Discovery 5: Create a VLAN Template Service

Introduction

In this activity, you will learn how to create a VLAN service. After completing this activity, you will be able to meet these objectives:

- Use the Cisco NSO CLI to configure a sample service
- Create the service template
- Compile and deploy the service
- Troubleshoot the service

Job Aids

The following job aids are available to help you complete the lab activities:

- This lab guide
- Student guide, for general explanations

Job Aids for Task 1

The following table describes the service characteristics and requirements. Use this information as a starting point for creating a service model in the first task.

Attribute	Name	Data Type	Restriction	Other
Service name	vlan	list	—	—
Service instance name	*name*	string	—	key for list vlan
VLAN ID	*vlan-id*	uint32	Range: 1 – 4096	unique value

Attribute	Name	Data Type	Restriction	Other
Device	*device*	list	—	—
Device name	*name*	leafref	—	key for list device
Interface type	*intf-type*	enumeration	Options: FastEthernet GigabitEthernet Ethernet	intf-type and intf-id combination must be unique
Interface ID	*intf-id*	string	—	intf-type and intf-id combination must be unique

Job Aids for Task 2

A network engineer has provided you with the actual configuration for the new service, one for each device type.

Cisco IOS platform (device SW31):

```
vlan 102
interface FastEthernet0/0
 switchport access vlan 102
 switchport mode access
!
```

Cisco NX-OS platform (device SW32):

```
vlan 102
interface Ethernet1/1
 switchport access vlan 102
 switchport mode access
!
```

Required Resources

The following resources and equipment are required for completing the activities in this lab guide:

- PC or laptop with a web browser
- Access to the internet

Command Syntax Reference

This lab guide uses the following conventions for command syntax:

Formatting	Description and Examples
show running config	Commands in steps use this formatting.
Example	Type **show running config**
Example	Use the **name** command.
`show running config`	Commands in CLI outputs and configurations use this formatting.
highlight	CLI output that is important is highlighted.

Formatting	Description and Examples
Example	`student@student-vm:~$ ncs --version` `5.8.2.1`
Example	Save your current configuration as the default **startup config**. `Router Name copy running startup`
brackets ([ ])	Indicates the optional element. You can choose one of the options.
Example:	`(config-if) frame-relay lmi-type {ansi|cisco|q933a}`
italics font	Arguments for which you supply values.
Example	Open file **ip tcp window-size** *bytes*
angle brackets (<>)	In contexts that do not allow italics, arguments for which you supply values are enclosed in angle brackets [<>]. Do not type the brackets when entering the command.
Example	If the command syntax is **ping** *<ip_address>*, you enter ping *192.32.10.12*
string	A non-quoted set of characters. Type the characters as-is.
Example	(config) **hostname MyRouter**
vertical line (|)	Indicates that you enter one of the choices. The vertical line separates choices. Do not type the vertical line when entering the command.
Example	If the command syntax is **show ip route|arp**, you enter either **show ip route** or **show ip arp**, but not both.

Command List

The following are the most common commands that you will need:

Linux Shell:

Command	Comment
source /home/student/nso-5.8/ncsrc	Source NSO environmental variables.
ls|ll	Display contents of the current directory.
cd	Move directly to user home directory.
cd ..	Exit the current directory.
cd test	Move into folder "test," which is a subfolder of the current directory.
cd /home/student/test	Move into folder "test" by specifying direct path to it, starting from the root of directory system.
ncs_cli -Cu admin	Log in to NSO CLI directly from local server.

NSO CLI:

Command	Comment
switch cli	Change CLI style.
show ?	Display all command options for current mode.
configure	Enter configuration mode.

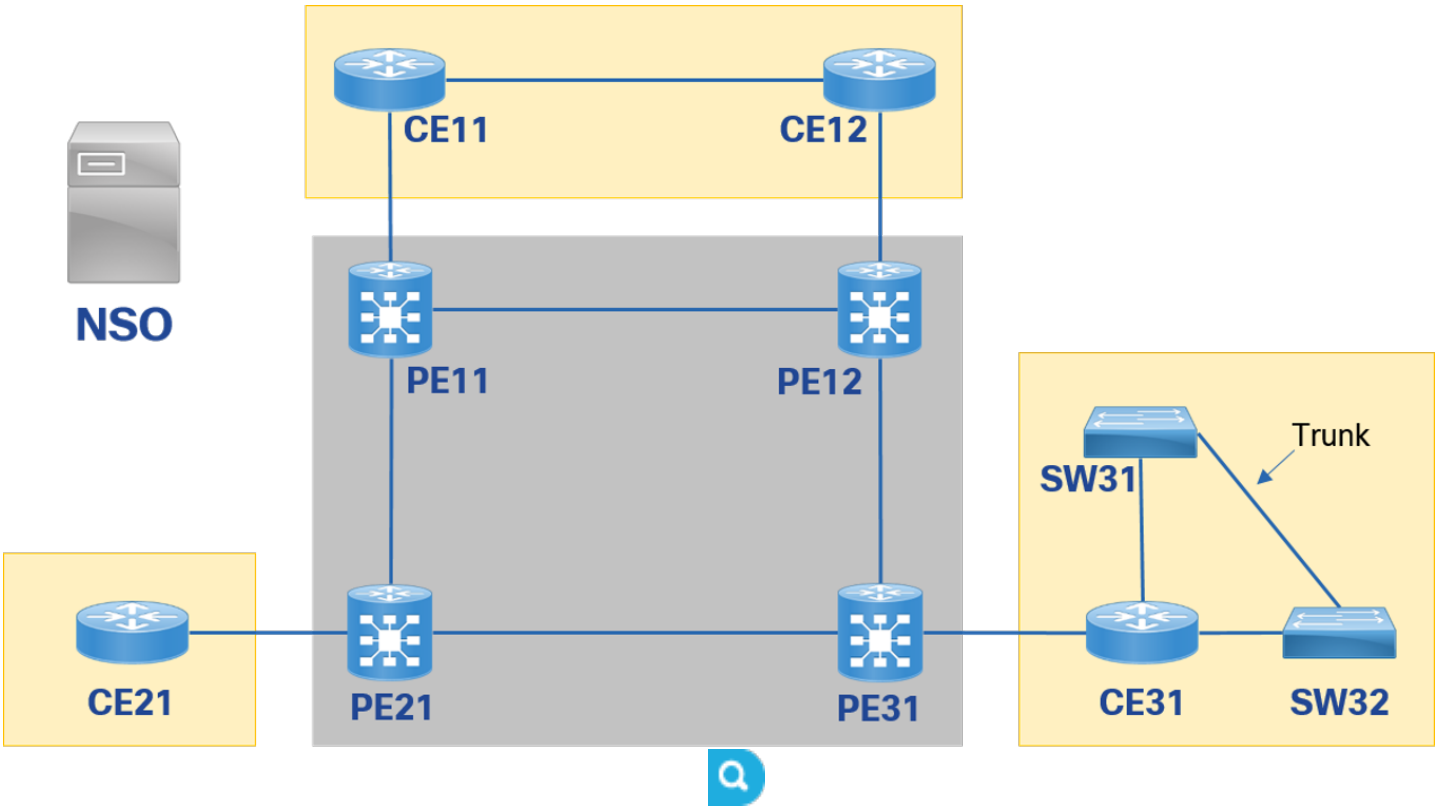
Command	Comment
commit	Commit new configuration (configuration mode only command).
show configuration	Display new configuration that has not yet been committed (configuration mode only command).

