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
    </vpn-policies>
    </vpn-targets>
    </address-family>
    </vpn-instance-profile>
    </active-vpn-instance-profiles>
    <status>
    <admin-status>
    <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
    </status>
    <vpn-network-accesses>

```

```

    <vpn-network-access>
      <id>int_2210_1</id>
      <interface-id>1/1/c1/1</interface-id>
      <description>L3VPN02-single-instance-2210</description>
      <connection>
        <encapsulation>
          <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
          <dot1q>
            <cvlan-id>2210</cvlan-id>
          </dot1q>
        </encapsulation>
      </connection>
      <ip-connection>
        <ipv4>
          <local-address>1.1.1.1</local-address>
          <prefix-length>24</prefix-length>
        </ipv4>
      </ip-connection>
      <service>
        <mtu>1492</mtu>
      </service>
    </vpn-network-access>
  </vpn-network-accesses>
</vpn-node>
<vpn-node>
  <vpn-node-id>L3VPN02-single-instance-2210-2</vpn-node-id>
  <description>L3VPN02-single-instance-2210</description>
  <ne-id>10.10.10.2</ne-id>
  <router-id>10.10.10.2</router-id>
  <active-vpn-instance-profiles>
    <vpn-instance-profile>
      <profile-id>profile1</profile-id>
      <rd>0:65000:2210</rd>
      <address-family>
        <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
      <vpn-targets>

```

```

        <vpn-policies>
            <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
            <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
        </vpn-policies>
    </vpn-targets>
</address-family>
</vpn-instance-profile>
</active-vpn-instance-profiles>
<status>
    <admin-status>
        <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
</status>
<vpn-network-accesses>
    <vpn-network-access>
        <id>int_2210_1</id>
        <interface-id>1/1/6</interface-id>
        <description>L3VPN02-single-instance-2210</description>
        <ip-connection>
            <ipv4>
                <local-address>2.2.2.2</local-address>
                <prefix-length>24</prefix-length>
            </ipv4>
        </ip-connection>
        <service>
            <mtu>1492</mtu>
        </service>
    </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
    <vpn-node-id>L3VPN02-single-instance-2210-6</vpn-node-id>
    <description>L3VPN02-single-instance-2210</description>
    <ne-id>10.10.10.6</ne-id>
    <router-id>10.10.10.6</router-id>
    <active-vpn-instance-profiles>

```



```

    <vpn-instance-profile>
      <profile-id>profile1</profile-id>
      <rd>0:65000:2210</rd>
      <address-family>
        <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
          <vpn-targets>
            <vpn-policies>
              <import-policy>L3VPN02-single-instance-2210-
IMPORT</import-policy>
              <export-policy>L3VPN02-single-instance-2211-
EXPORT</export-policy>
            </vpn-policies>
          </vpn-targets>
        </address-family>
      </vpn-instance-profile>
    </active-vpn-instance-profiles>
    <status>
      <admin-status>
        <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
      </admin-status>
    </status>
    <vpn-network-accesses>
      <vpn-network-access>
        <id>int_2210_1</id>
        <interface-id>1/1/10</interface-id>
        <description>L3VPN02-single-instance-2210</description>
        <connection>
          <encapsulation>
            <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
            <dot1q>
              <cvlan-id>2210</cvlan-id>
            </dot1q>
          </encapsulation>
        </connection>
        <ip-connection>
          <ipv4>

```

```

        <local-address>6.6.6.6</local-address>
        <prefix-length>24</prefix-length>
    </ipv4>
</ip-connection>
<service>
    <mtu>1492</mtu>
</service>
</vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
    <vpn-node-id>L3VPN02-single-instance-2210-7</vpn-node-id>
    <description>L3VPN02-single-instance-2210</description>
    <ne-id>10.10.10.7</ne-id>
    <router-id>10.10.10.7</router-id>
    <active-vpn-instance-profiles>
        <vpn-instance-profile>
            <profile-id>profile1</profile-id>
            <rd>0:65000:2210</rd>
            <address-family>
                <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
            </address-family>
            <vpn-targets>
                <vpn-policies>
                    <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
                    <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
                </vpn-policies>
            </vpn-targets>
        </address-family>
    </vpn-instance-profile>
</active-vpn-instance-profiles>
<status>
    <admin-status>
        <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
</status>

```

```

    <vpn-network-accesses>
      <vpn-network-access>
        <id>int_2210_1</id>
        <interface-id>1/1/10</interface-id>
        <description>L3VPN02-single-instance-2210</description>
        <connection>
          <encapsulation>
            <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
            <dot1q>
              <cvlan-id>2210</cvlan-id>
            </dot1q>
          </encapsulation>
        </connection>
        <ip-connection>
          <ipv4>
            <local-address>7.7.7.7</local-address>
            <prefix-length>24</prefix-length>
          </ipv4>
        </ip-connection>
        <service>
          <mtu>1492</mtu>
        </service>
      </vpn-network-access>
    </vpn-network-accesses>
  </vpn-node>
</vpn-nodes>
</vpn-service>
</vpn-services>
</l3vpn-ntw>
</config>

```

Creating L3VPN-NM service with NSP controller set during service creation

The following is a sample payload to create the L3VPN-NM service. The service configuration is pushed to the controller specified in the payload during service creation.

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <l3vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw">
    <vpn-services>

```

```

<vpn-service>
  <vpn-id>L3VPN02-single-instance-2210</vpn-id>
  <vpn-name>L3VPN02-single-instance-2210</vpn-name>
  <vpn-description>L3VPN02-single-instance-2210</vpn-description>
  <customer-name>1</customer-name>
  <vpn-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:l3vpn</vpn-type>
  <vpn-service-topology xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:hub-spoke</vpn-
service-topology>
  <underlay-transport>
    <protocol xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:sr-mpls</protocol>
  </underlay-transport>
  <vpn-nodes>
    <vpn-node>
      <vpn-node-id>L3VPN02-single-instance-2210-1</vpn-node-id>
      <description>L3VPN02-single-instance-2210</description>
      <ne-id>10.10.10.1</ne-id>
      <router-id>10.10.10.1</router-id>
      <active-vpn-instance-profiles>
        <vpn-instance-profile>
          <profile-id>profile1</profile-id>
          <rd>0:65000:2210</rd>
          <address-family>
            <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
            <vpn-targets>
              <vpn-policies>
                <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
                <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
              </vpn-policies>
            </vpn-targets>
          </address-family>
        </vpn-instance-profile>
      </active-vpn-instance-profiles>
      <status>

```

```

    <admin-status>
      <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
  </status>
  <vpn-network-accesses>
    <vpn-network-access>
      <id>int_2210_1</id>
      <interface-id>1/1/c1/1</interface-id>
      <description>L3VPN02-single-instance-2210</description>
      <connection>
        <encapsulation>
          <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
          <dot1q>
            <cvlan-id>2210</cvlan-id>
          </dot1q>
        </encapsulation>
      </connection>
      <ip-connection>
        <ipv4>
          <local-address>1.1.1.1</local-address>
          <prefix-length>24</prefix-length>
        </ipv4>
      </ip-connection>
      <service>
        <mtu>1492</mtu>
      </service>
    </vpn-network-access>
  </vpn-network-accesses>
</vpn-node>
<vpn-node>
  <vpn-node-id>L3VPN02-single-instance-2210-2</vpn-node-id>
  <description>L3VPN02-single-instance-2210</description>
  <ne-id>10.10.10.2</ne-id>
  <router-id>10.10.10.2</router-id>
  <active-vpn-instance-profiles>
    <vpn-instance-profile>
      <profile-id>profile1</profile-id>
    </vpn-instance-profile>
  </active-vpn-instance-profiles>
</vpn-node>

```

```

        <rd>0:65000:2210</rd>
        <address-family>
            <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>

            <vpn-targets>
                <vpn-policies>
                    <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
                    <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
                </vpn-policies>
            </vpn-targets>
        </address-family>
    </vpn-instance-profile>
</active-vpn-instance-profiles>
<status>
    <admin-status>
        <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
    </admin-status>
</status>
<vpn-network-accesses>
    <vpn-network-access>
        <id>int_2210_1</id>
        <interface-id>1/1/6</interface-id>
        <description>L3VPN02-single-instance-2210</description>
        <ip-connection>
            <ipv4>
                <local-address>2.2.2.2</local-address>
                <prefix-length>24</prefix-length>
            </ipv4>
        </ip-connection>
        <service>
            <mtu>1492</mtu>
        </service>
    </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>

```

```

<vpn-node-id>L3VPN02-single-instance-2210-6</vpn-node-id>
<description>L3VPN02-single-instance-2210</description>
<ne-id>10.10.10.6</ne-id>
<router-id>10.10.10.6</router-id>
<active-vpn-instance-profiles>
  <vpn-instance-profile>
    <profile-id>profile1</profile-id>
    <rd>0:65000:2210</rd>
    <address-family>
      <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
        <vpn-targets>
          <vpn-policies>
            <import-policy>L3VPN02-single-instance-2210-
IMPORT</import-policy>
            <export-policy>L3VPN02-single-instance-2211-
EXPORT</export-policy>
          </vpn-policies>
        </vpn-targets>
      </address-family>
    </vpn-instance-profile>
  </active-vpn-instance-profiles>
<status>
  <admin-status>
    <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
  </admin-status>
</status>
<vpn-network-accesses>
  <vpn-network-access>
    <id>int_2210_1</id>
    <interface-id>1/1/10</interface-id>
    <description>L3VPN02-single-instance-2210</description>
    <connection>
      <encapsulation>
        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
        <dot1q>
          <cvlan-id>2210</cvlan-id>

```

```

        </dot1q>
    </encapsulation>
</connection>
<ip-connection>
    <ipv4>
        <local-address>6.6.6.6</local-address>
        <prefix-length>24</prefix-length>
    </ipv4>
</ip-connection>
<service>
    <mtu>1492</mtu>
</service>
</vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
    <vpn-node-id>L3VPN02-single-instance-2210-7</vpn-node-id>
    <description>L3VPN02-single-instance-2210</description>
    <ne-id>10.10.10.7</ne-id>
    <router-id>10.10.10.7</router-id>
    <active-vpn-instance-profiles>
        <vpn-instance-profile>
            <profile-id>profile1</profile-id>
            <rd>0:65000:2210</rd>
            <address-family>
                <address-family xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:ipv4</address-
family>
                <vpn-targets>
                    <vpn-policies>
                        <import-policy>L3VPN02-single-instance-2211-
IMPORT</import-policy>
                        <export-policy>L3VPN02-single-instance-2210-
EXPORT</export-policy>
                    </vpn-policies>
                </vpn-targets>
            </address-family>
        </vpn-instance-profile>
    </active-vpn-instance-profiles>

```



```
<status>
  <admin-status>
    <status xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-
common">vpn-common:admin-up</status>
  </admin-status>
</status>
<vpn-network-accesses>
  <vpn-network-access>
    <id>int_2210_1</id>
    <interface-id>1/1/10</interface-id>
    <description>L3VPN02-single-instance-2210</description>
    <connection>
      <encapsulation>
        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:dot1q</type>
        <dot1q>
          <cvlan-id>2210</cvlan-id>
        </dot1q>
      </encapsulation>
    </connection>
    <ip-connection>
      <ipv4>
        <local-address>7.7.7.7</local-address>
        <prefix-length>24</prefix-length>
      </ipv4>
    </ip-connection>
    <service>
      <mtu>1492</mtu>
    </service>
  </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
</vpn-nodes>
<hco-controller xmlns="http://cisco.com/ns/nso/fp/cisco-hco-nm">cnc-
0</hco-controller>
</vpn-service>
</vpn-services>
</l3vpn-ntw>
</config>
```

2. Load merge the payload and commit the transaction to create the service. The following is an example.

```
admin@ncs% load merge L3-NSP_payload.xml
[ok]
admin@ncs% commit
Commit complete.
[ok]
```

3. Display the service plan to view and verify the status of the service as shown in the following example.

```
admin@ncs% run show l3vpn-ntw vpn-services vpn-service-plan L3VPN02-single-
instance-2210 plan
```

TYPE	NAME	BACK TRACK	GOAL	STATE	STATUS	WHEN	ref	POST ACTION STATUS	MESSAGE
self	self	false	-	init	reached	2023-10-12T16:55:32	-	-	-
				cisco-hco-l3vpn-nano:config-apply	reached	2023-10-12T16:55:32	-	-	-
				cisco-hco-l3vpn-nano:provisioned	not-reached	-	-	-	WAITING FOR KAFKA
NOTIFICATION				ready	not-reached	-	-	-	

The plan is in the **not-reached** state when it is awaiting Kafka notifications from the nsp device. The provisioned state indicates if the plan received the Kafka notifications. The plan updates to reached state upon receiving the Kafka notifications. A successful plan in the ready-reached state indicates the l3vpn-service definition is pushed onto the NSP controller.

