Cisco NSO T-SDN Function Pack Bundle

Installation Guide

Version 6.0.0

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 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

Abstract

This **Cisco Network Service Orchestrator Transport-SDN Function Pack Bundle Installation Guide** includes information to help you to install Cisco NSO Transport SDN Function Pack (T-SDN FP) Bundle.

Audience

This document is intended for Cisco Advanced Services developers, network engineers, and system engineers to install the T-SDN automation functionalities to Cisco customers.

Additional Documentation

This documentation requires the reader to have a good understanding of NSO and its usage as described in the NSO documentation.

Sl. No.	Documentation
1.	Cisco Transport SDN Function Pack Bundle User Guide
2.	NSO Installation Guide
3.	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.

Installation Methods

You can perform T-SDN FP Bundle installation on NSO in two ways:

- System Installation
- Local Installation

The system installation is for a real time production environment and is the preferred installation method.

The local installation is the demo version of the installation.

You must have **sudo** user privileges to perform the installation and run the installation commands. You can perform T-SDN FP Bundle system installation on a single machine or multiple machines at a given time. System installation is used to install NSO on multiple hosts/VMs from a Single Controller host.

This documentation describes how to perform the T-SDN FP Bundle system installation. For information on local installation, contact your Cisco representative.

Installation – Overview

The NSO T-SDN FP Bundle system installation allows you to install the Core Function Packs and the Example Function Packs.

The Core Function Packs are productized and supported implementation of Cisco SR-ODN, Cisco SR-Policy, Circuit-style policy services, IETF-L2VPN-NM, and IETF-L3VPN-NM services.

The services in the Example Function Packs are customized for specific requirements.

For a detailed overview of this product, see the ***Cisco NSO T-SDN FP Bundle User Guide***.

Perform the NSO T-SDN FP Bundle system installation by using the Layered Service Architecture (LSA) deployment model or on a single NSO instance. This documentation discusses each installation method in detail.

For information on how to install T-SDN FP Bundle on a single NSO instance, see **Installing and Uninstalling Cisco NSO T-SDN Function Pack Bundle on a Single NSO Instance**.

For information on how to install T-SDN FP Bundle in the LSA Model, see **Installing and Uninstalling T-SDN FP Bundle in the LSA Model.**

Install the Example Function Packs (flavors) on top of the base flavor SR-TE CFP-IOSXR CLI. To install a flavor, copy the required packages for the flavor either during or after the SR-TE CFP-IOSXR CLI installation. For more information on how to install different flavors, see **Installing Example Function Packs on a Single NSO Instance**.

Installation Requirements

This topic discusses the system requirements, NSO versions, and the NED versions required to install T-SDN FP Bundle.

System Requirements

The following table outlines the hardware requirements, software requirements, and platform dependencies to successfully install Cisco NSO T-SDN FP Bundle v6.0.0 on NSO v6.1.4.

Item	Requirement
Operating systems	NSO and T-SDN FP Bundle are available for all Linux distributions and supported on the following operating systems: Ubuntu 22.04 LTS or later Rocky Linux v8.6 or later RedHat Linux v8.7 or later
Software	Open JDK v17 or higher Python v3.8 or higher
ulimit value for NSO	64000 (minimum)

Cisco NSO and Cisco NED Requirements

Software/Driver	Versions
Cisco NSO	6.1.4
IOSXR CLI NED (default)	7.46, 7.52
IOSXE CLI NED (for Multi-Vendor Extension Example)	6.86, 6.100

The IOSXR CLI NED is the default NED. The IOSXR CLI NED and the IOSXE CLI NED are packaged with the installation tar file.

Note: T-SDN FP Bundle v6.0.0 supports two versions of IOSXR CLI and IOSXE CLI NEDs. The IOSXR CLI v7.46 and IOSXE CLI v6.86 NEDs support only T-SDN FP Bundle v5.0.0 features on T-SDN FP v6.0.0. You can choose to upgrade these NEDs to IOSXR CLI v7.52 and IOSXE CLI v6.100, as required, to avail the latest features. It is recommended to use the latest NED versions while onboarding new devices to NSO.

The following IOSXR Netconf NEDs are downloadable from the Cisco website.

IOSXR Netconf NED (for Multi-Vendor Extension Example)	7.3.2, 7.4.2. 7.5.2, 7.6.2, 7.7.2, 7.8.2, 7.9.2, 7.10.1, 7.11.1

Preparing the NSO Environment to Install the Cisco T-SDN FP Bundle

To install T-SDN FP Bundle on NSO, you must first prepare the NSO environment.
The information in this section is applicable to install Cisco T-SDN FP Bundle on a single NSO instance and in the LSA model.

To prepare the NSO environment:

1. Obtain the NSO 6.1.4 installation bin file and follow the steps described in the ***NSO Installation Guide – System Installation Guide*** to install NSO.

2. Verify NSO version.

   ```
   $ ncs --version
   6.1.4
   ```

3. Make sure the version for Python and Python 3 installed is 3.8 or higher. See Python documentation for more information about how to install Python.

   ```
   $ python --version
   3.8
   ```

   The default Python should point to Python 3. If you cannot change the default Python to Python 3, change the Python startup command configuration. For more information, see **Appendix A: Changing Python Startup Command Configuration**.

4. Verify OpenJDK 11 or higher is installed.

   ```
   $ Java --version
   openjdk 17
   ```

5. Set the overcommit memory settings to the default value 0.

   ```
   $ cat /proc/sys/vm/overcommit_memory
   0
   ```

6. Make sure to have **sudo** user privileges to perform the CFP installation. This user must also be part of the **ncsadmin** group.

7. (*LSA installation only*) Configure Network Time Protocol (NTP).

8. Add the ulimit level value for NSO and the operating system in **/etc/init.d/ncs** as follows:

   ```
   ...
   ncsdir=/opt/ncs/current
   confdir=/etc/ncs
   rundir=/var/opt/ncs
   logdir=/var/log/ncs

   ncs=${ncsdir}/bin/ncs
   ulimit -n 65535
   prog=ncs
   conf="-c ${confdir}/ncs.conf"
   heart="--heart"
   ...
   ```

9. Add the ulimit value for the operating system. The following is an example.

    a. Edit the **/etc/security/limits.conf** file and add the following lines:

    ```
    * soft nproc 65535
    * hard nproc 65535
    * soft nofile 65535
    * hard nofile 65535
    * hard memlock 65535
    * soft memlock 65535
    ```

    b. Run the **sysctl -p** script to set the parameters.

    c. Log out of the system and log in again to apply the new values.

10. Verify the ulimit values are applied.

```
[admin@cnso-60-52 ~]$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 95697
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files                      (-n) 65535
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 4096
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

11. Set the timeout value for the ncs services, if required. The default timeout value for the ncs services is 300 seconds.