Lab Topology Information

Your lab session is your own personal sandbox. Whatever you do in your session will not be reflected in anyone else's session. There are two topologies. The general one is your lab environment, and it has your workstation and a Linux server. On the Linux server within that topology is your second topology with your NSO installation, together with numerous netsim routers and switches that are logically grouped into a network topology. This network will be the one that you will orchestrate with your NSO.

- The network topology is designed to cover both the service provider and enterprise use cases. It is a simulated netsim network; devices have no control or data plane. Devices will, however, accept or reject a configuration sent by the NSO, just as real devices would.

Topology



Visual Objective

The following figure illustrates what you will accomplish in this activity. In this lab, you will provide a service to create and configure a VLAN between the two switches in the network.

The graphic provides a visual aid for this activity.

Task 1: Design a Service Model

The first task requires you to design a service model based on the provided high-level service requirements.

> The final solutions for all labs, including this one, are located in the ~/solutions directory. You can use them for copy-pasting longer pieces of code and as a reference point for troubleshooting your packages.

Activity

Complete these steps:

Step 1

Connect to the NSO server by clicking the NSO icon.

Step 2

Open the terminal window by clicking the Terminal icon on the bottom bar.

```
student@student-vm:~$
```

Step 3

Change your current location to the directory **$HOME/nso-run/packages**.

```
student@student-vm:~$ cd $HOME/nso-run/packages
```

Step 4

Create a template-based service skeleton, using the **ncs-make-package** command. Use **vlan** as the name of the new service.

> You can use **ncs-make-package --help** or **man ncs-make-package** to learn about all the available options.

```
student@student-vm:~/nso-run/packages$ ls
cisco-ios-cli-6.85  cisco-iosxr-cli-7.41  cisco-nx-cli-5.23  loopback
student@student-vm:~/nso-run/packages$ ncs-make-package --service-skeleton template vlan
student@student-vm:~/nso-run/packages$ ls
cisco-ios-cli-6.85  cisco-iosxr-cli-7.41  cisco-nx-cli-5.23  loopback  vlan
student@student-vm:~/nso-run/packages$
```

Step 5

Navigate to the vlan/src/yang directory and display the YANG service model.

This is how the file should look when you open it for the first time:

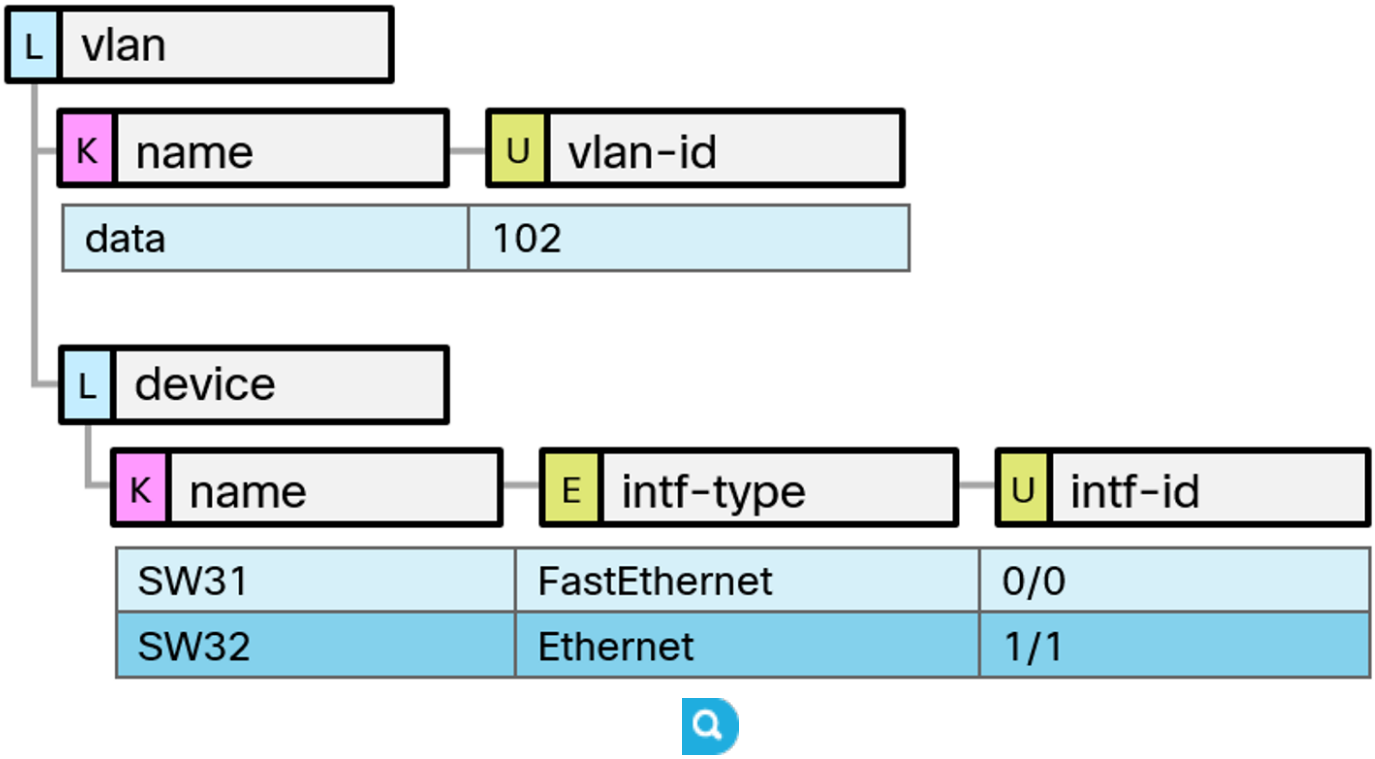
```
student@student-vm:~/nso-run/packages$ cd vlan/src/yang
student@student-vm:~/nso-run/packages/vlan/src/yang$ cat vlan.yang
module vlan {
  namespace "http://com/example/vlan";
```

```
  prefix vlan;

  import ietf-inet-types {
    prefix inet;
  }
  import tailf-ncs {
    prefix ncs;
  }

  list vlan {
    key name;

    uses ncs:service-data;
    ncs:servicepoint "vlan";

    leaf name {
      type string;
    }

    // may replace this with other ways of refering to the devices.
    leaf-list device {
      type leafref {
        path "/ncs:devices/ncs:device/ncs:name";
      }
    }

    // replace with your own stuff here
    leaf dummy {
      type inet:ipv4-address;
    }
  }
}
```

Step 6

Refer to the following table to fine-tune the service attributes that you will use in your YANG service model.