The timeout value to receive the Kafka notification is set in the kafka-notif-timeout leaf in the Kafka monitor.

The timeout value is set in the **kafka-notif-timeout** leaf in the Kafka monitor.

```
admin@ncs% set hco-nm-settings kafka-notif-timeout 360
```

If the Kafka notification is not received within this set timeout timeframe, the plan fails for service creation, updation, and/or deletion.

Deploying L3VPN-NM service with CNC controller set during service creation.

The following is a sample payload to create the L3VPN-NM service with CNC controller set during service creation. In this example, the service configuration is pushed to the cnc controller specified in the payload.

Load merge the payload and commit the transaction to create the service. The following is an example.

```
admin@ncs% load merge l3.xml
[ok]
```

```
[edit]
admin@ncs% commit
Commit complete.
[ok]
```

4. Display the service plan to view and verify the status of the service. The following is an example.

```
admin@ncs% run show l3vpn-ntw vpn-services vpn-service-plan L3VPN03-multi-
instance-InterAS-A-2330 plan
```

BACK					POST	ACTION				
TYPE	NAME	TRACK	GOAL	STATE	STATUS	WHEN	ref	STATUS	MESSAGE	
self	self	false	-	init	reached	2024-02-22T04:57:35	-	-	-	
				cisco-hco-l3vpn-nano:config-apply	reached	2024-02-22T04:57:35	-	-	-	
				cisco-hco-l3vpn-nano:provisioned	not-reached	-	-	-	WAITING FOR CNC	
NOTIFICATION										
				ready	not-reached	-	-	-	-	

The plan is in the **not-reached** state when it is awaiting CNC notifications from the CNC device. The provisioned state indicates if the plan received the CNC notifications and updates to **reached** state upon receiving the CNC notifications. A successful plan in the **ready reached** state indicates the l2vpn-service is pushed to the NSP controller.

The timeout value is set in the kafka-notif-timeout leaf in the Kafka monitor. If the Kafka notification is not received within this set timeout timeframe, the plan fails for service creation, updation, and/or deletion.

Deploying L3VPN Services on CNC Controller

Deploying L3VPN service on a default CNC controller

Create the L3VPN-NM service and push the configuration to a default CNC controller.

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <l3vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw">
    <vpn-services>
      <vpn-service>
        <vpn-id> L3VPN03-multi-instance-InterAS-A-2330</vpn-id>
        <vpn-instance-profiles>
          <vpn-instance-profile>
            <profile-id>any-to-any-profile</profile-id>
            <rd>0:4445:8</rd>
            <address-family>
              <address-family>ipv4</address-family>
            <vpn-targets>
              <vpn-target>
```

```

        <id>1</id>
        <route-targets>
            <route-target>0:4445:8</route-target>
        </route-targets>
        <route-target-type>both</route-target-type>
    </vpn-target>
</vpn-targets>
</address-family>
</vpn-instance-profile>
</vpn-instance-profiles>
<vpn-nodes>
    <vpn-node>
        <vpn-node-id>PIOSXR-0</vpn-node-id>
        <!-- <description>L3VPN-Acme_Inc-78000</description> -->
        <!-- <ne-id>63.130.123.143</ne-id> -->
        <local-as>4445</local-as>
        <active-vpn-instance-profiles>
            <vpn-instance-profile>
                <profile-id>any-to-any-profile</profile-id>
            </vpn-instance-profile>
        </active-vpn-instance-profiles>
        <vpn-network-accesses>
            <vpn-network-access>
                <id>Acme_Inc:3C0780005</id>
                <interface-id>GigabitEthernet0/0/0/0</interface-id>
                <connection>
                    <encapsulation>
                        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:untagged-int</type>
                    </encapsulation>
                </connection>
                <ip-connection>
                    <ipv4>
                        <local-address>10.0.0.10</local-address>
                        <prefix-length>30</prefix-length>
                    </ipv4>
                </ip-connection>
                <routing-protocols>
                    <routing-protocol>

```

```

        <id>L3VPN-BGP-Acme_Inc</id>
        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:bgp-routing</type>
        <bgp>
            <peer-as>65001</peer-as>
            <address-family>ipv4</address-family>
            <neighbor>10.0.0.9</neighbor>
        </bgp>
    </routing-protocol>
</routing-protocols>
<service>
    <qos>
        <qos-profile>
            <qos-profile>
                <profile>GF-VPN-STANDARD-EGRESS</profile>
                <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:wan-to-
site</direction>
            </qos-profile>
            <qos-profile>
                <profile>GF-VPN-STANDARD-INGRESS</profile>
                <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:site-to-
wan</direction>
            </qos-profile>
        </qos-profile>
    </qos>
</service>
</vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
    <vpn-node-id>PIOSXR-1</vpn-node-id>
    <local-as>4445</local-as>
    <active-vpn-instance-profiles>
        <vpn-instance-profile>
            <profile-id>any-to-any-profile</profile-id>
        </vpn-instance-profile>
    </active-vpn-instance-profiles>
    <vpn-network-accesses>

```

```

    <vpn-network-access>
      <id>Acme_Inc:3C0780005</id>
      <interface-id>GigabitEthernet0/0/0/0</interface-id>
      <connection>
        <encapsulation>
          <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:untagged-int</type>
        </encapsulation>
      </connection>
      <ip-connection>
        <ipv4>
          <local-address>10.0.0.2</local-address>
          <prefix-length>30</prefix-length>
        </ipv4>
      </ip-connection>
      <routing-protocols>
        <routing-protocol>
          <id>L3VPN-BGP-Acme_Inc</id>
          <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:bgp-routing</type>
          <bgp>
            <peer-as>65001</peer-as>
            <address-family>ipv4</address-family>
            <neighbor>10.0.0.1</neighbor>
          </bgp>
        </routing-protocol>
      </routing-protocols>
      <service>
        <qos>
          <qos-profile>
            <qos-profile>
              <profile>GF-VPN-STANDARD-EGRESS</profile>
              <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:wana-to-
site</direction>
            </qos-profile>
          <qos-profile>
            <profile>GF-VPN-STANDARD-INGRESS</profile>

```

```

        <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:site-to-
wan</direction>
        </qos-profile>
        </qos-profile>
        </qos>
        </service>
        </vpn-network-access>
        </vpn-network-accesses>
    </vpn-node>
</vpn-nodes>
</vpn-service>
</vpn-services>
</l3vpn-ntw>
</config>

```

Creating L3VPN-NM service with a CNC controller set during service creation.

The following is a sample payload to create the L3VPN-NM service. The service configuration is pushed to the controller specified in the payload during service creation.

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <l3vpn-ntw xmlns="urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw">
    <vpn-services>
      <vpn-service>
        <vpn-id>L3VPN03-multi-instance-InterAS-A-2330</vpn-id>
        <vpn-instance-profiles>
          <vpn-instance-profile>
            <profile-id>any-to-any-profile</profile-id>
            <rd>0:4445:8</rd>
            <address-family>
              <address-family>ipv4</address-family>
            <vpn-targets>
              <vpn-target>
                <id>1</id>
                <route-targets>
                  <route-target>0:4445:8</route-target>
                </route-targets>
                <route-target-type>both</route-target-type>
              </vpn-target>
            </vpn-targets>
          </vpn-instance-profile>
        </vpn-instance-profiles>
      </vpn-service>
    </vpn-services>
  </l3vpn-ntw>
</config>

```

```

        </vpn-target>
    </vpn-targets>
</address-family>
</vpn-instance-profile>
</vpn-instance-profiles>
<vpn-nodes>
    <vpn-node>
        <vpn-node-id>PIOSXR-0</vpn-node-id>
        <!-- <description>L3VPN-Acme_Inc-78000</description> -->
        <!-- <ne-id>63.130.123.143</ne-id> -->
        <local-as>4445</local-as>
        <active-vpn-instance-profiles>
            <vpn-instance-profile>
                <profile-id>any-to-any-profile</profile-id>
            </vpn-instance-profile>
        </active-vpn-instance-profiles>
        <vpn-network-accesses>
            <vpn-network-access>
                <id>Acme_Inc:3C0780005</id>
                <interface-id>GigabitEthernet0/0/0/0</interface-id>
                <connection>
                    <encapsulation>
                        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:untagged-int</type>
                    </encapsulation>
                </connection>
                <ip-connection>
                    <ipv4>
                        <local-address>10.0.0.10</local-address>
                        <prefix-length>30</prefix-length>
                    </ipv4>
                </ip-connection>
                <routing-protocols>
                    <routing-protocol>
                        <id>L3VPN-BGP-Acme_Inc</id>
                        <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:bgp-routing</type>
                    <bgp>
                        <peer-as>65001</peer-as>
                    </bgp>
                </routing-protocol>
            </routing-protocols>
        </vpn-network-access>
    </vpn-network-accesses>
</vpn-node>
</vpn-nodes>
</vpn-instance-profiles>
</address-family>
</vpn-targets>
</vpn-target>
</vpn-targets>
</address-family>
</vpn-instance-profile>
</vpn-instance-profiles>
</vpn-nodes>
</vpn>