```
sudo mkdir /etc/systemd/system/ncs.service.d
echo -e "[Service]\nTimeoutStartSec=<timeout_in_seconds>" | sudo tee
/etc/systemd/system/ncs.service.d/startup-timeout.conf
sudo systemctl daemon-reload
```

12. Verify the timeout value is set.

```
sudo systemctl show ncs | grep ^Timeout
```

Installing and Uninstalling Cisco NSO T-SDN Function Pack Bundle on a Single NSO Instance

This topic discusses the packages required to install the NSO-T-SDN FP Bundle on a single NSO instance and the information to prepare the NSO environment for such an installation.

Installing on a Single NSO Instance

This section discusses the packages required and the installation procedure to install T-SDN FP Bundle and the Example function packs on a single NSO instance.

Package Categories and Packages on a Single NSO Instance

This section discusses the required package categories and the associated packages. The following tables shows the packages in the Core Function Pack category and the Example Packages category.

IOSXR CLI NED is the default NED and is packaged with the installation tar file. IOSXR Netconf NEDs are downloadable from the Cisco website.

Core Function Pack packages are required to install SR-TE CFP. Example packages comprises the packages for the Example function packs you choose to install and can be extracted either during or after SR-TE CFP-IOSXR CLI installation.

Note: The Circuit-Style-Segment Routing-Traffic Engineering-Core Function Pack (**cs-sr-te-cfp)** package in SR-TE CFP-IOSXR CLI is supported only on IOSXR CLI 7.46 NED, IOSXR CLI 7.52, IOSXR NC 7.8 (or later) NED.

Core Function Pack	
Package Category	**Packages**
T-SDN FP Bundle packages	ncs-6.1.4-cisco-sr-te-cfp-6.0.0.tar.gz ncs-6.1.4-cisco-sr-te-cfp-internal-6.0.0.tar.gz ncs-6.1.4-cisco-cs-sr-te-cfp-6.0.0.tar.gz
T-SDN FP Bundle L2NM Packages	ncs-6.1.4-cisco-L2vpn-fp-internal-6.0.0.tar.gz ncs-6.1.4-ietf-l2vpn-nm-6.0.0.tar.gz
T-SDN FP Bundle L3NM Packages	ncs-6.1.4-cisco-L3vpn-fp-internal-6.0.0.tar.gz ncs-6.1.4-ietf-l3vpn-nm-6.0.0.tar.gz
T-SDN FP Bundle common packages	ncs-6.1-core-fp-plan-notif-generator-1.0.10.tar.gz ncs-6.1-custom-template-utils-2.0.13.tar.gz ncs-6.1-lsa-utils-1.0.4.tar.gz ncs-6.1-core-fp-common-1.33.0.tar.gz ncs-6.1.4-cisco-tsdn-core-fp-common-6.0.0.tar.gz ncs-6.1.3-resource-manager-4.2.1.tar.gz ncs-6.1-core-fp-delete-tag-service-1.0.6.tar.gz ncs-6.1.3-cisco-cfp-jwt-auth-1.0.0.tar.gz (RBAC PAM

	package) ncs-6.1.4-cw-device-auth-6.0.0.9.tar.gz
T-SDN FP Bundle Multi-Vendor for IOSXE	ncs-6.1.4-sr-te-multi-vendors-6.0.0.tar.gz ncs-6.1.4-l2vpn-multi-vendors-6.0.0.tar.gz ncs-6.1.4-l3vpn-multi-vendors-6.0.0.tar.gz ncs-6.1.4-rsvp-te-multi-vendors-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-pm-multi-vendors-EXAMPLE-6.0.0.tar.gz
CNC user specific packages	ncs-6.1.4-dlm-svc-6.0.0-22.tar.gz nso-6.1.4-cw-device-auth-6.0.0-9.tar.gz ncs-6.1.4-cisco-tm-tc-fp-6.0.0-9.tar.gz
NEDs	**IOSXR CLI NED:** ncs-6.1.4-cisco-iosxr-7.52.2.tar.gz ncs-6.0.3-cisco-iosxr-7.46.3.tar.gz **IOSXE CLI NED:** ncs-6.1-cisco-ios-6.86.6.tar.gz ncs-6.1-cisco-ios-6.100.tar.gz

Example Packages	
Package Category	**Package Name**
Automated Assurance Example package for L2NM,L3NM	ncs-6.1.4-cisco-aa-service-assurance-6.0.0.tar.gz
IETF Network Slice Service Example package	ncs-6.1.4-ietf-network-slice-service-EXAMPLE-6.0.0.tar.gz
IETF-TE Example package with IOSXR CLI NED	ncs-6.1.4-ietf-te-fp-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-cisco-rsvp-te-fp-EXAMPLE-6.0.0.tar.gz
Performance Measurement Example package	ncs-6.1.4-cisco-pm-fp-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-cisco-pm-fp-internal-EXAMPLE-6.0.0.tar.gz

Editing the NCS Configuration File on a Single NSO Instance

Edit the NCS configuration file to add or update configurations for Cisco NSO T-SDN Function Pack Bundle. For more information about the ncs.conf file, see the **nso_man-<*version*>.pdf** documentation in **volume5**.

Edit the /etc/ncs/ncs.conf file as follows:

1. Back up the current **ncs.conf** file before editing the file.
2. Add **service-state-changes** under **<stream>** to generate notifications for any changes in the service state.

```
<notifications>
```

```
<event-streams>
    <stream>
        <name>service-state-changes</name>
        <description>Plan state transitions according to
  tailf-ncs-plan.yang</description>
        <replay-support>true</replay-support>
        <builtin-replay-store>
            <enabled>true</enabled>
            <dir>${NCS_RUN_DIR}/state</dir>
            <max-size>S10M</max-size>
            <max-files>50</max-files>
        </builtin-replay-store>
    </stream>
```

3. If you choose to install Automated Assurance (AA), add AA notifications under **<notifications>** -> **<event streams>** to notify any AA configuration changes.

```
<stream>
  <name>service-aa-changes</name>
  <description>Notifications relating to the service aa configuration
change</description>
  <replay-support>true</replay-support>
  <builtin-replay-store>
    <enabled>true</enabled>
    <dir>${NCS_RUN_DIR}/state</dir>
    <max-size>S10M</max-size>
    <max-files>50</max-files>
  </builtin-replay-store>
</stream>
```

4. Append the **<hide-group>** information as follows.