Attribute	**Name**	**Data Type**	**Restriction**	**Other**
Service name	vlan	list	—	—
Service instance name	*name*	string	—	key for list vlan
VLAN ID	*vlan-id*	uint32	Range: 1 – 4096	unique value
Device	*device*	list	—	—
Device name	*name*	leafref	—	key for list device
Interface type	*intf-type*	enumeration	Options: FastEthernet GigabitEthernet Ethernet	intf-type and intf-id combination must be unique
Interface ID	*intf-id*	string	—	intf-type and intf-id combination must be unique



You can find the data table in the Job Aids.

Step 7

Open the YANG service model for editing with the **code vlan.yang** command.

```
student@student-vm:~/nso-run/packages/vlan/src/yang$ code vlan.yang
```

You can edit the YANG service model YANG file by using Visual Studio Code or any editor of your choice. Visual Studio Code text editor on the workstation will provide you with syntax highlighting for YANG.

Step 8

Rename the namespace to http://cisco.com/example/vlan, remove the leaf-list *device*, leaf *dummy*, and import statement for *ietf-inet-types*. They will not be used for the vlan service and will be replaced by other statements. Add an augment */ncs:services* statement, to include the *list vlan* in the *services* branch of the CDB.

```
module vlan {
  namespace "http://cisco.com/example/vlan";
  prefix vlan;

  import tailf-ncs {
    prefix ncs;
  }

 augment /ncs:services {
  list vlan {
    key name;

    uses ncs:service-data;
    ncs:servicepoint "vlan";
```

```
    leaf name {
      type string;
    }
  }
 }
}
```

YANG allows a module to insert additional nodes into existing data models. The "augment" statement defines the location in the data model hierarchy where new nodes are inserted. In your case, the location is under the services branch of the CDB.

Step 9

Add a list, that is used to support multiple instances of the VLAN service. The list records are uniquely identified, using the key attribute (name). Add another import statement for **tailf-common**, to support YANG annotations. You can also add a custom description and revision information.

```
module vlan {
  namespace "http://cisco.com/example/vlan";
  prefix vlan;

  import tailf-ncs {
    prefix ncs;
  }

  import tailf-common {
    prefix tailf;
  }

  description "This is a VLAN service package.";

  revision 2022-09-01 {
    description
      "Initial revision.";
  }

 augment /ncs:services {
  list vlan {
    tailf:info "VLAN Service";
    key name;

    uses ncs:service-data;
    ncs:servicepoint "vlan";

    leaf name {
      tailf:info "Service Instance Name";
      type string;
    }
  }
 }
}
```

Step 10

Create a unique VLAN ID for each VLAN within a standard VLAN range.

```
module vlan {

 ...
```

```
 augment /ncs:services {
  list vlan {
    tailf:info "VLAN Service";
    key name;
    unique vlan-id;

    uses ncs:service-data;
    ncs:servicepoint "vlan";

    leaf name {
      tailf:info "Service Instance Name";
      type string;
    }

    leaf vlan-id {
      tailf:info "Unique VLAN ID";
      type uint32 {
        range "1..4096";
      }
    }

  }
 }
}
```

Step 11

Add a list of devices. Each VLAN can be configured on multiple devices, which are represented with the list device. Each device in the list is identified by a unique name.

```
module vlan {

 ...

 augment /ncs:services {
  list vlan {
    tailf:info "VLAN Service";
    key name;
    unique vlan-id;

    uses ncs:service-data;
    ncs:servicepoint "vlan";

    leaf name {
      tailf:info "Service Instance Name";
      type string;
    }

    leaf vlan-id {
      tailf:info "Unique VLAN ID";
      type uint32 {
        range "1..4096";
      }
    }

    list device {
      tailf:info "L3 Switch";
      key name;

      leaf name {
        tailf:info "Device Name";
        type leafref {
          path "/ncs:devices/ncs:device/ncs:name";
```

```
        }
      }
    }

  }
 }
}
```

Step 12

Create a list of interfaces. On each device, you can configure multiple interfaces, which are represented with the list interface. Each interface can be one of the three available types, and it must have a unique identifier (intf-id).

```
module vlan {

 ...

 augment /ncs:services {
  list vlan {

    ...

    list device {
      tailf:info "L3 Switch";
      key name;

      leaf name {
        tailf:info "Device Name";
        type leafref {
          path "/ncs:devices/ncs:device/ncs:name";
        }
      }

      list interface {
        tailf:info "Ethernet Interface";
        key "intf-type intf-id";

        leaf intf-type {
          tailf:info "Ethernet Interface Type";
          type enumeration {
            enum Ethernet;
            enum FastEthernet;
            enum GigabitEthernet;
          }
        }

        leaf intf-id {
          tailf:info "Ethernet Interface ID";
          type string;
        }
      }

    }

  }
 }
}
```

Step 13

Save the file.

Step 14

You can validate your YANG file by using **pyang**. Keep making corrections to your YANG file until pyang no longer produces any output for *vlan.yang* (that is, until there are no errors).

```
student@student-vm:~/nso-run/packages/vlan/src/yang$ pyang vlan.yang
/home/student/nso-5.8/src/ncs/yang/tailf-ncs-cluster.yang:224: error: unexpected keyword
"default"
/home/student/nso-5.8/src/ncs/yang/tailf-ncs-cluster.yang:225: error: unexpected keyword
"default"
/home/student/nso-5.8/src/ncs/yang/tailf-ncs-cluster.yang:226: error: unexpected keyword
"default"
...
/home/student/nso-5.8/src/ncs/yang/tailf-ncs-packages.yang:1006: warning: node "tailf-
ncs::high-availability" is not found in module "tailf-ncs-packages"
student@student-vm:~/nso-run/packages/vlan/src/yang$
```

Pyang can show errors and warnings in its output. In the lab, *only errors* pertaining to **vlan.yang** must to be fixed.