```



```

        <address-family>ipv4</address-family>
        <neighbor>10.0.0.9</neighbor>
    </bgp>
</routing-protocol>
</routing-protocols>
<service>
    <qos>
        <qos-profile>
            <qos-profile>
                <profile>GF-VPN-STANDARD-EGRESS</profile>
                <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:wan-to-
site</direction>
            </qos-profile>
            <qos-profile>
                <profile>GF-VPN-STANDARD-INGRESS</profile>
                <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:site-to-
wan</direction>
            </qos-profile>
        </qos-profile>
    </qos>
</service>
</vpn-network-access>
</vpn-network-accesses>
</vpn-node>
<vpn-node>
    <vpn-node-id>PIOSXR-1</vpn-node-id>
    <local-as>4445</local-as>
    <active-vpn-instance-profiles>
        <vpn-instance-profile>
            <profile-id>any-to-any-profile</profile-id>
        </vpn-instance-profile>
    </active-vpn-instance-profiles>
    <vpn-network-accesses>
        <vpn-network-access>
            <id>Acme_Inc:3C0780005</id>
            <interface-id>GigabitEthernet0/0/0/0</interface-id>
            <connection>

```

```

        <encapsulation>
            <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:untagged-int</type>
        </encapsulation>
    </connection>
    <ip-connection>
        <ipv4>
            <local-address>10.0.0.2</local-address>
            <prefix-length>30</prefix-length>
        </ipv4>
    </ip-connection>
    <routing-protocols>
        <routing-protocol>
            <id>L3VPN-BGP-Acme_Inc</id>
            <type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-
vpn-common">vpn-common:bgp-routing</type>
            <bgp>
                <peer-as>65001</peer-as>
                <address-family>ipv4</address-family>
                <neighbor>10.0.0.1</neighbor>
            </bgp>
        </routing-protocol>
    </routing-protocols>
    <service>
        <qos>
            <qos-profile>
                <qos-profile>
                    <profile>GF-VPN-STANDARD-EGRESS</profile>
                    <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:wan-to-
site</direction>
                </qos-profile>
            </qos-profile>
            <qos-profile>
                <profile>GF-VPN-STANDARD-INGRESS</profile>
                <direction xmlns:vpn-
common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:site-to-
wan</direction>
            </qos-profile>
        </qos>
    </service>

```

```

        </service>
    </vpn-network-access>
</vpn-network-accesses>
</vpn-node>
</vpn-nodes>
    <hco-controller xmlns="http://cisco.com/ns/nso/fp/cisco-hco-nm">cnc-
0</hco-controller>
    </vpn-service>
</vpn-services>
</l3vpn-ntw>
</config>

```

2. Load merge the payload and commit the transaction to create the service. The following is an example.

```

admin@ncs% load merge L3VPN03-multi-instance-InterAS-A-2330.xml
[ok]
[edit]
admin@ncs% commit
Commit complete.
[ok]

```

3. Display the service plan to view and verify the status of the service as shown in the following example.

```

admin@ncs% run show l3vpn-ntw vpn-services vpn-service-plan L3VPN03-multi-
instance-InterAS-A-2330 plan | tab

```

TYPE	NAME	BACK TRACK	GOAL	STATE	STATUS	WHEN	ref	POST ACTION STATUS	MESSAGE
self	self	false	-	init	reached	2023-11-18T01:24:48	-	-	-
				cisco-hco-l3vpn-nano:config-apply	reached	2023-11-18T01:24:48	-	-	-
				cisco-hco-l3vpn-nano:provisioned	not-reached	-	-	-	WAITING FOR CNC
NOTIFICATION				ready	not-reached	-	-	-	-

The plan is in the **not-reached** state when it is awaiting CNC notifications from the CNC device. The provisioned state indicates if the plan received the CNC notifications. The plan updates to **reached** state upon receiving the CNC notifications. A successful plan in the **ready-reached** state indicates the l3vpn-service definition is pushed onto the CNC controller.

Working with Action Commands

Use the action commands to perform specific tasks on services, such as to purge and clean up a service.

Purge Action

When you try to perform a regular service deletion, it may fail for several factors, such as the device being unreachable. This leaves the internal CDB with stale entries of the service and the plan may display failures. When the regular service deletion fails to delete the service elements, use the **Purge** action to clean up all the data created by the service.

Note: You must use the **Purge** action only if the regular service deletion fails and no other recovery mechanisms are available.

Use the following command to purge a service.

```
admin@ncs% request <service> vpn-services vpn-service-plan <service_name> purge
```

Example

```
admin@ncs% request l3vpn-ntw vpn-services vpn-service-plan L3VPN03-single-
instance-2310 purge
#####
#                               Warning                               #
#####
You are about to forcefully purge a vpn-service-plan.
Are you sure you want to proceed? [no,yes] yes
result true
info L3VPN03-single-instance-2310 plan-status operational data successfully
purged.
[ok]
```

Managing CHCO FP

Use the information in this section to understand the log directories for CHCO FP and how to delete the L2VPN/L3VPN services.

Maintaining Logs

The log data contains information about the operation and the usage patterns of the services.

The NED logs are available in the **/var/log/ncs** folder.

Deleting Services

Use the following command to delete L2VPN/L3VPN services. The service plan data is automatically removed if the service deletion is successful.

```
admin@ncs% delete <service> vpn-services vpn-service <l2vpn-service-name>
```

Example

```
admin@ncs% delete l2vpn-ntw vpn-services vpn-service PW01-tldp-interAS-A-55000-  
1130  
[ok]  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok]  
[edit]  
admin@ncs% delete l3vpn-ntw vpn-services vpn-service L3VPN04-multi-instance-  
2421  
[ok]  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok]
```

Appendix A - Yang Models

This topic contains the yang models for the CHCO FP services.

hco-nm.yang

```
module cisco-hco-nm {  
  
    namespace "http://cisco.com/ns/nso/fp/cisco-hco-nm";  
    prefix cisco-hco-nm;  
    yang-version 1.1;  
  
    import tailf-ncs {  
        prefix ncs;  
    }  
  
    import tailf-common {  
        prefix tailf;  
    }  
    import ietf-l2vpn-ntw {  
        prefix l2vpn-ntw;  
    }  
    import ietf-l3vpn-ntw {  
        prefix l3nm;  
    }  
    import ietf-yang-types {  
        prefix yang;  
    }  
  
    import ietf-inet-types {  
        prefix inet;  
    }  
  
    import tailf-kicker {  
        prefix kicker;  
    }  
  
    revision 2023-09-29 {  
        description  
            "Add support for NSP Notifications (Kafka)";  
    }  
  
    revision 2023-03-27 {  
        description  
            "Initial revision.";  
    }  
  
    revision 2023-10-23 {
```

```

    description
      "HCO 8.0.0 release.
      Added 'local-user' leaf in container 'hco-nm-settings'.";
  }

  typedef kafka-authentication-method {
    type enumeration {
      enum no-auth {
        tailf:info "Kafka Listener is not using authentication,
usually used in no production environment";
        description
          "Kafka Listener is not using authentication, usually
used in no production environment";
      }
      enum SSL {
        tailf:info "Kafka Listener is using certificate to
authenticate and encrypt the communication";
        description
          "Kafka Listener is using certificate to authenticate
and encrypt the communication";
      }
      enum SASL_SSL {
        tailf:info "Kafka Listener is using a combination of
SSL with User and password for authentication";
        description
          "Kafka Listener is using a combination of SSL with
User and password for authentication";
      }
      enum SASL_PLAINTEXT {
        tailf:info "Kafka Listener is simple User and password
for authentication";
        description
          "Kafka Listener is simple User and password for
authentication";
      }
    }
  }

  list service-monitor-data {
    tailf:info "l3nm/l2nm service monitor data";
    description "l3nm/l2nm service monitor data";
    config false;
    tailf:cdb-oper {
      tailf:persistent true;
    }
    key "uuid";
    leaf uuid {
      type string;
    }
    leaf service-kp {

```

```

        type string;
    }
    leaf action-type {
        type string;
    }
    leaf timestamp {
        type yang:date-and-time;
    }
}
container vpn-actions {
    tailf:action remove-from-service-monitor {
        tailf:actionpoint cisco-hco-nm-remove-service-monitor-
actionpoint;
        input {
            uses ncs:post-action-input-params;
        }
    }
    action cnc-notif-handler {
        //action to handle CNC notification via notification-kicker
        // tailf:hidden full;
        description
            "Internal action used by cnc notif kicker for
            'service-state-changes' stream to update hco service
plan.";
        tailf:info "Internal action invoked by cnc-hco-kicker to
update cnc plan";
        tailf:actionpoint cnc-notif-handler;
        input {
            uses kicker:action-input-params;
        }
    }
}

grouping vpn-notifications {
    // kafka notifications logger
    container vpn-notifications {
        list change-log-list {
            key "offset-id";
            leaf offset-id {
                tailf:info "log id of parsed notifications from
kafka";
                description
                    "log id of parsed notifications from kafka";
                type string;
            }
            leaf instance-id {

```



```

notification";
    tailf:info "service id of parsed kafka
description
"service id of parsed kafka notification";
type string;
}
leaf time-stamp {
notification";
    tailf:info "event-time of parsed kafka
description
"event-time of parsed kafka notification";
type yang:date-and-time;
}
leaf notif-content {
from kafka notification";
    tailf:info "the complete JSON notification string
description
"the complete JSON notification string from kafka
notification";
type string;
}
leaf nsp-device-name {
notification from";
    tailf:info "NSP device name which we received kafka
description
"NSP device name which we received kafka
notification from";
type string;
}
}
}
}

grouping action-output {
    leaf result {
        type boolean;
        mandatory true;
    }
    leaf info {
        type string;
    }
}

grouping purge-action {
    action purge {
        description
        "Action to purge vpn-service-plan operational data.";
        tailf:info "Action to purge vpn-service-plan operational
data.";
        tailf:actionpoint hco-vpn-plan-purge;
    }
}

```

```

        output {
            uses action-output;
        }
        tailf:confirm-text
        "#####\n"
+ "#                               Warning                               #\n"
+ "#####\n"
+ "You are about to forcefully purge a vpn-service-plan.\n"
+ "Are you sure you want to proceed?" {
            tailf:confirm-default false;
        }
    }
}

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-ntw:vpn-
service" {
    // TODO: to be updated later
    leaf hco-controller {
        tailf:info "HCO controller name";
        description
            "Nokia NSP controller name";
        type leafref {
            path "/ncs:devices/ncs:device/ncs:name";
        }
        // tailf:default-ref "../../../../../nsp-nm/default-nsp-
controller";
    }
}

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services" {
    list vpn-service-plan {
        config false;
        tailf:cdb-oper {
            tailf:persistent true;
        }
        key "vpn-id";
        leaf vpn-id {
            type string;
        }
        leaf delete-notif-received {
            config false;
            tailf:cdb-oper {
                tailf:persistent true;
            }
            type boolean;
            default false;
        }
        uses cisco-hco-nm:vpn-notifications;
    }
}

```

```

        uses ncs:nano-plan-data;
        uses ncs:nano-plan-history;
        uses purge-action;
    }
}

augment "/l3nm:l3vpn-ntw/l3nm:vpn-services/l3nm:vpn-service" {
    // TODO: to be updated later
    leaf hco-controller {
        tailf:info "Nokia NSP controller name";
        description
            "Nokia NSP controller name";
        type leafref {
            path "/ncs:devices/ncs:device/ncs:name";
        }
        // tailf:default-ref "../..../..../nsp-nm/default-nsp-
controller";
    }
}

augment "/l3nm:l3vpn-ntw/l3nm:vpn-services" {
    list vpn-service-plan {
        config false;
        tailf:cdb-oper {
            tailf:persistent true;
        }
        key "vpn-id";
        leaf vpn-id {
            type string;
        }
        leaf delete-notif-received {
            config false;
            tailf:cdb-oper {
                tailf:persistent true;
            }
            type boolean;
            default false;
        }
        uses cisco-hco-nm:vpn-notifications;
        uses ncs:nano-plan-data;
        uses ncs:nano-plan-history;
        uses purge-action;
    }
}

augment "/ncs:devices/ncs:device" {
    when "starts-with(ncs:device-type/ncs:generic/ncs:ned-id,
'nokia-nsp')";
}