```
<hide-group>
    <name>debug</name>
</hide-group>
<hide-group>
    <name>tsdn</name>
</hide-group>
<hide-group>
    <name>fastmap-private</name>
</hide-group>
```

5. The SSH port configuration is optional for CLI, webui, and netconf northbound parameters. You can choose to either enable or disable the SSH port configuration, as required, for these

parameters.
By default, the SSH port configuration for these parameters is disabled. For more information on these parameters, see the ***NSO documentation***.

The following shows how to enable the SSH port configuration, if required. Provide the port numbers as per your requirement.

SSH port for CLI

```
<cli>
  <enabled>true</enabled>
  <!-- Use the builtin SSH server -->
  <ssh>
         <enabled>true</enabled>
         <ip>0.0.0.0</ip>
          <port>${North_Bound_CLI_SSH_Port}</port>
          <extra-listen>
         <ip>::</ip>
         <port>${North_Bound_CLI_SSH_Port}</port>
       </extra-listen>
  </ssh>
```

webui

Enable webui in either TCP or SSL.

```
<webui>
<enabled>true</enabled>
  <transport>
    <tcp>
       <enabled>true</enabled>
        <ip>0.0.0.0</ip>
        <port>${North_Bound_Web_UI_Port}</port>
    </tcp>

       <ssl>
           <enabled>true</enabled>
           <ip>0.0.0.0</ip>
           <port>${SSL_port}</port>
           <key-file>${NCS_CONFIG_DIR}/ssl/cert/host.key</key-file>
           <cert-file>${NCS_CONFIG_DIR}/ssl/cert/host.cert</cert-file>
       </ssl>
     </transport>
```

netconf northbound

```
<netconf-north-bound>
```

```
        <enabled>true</enabled>
         <transport>
    <ssh>
    <enabled>true</enabled>
    <ip>0.0.0.0</ip>
    <port>${Netconf_North_Bound_port}</port>
     </ssh>
```

6. (*RBAC PAM installation only*) Set **<package-authentication>** to true under **<aaa>** to use the RBAC PAM feature.

```
    <aaa>
       <package-authentication>
         <enabled>true</enabled>
         <packages>
           <package>cisco-cfp-jwt-auth</package>
         </packages>
       </package-authentication>
     </aaa>
```

7. Add/update start-timeout in **ncs.conf** under **<python-vm>**.

```
<python-vm>
  <start-timeout>PT300S</start-timeout>
</python-vm>
```

Installing Core Function Packs on a Single NSO Instance

To install Cisco NSO T-SDN FP Bundle, you must install SR-TE CFP-IOSXR CLI. The SR-TE CFP-IOSXR CLI is the primary CFP or the main component in T-SDN FP Bundle and includes the **cs-sr-te-cfp** package. This is the base flavor on which the other CFPs (such as SR-TE CFP-IOSXR NC and SR-TE CFP IETF-TE) and Example Function Packs are installed.

Note: The **cs-sr-te-cfp** package in SR-TE CFP-IOSXR CLI is supported only on IOSXR CLI 7.46/7.52 NED and IOSXR NC 7.8 (and later) NED.

This topic discusses the procedure to install the Core Function Packs. For information on how to install the Example Function Packs, see **Installing Example Function Packs on a Single NSO Instance.**

For information on how to install the Core Function Packs using Netconf NED, see **Appendix C: Using Netconf NED in T-SDN FP Bundle**.

Installing SR-TE CFP

Use the information in this section to install SR-TE CFP with IOSXR CLI NED and IOSXE CLI NED. SR-TE CFP with IOSXR CLI NED installation is the base for all other installation flavors. You must, therefore, first install SR-TE CFP-IOSXR CLI and then perform SR-TE CFP IOSXE CLI installation.

Note: **cs-sr-te-cfp** is not supported on IOSXE CLI.

Do the following to install SR-TE CFP:

Steps 1 to 8 describe the SR-TE CFP - IOSXR CLI installation procedure. If it is already installed, skip to step 9 for the SR-TE CFP-IOSXE CLI installation procedure.

1. Make sure to have performed the tasks described in the following sections:
   - **Preparing the NSO Environment to Install the Cisco T-SDN FP Bundle**
   - **Editing the NCS Configuration File on a Single NSO Instance**
2. Log in to the host machine as the **sudo** user, who is also part of the **ncsadmin** user group.
3. Obtain and download the signed bin file **tsdn-<*version*>-nso-<*version*>.signed.bin** package from Cisco website and copy it to the host server.

   For example, obtain and download the **tsdn-6.0.0-nso-6.1.4.signed.bin** package from Cisco website and copy it to the host server.
4. Extract the content of the bin file to the current directory.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin
   ```

   This verifies the authenticity of the product. If you encounter any network connectivity issues, run the following command to skip this verification.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin --skip-verification
   ```
5. Untar the T-SDN FP Bundle installer **tar.gz** file to the current directory. If the folder already exists, make sure to create a backup of the existing folder.

   ```
   $ tar -xvf tsdn-<version>-nso-<version>.tar.gz
   ```
6. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```
7. Copy and link the following packages to install SR-TE CFP-IOSXR CLI. The other packages are for the different installation flavors.

   ```
   sudo cp ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz
   /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz

   sudo cp ncs-<version>-cisco-sr-te-cfp-internal-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-
   internal-<version>.tar.gz

   sudo cp ncs-<version>-cisco-cs-sr-te-cfp-<version>.tar.gz /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
   <version>.tar.gz var/opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
   <version>.tar.gz
   ```

```
sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz

sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-
common-<version>.tar.gz

sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz

sudo cp ncs-<version>-cisco-iosxr-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz

sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-
generator-<version>.tar.gz

sudo cp ncs-<version>-resource-manager-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz

sudo cp ncs-<version>-lsa-utils-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz

sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-lsa-utils-
<version>.tar.gz

sudo cp ncs-<version>-cisco-cfp-jwt-auth-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-cfp-jwt-auth-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-cfp-jwt-auth-
<version>.tar.gz
```

Note: To install one or more flavors, copy and link the required packages for the flavor. For more information on the flavors and the required packages, see **Package Categories and Packages on a Single NSO Instance**.

8. (*Optional*) If the ncs services fail after the default timeout of 300 seconds, change the default timeout for ncs services.

```
sudo mkdir /etc/systemd/system/ncs.service.d
```

```
echo -e "[Service]\nTimeoutStartSec=<timeout_in_seconds>" | sudo tee
/etc/systemd/system/ncs.service.d/startup-timeout.conf
```

```
sudo systemctl daemon-reload
```

Verify the new timeout is applied.

```
sudo systemctl show ncs | grep ^Timeout
```

9. To install SR-TE CFP-IOSXE CLI, continue with step 10 , else skip to step 13 to complete the SR-TE CFP IOSXR CLI installation.

Note: Be sure to have completed steps 1-8 (SR-TE CFP-IOSXR CLI installation) before performing SR-TE CFP-IOSXE CLI installation.

10. Obtain and load the required IOSXE CLI NEDs into NCS. For more information, see **Cisco NSO and Cisco NED Requirements**.

11. Go to the packages directory.