Step 15

Change the directory to the parent directory where the Makefile is located. Use the **make** command to compile the package.

```
student@student-vm:~/nso-run/packages/vlan/src/yang$ cd ..
student@student-vm:~/nso-run/packages/vlan/src$ make
mkdir -p ../load-dir
/home/student/nso-5.8/bin/ncsc `ls vlan-ann.yang  > /dev/null 2>&1 && echo "-a vlan-ann.yang"`
\
        --fail-on-warnings \
         \
        -c -o ../load-dir/vlan.fxs yang/vlan.yang
```

Step 16

After the package is compiled, log in to the NSO CLI and issue the **packages reload** command.

```
student@student-vm:~/nso-run/packages/vlan/src$ ncs_cli -Cu admin
admin@ncs packages reload

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
    package cisco-ios-cli-6.85
    result true
}
reload-result {
    package cisco-iosxr-cli-7.41
    result true
}
reload-result {
    package cisco-nx-cli-5.23
    result true
}
reload-result {
    package loopback
    result true
}
```

```
reload-result {
    package vlan
    result true
}
admin@ncs
System message at 2022-09-17 15:55:58...
    Subsystem stopped: ncs-dp-4-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 15:55:58...
    Subsystem stopped: ncs-dp-3-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 15:55:58...
    Subsystem started: ncs-dp-5-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 15:55:58...
    Subsystem started: ncs-dp-6-cisco-nx-cli-5.23:NexusDp
admin@ncs
```

Step 17

Check the availability of CLI commands to manage the service. Use the **services vlan** command, followed by a question mark, to investigate the new CLI options that are available for the provisioning of service instances.

```
admin@ncs services ?
Possible completions:
  check-sync   Check if device configuration is according to the services
  loopback     Loopback Service
  vlan         VLAN Service
admin@ncs services vlan ?
% No entries found
admin@ncs
```

Step 18

Exit NSO CLI. Verify that the directory structure and file templates for the new service match the expected structure.

```
student@student-vm:~/nso-run/packages/vlan/src$ cd $HOME/nso-run/packages
student@student-vm:~/nso-run/packages$ ls
cisco-ios-cli-6.85  cisco-iosxr-cli-7.41  cisco-nx-cli-5.23  loopback  vlan
student@student-vm:~/nso-run/packages$ ls -l vlan
total 20
drwxrwxr-x 2 student student 4096 Sep 17 15:54 load-dir
-rw-rw-r-- 1 student student  366 Sep 17 15:31 package-meta-data.xml
drwxr-xr-x 3 student student 4096 Sep 17 15:31 src
drwxr-xr-x 2 student student 4096 Sep 17 15:31 templates
drwxr-xr-x 3 student student 4096 Jul  8 09:56 test
student@student-vm:~/nso-run/packages$ ls -l vlan/src/yang
total 4
-rw-rw-r-- 1 student student 1344 Sep 17 15:50 vlan.yang
student@student-vm:~/nso-run/packages$ ls -l vlan/templates
total 4
-rw-rw-r-- 1 student student 732 Sep 17 15:31 vlan-template.xml
student@student-vm:~/nso-run/packages$
```

Activity Verification

You have completed this task when you attain this result:

- You have a directory structure and file templates for a new service that matches the expected structure.

You are still unable to provision services, because the service template is only a placeholder at this point. You will develop it properly in the next task.

Task 2: Create Configuration Templates

In this task, you will configure the sample VLAN service from NSO CLI, in order to obtain the device-specific configuration in XML format. Then you can create a device template for the Cisco IOS and Cisco NX-OS platforms.

Activity

Complete these steps:

Step 1

Connect to the NSO CLI.

```
student@student-vm:~/nso-run/packages$ ncs_cli -Cu admin
admin@ncs
```

Step 2

A network engineer has provided you with the actual configuration for the new service, one for each device type.

Cisco IOS platform (device SW31):

```
vlan 102
interface FastEthernet0/0
 switchport access vlan 102
 switchport mode access
!
```

Cisco NX-OS platform (device SW32):

```
vlan 102
interface Ethernet1/1
 switchport access vlan 102
 switchport mode access
!
```

You can find the configuration for the new service in the Job Aids.

Step 3

Configure the preceding examples from NSO CLI.

Use the **devices device SW31** configuration command as a starting point to configure the Cisco IOS sample.

Use the **devices device SW32** configuration command as a starting point to configure the Cisco NX-OS sample.

```
admin@ncs config
Entering configuration mode terminal
admin@ncs(config) devices device SW31 config
admin@ncs(config-config) vlan 102
admin@ncs(config-vlan) exit
```

```
admin@ncs(config-config) interface FastEthernet 0/0
admin@ncs(config-if) switchport mode access
admin@ncs(config-if) switchport access vlan 102
admin@ncs(config-config) top
admin@ncs(config) devices device SW32 config
admin@ncs(config-config) vlan 102
admin@ncs(config-vlan) exit
admin@ncs(config-config) interface Ethernet 1/1
admin@ncs(config-if) switchport mode access
admin@ncs(config-if) switchport access vlan 102
admin@ncs(config-if) top
admin@ncs(config)
```

Step 4

Use the **commit dry-run outformat xml** command to retrieve the XML version of the configuration.

Verify that the output contains all the configured parameters. If some or all of the changes were committed earlier, the output will be missing those parts.

```
admin@ncs(config) commit dry-run outformat xml
result-xml {
    local-node {
        data <devices xmlns="http://tail-f.com/ns/ncs">
               <device>
                 <name>SW31</name>
                 <config>
                   <vlan xmlns="urn:ios">
                     <vlan-list>
                       <id>102</id>
                     </vlan-list>
                   </vlan>
                   <interface xmlns="urn:ios">
                     <FastEthernet>
                       <name>0/0</name>
                       <switchport>
                         <mode>
                           <access/>
                         </mode>
                         <access>
                           <vlan>102</vlan>
                         </access>
                       </switchport>
                     </FastEthernet>
                   </interface>
                 </config>
               </device>
               <device>
                 <name>SW32</name>
                 <config>
                   <vlan xmlns="http://tail-f.com/ned/cisco-nx">
                     <vlan-list>
                       <id>102</id>
                     </vlan-list>
                   </vlan>
                   <interface xmlns="http://tail-f.com/ned/cisco-nx">
                     <Ethernet>
                       <name>1/1</name>
                       <switchport>
                         <mode>access</mode>
                         <access>
                           <vlan>102</vlan>
                         </access>
```

```
                      </switchport>
                    </Ethernet>
                  </interface>
                </config>
              </device>
            </devices>
    }
}
admin@ncs(config) abort
admin@ncs
```

Step 5

Go to the templates subdirectory of your package skeleton (for example, $HOME/nso-run/packages/vlan/templates).

```
student@student-vm:~$ cd $HOME/nso-run/packages/vlan/templates
student@student-vm:~/nso-run/packages/vlan/templates$
```

Step 6

Open the template by using the **code vlan-template.xml** command.