```

```

container ns-kafka-settings {
    tailf:info "Nokia NSP Notification System";
    description
        "Nokia NSP Notification System";

    leaf active {
        tailf:info "Enable message filter for NSP
notification";
        description
            "Kafka deployment feedback Enable/Disable";
        type boolean;
        default false;
    }

    leaf listeners {
        when "../active = 'true'";
        type string;
    }

    leaf topic-name {
        when "../active = 'true'";
        type string;
        default "nsp-yang-model.change-notif";
    }

    container authentication {
        when "../active = 'true'";
        leaf authentication-type {
            type kafka-authentication-method;
            // mandatory true;
        }

        leaf ssl-key-location {
            when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
            type string;
        }

        leaf ssl-key-password {
            when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
            tailf:suppress-echo "true";
            type tailf:aes-256-cfb-128-encrypted-string;
        }

        leaf ssl-ca-location {
            when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
            type string;
        }
    }
}

```

```

    }

    leaf ssl-certificate-location {
        when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
        type string;
    }

    leaf ssl.endpoint.identification.algorithm {
        when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
        type enumeration {
            enum none;
            enum HTTPS;
        }
        default "none";
    }

    leaf username {
        when "../authentication-type = 'SASL_SSL'";
        type string;
    }

    leaf password {
        when "../authentication-type = 'SASL_SSL'";
        tailf:suppress-echo "true";
        type tailf:aes-256-cfb-128-encrypted-string;
    }
}
}
}

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/vpn-service-
plan/plan/component/state" {
    leaf message {
        tailf:info "additional state leaf to display messages.";
        description
            "additional state leaf to display messages.";
        tailf:display-column-name "MESSAGE";
        type string;
    }
}

augment "/l3nm:l3vpn-ntw/l3nm:vpn-services/vpn-service-
plan/plan/component/state" {
    leaf message {
        tailf:info "additional state leaf to display messages.";
        description
            "additional state leaf to display messages.";
    }
}

```

```

        tailf:display-column-name "MESSAGE";
        type string;
    }
}
container hco-nm-settings {
    tailf:info "cisco hco-nm function pkg settings";
    description
        "cisco hco-nm function pkg settings";
    leaf controller-type {
        type enumeration {
            enum CNC;
            enum NSP;
        }
        default "NSP";
    }
    // TODO: to be updated later
    leaf default-hco-controller {
        tailf:info "Default HCO(NSP/CNC) controller device based on
controller-type";
        description
            "Default HCO(NSP/CNC) controller device";
        type leafref {
            path "/ncs:devices/ncs:device/ncs:name";
        }
        must "(
            (../controller-type = 'CNC')
            and

            (contains(/ncs:devices/ncs:device[ncs:name=current()]/ncs:device-
type/ncs:netconf/ncs:ned-id , 'cisco-nso-nc')
                or

            contains(/ncs:devices/ncs:device[ncs:name=current()]/ncs:device-
type/ncs:generic/ncs:ned-id , 'cisco-cnc_rc-gen'))
                or
                ((../controller-type = 'NSP')
                    and

            (contains(/ncs:devices/ncs:device[ncs:name=current()]/ncs:device-
type/ncs:generic/ncs:ned-id , 'nokia-nsp_'))));";
    }
    leaf kafka-notif-timeout {
        tailf:info "Kafka notification timeout in seconds";
        type uint16 {
            range "60..65535";
        }
        default "180";
    }
}
leaf nsp-notif-processing {

```

```

        tailf:info "process nsp Kafka notification after this
seconds delay.";
        type uint16 {
            range "10..65535";
        }
        default 10;
    }
    leaf local-user {
        description
            "Map NCS user with sufficient permissions to read and to
write into the cdb.";
        tailf:info
            "Map NCS user with sufficient permissions to read and to
write into the cdb.";
        type string;
        default "admin";
    }
    action sleep-change {
        tailf:hidden full;
        description
            "Internal action used by sleep_setting_kicker to update
sleep dict object in python.";
        tailf:info "Internal action used by sleep_setting_kicker to
update sleep dict object in python.";
        tailf:actionpoint hco-nm-sleep-change;
        input {
            uses kicker:action-input-params;
        }
    }
}

// TODO: name change TBD
container hco-nm-device-mapping {
    tailf:info "Cisco nsp-nm service to device mapping";
    description
        "Cisco nsp-nm service to device mapping";
    tailf:hidden debug;
    list l2vpn-ntw {
        key "vpn-id";
        leaf vpn-id {
            tailf:info
                "A VPN identifier that uniquely identifies a VPN.";
            description
                "A VPN identifier that uniquely identifies a VPN.";
            type string;
        }
        leaf hco-controller {
            tailf:info "HCO controller name";
            description

```

```

        "HCO controller name";
        type leafref {
            path "/ncs:devices/ncs:device/ncs:name";
        }
    }
}
list l3vpn-ntw {
    key "vpn-id";
    leaf vpn-id {
        tailf:info
"A VPN identifier that uniquely identifies a VPN.";
        description
            "A VPN identifier that uniquely identifies a VPN.";
        type string;
    }
    leaf hco-controller {
        tailf:info "HCO controller name";
        description
            "HCO controller name";
        type leafref {
            path "/ncs:devices/ncs:device/ncs:name";
        }
    }
}
}
}
}

```

ietf-l2vpn-ntw.yang

```

module: ietf-l2vpn-ntw
+--rw l2vpn-ntw
  +--rw vpn-profiles
  | +--rw valid-provider-identifiers
  |   +--rw qos-profile-identifier* [id]
  |   +--rw routing-profile-identifier* [id]
  +--rw vpn-services
    +--rw vpn-service* [vpn-id]
      +--rw vpn-id                               vpn-common:vpn-id
      +--rw vpn-name?                            string
      +--rw vpn-description?                     string

```



```

+---rw customer-name?                string
+---rw parent-service-id?            vpn-common:vpn-id
+---rw vpn-type?                     identityref
+---rw vpn-service-topology?         identityref
+---rw bgp-ad-enabled?               boolean
+---rw signaling-type?               identityref
+---rw global-parameters-profiles
|  +---rw global-parameters-profile* [profile-id]
|    +---rw profile-id                string
|    +---rw (rd-choice)?
|      | +---:(directly-assigned)
|      | | +---rw rd?                 rt-types:route-
distinguisher
|      | | +---:(directly-assigned-suffix)
|      | | | +---rw rd-suffix?       uint16
|      | | +---:(auto-assigned)
|      | | | +---rw rd-auto
|      | | |   +---rw (auto-mode)?
|      | | |   | +---:(from-pool)
|      | | |   | | +---rw rd-pool-name? string
|      | | |   | +---:(full-auto)
|      | | |   |   +---rw auto?      empty
|      | | |   +---ro auto-assigned-rd? rt-types:route-
distinguisher
|      | | +---:(auto-assigned-suffix)
|      | | | +---rw rd-auto-suffix
|      | | |   +---rw (auto-mode)?
|      | | |   | +---:(from-pool)
|      | | |   | | +---rw rd-pool-name? string
|      | | |   | +---:(full-auto)
|      | | |   |   +---rw auto?      empty
|      | | |   +---ro auto-assigned-rd-suffix? uint16
|      | +---:(no-rd)

```

```

|   |   +---rw no-rd?           empty
|   +---rw vpn-target* [id]
|   |   +---rw id               uint8
|   |   +---rw route-targets* [route-target]
|   |   |   +---rw route-target  rt-types:route-target
|   |   +---rw route-target-type  rt-types:route-target-
type
|   +---rw vpn-policies
|   |   +---rw import-policy?   string
|   |   +---rw export-policy?   string
|   +---rw local-autonomous-system?  inet:as-number
|   +---rw svc-mtu?             uint32
|   +---rw ce-vlan-preservation?    boolean
|   +---rw ce-vlan-cos-preservation? boolean
|   +---rw control-word-negotiation? boolean
|   +---rw mac-policies
|   |   +---rw mac-addr-limit
|   |   |   +---rw limit-number?   uint16
|   |   |   +---rw time-interval?  uint32
|   |   |   +---rw action?         identityref
|   |   +---rw mac-loop-prevention
|   |   |   +---rw window?         uint32
|   |   |   +---rw frequency?      uint32
|   |   |   +---rw retry-timer?    uint32
|   |   |   +---rw protection-type? identityref
|   +---rw multicast {vpn-common:multicast}?
|   |   +---rw enabled?           boolean
|   |   +---rw customer-tree-flavors
|   |   |   +---rw tree-flavor*    identityref
+---rw underlay-transport
|   +---rw (type)?
|   +---:(abstract)
|   |   +---rw transport-instance-id?  string

```

```

|   | +--rw instance-type?          identityref
|   +--:(protocol)
|       +--rw protocol*             identityref
+--rw status
|   +--rw admin-status
|   |   +--rw status?               identityref
|   |   +--rw last-change?          yang:date-and-time
|   +--ro oper-status
|       +--ro status?               identityref
|       +--ro last-change?          yang:date-and-time
+--rw vpn-nodes
|   +--rw vpn-node* [vpn-node-id]
|       +--rw vpn-node-id           vpn-
common:vpn-id
|       +--rw description?           string
|       +--rw ne-id?                 string
|       +--rw role?                  identityref
|       +--rw router-id?             rt-
types:router-id
|       +--rw active-global-parameters-profiles
|           +--rw global-parameters-profile* [profile-id]
|               +--rw profile-id     ->
../../../../../../../../global-parameters-profiles/global-parameters-
profile/profile-id
|           +--rw local-autonomous-system?  inet:as-number
|           +--rw svc-mtu?                 uint32
|           +--rw ce-vlan-preservation?     boolean
|           +--rw ce-vlan-cos-preservation? boolean
|           +--rw control-word-negotiation? boolean
|           +--rw mac-policies
|               +--rw mac-addr-limit
|                   +--rw limit-number?     uint16
|                   +--rw time-interval?    uint32
|                   +--rw action?           identityref

```

```

| | +--rw mac-loop-prevention
| |   +--rw window?          uint32
| |   +--rw frequency?       uint32
| |   +--rw retry-timer?     uint32
| |   +--rw protection-type? identityref
| +--rw multicast {vpn-common:multicast}?
|   +--rw enabled?           boolean
|   +--rw customer-tree-flavors
|     +--rw tree-flavor*    identityref
+--rw status
| +--rw admin-status
| | +--rw status?          identityref
| | +--rw last-change?    yang:date-and-time
| +--ro oper-status
|   +--ro status?         identityref
|   +--ro last-change?    yang:date-and-time
+--rw bgp-auto-discovery
| +--rw (bgp-type)?
| | +--:(l2vpn-bgp)
| | | +--rw vpn-id?       vpn-common:vpn-id
| | +--:(evpn-bgp)
| |   +--rw evpn-type?    ->
../../../../../../../../vpn-type
| |   +--rw auto-rt-enable? boolean
| |   +--ro auto-route-target? rt-types:route-
target
| +--rw (rd-choice)?
| | +--:(directly-assigned)
| | | +--rw rd?           rt-types:route-
distinguisher
| | | +--:(directly-assigned-suffix)
| | | | +--rw rd-suffix?   uint16
| | | +--:(auto-assigned)
| | | | +--rw rd-auto