```
$ cd tsdn-<version>-nso-<version>/standalone/packages
```

12. Copy and link the following packages to install SR-TE CFP-IOSXE CLI.

```
sudo cp ncs-<version>-sr-te-multi-vendors-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

13. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
                     [  OK  ]
```

14. Verify the installation and make sure the packages are up and running, and then perform the post installation tasks.

```
admin@ncs> show packages package package-version | select build-info ncs
version | select build-info file | select build-info package sha1 | select
oper-status error-info | select oper-status up | tab
```

Performing Post Installation Tasks for SR-TE CFP

Do the following:

To perform the post installation tasks for SR-TE CFP – IOSXR CLI and SR-TE CFP – IOSXE CLI, change the current directory as follows:

```
$ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data

$ ncs_cli -u admin

admin@ncs> configure

unhide debug
```

1. Perform the following post installation tasks for SR-TE CFP – IOSXR CLI installation. If you have already installed and performed the post installation tasks for SR-TE CFP-IOSXR CLI CFP, skip to step 2 for the post installation tasks of SR-TE CFP – IOSXE CLI installation.

   a. Load-merge the following plan notification files to activate notifications.

   ```
   admin@ncs% load merge SR-plan-notification-settings.xml
   admin@ncs% load merge CS-SR-plan-notification-settings.xml

   admin@ncs% commit
   ```

   b. Load-merge the following files to activate status-codes.

   ```
   $ ncs_cli -u admin

   admin@ncs> configure

   admin@ncs% unhide debug

   admin@ncs% load merge SR-status-codes.xml
   admin@ncs% load merge CS-SR-status-codes.xml

   admin@ncs% commit
   ```

   c. Load-merge the following files to activate kickers.

   ```
   $ ncs_cli -u admin

   admin@ncs> configure

   admin@ncs% unhide debug

   admin@ncs% load merge SR-internal-plan-kicker.xml

   admin@ncs% load merge 1_SR-cfp-configuration-kicker.xml
   admin@ncs% load merge CS-SR-internal-plan-kicker.xml

   admin@ncs% commit
   ```

   d. Configure resource-pools.

   ```
   load merge xr-bidirectional-association-id-resource-pool.xml

   load merge xr-color-resource-pool.xml

   load merge xr-disjoint-group-id-resource-pool.xml

   commit
   ```

   e. Configure the following common bootstrap data.

      - Commit queue settings

      ```
      admin@ncs% load merge commit-queue-settings.xml

      admin@ncs% commit
      ```

      - Sync dispatch-map.

      ```
      admin@ncs% unhide debug
      ```

```
load merge dispatch-map-settings.xml
commit request devices lsa dispatch-map sync
show devices lsa dispatch-map
```

- Set NACM rules for the user.

  The user must be a **sudo** user, who is also part of the **ncsadmin** group.

  ```
  admin@ncs% set nacm groups group ncsadmin user-name [ <user-name>
  private ]
  admin@ncs% commit
  ```

- Configure ssh-rsa algorithms public key. You can configure this either at the global level or for a specific device that will be onboarded.

  Setting the algorithm globally

  ```
  % show devices global-settings ssh-algorithms public-key
  public-key [ ssh-ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384
  ecdsa-sha2-nistp521 rsa-sha2-512 rsa-sha2-256 ];

  % set devices global-settings ssh-algorithms public-key [ ssh-
  ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521
  rsa-sha2-512 rsa-sha2-256 ssh-rsa ]
  % commit

  % show device global-settings ssh-algorithms public-key
  public-key [ ssh-ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384
  ecdsa-sha2-nistp521 rsa-sha2-512 rsa-sha2-256 ssh-rsa ];
  ```

  Setting the algorithm for a specific device

  ```
  % set devices device <DEVICE_NAME> ssh-algorithms public-key [ ssh-
  rsa ]
  % commit

  % show device device <DEVICE_NAME> ssh-algorithms public-key
  public-key [ ssh-rsa ];
  ```

- If you installed RBAC PAM, configure the bootstrap information for RBAC PAM as follows:

  ```
  set jwt-auth cnc-host <ip-address/FQDN>
  set jwt-auth port 30603
  set jwt-auth pem-key-path /home/cisco/cert.pem
  commit
  ```

  where cnc-host is the IP address of the Authentication Provider, port is the port number used for the Authentication Provider, and pem-key-path is the file path to the certificate file key used for authentication.

2. (*SR-TE CFP - IOSXE CLI only*) Load-merge the **2_SR-multi-vendor-iosxe-cli.xml** file to configure dynamic-mapping.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% load merge 2_SR-multi-vendor-iosxe-cli.xml
     admin@ncs% commit
```

Verifying the Post Installation Tasks for SR-TE CFP

Verify the post installation tasks you performed are correct.

Do the following:

1. Verify SR-TE CFP-IOSXR CLI installation as follows. If you have already verified SR-TE CFP-IOSXR CLI installation, skip to step 2 to verify SR-TE CFP-IOSXE CLI installation.

   a. Verify the kickers configuration.

```
admin@ncs% unhide debug
admin@ncs% show kickers
data-kicker sr-te-cfp-configuration-kicker {
    monitor     /cisco-sr-te-cfp:cfp-configurations;
    kick-node   /cisco-sr-te-cfp:sr-te;
    action-name update-internal-cfp-configurations;
}
data-kicker sr-te-odn-internal-plan-kicker {
    monitor     /cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-odn-
internal:odn/cisco-sr-te-cfp-sr-odn-internal:odn-template-plan;
    kick-node   /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-odn:odn/cisco-
sr-te-cfp-sr-odn:actions;
    action-name internal-plan-change-handler;
}
data-kicker sr-te-policy-internal-plan-kicker {
    monitor     /cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-
policies-internal:policies/cisco-sr-te-cfp-sr-policies-internal:policy-
plan;
    kick-node   /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
policies:policies/cisco-sr-te-cfp-sr-policies:actions;
    action-name internal-plan-change-handler;
}
data-kicker cs-sr-te-internal-plan-kicker {
    monitor     /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
policies:policies/policy-plan;
    kick-node   /cisco-cs-sr-te-cfp:cs-sr-te-actions;
    action-name internal-plan-change-handler;
```

b. Verify the status-codes.

```
admin@ncs% show status-codes
core-function-pack SR {
    status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/sr_te_status_codes;
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and
perform recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and
perform recovery steps.";
    }
…
…
    }
core-function-pack CS-SR {
    status-code 400 {
        reason              "Status code mapping has not been loaded for
function pack during install";
        category            user;
        severity            ERROR;
        recommended-actions "Bootstrap status code mapping";
    }
}
[ok]
```