This is how the template should look when you open it for the first time:

```
<config-template xmlns="http://tail-f.com/ns/config/1.0"
                 servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <device>
      <!--
          Select the devices from some data structure in the service
          model. In this skeleton the devices are specified in a leaf-list.
          Select all devices in that leaf-list:
      -->
      <name>{/device}</name>
      <config>
        <!--
            Add device-specific parameters here.
            In this skeleton the service has a leaf "dummy"; use that
            to set something on the device e.g.:
            <ip-address-on-device>{/dummy}</ip-address-on-device>
        -->
      </config>
    </device>
  </devices>
</config-template>
```

Step 7

Because the service model supports two types of devices, the XML configuration changes segment must accommodate both of them. Remove the provided comments under the <device> and <config> tags and add your own (DEVICE, IOS, NX-OS).

```
<config-template xmlns="http://tail-f.com/ns/config/1.0"
                 servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device}</name>
      <config>
```

```
        <!-- IOS -->

        <!-- NX-OS -->

      </config>
    </device>
  </devices>
</config-template>
```

Step 8

Insert the XML configuration created with **commit dry-run outformat xml** in the modified XML skeleton. Edit the segment with XML configuration code. Add both the Cisco IOS and Cisco NX-OS parts of the configuration. Remove the provided comments under the **<device>** and **<config>** tags.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device}</name>
      <config>

        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>102</id>
          </vlan-list>
        </vlan>
        <interface xmlns="urn:ios">
          <FastEthernet>
            <name>0/0</name>
            <switchport>
              <mode>
                <access/>
              </mode>
              <access>
                <vlan>102</vlan>
              </access>
            </switchport>
          </FastEthernet>
        </interface>

        <!-- NX-OS -->
        <vlan xmlns="http://tail-f.com/ned/cisco-nx">
          <vlan-list>
            <id>102</id>
          </vlan-list>
        </vlan>
        <interface xmlns="http://tail-f.com/ned/cisco-nx">
          <Ethernet>
            <name>1/1</name>
            <switchport>
              <mode>access</mode>
              <access>
                <vlan>102</vlan>
              </access>
            </switchport>
          </Ethernet>
        </interface>

      </config>
    </device>
  </devices>
```

```
</config-template>
```

Step 9

Replace all the static parameters with variables, which reference service attributes according to the hierarchy of the YANG data model.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>

        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <interface xmlns="urn:ios">
          <FastEthernet>
            <name>{intf-id}</name>
            <switchport>
              <mode>
                <access/>
              </mode>
              <access>
                <vlan>{/vlan-id}</vlan>
              </access>
            </switchport>
          </FastEthernet>
        </interface>

        <!-- NX-OS -->
        <vlan xmlns="http://tail-f.com/ned/cisco-nx">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <interface xmlns="http://tail-f.com/ned/cisco-nx">
          <Ethernet>
            <name>{intf-id}</name>
            <switchport>
              <mode>access</mode>
              <access>
                <vlan>{/vlan-id}</vlan>
              </access>
            </switchport>
          </Ethernet>
        </interface>

      </config>
    </device>
  </devices>
</config-template>
```

Step 10

Because you must support two interface types for the Cisco IOS platform, add a condition to check the interface type and then add a separate section for the GigabitEthernet interface type.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>

        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <interface xmlns="urn:ios">
          <?if {intf-type='FastEthernet'}?>
            <FastEthernet>
              <name>{intf-id}</name>
              <switchport>
                <mode>
                  <access/>
                </mode>
                <access>
                  <vlan>{/vlan-id}</vlan>
                </access>
              </switchport>
            </FastEthernet>
          <?end?>
          <?if {intf-type='GigabitEthernet'}?>
            <GigabitEthernet>
              <name>{intf-id}</name>
              <switchport>
                <mode>
                  <access/>
                </mode>
                <access>
                  <vlan>{/vlan-id}</vlan>
                </access>
              </switchport>
            </GigabitEthernet>
          <?end?>
        </interface>

        <!-- NX-OS -->

        ...

      </config>
    </device>
  </devices>
</config-template>
```

Step 11

Because the service model supports multiple interfaces, add a **foreach** loop, which will loop through the interface list and apply the configuration for all selected interfaces.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>

        <!-- IOS -->
```

```
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <?foreach {interface}?>
          <interface xmlns="urn:ios">
            <?if {intf-type='FastEthernet'}?>
              <FastEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{/vlan-id}</vlan>
                  </access>
                </switchport>
              </FastEthernet>
            <?end?>
            <?if {intf-type='GigabitEthernet'}?>
              <GigabitEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{/vlan-id}</vlan>
                  </access>
                </switchport>
              </GigabitEthernet>
            <?end?>

          </interface>
        <?end?>

        <!-- NX-OS -->
        <vlan xmlns="http://tail-f.com/ned/cisco-nx">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <?foreach {interface}?>
          <interface xmlns="http://tail-f.com/ned/cisco-nx">
            <Ethernet>
              <name>{intf-id}</name>
              <switchport>
                <mode>access</mode>
                <access>
                  <vlan>{/vlan-id}</vlan>
                </access>
              </switchport>
            </Ethernet>
          </interface>
        <?end?>

      </config>
    </device>
  </devices>
</config-template>
```

Step 12

Save the file.

Step 13

Connect to NSO CLI and reload the packages.

```
student@student-vm:~/nso-run/packages/vlan/templates$ ncs_cli -Cu admin
admin@ncs packages reload
reload-result {
    package cisco-ios-cli-6.85
    result true
}
reload-result {
    package cisco-iosxr-cli-7.41
    result true
}
reload-result {
    package cisco-nx-cli-5.23
    result true
}
reload-result {
    package loopback
    result true
}
reload-result {
    package vlan
    result true
}
admin@ncs
System message at 2022-09-17 16:41:12...
    Subsystem stopped: ncs-dp-6-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 16:41:12...
    Subsystem stopped: ncs-dp-5-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 16:41:12...
    Subsystem started: ncs-dp-7-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 16:41:12...
    Subsystem started: ncs-dp-8-cisco-nx-cli-5.23:NexusDp
admin@ncs
```

Troubleshooting Hints

- The variables must reference data according to the hierarchy of the YANG data model.
- You will learn more about troubleshooting in a later task.