```

```

    | | | +--rw (auto-mode)?
    | | | | +--:(from-pool)
    | | | | | +--rw rd-pool-name? string
    | | | | +--:(full-auto)
    | | | | +--rw auto? empty
    | | | +--ro auto-assigned-rd? rt-
types:route-distinguisher
    | | +--:(auto-assigned-suffix)
    | | | +--rw rd-auto-suffix
    | | | +--rw (auto-mode)?
    | | | | +--:(from-pool)
    | | | | | +--rw rd-pool-name? string
    | | | | +--:(full-auto)
    | | | | +--rw auto? empty
    | | | +--ro auto-assigned-rd-suffix? uint16
    | | +--:(no-rd)
    | | +--rw no-rd? empty
    | +--rw vpn-target* [id]
    | | +--rw id uint8
    | | +--rw route-targets* [route-target]
    | | | +--rw route-target rt-types:route-target
    | | +--rw route-target-type rt-types:route-
target-type
    | +--rw vpn-policies
    | +--rw import-policy? string
    | +--rw export-policy? string
+--rw signaling-option
    | +--rw advertise-mtu? boolean
    | +--rw mtu-allow-mismatch? boolean
    | +--rw signaling-type?
identityref
    | +--rw (signaling-option)?
    | +--:(bgp)
    | | +--rw (bgp-type)?

```

```

| | +---:(l2vpn-bgp)
| | | +---rw ce-range? uint16
identityref | | | +---rw pw-encapsulation-type?
| | | +---rw vpls-instance
| | | +---rw vpls-edge-id? uint16
| | | +---rw vpls-edge-id-range? uint16
| | +---:(evpn-bgp)
| | +---rw evpn-type? ->
../../../../bgp-auto-discovery/evpn-type
identityref | | +---rw service-interface-type?
| | +---rw evpn-policies
identityref | | +---rw mac-learning-mode?
boolean | | +---rw ingress-replication?
boolean | | +---rw p2mp-replication?
| | +---rw arp-proxy {vpn-common:ipv4}?
boolean | | | +---rw enable?
boolean | | | +---rw arp-suppression?
uint16 | | | +---rw ip-mobility-threshold?
interval? | | | +---rw duplicate-ip-detection-
uint16 | | +---rw nd-proxy {vpn-common:ipv6}?
boolean | | | +---rw enable?
boolean | | | +---rw nd-suppression?
uint16 | | | +---rw ip-mobility-threshold?
interval? | | | +---rw duplicate-ip-detection-
uint16

```

```

boolean          |      |          +---rw underlay-multicast?

suppression?    |      |          +---rw flood-unknown-unicast-
boolean         |      |

boolean         |      |          +---rw vpws-vlan-aware?

boolean         |      |          +---rw bum-management

boolean         |      |          |  +---rw discard-broadcast?

boolean         |      |          |  +---rw discard-unknown-multicast?

boolean         |      |          |  +---rw discard-unknown-unicast?

boolean         |      |          +---rw pbb

yang:mac-address |      |          +---rw backbone-src-mac?

                |      |          +---:(ldp-or-l2tp)
                |      |          +---rw ldp-or-l2tp

types:route-distinguisher |      |          +---rw agi?          rt-

                |      |          +---rw sai?          uint32

                |      |          +---rw remote-targets* [taii]

                |      |          |  +---rw taii          uint32

                |      |          |  +---rw peer-addr?  inet:ip-address

                |      |          +---rw (ldp-or-l2tp)?

                |      |          +---:(ldp)

identityref     |      |          |  +---rw t-ldp-pw-type?

identityref     |      |          |  +---rw pw-type?

                |      |          |  +---rw pw-description?  string

                |      |          |  +---rw mac-addr-withdraw?  boolean

id]            |      |          |  +---rw pw-peer-list* [peer-addr vc-

address        |      |          |  |  +---rw peer-addr  inet:ip-

                |      |          |  |  +---rw vc-id    string

```

```

| | +--rw pw-priority? uint32
| | +--rw qinq
| | +--rw s-tag dot1q-types:vlanid
| | +--rw c-tag dot1q-types:vlanid
| +--:(l2tp)
| +--rw router-id? rt-
types:router-id
| +--rw pseudowire-type?
identityref
+--rw vpn-network-accesses
+--rw vpn-network-access* [id]
+--rw id vpn-
common:vpn-id
+--rw description?
string
+--rw interface-id?
string
+--rw active-vpn-node-profile? ->
../../../../active-global-parameters-profiles/global-parameters-
profile/profile-id
+--rw status
| +--rw admin-status
| | +--rw status? identityref
| | +--rw last-change? yang:date-and-time
| +--ro oper-status
| +--ro status? identityref
| +--ro last-change? yang:date-and-time
+--rw connection
| +--rw l2-termination-point? string
| +--rw local-bridge-reference? string
| +--rw bearer-reference? string {vpn-
common:bearer-reference}?
| +--rw encapsulation
| | +--rw encap-type? identityref
| | +--rw dot1q

```



```

| | | +--rw tag-type?          identityref
| | | +--rw cvlan-id?         dot1q-
types:vlanid
| | | +--rw tag-operations
| | |   +--rw (op-choice)?
| | |     | +--:(pop)
| | |     | | +--rw pop?      empty
| | |     | +--:(push)
| | |     | | +--rw push?    empty
| | |     | +--:(translate)
| | |     |   +--rw translate? empty
| | |   +--rw tag-1?         dot1q-
types:vlanid
| | |   +--rw tag-1-type?    dot1q-
types:dot1q-tag-type
| | |   +--rw tag-2?         dot1q-
types:vlanid
| | |   +--rw tag-2-type?    dot1q-
types:dot1q-tag-type
| | +--rw priority-tagged
| | | +--rw tag-type?      identityref
| | +--rw qinq
| |   +--rw tag-type?      identityref
| |   +--rw svlan-id       dot1q-
types:vlanid
| |   +--rw cvlan-id       dot1q-
types:vlanid
| |   +--rw tag-operations
| |     +--rw (op-choice)?
| |       | +--:(pop)
| |       | | +--rw pop?    uint8
| |       | +--:(push)
| |       | | +--rw push?   empty
| |       | +--:(translate)
| |       |   +--rw translate? uint8

```

```

types:vlanid          | |          +---rw tag-1?          dot1q-
types:dot1q-tag-type  | |          +---rw tag-1-type?       dot1q-
types:vlanid          | |          +---rw tag-2?          dot1q-
types:dot1q-tag-type  | |          +---rw tag-2-type?       dot1q-
interface)?          | +---rw lag-interface {vpn-common:lag-
                    | +---rw lag-interface-id?  string
                    | +---rw lacp
                    | | +---rw lacp-state?      boolean
                    | | +---rw mode?            identityref
                    | | +---rw speed?          uint32
                    | | +---rw mini-link-num?   uint32
                    | | +---rw system-id?       yang:mac-
address              | | +---rw admin-key?      uint16
                    | | +---rw system-priority? uint16
                    | | +---rw member-link-list
                    | | | +---rw member-link* [name]
                    | | |   +---rw name          string
                    | | |   +---rw speed?       uint32
                    | | |   +---rw mode?
identityref          | | | +---rw link-mtu?      uint32
                    | | | +---rw oam-802.3ah-link {oam-3ah}?
                    | | |   +---rw enable?      boolean
                    | | +---rw flow-control?    boolean
                    | | +---rw lldp?           boolean
                    | +---rw split-horizon
                    |   +---rw group-name?     string
+---rw (signaling-option)?
| +---: (bgp)

```

```

|      +---rw (bgp-type)?
|      |      +---:(l2vpn-bgp)
|      |      |      +---rw ce-id?
uint16
|      |      |      +---rw remote-ce-id?
uint16
|      |      |      +---rw vpls-instance
|      |      |      |      +---rw vpls-edge-id?      uint16
|      |      +---:(evpn-bgp)
|      |      +---rw df-preference?
uint16
|      |      +---rw vpws-service-instance
|      |      |      +---rw (local-vsi-choice)?
|      |      |      |      +---:(directly-assigned)
|      |      |      |      |      +---rw local-vpws-service-
instance?      uint32
|      |      |      |      |      +---:(auto-assigned)
|      |      |      |      |      |      +---rw local-vsi-auto
|      |      |      |      |      |      |      +---rw (auto-mode)?
|      |      |      |      |      |      |      |      +---:(from-pool)
|      |      |      |      |      |      |      |      |      +---rw vsi-pool-
name?      string
|      |      |      |      |      |      |      |      |      +---:(full-auto)
|      |      |      |      |      |      |      |      |      +---rw auto?
empty
|      |      |      |      |      |      |      |      |      +---ro auto-local-vsi?
uint32
|      |      |      |      |      |      |      |      |      +---rw (remote-vsi-choice)?
|      |      |      |      |      |      |      |      |      +---:(directly-assigned)
|      |      |      |      |      |      |      |      |      |      +---rw remote-vpws-service-
instance?      uint32
|      |      |      |      |      |      |      |      |      +---:(auto-assigned)
|      |      |      |      |      |      |      |      |      |      +---rw remote-vsi-auto
|      |      |      |      |      |      |      |      |      |      +---rw (auto-mode)?
|      |      |      |      |      |      |      |      |      |      |      +---:(from-pool)

```

```

name?  string      | | +---rw vsi-pool-
| | | | |
| | | | | +---:(full-auto)
| | | | | +---rw auto?
empty
| | | | | +---ro auto-remote-vsi?
uint32
+---rw group* [group-id]
| +---rw group-id          string
| +---rw precedence?
identityref
| +---rw ethernet-segment-identifier?  string
+---rw ethernet-service-oam
| +---rw md-name?          string
| +---rw md-level?        uint8
| +---rw cfm-802.1-ag
| | +---rw n2-uni-c* [maid]
| | | +---rw maid          string
| | | +---rw mep-id?       uint32
| | | +---rw mep-level?    uint32
| | | +---rw mep-up-down?  enumeration
| | | +---rw remote-mep-id? uint32
| | | +---rw cos-for-cfm-pdus? uint32
| | | +---rw ccm-interval?  uint32
| | | +---rw ccm-holdtime?  uint32
| | | +---rw ccm-p-bits-pri? ccm-priority-
type
| | +---rw n2-uni-n* [maid]
| |   +---rw maid          string
| |   +---rw mep-id?       uint32
| |   +---rw mep-level?    uint32
| |   +---rw mep-up-down?  enumeration
| |   +---rw remote-mep-id? uint32
| |   +---rw cos-for-cfm-pdus? uint32