c. Verify the plan-notifications.

```
admin@ncs% run show configuration services plan-notifications
subscription cs-sr-te-notif {
    service-type /cisco-cs-sr-te-cfp:cs-sr-te-policy;
}
subscription sr-odn-notif {
    service-type /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
odn:odn/cisco-sr-te-cfp-sr-odn:odn-template;
}
subscription sr-policy-notif {
    service-type /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
policies:policies/cisco-sr-te-cfp-sr-policies:policy;
}
[ok]
admin@ncs% run show configuration plan-path-for-notification
plan-path-for-notification /cisco-cs-sr-te-cfp:cs-sr-te-plan {
    service-path         /cisco-cs-sr-te-cfp:cs-sr-te-policy;
    service-key-elements [ name ];
}
plan-path-for-notification /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
odn:odn/odn-template-plan {
    service-path         /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
odn:odn/odn-template;
    service-key-elements [ name ];
}
plan-path-for-notification /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
policies:policies/policy-plan {
    service-path         /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-
policies:policies/policy;
    service-key-elements [ name ];
}
```

d. Check the device list. The list displays the devices configured. The devices in this list must also be populated in the dispatch-map. If no devices are configured, the list is empty.

```
admin@ncs> show devices list
```

e. Verify the bootstrap data is successfully loaded.

Verify the dispatch-map. The dispatch-map is populated with the devices from the device list. If there are no devices in the device list, the dispatch-map is empty.

The following is a sample output for a correctly configured dispatch-map for two PIOSXR devices.

```
    admin@ncs% show devices lsa dispatch-map PIOSXR-0 {
        ned-id cisco-iosxr-cli-7.33:cisco-iosxr-cli-7.33;
    }
    dispatch-map PIOSXR-1 {
        ned-id cisco-iosxr-cli-7.33:cisco-iosxr-cli-7.33;
    }
```

f. Verify the commit-queue setting configuration.

```
admin@ncs% show devices global-settings commit-queue
enabled-by-default false;
```

```
        async;
        atomic              false;
        retry-attempts      0;
        retry-timeout       30;
        error-option        stop-on-error;
        [ok]
```

2. (*SR-TE CFP-IOSXE CLI only*) Verify the dynamic-mapping for the IOSXE CLI NEDs installed.

   ```
   admin@ncs% show cisco-sr-te-cfp:cfp-configurations
   dynamic-device-mapping cisco-ios-cli-<version>:cisco-ios-cli-<version> {
       python-impl-class-name sr_te_multi_vendors.IosXE;
   }
   [ok]
   ```

Installing IETF-L2VPN-NM Services

L2VPN-NM (or L2NM) uses the standardized IETF version of L2VPN. This section discusses the procedure and the packages required to install the L2NM service either on SR-TE CFP or as a standalone flavor, and the procedure to verify the same.

Note: To install L2VPN-NM-IOSXE CLI, you must first install L2VPN-NM-IOSXR CLI.

To install L2NM service on SR-TE CFP, make sure SR-TE CFP-IOSXR CLI is installed and then continue to perform the tasks mentioned in this topic. To install IETF-L2VPN-NM-IOSXR CLI as a standalone flavor, perform the tasks from **step 1** to **step 5** in section **Installing SR-TE CFP** and then continue to perform the tasks mentioned in this topic.

To install IETF-L2VPN-NM CFP:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

2. Do one of the following to install L2NM-IOSXR CLI. If L2NM-IOSXR CLI is already installed, skip to step 4 to install L2NM-IOSXE CLI.

   a. Copy and link the following packages to install L2NM-IOSXR CLI on SR-TE CFP-IOSXR CLI.

      ```
      sudo cp ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz /opt/ncs/packages/
      sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-
      <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-
      <version>.tar.gz
      sudo cp ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz
      /opt/ncs/packages/
      sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-
      <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-
      internal-<version>.tar.gz
      ```

```
sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-
service-<version>.tar.gz
```

(*Optional*) To install the AA feature, copy and link the AA package.

```
sudo cp ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-aa-service-
assurance-<version>.tar.gz
```

b. Copy and link the following packages to install L2NM-IOSXR CLI as a standalone flavor.

```
sudo cp ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-
internal-<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-
generator-<version>.tar.gz
```

```
sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-
service-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-iosxr-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz /var/opt/ncs/ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz

sudo cp ncs-<version>-lsa-utils-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz

sudo cp ncs-<version>-resource-manager-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz

sudo cp ncs-<version>-cisco-cfp-jwt-auth-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-cfp-jwt-auth-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-cfp-jwt-auth-<version>.tar.gz
```

3. To install L2NM-IOSXE CLI service, continue with step 4, else skip to step 5 to complete installing L2NM-IOSXR CLI service.

4. Copy and link the following packages to install L2NM-IOSXE CLI:

```
sudo cp ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz

sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

5. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart

Restarting ncs (via systemctl):

                      [  OK  ]
```

   The L2NM installation is now complete.

6. Verify the installation and make sure the packages are up and running, and then perform the post installation tasks.

```
admin@ncs% run show packages package package-version | select build-info ncs version | select build-info file | select build-info package sha1 | select oper-status error-info | select oper-status up | tab
```

   Note: For standalone installation, the cisco-sr-te-cfp package is not displayed in the output.

Performing Post Installation Tasks for IETF-L2VPN-NM Service

Do the following after installing L2NM service:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. Do the following post installation tasks for L2NM-IOSXR CLI installation. If you have already installed and performed the post installation tasks for L2NM-IOSXR CLI, skip to step 3 to perform post installation tasks for L2NM-IOSXE CLI.

   a. Load-merge the **IETF-L2NM-plan-notification-settings.xml** file to activate notifications.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% load merge IETF-L2NM-plan-notification-settings.xml
   admin@ncs% commit
   ```

   b. Load-merge the **IETF-L2NM-status-codes.xml** file to activate status-codes.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% load merge IETF-L2NM-status-codes.xml
   admin@ncs% commit
   ```

   c. Load-merge the following xml files to activate kicker settings.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% load merge IETF-L2NM-internal-plan-kicker.xml
   admin@ncs% load merge 1_IETF-L2NM-cfp-configuration-kicker.xml
   admin@ncs% commit
   admin@ncs% load merge IETF-L2NM-route-policy-kicker.xml
   admin@ncs% commit
   ```

   d. If you have installed AA, load merge the **IETF-L2NM-AA-notification-settings.xml** to activate AA notifications.

   ```
   $ ncs_cli -u admin
   configure
   unhide tsdn
   load merge IETF-L2NM-AA-notification-settings.xml
   commit
   ```

   e. Configure SMAN ID resource pool for Y1731.

   ```
   $ ncs_cli -u admin
   configure
   load merge xr-sman-id-resource-pool.xml
   ```