Activity Verification

You have completed this task when you attain this result:

- You have successfully developed the **vlan-template.xml**, and the **packages reload** operation was successful.

Task 3: Deploy a Service Instance

In this task, you will deploy a service instance based on your newly installed service.

Activity

Complete these steps:

Step 1

Connect to the NSO CLI.

```
student@student-vm:~/nso-run/packages$ ncs_cli -Cu admin
admin@ncs
```

Step 2

Provision the first service instance.

```
admin@ncs config
Entering configuration mode terminal
admin@ncs(config) services vlan ?
% No entries found
Possible completions:
  Service Instance Name
admin@ncs(config) services vlan Internal
admin@ncs(config-vlan-Internal) vlan-id 102
admin@ncs(config-vlan-Internal) device SW31
admin@ncs(config-device-SW31) interface FastEthernet 0/0
admin@ncs(config-interface-FastEthernet/0/0) exit
admin@ncs(config-device-SW31) exit
admin@ncs(config-vlan-Internal) device SW32
admin@ncs(config-device-SW32) interface Ethernet 1/1
admin@ncs(config-interface-Ethernet/1/1) top
```

Step 3

When you have entered all the service parameters, preview the device changes by using the **commit dry-run** or **commit dry-run outformat native** command.

```
admin@ncs(config) commit dry-run
cli {
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          vlan {
            +                 vlan-list 102 {
            +                 }
                          }
                          interface {
                              FastEthernet 0/0 {
            +                     switchport {
            +                         mode {
            +                             access {
            +                             }
            +                         }
            +                         access {
            +                             vlan 102;
            +                         }
            +                     }
                              }
                          }
                      }
                  }
                  device SW32 {
                      config {
                          vlan {
            +                 vlan-list 102 {
            +                 }
                          }
                          interface {
```

```
                                Ethernet 1/1 {
                                    switchport {
              -                         mode trunk;
              +                         mode access;
                                        access {
              +                             vlan 102;
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
                services {
              +     vlan Internal {
              +         vlan-id 102;
              +         device SW31 {
              +             interface FastEthernet 0/0;
              +         }
              +         device SW32 {
              +             interface Ethernet 1/1;
              +         }
              +     }
                }
    }
}
```

Step 4

Commit the service instance creation by using the **commit** command, and exit the configuration mode.

```
admin@ncs(config) commit
Commit complete.
admin@ncs(config) exit
admin@ncs
```

Step 5

Verify that the service has been successfully deployed.

```
admin@ncs show running-config services vlan
services vlan Internal
 vlan-id 102
 device SW31
  interface FastEthernet 0/0
  !
 !
 device SW32
  interface Ethernet 1/1
  !
 !
!
admin@ncs show running-config services vlan | display xpath
/services/vlan:vlan[name='Internal']/vlan-id 102
/services/vlan:vlan[name='Internal']/device[name='SW31']/interface[intf-type='FastEthernet']
[intf-id='0/0']
/services/vlan:vlan[name='Internal']/device[name='SW32']/interface[intf-type='Ethernet'][intf-
id='1/1']
admin@ncs show running-config services vlan | tab
          VLAN                      INTF
NAME      ID    NAME  INTF TYPE     ID
-----------------------------------------
```

```
Internal  102   SW31  FastEthernet  0/0
                SW32  Ethernet      1/1
```

Activity Verification

You have completed this task when you attain this result:

- The service has been successfully deployed.

Task 4: Template Troubleshooting

In this task, you will intentionally misconfigure the template, view the output and then repair it.

Activity

Complete these steps:

Since you will be actively observing the log files while configuring NSO, it is very beneficial to have at least two console windows or tabs open.

Step 1

Delete the service that you have just created.

```
admin@ncs config
Entering configuration mode terminal
admin@ncs(config) no services vlan Internal
admin@ncs(config)
```

Step 2

Check what modifications will be done by using the **commit dry-run** command.

```
admin@ncs(config) commit dry-run
cli {
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          vlan {
            -                 vlan-list 102 {
            -                 }
                          }
                          interface {
                              FastEthernet 0/0 {
            -                     switchport {
            -                         mode {
            -                             access {
            -                             }
            -                         }
            -                         access {
            -                             vlan 102;
            -                         }
            -                     }
                              }
                          }
                      }
                  }
                  device SW32 {
                      config {
```

```
                        vlan {
-                           vlan-list 102 {
-                           }
                        }
                        interface {
                            Ethernet 1/1 {
                                switchport {
-                                   mode access;
+                                   mode trunk;
                                    access {
-                                       vlan 102;
                                    }
                                }
                            }
                        }
                    }
                }
            }
            services {
-               vlan Internal {
-                   vlan-id 102;
-                   device SW31 {
-                       interface FastEthernet 0/0;
-                   }
-                   device SW32 {
-                       interface Ethernet 1/1;
-                   }
-               }
            }
    }
}
```

Step 3

Commit the changes and exit NSO CLI.

```
admin@ncs(config) commit
Commit complete.
admin@ncs(config) exit
admin@ncs exit
student@student-vm:~/nso-run/packages$
```

Step 4

Open the *vlan-template.xml* in Visual Studio Code by using the **code vlan-template.xml** command.

```
student@student-vm:~/nso-run/packages$ cd $HOME/nso-run/packages/vlan/templates
student@student-vm:~/nso-run/packages/vlan/templates$ code vlan-template.xml
```

Step 5

In the IOS portion of the template, change the *{/vlan-id}* variable to *{vlan-id}* (remove the preceding "/"). Additionally, change the closing tag from *</GigabitEthernet>* to *</FastEthernet>*.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>
```

```
        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{vlan-id}</id>
          </vlan-list>
        </vlan>
        <?foreach {interface}?>
          <interface xmlns="urn:ios">
            <?if {intf-type='FastEthernet'}?>
              <FastEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{vlan-id}</vlan>
                  </access>
                </switchport>
              </FastEthernet>
            <?end?>
            <?if {intf-type='GigabitEthernet'}?>
              <GigabitEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{vlan-id}</vlan>
                  </access>
                </switchport>
              </FastEthernet>
            <?end?>

          </interface>
        <?end?>

        <!-- NX-OS -->

        ...

      </config>
    </device>
  </devices>
</config-template>
```

Step 6

Save the file.

Step 7

Switch back to terminal application, connect to NSO CLI and reload the packages.