```

```

| | +--rw ccm-interval?      uint32
| | +--rw ccm-holdtime?     uint32
| | +--rw ccm-p-bits-pri?   ccm-priority-
type
| +--rw y-1731* [maid]
|   +--rw maid              string
|   +--rw mep-id?          uint32
identityref
|   +--rw pm-type?
|
|   +--rw remote-mep-id?   uint32
|   +--rw message-period?  uint32
|   +--rw measurement-interval?  uint32
|   +--rw cos?             uint32
boolean
|   +--rw loss-measurement?
boolean
|   +--rw synthetic-loss-measurement?
|
|   +--rw delay-measurement
|   | +--rw enable-dm?     boolean
|   | +--rw two-way?       boolean
|   +--rw frame-size?     uint32
enumeration
|   +--rw session-type?
+--rw service
  +--rw mtu?
uint32
+--rw svc-pe-to-ce-bandwidth {vpn-
common:inbound-bw}?
| +--rw pe-to-ce-bandwidth* [bw-type]
|   +--rw bw-type          identityref
|   +--rw (type)?
|     +--:(per-cos)
|       | +--rw cos* [cos-id]
|       |   +--rw cos-id   uint8
|       |   +--rw cir?     uint64

```

```

|         |         +---rw cbs?      uint64
|         |         +---rw eir?      uint64
|         |         +---rw ebs?      uint64
|         |         +---rw pir?      uint64
|         |         +---rw pbs?      uint64
|         +---:(other)
|             +---rw cir?    uint64
|             +---rw cbs?    uint64
|             +---rw eir?    uint64
|             +---rw ebs?    uint64
|             +---rw pir?    uint64
|             +---rw pbs?    uint64
+---rw svc-ce-to-pe-bandwidth {vpn-
common:outbound-bw}?
| +---rw ce-to-pe-bandwidth* [bw-type]
|   +---rw bw-type          identityref
|   +---rw (type)?
|     +---:(per-cos)
|       | +---rw cos* [cos-id]
|       |   +---rw cos-id    uint8
|       |   +---rw cir?      uint64
|       |   +---rw cbs?      uint64
|       |   +---rw eir?      uint64
|       |   +---rw ebs?      uint64
|       |   +---rw pir?      uint64
|       |   +---rw pbs?      uint64
|       +---:(other)
|         +---rw cir?    uint64
|         +---rw cbs?    uint64
|         +---rw eir?    uint64
|         +---rw ebs?    uint64
|         +---rw pir?    uint64
|         +---rw pbs?    uint64

```

```

+---rw qos {vpn-common:qos}?
| +---rw qos-classification-policy
| | +---rw rule* [id]
| |   +---rw id
string
| |   +---rw (match-type)?
| |   | +---:(match-flow)
| |   | | +---rw match-flow
| |   | |   +---rw dscp?
inet:dscp
| |   | |   +---rw dot1q?
uint16
| |   | |   +---rw pcpc?
uint8
| |   | |   +---rw src-mac-address?
yang:mac-address
| |   | |   +---rw dst-mac-address?
yang:mac-address
| |   | |   +---rw color-type?
identityref
| |   | |   +---rw any?
empty
| |   | | +---:(match-application)
| |   | |   +---rw match-application?
identityref
| |   +---rw target-class-id?
string
| +---rw qos-profile
|   +---rw qos-profile* [profile]
|     +---rw profile      string
|     +---rw direction?  identityref
+---rw mac-policies
| +---rw access-control-list* [name]
| | +---rw name                string
| | +---rw src-mac-address*    yang:mac-
address

```

```

address          | | +--rw src-mac-address-mask*  yang:mac-
address          | | +--rw dst-mac-address*       yang:mac-
address          | | +--rw dst-mac-address-mask*  yang:mac-
identityref      | | +--rw action?
                 | | +--rw rate-limit?           decimal64
                 | +--rw mac-loop-prevention
                 | | +--rw window?             uint32
                 | | +--rw frequency?          uint32
                 | | +--rw retry-timer?        uint32
                 | | +--rw protection-type?     identityref
                 | +--rw mac-addr-limit
                 |   +--rw limit-number?       uint16
                 |   +--rw time-interval?      uint32
                 |   +--rw action?             identityref
+--rw broadcast-unknown-unicast-multicast
   +--rw multicast-site-type?
enumeration
   +--rw multicast-gp-address-mapping* [id]
   | +--rw id                               uint16
   | +--rw vlan-id                           uint32
   | +--rw mac-gp-address                    yang:mac-
address
   | +--rw port-lag-number?                  uint32
+--rw bum-overall-rate?
uint64

```

ietf-l3vpn-ntw.yang

```

module: ietf-l3vpn-ntw
+--rw l3vpn-ntw
  +--rw vpn-profiles
  | +--rw valid-provider-identifiers
  |   +--rw qos-profile-identifier* [id]
  |   +--rw routing-profile-identifier* [id]

```



```

+---rw vpn-services
  +---rw vpn-service* [vpn-id]
    +---rw vpn-id                vpn-common:vpn-id
    +---rw vpn-name?             string
    +---rw vpn-description?      string
    +---rw customer-name?       string
    +---rw parent-service-id?   vpn-common:vpn-id
    +---rw vpn-type?            identityref
    +---rw vpn-service-topology? identityref
    +---rw status
      | +---rw admin-status
      | | +---rw status?        identityref
      | | +---rw last-change?   yang:date-and-time
      | +---ro oper-status
      |   +---ro status?        identityref
      |   +---ro last-change?   yang:date-and-time
    +---rw vpn-instance-profiles
      | +---rw vpn-instance-profile* [profile-id]
      |   +---rw profile-id      string
      |   +---rw role?           identityref
      |   +---rw local-as?       inet:as-number {vpn-common:rtg-
bgp}?
      |   +---rw (rd-choice)?
      |     | +---:(directly-assigned)
      |     | | +---rw rd?       rt-types:route-distinguisher
      |     | | +---:(directly-assigned-suffix)
      |     | | | +---rw rd-suffix? uint16
      |     | | +---:(auto-assigned)
      |     | | | +---rw rd-auto
      |     | | |   +---rw (auto-mode)?
      |     | | |     | +---:(from-pool)
      |     | | |     | | +---rw rd-pool-name? string
      |     | | |     | | +---:(full-auto)
      |     | | |     |   +---rw auto?         empty
      |     | | |     +---ro auto-assigned-rd? rt-types:route-
distinguisher
      |     | | +---:(auto-assigned-suffix)
      |     | | | +---rw rd-auto-suffix
      |     | | |   +---rw (auto-mode)?
      |     | | |     | +---:(from-pool)
      |     | | |     | | +---rw rd-pool-name? string
      |     | | |     | | +---:(full-auto)
      |     | | |     |   +---rw auto?         empty
      |     | | |     +---ro auto-assigned-rd-suffix? uint16
      |     | | +---:(no-rd)

```

```

|     |     +---rw no-rd?          empty
| +---rw address-family* [address-family]
|     | +---rw address-family    identityref
|     | +---rw vpn-targets
|     | | +---rw vpn-target* [id]
|     | | | +---rw id              uint8
|     | | | +---rw route-targets* [route-target]
|     | | | | +---rw route-target  rt-types:route-target
|     | | | +---rw route-target-type  rt-types:route-target-type
|     | | +---rw vpn-policies
|     | |   +---rw import-policy?  string
|     | |   +---rw export-policy?  string
|     | +---rw maximum-routes* [protocol]
|     |   +---rw protocol          identityref
|     |   +---rw maximum-routes?  uint32
| +---rw multicast {vpn-common:multicast}?
|     +---rw tree-flavor?  identityref
|     +---rw rp
|     | +---rw rp-group-mappings
|     | | +---rw rp-group-mapping* [id]
|     | | | +---rw id              uint16
|     | | | +---rw provider-managed
|     | | | | +---rw enabled?          boolean
|     | | | | +---rw rp-redundancy?    boolean
|     | | | | +---rw optimal-traffic-delivery?  boolean
|     | | | | +---rw anycast
|     | | | |   +---rw local-address?  inet:ip-address
|     | | | |   +---rw rp-set-address*  inet:ip-address
|     | | | +---rw rp-address          inet:ip-address
|     | | | +---rw groups
|     | | |   +---rw group* [id]
|     | | |   +---rw id              uint16
|     | | |   +---rw (group-format)
|     | | |   +---:(group-prefix)
|     | | |   | +---rw group-address?  inet:ip-prefix
|     | | |   +---:(startend)
|     | | |   +---rw group-start?    inet:ip-address
|     | | |   +---rw group-end?      inet:ip-address
|     | +---rw rp-discovery
|     |   +---rw rp-discovery-type?  identityref
|     |   +---rw bsr-candidates
|     |     +---rw bsr-candidate-address*  inet:ip-address
| +---rw igmp {vpn-common:igmp and vpn-common:ipv4}?
|     | +---rw static-group* [group-addr]

```

```

address      |      | | +--rw group-addr      rt-types:ipv4-multicast-group-
address      |      | | +--rw source-addr?    rt-types:ipv4-multicast-source-
              |      | +--rw max-groups?      uint32
              |      | +--rw max-entries?    uint32
              |      | +--rw version?      identityref
              |      +--rw mld {vpn-common:mld and vpn-common:ipv6}?
              |      | +--rw static-group* [group-addr]
address      |      | | +--rw group-addr      rt-types:ipv6-multicast-group-
address      |      | | +--rw source-addr?    rt-types:ipv6-multicast-source-
              |      | +--rw max-groups?      uint32
              |      | +--rw max-entries?    uint32
              |      | +--rw version?      identityref
              |      +--rw pim {vpn-common:pim}?
              |          +--rw hello-interval?  rt-types:timer-value-seconds16
              |          +--rw dr-priority?    uint32
+--rw underlay-transport
| +--rw (type)?
|   +---:(abstract)
|   | +--rw transport-instance-id?  string
|   | +--rw instance-type?          identityref
|   +---:(protocol)
|     +--rw protocol*                identityref
+--rw external-connectivity {vpn-common:external-connectivity}?
| +--rw (profile)?
|   +---:(profile)
|     +--rw profile-name?  string
+--rw vpn-nodes
  +--rw vpn-node* [vpn-node-id]
    +--rw vpn-node-id          vpn-common:vpn-id
    +--rw description?         string
    +--rw ne-id?               string
    +--rw local-as?            inet:as-number {vpn-
common:rtg-bgp}?
    +--rw router-id?           rt-types:router-id
    +--rw active-vpn-instance-profiles
    | +--rw vpn-instance-profile* [profile-id]
    |   +--rw profile-id        string
    |   +--rw router-id* [address-family]
    |     | +--rw address-family  identityref
    |     | +--rw router-id?      inet:ip-address
    |     +--rw local-as?        inet:as-number {vpn-
common:rtg-bgp}?