```
commit
```

3. (*L2NM-IOSXE CLI only*) Load-merge the L2VPN-multi-vendor-iosxe-cli.xml file to configure dynamic-mapping.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% load merge 2_IETF-L2NM-multi-vendor-iosxe-cli.xml
admin@ncs% commit
```

Verifying the Post Installation Tasks for L2NM Service

Verify the post installation tasks as follows:

1. Verify the L2NM-IOSXR CLI installation as follows. If you have already verified L2NM-IOSXR CLI installation, skip to step 2 to verify the L2NM-IOSXE CLI installation.

   a. Verify the kickers configuration.

```
unhide debug
admin@ncs% show kickers
data-kicker flat-L2vpn-internal-remote-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-internal-remote-site/cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker flat-L2vpn-internal-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-internal-site/cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker flat-L2vpn-internal-local-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-internal-local-site/cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker plan-notification-kicker-/l2vpn-ntw:l2vpn-ntw/vpn-services/cisco-l2vpn-ntw:vpn-service-plan {
    monitor     /l2vpn-ntw:l2vpn-ntw/vpn-services/cisco-l2vpn-ntw:vpn-service-plan;
    kick-node   /action;
    action-name generate-plan-notifications;
data-kicker l2nm-defined-set-kicker {
    monitor     /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-
l2vpn-routing-policy:defined-sets;
    kick-node   /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-
l2vpn-routing-policy:policy-definitions;
    action-name internal-defined-sets-change-handler;
}
data-kicker l2nm-route-policy-kicker {
    monitor     /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-
l2vpn-routing-policy:policy-definitions/cisco-l2vpn-routing-
policy:policy-definition;
    kick-node   /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-
ntw:vpn-service[vpn-nodes/vpn-node/te-service-mapping/te-
mapping/odn/route-policy=current()/name];
    action-name reactive-re-deploy;
}
data-kicker service-assurance-subsciption-kicker-/l2vpn-ntw:l2vpn-
ntw/vpn-services/vpn-service {
    monitor     /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service;
    kick-node   /service-assurance;
    action-name service-assurance-action;
}
```

b. Verify the status-codes.

```
admin@ncs% show status-codes core-function-pack IETF-L2NM
status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/ietf_l2vpn_nm_statu
s_codes;
    status-code 400 {
        reason              "Status code mapping has not been loaded for
function pack during install";
        category            user;
        severity            ERROR;
        recommended-actions "Bootstrap status code mapping";
    }
    status-code 404 {
        reason              " The value for the input element is not
supported ";
        category            validation;
        severity            ERROR;
        recommended-actions "Verify that input element's value is
supported in the payload";
```

```
        }
…
        }
}
[ok]
```

c. Verify the plan-notifications. Plan-notifications display AA models only if the AA packages are installed.

```
admin@ncs% run show configuration services plan-notifications

subscription l2nm-notif {

    service-type /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-
ntw:vpn-service;

}

[ok]

admin@ncs% run show plan-path-for-notification

plan-path-for-notification /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-
service-plan {

    service-path         /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service;

    service-key-elements [ vpn-id ];

}

[ok]
```

d. Verify SMAN ID configuration for Y1731.

```
admin@ncs% show resource-pools id-pool sman-id-pool

range {

    start 1;

    end   65535;

}
```

e. If you have installed AA, verify the bootstrap data is successfully loaded for AA notification settings.

```
admin@ncs% unhide tsdn

admin@ncs% run show service-path-for-subscription


                                                         LSA                        LSA       DEVICE   CUSTOMER  PLAN                                       CONFIG
SERVICE PATH                                    DEVICES  SERVICES  SERVICES  DEVICES  SERVICES  SERVICES  LIST      SERVICE   LOCATION  ID  STATUS  NAME  TIME  DATA    ERROR  WHEN  TYPE  LEVEL  MESSAGE
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
/l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service   -        -         -         -        -         -         -         -         -
[ok]

admin@ncs% hide tsdn
```

2. (*L2NM-IOSXE CLI only)* Verify the dynamic mapping for the IOSXE CLI NEDs you installed are listed. The following shows an example for IOSXE CLI 6.100:

```
unhide tsdn
admin@ncs% show l2vpn-ntw cfp-configurations
dynamic-device-mapping cisco-ios-cli-6.100:cisco-ios-cli-6.100 {
python-impl-class-name flat_l3vpn_multi_vendors.IosXE;
}
dynamic-device-mapping cisco-ios-cli-6.100:cisco-ios-cli-6.100 {
    python-impl-class-name flat_l2vpn_multi_vendors.IosXE;
}
dynamic-device-mapping cisco-ios-cli-6.86:cisco-ios-cli-6.86 {
    python-impl-class-name flat_l2vpn_multi_vendors.IosXE;
}
```

Installing IETF-L3VPN-NM Services

L3VPN-NM (L3NM) picks up the standardized IETF version of L3VPN implementation. This section discusses the procedure and packages you must copy to install the L3NM service either on SR-TE CFP or as a standalone flavor and the procedure to verify the same.

Note: To install L3VPN-NM-IOSXE CLI service, you must first install L3VPN-NM-IOSXR CLI service.

To install L3NM service on SR-TE CFP, make sure SR-TE CFP-IOSXR CLI is installed and then continue to perform the tasks mentioned in this topic. To install IETF-L3VPN-NM-IOSXR CLI as a standalone flavor, perform the tasks from **step 1** to **step 5** in section **Installing SR-TE CFP** and then continue to perform the tasks mentioned in this topic.

To install IETF-L3VPN-NM CFP:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

2. Do one of the following to install L3NM-IOSXR CLI service. If you have already installed L3NM-IOSXR CLI, skip to step 4 to install L3NM-IOSXE CLI service.

   a. Copy and link the following packages to install IETF-L3VPN-NM-IOSXR CLI on SR-TE CFP-IOSXR CLI.