```
student@student-vm:~/nso-run/packages/vlan/templates$ ncs_cli -Cu admin
admin@ncs packages reload
reload-result {
    package cisco-ios-cli-6.85
    result true
}
reload-result {
    package cisco-iosxr-cli-7.41
    result true
```

```
}
reload-result {
    package cisco-nx-cli-5.23
    result true
}
reload-result {
    package loopback
    result true
}
reload-result {
    package vlan
    result false
    info vlan-template.xml:40: The end tag 'FastEthernet' does not match the start tag
'GigabitEthernet'.
}
admin@ncs
System message at 2022-09-17 17:06:24...
    Subsystem stopped: ncs-dp-8-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 17:06:24...
    Subsystem stopped: ncs-dp-7-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 17:06:24...
    Subsystem started: ncs-dp-9-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 17:06:24...
    Subsystem started: ncs-dp-10-cisco-nx-cli-5.23:NexusDp
admin@ncs *** ALARM package-load-failure: vlan-template.xml:40: The end tag 'FastEthernet'
does not match the start tag 'GigabitEthernet'.
admin@ncs
```

Step 8

Switch back to the *vlan-template.xml* and fix the reported error so that the end tag matches the start tag.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>

        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{vlan-id}</id>
          </vlan-list>
        </vlan>
        <?foreach {interface}?>
          <interface xmlns="urn:ios">
            <?if {intf-type='FastEthernet'}?>
              <FastEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{vlan-id}</vlan>
                  </access>
                </switchport>
              </FastEthernet>
            <?end?>
            <?if {intf-type='GigabitEthernet'}?>
```

```
              <GigabitEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{vlan-id}</vlan>
                  </access>
                </switchport>
              </GigabitEthernet>
            <?end?>

          </interface>
        <?end?>

        <!-- NX-OS -->

        ...

      </config>
    </device>
  </devices>
</config-template>
```

Step 9

Save the file.

Step 10

Switch to NSO CLI and reload the packages.

```
admin@ncs packages reload
reload-result {
    package cisco-ios-cli-6.85
    result true
}
reload-result {
    package cisco-iosxr-cli-7.41
    result true
}
reload-result {
    package cisco-nx-cli-5.23
    result true
}
reload-result {
    package loopback
    result true
}
reload-result {
    package vlan
    result true
}
admin@ncs
System message at 2022-09-17 17:17:11...
    Subsystem started: ncs-dp-1-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 17:17:11...
    Subsystem started: ncs-dp-2-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 17:17:11...
    Subsystem stopped: ncs-dp-2-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 17:17:11...
```

```
    Subsystem stopped: ncs-dp-1-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 17:17:11...
    Subsystem started: ncs-dp-4-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 17:17:11...
    Subsystem started: ncs-dp-3-cisco-ios-cli-6.85:IOSDp
```

Step 11

View the output. Even though one of the variables in the template does not match the leaf node in the YANG file, we are not informed about that just yet.

Step 12

Start configuring the service. After entering the interface, use the command **commit dry-run** command to check the output.

```
admin@ncs config
Entering configuration mode terminal
admin@ncs(config) services vlan Internal
admin@ncs(config-vlan-Internal) vlan-id 102
admin@ncs(config-vlan-Internal) device SW31
admin@ncs(config-device-SW31) interface FastEthernet 0/0
admin@ncs(config) top
admin@ncs(config-interface-FastEthernet/0/0) commit dry-run
cli {
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          interface {
                              FastEthernet 0/0 {
             +                    switchport {
             +                        mode {
             +                            access {
             +                            }
             +                        }
             +                    }
                              }
                          }
                      }
                  }
              }
              services {
             +    vlan Internal {
             +        vlan-id 102;
             +        device SW31 {
             +            interface FastEthernet 0/0;
             +        }
             +    }
              }
    }
}
```

Notice that you do not have any errors.

Step 13

Compare the output to the previous one.

```
cli {
```

```
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          vlan {
              +               vlan-list 102 {
              +               }
                          }
                          interface {
                              FastEthernet 0/0 {
              +                   switchport {
              +                       mode {
              +                           access {
              +                           }
              +                       }
              +                       access {
              +                           vlan 102;
              +                       }
              +                   }
                              }
                          }
                      }
                  }

                  ...
              }
    }
}
```

If you look at the output more closely, you will notice that VLAN configuration items are missing due to XPath variable misconfiguration. To troubleshoot this further you need to debug the template.

Step 14

Use the **commit dry-run** command with additional instruction **debug xpath** to debug XPath expressions.

Examine the output. Because you deleted the "/" character in front of the **vlan-id** variable in the template, the mapping logic is no longer looking for the **vlan-id** node on the root level of the data tree in the YANG file.

```
admin@ncs(config) commit dry-run | debug xpath
 2022-09-17T17:28:38.177 xpath: evaluating: /services/vlan:vlan[name='Internal']/
device[name='SW31']/name: /ncs:devices/ncs:device/ncs:name
 2022-09-17T17:28:38.178 xpath: exists(/devices/device[name='SW31']) = true
 2022-09-17T17:28:38.178 xpath: result: /services/vlan:vlan[name='Internal']/
device[name='SW31']/name: true (0.000 s)
 2022-09-17T17:28:38.187 xpath: template vlan-template: evaluating: /device/name
 2022-09-17T17:28:38.187 xpath: template vlan-template: match: /services/
vlan:vlan[name='Internal']/device[name='SW31']/name
 2022-09-17T17:28:38.187 xpath: template vlan-template: result: one node (0.000 s)
 2022-09-17T17:28:38.192 xpath: template vlan-template: evaluating: vlan-id
 2022-09-17T17:28:38.192 xpath: template vlan-template: result: no nodes (0.000 s)
 2022-09-17T17:28:38.192 xpath: template vlan-template: evaluating: interface
 2022-09-17T17:28:38.192 xpath: template vlan-template: match: /services/
vlan:vlan[name='Internal']/device[name='SW31']/interface[intf-type='FastEthernet'][intf-
id='0/0']
 2022-09-17T17:28:38.192 xpath: template vlan-template: result: one node (0.000 s)
 2022-09-17T17:28:38.192 xpath: template vlan-template: evaluating: string(boolean(intf-
type='FastEthernet'))
 2022-09-17T17:28:38.193 xpath: template vlan-template: result: true (0.000 s)
 2022-09-17T17:28:38.193 xpath: template vlan-template: evaluating: intf-id
 2022-09-17T17:28:38.193 xpath: template vlan-template: match: /services/
vlan:vlan[name='Internal']/device[name='SW31']/interface[intf-type='FastEthernet'][intf-
id='0/0']/intf-id
 2022-09-17T17:28:38.193 xpath: template vlan-template: result: one node (0.000 s)
```

```
 2022-09-17T17:28:38.208 xpath: template vlan-template: evaluating: vlan-id
 2022-09-17T17:28:38.208 xpath: template vlan-template: result: no nodes (0.000 s)
 2022-09-17T17:28:38.208 xpath: template vlan-template: evaluating: string(boolean(intf-
type='GigabitEthernet'))
...
cli {
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          interface {
                              FastEthernet 0/0 {
             +                    switchport {
             +                        mode {
             +                            access {
             +                            }
             +                        }
             +                    }
                              }
                          }
                      }
                  }
              }
              services {
             +    vlan Internal {
             +        vlan-id 102;
             +        device SW31 {
             +            interface FastEthernet 0/0;
             +        }
             +    }
              }
    }
}
admin@ncs(config)
```