```

```

|      +---rw (rd-choice)?
|      | +---:(directly-assigned)
|      | | +---rw rd?                rt-types:route-
distinguisher
|      | +---:(directly-assigned-suffix)
|      | | +---rw rd-suffix?         uint16
|      | +---:(auto-assigned)
|      | | +---rw rd-auto
|      | |   +---rw (auto-mode)?
|      | |   | +---:(from-pool)
|      | |   | | +---rw rd-pool-name?  string
|      | |   | +---:(full-auto)
|      | |   |   +---rw auto?         empty
|      | |   +---ro auto-assigned-rd?  rt-types:route-
distinguisher
|      | +---:(auto-assigned-suffix)
|      | | +---rw rd-auto-suffix
|      | |   +---rw (auto-mode)?
|      | |   | +---:(from-pool)
|      | |   | | +---rw rd-pool-name?  string
|      | |   | +---:(full-auto)
|      | |   |   +---rw auto?         empty
|      | |   +---ro auto-assigned-rd-suffix?  uint16
|      | +---:(no-rd)
|      |   +---rw no-rd?              empty
|      +---rw address-family* [address-family]
|      | +---rw address-family      identityref
|      | +---rw vpn-targets
|      | | +---rw vpn-target* [id]
|      | | | +---rw id                uint8
|      | | | +---rw route-targets* [route-target]
|      | | | | +---rw route-target    rt-types:route-target
|      | | | | +---rw route-target-type  rt-types:route-
target-type
|      | | +---rw vpn-policies
|      | |   +---rw import-policy?    string
|      | |   +---rw export-policy?    string
|      | +---rw maximum-routes* [protocol]
|      |   +---rw protocol            identityref
|      |   +---rw maximum-routes?    uint32
|      +---rw multicast {vpn-common:multicast}?
|      | +---rw tree-flavor?         identityref
|      | +---rw rp
|      | | +---rw rp-group-mappings
|      | | | +---rw rp-group-mapping* [id]

```

```

|         | |         +---rw id                               uint16
|         | |         +---rw provider-managed
|         | |         | +---rw enabled?                       boolean
|         | |         | +---rw rp-redundancy?                 boolean
|         | |         | +---rw optimal-traffic-delivery?     boolean
|         | |         | +---rw anycast
|         | |         | +---rw local-address?                 inet:ip-
address
|         | |         | +---rw rp-set-address*                inet:ip-
address
|         | |         +---rw rp-address                       inet:ip-address
|         | |         +---rw groups
|         | |         +---rw group* [id]
|         | |         +---rw id                               uint16
|         | |         +---rw (group-format)
|         | |         +---:(group-prefix)
|         | |         | +---rw group-address?                 inet:ip-
prefix
|         | |         +---:(startend)
|         | |         +---rw group-start?                     inet:ip-
address
|         | |         +---rw group-end?                       inet:ip-
address
|         | +---rw rp-discovery
|         | +---rw rp-discovery-type? identityref
|         | +---rw bsr-candidates
|         | +---rw bsr-candidate-address* inet:ip-
address
|         +---rw igmp {vpn-common:igmp and vpn-common:ipv4}?
|         | +---rw static-group* [group-addr]
|         | | +---rw group-addr rt-types:ipv4-multicast-
group-address
|         | | +---rw source-addr? rt-types:ipv4-multicast-
source-address
|         | +---rw max-groups? uint32
|         | +---rw max-entries? uint32
|         | +---rw version? identityref
|         +---rw mld {vpn-common:mld and vpn-common:ipv6}?
|         | +---rw static-group* [group-addr]
|         | | +---rw group-addr rt-types:ipv6-multicast-
group-address
|         | | +---rw source-addr? rt-types:ipv6-multicast-
source-address
|         | +---rw max-groups? uint32
|         | +---rw max-entries? uint32
|         | +---rw version? identityref
|         +---rw pim {vpn-common:pim}?

```

```

seconds16      |             +--rw hello-interval?   rt-types:timer-value-
|
|             +--rw dr-priority?      uint32
+--rw msdp {msdp}?
|  +--rw peer?                          inet:ipv4-address
|  +--rw local-address?                 inet:ipv4-address
|  +--rw status
|    +--rw admin-status
|    |  +--rw status?                   identityref
|    |  +--rw last-change?              yang:date-and-time
|    +--ro oper-status
|      +--ro status?                    identityref
|      +--ro last-change?                yang:date-and-time
+--rw groups
|  +--rw group* [group-id]
|    +--rw group-id                     string
+--rw status
|  +--rw admin-status
|  |  +--rw status?                     identityref
|  |  +--rw last-change?                yang:date-and-time
|  +--ro oper-status
|    +--ro status?                      identityref
|    +--ro last-change?                  yang:date-and-time
+--rw vpn-network-accesses
  +--rw vpn-network-access* [id]
    +--rw id                             vpn-common:vpn-id
    +--rw interface-id?                  string
    +--rw description?                   string
    +--rw vpn-network-access-type?       identityref
    +--rw vpn-instance-profile?          string
    +--rw status
    |  +--rw admin-status
    |  |  +--rw status?                  identityref
    |  |  +--rw last-change?              yang:date-and-time
    |  +--ro oper-status
    |    +--ro status?                    identityref
    |    +--ro last-change?                yang:date-and-time
    +--rw connection
    |  +--rw encapsulation
    |  |  +--rw type?                    identityref
    |  |  +--rw dot1q
    |  |  |  +--rw tag-type?              identityref
    |  |  |  +--rw cvlan-id?              uint16
    |  |  +--rw priority-tagged

```

```

| | | +--rw tag-type?  identityref
| | +--rw qinq
| |   +--rw tag-type?  identityref
| |   +--rw svlan-id   uint16
| |   +--rw cvlan-id   uint16
| +--rw (l2-service)?
| | +--:(l2-tunnel-service)
| | | +--rw l2-tunnel-service
| | |   +--rw type?      identityref
| | |   +--rw pseudowire
| | |     | +--rw vcid?   uint32
| | |     | +--rw far-end? union
| | |     +--rw vpls
| | |       | +--rw vcid?   uint32
| | |       | +--rw far-end* union
| | |       +--rw vxlan
| | |         +--rw vni-id      uint32
| | |         +--rw peer-mode?  identityref
| | |         +--rw peer-ip-address* inet:ip-address
| | +--:(l2vpn)
| |   +--rw l2vpn-id?          vpn-common:vpn-id
| +--rw l2-termination-point?  string
| +--rw local-bridge-reference? string
| +--rw bearer-reference?      string {vpn-
common:bearer-reference}?
| +--rw lag-interface {vpn-common:lag-interface}?
|   +--rw lag-interface-id?  string
|   +--rw member-link-list
|     +--rw member-link* [name]
|       +--rw name          string
+--rw ip-connection
| +--rw l3-termination-point?  string
| +--rw ipv4 {vpn-common:ipv4}?
| | +--rw local-address?
inet:ipv4-address
| | +--rw prefix-length?
uint8
| | +--rw address-allocation-type?
identityref
| | +--rw (allocation-type)?
| |   +--:(provider-dhcp)
| |     | +--rw dhcp-service-type?
enumeration
| |     | +--rw (service-type)?
| |     |   +--:(relay)

```

```

inet:ipv4-address      | | | | +--rw server-ip-address*
                       | | | | +---:(server)
                       | | | |   +--rw (address-assign)?
                       | | | |     +---:(number)
uint16                 | | | | | +--rw number-of-dynamic-address?
                       | | | | | +---:(explicit)
                       | | | | |   +--rw customer-addresses
                       | | | | |     +--rw address-pool* [pool-id]
string                 | | | | |   +--rw pool-id
inet:ipv4-address     | | | | |   +--rw start-address
inet:ipv4-address     | | | | |   +--rw end-address?
                       | | | | +---:(dhcp-relay)
                       | | | | | +--rw customer-dhcp-servers
address               | | | | |   +--rw server-ip-address*  inet:ipv4-
                       | | | | +---:(static-addresses)
-> ../address/address-id | | | | |   +--rw primary-address?
                       | | | | |   +--rw address* [address-id]
                       | | | | |   +--rw address-id          string
address               | | | | |   +--rw customer-address?  inet:ipv4-
                       | | | | +--rw ipv6 {vpn-common:ipv6}?
address               | | | | |   +--rw local-address?      inet:ipv6-
                       | | | | |   +--rw prefix-length?      uint8
                       | | | | |   +--rw address-allocation-type?  identityref
                       | | | | |   +--rw (allocation-type)?
                       | | | | |     +---:(provider-dhcp)
                       | | | | |     | +--rw provider-dhcp
enumeration           | | | | |     |   +--rw dhcp-service-type?
                       | | | | |     |   +--rw (service-type)?
                       | | | | |     |   +---:(relay)
inet:ipv6-address     | | | | |     | | +--rw server-ip-address*
                       | | | | |     | | +---:(server)
                       | | | | |     | |   +--rw (address-assign)?
                       | | | | |     | |     +---:(number)
address?  uint16      | | | | |     | | | +--rw number-of-dynamic-
                       | | | | |     | | | +---:(explicit)

```



```

{vpn-common:ipv6}? | | +---rw ipv6-lan-prefixes* [lan next-hop]
| | +---rw lan inet:ipv6-prefix
| | +---rw lan-tag? string
| | +---rw next-hop union
common:bfd}? | | +---rw bfd-enable? boolean {vpn-
| | +---rw metric? uint32
| | +---rw preference? uint32
| | +---rw status
| | +---rw admin-status
| | | +---rw status? identityref
time | | | +---rw last-change? yang:date-and-
| | +---ro oper-status
| | +---ro status? identityref
time | | +---ro last-change? yang:date-and-
| +---rw bgp
| | +---rw description? string
| | +---rw local-as? inet:as-number
| | +---rw peer-as inet:as-number
| | +---rw address-family? identityref
| | +---rw local-address? union
| | +---rw neighbor* inet:ip-address
| | +---rw multihop? uint8
| | +---rw as-override? boolean
| | +---rw allow-own-as? uint8
| | +---rw prepend-global-as? boolean
| | +---rw send-default-route? boolean
origin | | +---rw site-of-origin? rt-types:route-
route-origin | | +---rw ipv6-site-of-origin? rt-types:ipv6-
| | +---rw redistribute-connected* [address-family]
| | | +---rw address-family identityref
| | | +---rw enable? boolean
| | +---rw bgp-max-prefix
| | | +---rw max-prefix? uint32
| | | +---rw warning-threshold? decimal64
| | | +---rw violate-action? enumeration
| | | +---rw restart-timer? uint32
| | +---rw bgp-timers
| | | +---rw keepalive? uint16
| | | +---rw hold-time? uint16
| | +---rw authentication

```

```

| | | +---rw enable?          boolean
| | | +---rw keying-material
| | |   +---rw (option)?
| | |     +---:(ao)
| | |       | +---rw enable-ao?      boolean
| | |       | +---rw ao-keychain?    key-
chain:key-chain-ref
| | |     +---:(md5)
| | |       | +---rw md5-keychain?   key-
chain:key-chain-ref
| | |     +---:(explicit)
| | |       | +---rw key-id?         uint32
| | |       | +---rw key?           string
| | |       | +---rw crypto-algorithm?
identityref
| | |     +---:(ipsec)
| | |       +---rw sa?              string
| | |   +---rw status
| | |     +---rw admin-status
| | |       | +---rw status?        identityref
| | |       | +---rw last-change?   yang:date-and-time
| | |     +---ro oper-status
| | |       +---ro status?         identityref
| | |       +---ro last-change?    yang:date-and-time
+---rw ospf
| | +---rw address-family?  identityref
| | +---rw area-id         yang:dotted-quad
| | +---rw metric?        uint16
| | +---rw sham-links {vpn-common:rtg-ospf-sham-
link)?
| | | +---rw sham-link* [target-site]
| | |   +---rw target-site   string
| | |   +---rw metric?      uint16
| | | +---rw max-lsa?       uint32
| | | +---rw authentication
| | |   +---rw enable?      boolean
| | |   +---rw keying-material
| | |     +---rw (option)?
| | |       +---:(auth-key-chain)
| | |       | +---rw key-chain?    key-
chain:key-chain-ref
| | |       +---:(auth-key-explicit)
| | |         | +---rw key-id?     uint32
| | |         | +---rw key?       string
| | |         | +---rw crypto-algorithm?
identityref