   ```
   sudo cp ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-
   <version>.tar.gz

   sudo cp ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-
   internal-<version>.tar.gz

   sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-
   service-<version>.tar.gz
   ```

   (*Optional*) To install the AA feature, copy and link the AA package.

```
sudo cp ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-aa-service-
assurance-<version>.tar.gz
```

b. Copy and link the following packages to install L3NM-IOSXR CLI as a standalone flavor.

```
sudo cp ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-
<version>.tar.gz

sudo cp ncs-<version>-cisco-L3vpn-fp-internal-core-fp-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-
internal-<version>.tar.gz

sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-common-
<version>.tar.gz

sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-
generator-<version>.tar.gz

sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-
utils-<version>.tar.gz

sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-
service-<version>.tar.gz

sudo cp ncs-<version>-cisco-iosxr-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz

sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz /var/opt/ncs/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz

sudo cp ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-aa-service-
assurance-<version>.tar.gz
```

```
sudo cp ncs-<version>-lsa-utils-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz
```

```
sudo cp ncs-<version>-resource-manager-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz
```

3. To install L3NM IOSXE CLI service, continue with step 4, else skip to step 5 to complete installing IETF-L3VPN-NM-IOSXR CLI service.
4. Copy and link the following packages to install L3NM-IOSXE CLI:

```
sudo cp ncs-<version>-l3vpn-multi-vendors-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-
<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

5. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
                      [  OK  ]
```

The L3NM installation is now complete.

6. Verify the installation and make sure the packages are up and running, and then perform the post installation tasks.

```
admin@ncs% run show packages package package-version | select build-info ncs
version | select build-info file | select build-info package sha1 | select
oper-status error-info | select oper-status up | tab
```

Note: For standalone installation, the cisco-sr-te-cfp package is not displayed in the output.

Performing Post Installation Tasks for IETF-L3VPN-NM Service

Do the following after installing L3NM service:

1. Change the current directory to:

```
$ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
```

2. Do the following post-installation tasks for L3NM-IOSXR CLI installation. If you have already installed L3NM-IOSXR CLI and performed the post installation tasks, skip to step 3 to perform the post installation tasks for L3NM-IOSXE CLI.

   a. Load-merge the **IETF-L3NM-plan-notification-settings.xml** file to activate notifications.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% load merge IETF-L3NM-plan-notification-settings.xml
   admin@ncs% commit
   ```

   b. Load-merge the **IETF-L3NM-status-codes.xml** file to activate status-codes.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% load merge IETF-L3NM-status-codes.xml
   admin@ncs% commit
   ```

   c. Load-merge the **IETF-L3NM-internal-plan-kicker.xml** file to activate kicker settings.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% load merge IETF-L3NM-internal-plan-kicker.xml
   admin@ncs% load merge 1_IETF-L3NM-cfp-configuration-kicker.xml
   admin@ncs% load merge IETF-L3NM-route-policy-kicker.xml
   admin@ncs% commit
   ```

   d. If AA is installed, load merge the **IETF-L3NM-AA-notification-settings.xml** file to configure AA notifications.

   ```
   /opt/ncs/current/bin/ncs_cli -u admin
   unhide tsdn
   configure
   load merge IETF-L3NM-AA-notification-settings.xml
   commit
   ```

3. (*L3NM-IOSXE CLI installation only*) Load-merge the **L3VPN-multi-vendor-iosxe-cli.xml** file to configure dynamic mapping.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% load merge 2_IETF-L3NM-multi-vendor-iosxe-cli.xml
admin@ncs% commit
```

Verifying the Post Installation Tasks for IETF-L3VPN-NM Service

Verify the post installation tasks as follows:

1. Verify the L3NM-IOSXR CLI installation as follows. If you have already verified the L3NM-IOSXR CLI installation, skip to step 2 to verify the L3NM-IOSXE CLI installation.

    a. Verify the kickers configuration.

```
unhide debug
admin@ncs% show kickers data-kicker ietf-l3nm-cfp-configuration-kicker {
    monitor     /l3nm:l3vpn-ntw/cisco-l3nm:cfp-configurations;
    kick-node   /l3nm:l3vpn-ntw/cisco-l3nm:l3nm-actions;
    action-name update-internal-cfp-configurations;
}
data-kicker l3nm-internal-plan-kicker {
    monitor     /cisco-flat-L3vpn-fp-internal:flat-L3vpn-internal/cisco-
flat-L3vpn-fp-internal:flat-L3vpn-plan;
    kick-node   /l3nm:l3vpn-ntw/cisco-l3nm:l3nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker l3nm-policy-definition-kicker {
    monitor     /cisco-l3vpn-routing-policy:l3vpn-routing-policy/policy-
definitions/cisco-l3vpn-routing-policy:policy-definition;
    kick-node   /cisco-l3vpn-routing-policy:l3vpn-routing-policy/cisco-
l3vpn-routing-policy:policy-definitions;
    action-name internal-policy-defs-change-handler;
}

data-kicker l3nm-defined-set-kicker {
    monitor     /cisco-l3vpn-routing-policy:l3vpn-routing-policy/cisco-
l3vpn-routing-policy:defined-sets;
    kick-node   /cisco-l3vpn-routing-policy:l3vpn-routing-policy/cisco-
l3vpn-routing-policy:policy-definitions;
    action-name internal-defined-sets-change-handler;
}
```

    b. Verify the status-codes.

```
admin@ncs% show status-codes
core-function-pack IETF-L3NM {
    status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/ietf_l3vpn_nm_statu
s_codes;
```

```
    status-code 400 {
        reason              "Status code mapping has not been loaded for
function pack during install";
        category            user;
        severity            ERROR;
        recommended-actions "Bootstrap status code mapping";
    }
    status-code 404 {
        reason              "Input element's value is not supported";
        category            validation;
        severity            ERROR;
        recommended-actions "Verify that input element's value is
supported in the payload";
    }
…
…
    }
}
[ok]
```

c. Verify the plan-notifications. The plan-notifications display AA models only if the AA package is installed.

```
admin@ncs% run show configuration services plan-notifications
subscription l3nm-notif {
    service-type /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services/l3vpn-
ntw:vpn-service;
}
[ok]

admin@ncs% show plan-path-for-notification
plan-path-for-notification /l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-
service-plan {
    service-path         /l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-service;
    service-key-elements [ vpn-id ];
}
[ok]
```

d. If you have installed AA, verify the bootstrap data is successfully loaded for AA notification settings.

```
admin@ncs% unhide tsdn
admin@ncs% run show service-path-for-subscription
```

```
                                                         LSA                         LSA      DEVICE  CUSTOMER  PLAN                            CONFIG
SERVICE PATH                                   DEVICES  SERVICES  SERVICES  DEVICES  SERVICES SERVICES  LIST    SERVICE   LOCATION  ID  STATUS  NAME  TIME  DATA    ERROR  WHEN  TYPE  LEVEL  MESSAGE
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
/l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-service -     -     -     -     -     -     -     -     -

[ok]

admin@ncs% hide tsdn
```

2. (*L3NM-IOSXE CLI only*) Verify the dynamic-mapping configured for IOSXE CLI NED. The following is an example for IOSXE CLI 6.86 NED.