Step 15

Fix the error in the template.

```
<config-template xmlns="http://tail-f.com/ns/config/1.0" servicepoint="vlan">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <!-- DEVICE -->
    <device>
      <name>{/device/name}</name>
      <config>

        <!-- IOS -->
        <vlan xmlns="urn:ios">
          <vlan-list>
            <id>{/vlan-id}</id>
          </vlan-list>
        </vlan>
        <?foreach {interface}?>
          <interface xmlns="urn:ios">
            <?if {intf-type='FastEthernet'}?>
              <FastEthernet>
                <name>{intf-id}</name>
                <switchport>
                  <mode>
                    <access/>
                  </mode>
                  <access>
                    <vlan>{/vlan-id}</vlan>
                  </access>
```

```
              </switchport>
            </FastEthernet>
          <?end?>
          <?if {intf-type='GigabitEthernet'}?>
            <GigabitEthernet>
              <name>{intf-id}</name>
              <switchport>
                <mode>
                  <access/>
                </mode>
                <access>
                  <vlan>{/vlan-id}</vlan>
                </access>
              </switchport>
            </GigabitEthernet>
          <?end?>

        </interface>
      <?end?>

      <!-- NX-OS -->

      ...

    </config>
  </device>
 </devices>
</config-template>
```

Because you deleted the "/" character in front of the vlan-id variable in the template, the mapping logic is no longer looking for the **vlan-id** node on the root level of the data tree in the YANG file.

Step 16

Exit the configuration mode and reload the packages. Make sure the vlan package reloads successfully.

```
admin@ncs(config) abort
admin@ncs packages reload
reload-result {
    package cisco-ios-cli-6.85
    result true
}
reload-result {
    package cisco-iosxr-cli-7.41
    result true
}
reload-result {
    package cisco-nx-cli-5.23
    result true
}
reload-result {
    package loopback
    result true
}
reload-result {
    package vlan
    result true
}
admin@ncs
System message at 2022-09-17 17:36:12...
    Subsystem stopped: ncs-dp-4-cisco-nx-cli-5.23:NexusDp
admin@ncs
System message at 2022-09-17 17:36:12...
    Subsystem stopped: ncs-dp-3-cisco-ios-cli-6.85:IOSDp
```

```
admin@ncs
System message at 2022-09-17 17:36:12...
    Subsystem started: ncs-dp-5-cisco-ios-cli-6.85:IOSDp
admin@ncs
System message at 2022-09-17 17:36:12...
    Subsystem started: ncs-dp-6-cisco-nx-cli-5.23:NexusDp
admin@ncs
```

Step 17

Provision the service instance again.

```
admin@ncs config
Entering configuration mode terminal
admin@ncs(config) services vlan Internal
admin@ncs(config-vlan-Internal) vlan-id 102
admin@ncs(config-vlan-Internal) device SW31
admin@ncs(config-device-SW31) interface FastEthernet 0/0
admin@ncs(config-interface-FastEthernet/0/0) exit
admin@ncs(config-device-SW31) exit
admin@ncs(config-vlan-Internal) device SW32
admin@ncs(config-device-SW32) interface Ethernet 1/1
admin@ncs(config-interface-Ethernet/1/1) top
```

Step 18

When you have entered all the service parameters, preview the device changes by using the **commit dry-run** or **commit dry-run** outformat native command.

```
admin@ncs(config) commit dry-run
cli {
    local-node {
        data  devices {
                  device SW31 {
                      config {
                          vlan {
            +                 vlan-list 102 {
            +                 }
                          }
                          interface {
                              FastEthernet 0/0 {
            +                     switchport {
            +                         mode {
            +                             access {
            +                             }
            +                         }
            +                         access {
            +                             vlan 102;
            +                         }
            +                     }
                              }
                          }
                      }
                  }
                  device SW32 {
                      config {
                          vlan {
            +                 vlan-list 102 {
            +                 }
                          }
                          interface {
                              Ethernet 1/1 {
```

```
                                    switchport {
              -                         mode trunk;
              +                         mode access;
                                        access {
              +                             vlan 102;
                                        }
                                    }
                                }
                            }
                        }
                    }
                 }
                 services {
              +      vlan Internal {
              +          vlan-id 102;
              +          device SW31 {
              +              interface FastEthernet 0/0;
              +          }
              +          device SW32 {
              +              interface Ethernet 1/1;
              +          }
              +      }
                 }
    }
}
```

Step 19

Commit the service instance creation by using the **commit** command and exit the configuration mode.

```
admin@ncs(config) commit
Commit complete.
admin@ncs(config) exit
admin@ncs
```

Step 20

Verify that the service has been successfully deployed.

```
admin@ncs show running-config services vlan
services vlan Internal
 vlan-id 102
 device SW31
  interface FastEthernet 0/0
  !
 !
 device SW32
  interface Ethernet 1/1
  !
 !
!
admin@ncs show running-config services vlan | display xpath
/services/vlan:vlan[name='Internal']/vlan-id 102
/services/vlan:vlan[name='Internal']/device[name='SW31']/interface[intf-type='FastEthernet']
[intf-id='0/0']
/services/vlan:vlan[name='Internal']/device[name='SW32']/interface[intf-type='Ethernet'][intf-
id='1/1']
admin@ncs show running-config services vlan | tab
          VLAN                      INTF
NAME      ID    NAME  INTF TYPE     ID
-----------------------------------------
Internal  102   SW31  FastEthernet  0/0
```

```
                SW32  Ethernet      1/1
```

Activity Verification

You have completed this task when you attain this result:

- The service has been successfully deployed.