```

```

| | | +---:(ipsec)
| | | +---rw sa? string
| | +---rw status
| | +---rw admin-status
| | | +---rw status? identityref
| | | +---rw last-change? yang:date-and-time
| | +---ro oper-status
| | +---ro status? identityref
| | +---ro last-change? yang:date-and-time
| +---rw isis
| | +---rw address-family? identityref
| | +---rw area-address area-address
| | +---rw level? identityref
| | +---rw metric? uint16
| | +---rw mode? enumeration
| | +---rw authentication
| | | +---rw enable? boolean
| | | +---rw keying-material
| | | +---rw (option)?
| | | +---:(auth-key-chain)
| | | | +---rw key-chain? key-
chain:key-chain-ref
| | | +---:(auth-key-explicit)
| | | +---rw key-id? uint32
| | | +---rw key? string
| | | +---rw crypto-algorithm?
identityref
| | +---rw status
| | +---rw admin-status
| | | +---rw status? identityref
| | | +---rw last-change? yang:date-and-time
| | +---ro oper-status
| | +---ro status? identityref
| | +---ro last-change? yang:date-and-time
| +---rw rip
| | +---rw address-family? identityref
| | +---rw timers
| | | +---rw update-interval? uint16
| | | +---rw invalid-interval? uint16
| | | +---rw holddown-interval? uint16
| | | +---rw flush-interval? uint16
| | +---rw default-metric? uint8
| | +---rw authentication
| | | +---rw enable? boolean
| | | +---rw keying-material

```

```

|         | |         +---rw (option)?
|         | |         +---:(auth-key-chain)
chain:key-chain-ref |         | |         | +---rw key-chain?         key-
|         | |         +---:(auth-key-explicit)
|         | |         +---rw key?         string
identityref         |         | |         +---rw crypto-algorithm?
|         | +---rw status
|         | +---rw admin-status
|         | | +---rw status?         identityref
|         | | +---rw last-change?    yang:date-and-time
|         | +---ro oper-status
|         | +---ro status?         identityref
|         | +---ro last-change?    yang:date-and-time
+---rw vrrp
|         +---rw address-family?    identityref
|         +---rw vrrp-group?        uint8
|         +---rw backup-peer?       inet:ip-address
|         +---rw virtual-ip-address* inet:ip-address
|         +---rw priority?          uint8
|         +---rw ping-reply?        boolean
|         +---rw status
|         +---rw admin-status
|         | +---rw status?         identityref
|         | +---rw last-change?    yang:date-and-time
|         +---ro oper-status
|         +---ro status?         identityref
|         +---ro last-change?    yang:date-and-time
+---rw oam
| +---rw bfd {vpn-common:bfd}?
|         +---rw session-type?      identityref
|         +---rw desired-min-tx-interval? uint32
|         +---rw required-min-rx-interval? uint32
|         +---rw local-multiplier?   uint8
|         +---rw holdtime?          uint32
|         +---rw profile?           string
|         +---rw authentication!
|         | +---rw key-chain?      key-chain:key-chain-ref
|         | +---rw meticulous?    boolean
|         +---rw status
|         +---rw admin-status
|         | +---rw status?        identityref
|         | +---rw last-change?   yang:date-and-time
|         +---ro oper-status

```

```

|           +---ro status?          identityref
|           +---ro last-change?     yang:date-and-time
+---rw security
| +---rw encryption {vpn-common:encryption}?
| | +---rw enabled?    boolean
| | +---rw layer?      enumeration
| +---rw encryption-profile
|   +---rw (profile)?
|     +---:(provider-profile)
|       | +---rw profile-name?      string
|       +---:(customer-profile)
|         +---rw customer-key-chain? key-chain:key-
chain-ref
+---rw service
+---rw pe-to-ce-bandwidth?  uint64 {vpn-
common:inbound-bw}?
+---rw ce-to-pe-bandwidth?  uint64 {vpn-
common:outbound-bw}?
+---rw mtu?                  uint32
+---rw qos {vpn-common:qos}?
| +---rw qos-classification-policy
| | +---rw rule* [id]
| |   +---rw id                string
| |   +---rw (match-type)?
| |     | +---:(match-flow)
| |     | | +---rw (l3)?
| |     | | | +---:(ipv4)
| |     | | | | +---rw ipv4
| |     | | | | | +---rw dscp?
inet:dscp
| |     | | | | | +---rw ecn?
uint8
| |     | | | | | +---rw length?
uint16
| |     | | | | | +---rw ttl?
uint8
| |     | | | | | +---rw protocol?
uint8
| |     | | | | | +---rw ihl?
uint8
| |     | | | | | +---rw flags?
bits
| |     | | | | | +---rw offset?
uint16
| |     | | | | | +---rw identification?
uint16
| |     | | | | | +---rw (destination-network)?

```


						+++rw (source-port)?
operator)						+++:(source-port-range-or-
range-or-operator						+++rw source-port-
or-operator)?						+++rw (port-range-
						+++:(range)
port inet:port-number						+++rw lower-
port inet:port-number						+++rw upper-
						+++:(operator)
operator? operator						+++rw
inet:port-number						+++rw port
						+++rw (destination-port)?
range-or-operator)						+++:(destination-port-
range-or-operator						+++rw destination-port-
or-operator)?						+++rw (port-range-
						+++:(range)
port inet:port-number						+++rw lower-
port inet:port-number						+++rw upper-
						+++:(operator)
operator? operator						+++rw
inet:port-number						+++rw port
						+++:(udp)
						+++rw udp
uint16						+++rw length?
						+++rw (source-port)?
operator)						+++:(source-port-range-or-
range-or-operator						+++rw source-port-
or-operator)?						+++rw (port-range-
						+++:(range)
port inet:port-number						+++rw lower-
port inet:port-number						+++rw upper-


```

| | | | | +--:(operator)
| | | | | +---rw
operator? operator
| | | | | +---rw port
inet:port-number
| | | | | +---rw (destination-port)?
| | | | | +--:(destination-port-
range-or-operator)
| | | | | +---rw destination-port-
range-or-operator
| | | | | +---rw (port-range-
or-operator)?
| | | | | +--:(range)
port inet:port-number | +---rw lower-
port inet:port-number | +---rw upper-
operator? operator | +--:(operator)
| | | | | +---rw
inet:port-number | +---rw port
| | | | +--:(match-application)
| | | | +---rw match-application? identityref
| | | +---rw target-class-id? string
| +---rw qos-action
| | +---rw rule* [id]
| | +---rw id string
| | +---rw target-class-id? string
| | +---rw inbound-rate-limit? decimal64
| | +---rw outbound-rate-limit? decimal64
| +---rw qos-profile
| +---rw qos-profile* [profile]
| +---rw profile string
| +---rw direction? identityref
+---rw carriers-carrier {vpn-common:carriers-carrier}?
| +---rw signaling-type? enumeration
+---rw ntp
| +---rw broadcast? enumeration
| +---rw auth-profile
| | +---rw profile-id? string
| +---rw status
| +---rw admin-status
| | +---rw status? identityref
| | +---rw last-change? yang:date-and-time
| +---ro oper-status
| +---ro status? identityref

```

```

|         +---ro last-change?   yang:date-and-time
+---rw multicast {vpn-common:multicast}?
  +---rw access-type?           enumeration
  +---rw address-family?        identityref
  +---rw protocol-type?         enumeration
  +---rw remote-source?         boolean
  +---rw igmp {vpn-common:igmp}?
  | +---rw static-group* [group-addr]
  | | +---rw group-addr        rt-types:ipv4-
multicast-group-address
  | | +---rw source-addr?     rt-types:ipv4-
multicast-source-address
  | +---rw max-groups?          uint32
  | +---rw max-entries?         uint32
  | +---rw max-group-sources?   uint32
  | +---rw version?             identityref
  | +---rw status
  |   +---rw admin-status
  |   | +---rw status?         identityref
  |   | +---rw last-change?    yang:date-and-time
  |   +---ro oper-status
  |     +---ro status?         identityref
  |     +---ro last-change?    yang:date-and-time
+---rw mld {vpn-common:mld}?
  | +---rw static-group* [group-addr]
  | | +---rw group-addr        rt-types:ipv6-
multicast-group-address
  | | +---rw source-addr?     rt-types:ipv6-
multicast-source-address
  | +---rw max-groups?          uint32
  | +---rw max-entries?         uint32
  | +---rw max-group-sources?   uint32
  | +---rw version?             identityref
  | +---rw status
  |   +---rw admin-status
  |   | +---rw status?         identityref
  |   | +---rw last-change?    yang:date-and-time
  |   +---ro oper-status
  |     +---ro status?         identityref
  |     +---ro last-change?    yang:date-and-time
+---rw pim {vpn-common:pim}?
  +---rw hello-interval?       rt-types:timer-value-
seconds16
  +---rw dr-priority?          uint32
  +---rw status

```

```

+--rw admin-status
| +--rw status?          identityref
| +--rw last-change?    yang:date-and-time
+--ro oper-status
   +--ro status?        identityref
   +--ro last-change?   yang:date-and-time

```

Augmentations Model for Kafka Settings

```

augment "/ncs:devices/ncs:device" {
    when "starts-with(ncs:device-type/ncs:generic/ncs:ned-id,
'nokia-nsp')";
    container ns-kafka-settings {
        tailf:info "Nokia NSP Notification System";
        description
            "Nokia NSP Notification System";

        leaf active {
            tailf:info "Enable message filter for NSP
notification";
            description
                "Kafka deployment feedback Enable/Disable";
            type boolean;
            default false;
        }

        leaf listeners {
            when "../active = 'true'";
            type string;
        }

        leaf topic-name {
            when "../active = 'true'";
            type string;
            default "nsp-yang-model.change-notif";
        }

        container authentication {
            when "../active = 'true'";
            leaf authentication-type {
                type kafka-authentication-method;
            }
        }
    }
}

```

```
// mandatory true;
}

leaf ssl-key-location {
    when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
    type string;
}

leaf ssl-key-password {
    when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
    tailf:suppress-echo "true";
    type tailf:aes-256-cfb-128-encrypted-string;
}

leaf ssl-ca-location {
    when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
    type string;
}

leaf ssl-certificate-location {
    when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
    type string;
}

leaf ssl.endpoint.identification.algorithm {
    when "../authentication-type = 'SSL' or
../authentication-type = 'SASL_SSL'";
    type enumeration {
        enum none;
        enum HTTPS;
    }
    default "none";
}

leaf username {
    when "../authentication-type = 'SASL_SSL'";
```

```
        type string;
    }

    leaf password {
        when "../authentication-type = 'SASL_SSL'";
        tailf:suppress-echo "true";
        type tailf:aes-256-cfb-128-encrypted-string;
    }
}
}
```