   ```
   unhide tsdn

   admin@ncs% show l3vpn-ntw cfp-configurations

   dynamic-device-mapping cisco-ios-cli-6.86:cisco-ios-cli-6.86 {

       python-impl-class-name flat_l3vpn_multi_vendors.IosXE;

   }
   ```

   Note: For standalone installation, the cisco-sr-te-cfp package is not displayed in the output.

Installing Automated Assurance Services

The Automated Assurance feature is an optional feature and is applicable only to IETF-L2VPN-L2NM and IETF-L3VPN-L3NM services. This feature is installed if the **ncs-<*version*>-cisco-aa-service-assurance-<*version*>.tar.gz** package is extracted and copied along with the packages to install the Example services.

Installing Example Function Packs on a Single NSO Instance

The SR-TE CFP-IOSXR-CLI is the base flavor and is the pre-requisite to install the Example Function Packs (flavors), except for standalone services.

You can choose to install one or more flavors on top of the base flavor SR-TE CFP-IOSXR CLI. To install a flavor, copy the packages for the flavor either during or after the SR-TE CFP-IOSXR CLI installation.

Additionally, the IOSXR-NC flavor or the IOSXE-CLI flavor for a service (IETF-TE) requires the IOSXR-CLI flavor for the service. For example, to install the IETF-TE-IOSXE flavor, you must:

1. Install the base flavor (SR-TE CFP-IOSXR CLI).
2. Install the IETF-TE-IOSXR CLI flavor on the base flavor .
3. Install the IETF-TE-IOSXE flavor.

Standalone Flavors

To perform standalone installations, do not copy the **cisco-sr-te-cfp** packages during the SR-TE CFP-IOSXR CLI installation. You can install the L2NM/L3NM/IETF-TE services with IOSXR-CLI, IOSXR-NC, or IOSXE-CLI as standalone services.

The standalone installation for the service with IOSXR-CLI is the base and is a pre-requisite to perform the standalone installations for the services with IOSXR-NC or IOSXE-CLI flavors.

For example, to install IETF-TE-IOSXE CLI as a standalone service, you must first install IETF-TE-IOSXR CLI as a standalone service and then install the IETF-TE-IOSXE CLI service.

Installing IETF-TE Services

This section discusses the procedure to copy the packages to install and verify the IETF-TE example service either on SR-TE CFP-IOSXR CLI or as a standalone flavor.

Before you install IETF-TE service, make sure SR-TE CFP-IOSXR CLI is installed and then continue to perform the tasks mentioned in this topic. To install IETF-TE-IOSXR CLI as a standalone flavor, perform the tasks from **step 1** to **step 5** in section **Installing SR-TE CFP** and then continue to perform the tasks mentioned in this topic.

Note: To install IETF-TE-IOSXE CLI service, you must first install IETF-TE-IOSXR CLI service.

To install IETF-TE service:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

2. Do one of the following as required to install IETF-TE-IOSXR CLI service. If the service is already installed, skip to step 3 to install IETF-TE-IOSXE CLI service.

   a. Copy and link the following packages to install IETF TE-IOSXR CLI on SR-TE CFP-IOSXR CLI:

   ```
   sudo cp ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-
   EXAMPLE-<version>.tar.gz

   sudo cp ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
   <version>.tar.gz
   ```

   b. Copy and link the following packages to install IETF-TE-IOSXR CLI as a standalone flavor.

   ```
   sudo cp ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-
   EXAMPLE-<version>.tar.gz
   ```

```
sudo cp ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-
service-<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-common-
<version>.tar.gz
```

```
sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-
generator-<version>.tar.gz
```

```
sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-
utils-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-iosxr-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz /var/opt/ncs/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz
```

3. To install IETF-TE-IOSXE CLI, continue with step 4, else skip to step 5 to complete IETF-TE-IOSXR CLI installation.

4. Copy and link the following packages to install IETF-TE-IOSXE CLI:

```
sudo cp ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-
EXAMPLE-<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

5. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                         [  OK  ]
   ```

   The IETF TE service installation is now complete.

6. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

   Note: The cisco-sr-te-cfp package is not displayed for standalone installation.

7. Perform the post installation tasks.

Performing Post Installation Tasks for IETF-TE Service

Do the following after installing IETF-TE service:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. (*standalone installation only*) Make sure to configure the common bootstrap data as described in section **Performing Post Installation Tasks for SR-TE CFP**.

3. Do the following post installation tasks for IETF-TE-IOSXR CLI installation. If you have already performed the post installation tasks for the installation, skip to step 4 to perform the post installation tasks for IETF-TE-IOSXE CLI installation.

   a. Load-merge the **IETF-TE-plan-notification-settings-example.xml** file to configure plan notifications.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% load merge IETF-TE-plan-notification-settings-example.xml
   admin@ncs% commit
   ```

   b. Load-merge the following xml files to configure the status-codes.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% load merge RSVP-TE-status-codes-example.xml
   load merge IETF-TE-status-codes-example.xml
   admin@ncs% commit
   ```

   c. Load-merge the following xml files to configure kickers.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% load merge IETF-TE-internal-plan-kicker-example.xml
admin@ncs% load merge 1_IETF-TE-cfp-configuration-kicker-example.xml
admin@ncs% commit
```

4. (*IETF-TE-IOSXE CLI only*) Load-merge the **2_IETF-TE-multi-vendor-iosxe-cli-example.xml** file to configure dynamic-mapping.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% load merge 2_IETF-TE-multi-vendor-iosxe-cli-example.xml
admin@ncs% commit
```

Verifying the Post Installation Tasks for IETF-TE Service

Do the following:

1. Verify IETF-TE-IOSXR CLI installation as follows. If the installation is verified, skip to step 2 to verify the IETF-TE-IOSXE CLI installation.

   a. Verify the kickers configuration.

```
admin@ncs% show kickers
data-kicker ietf-te-fp-configuration-kicker {
    monitor     /te:cfp-configurations;
    kick-node   /te:ietf-te-actions;
    action-name update-internal-fp-configurations;
}
data-kicker ietf-te-internal-plan-kicker {
    monitor     /cisco-rsvp-te-fp:rsvp-te/cisco-rsvp-te-fp:tunnel-te-
plan;
    kick-node   /te:ietf-te-actions;
    action-name internal-plan-change-handler;
}
[ok]
```

   b. Verify the status-codes.

```
admin@ncs% show status-codes
core-function-pack IETF-TE {
status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/ietf_te_status_code
s;
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and
perform recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and
perform recovery steps.";
    }
…
…   }
}
core-function-pack RSVP-TE {
status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/rsvp_te_status_code
s;
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and
perform recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and
perform recovery steps.";
    }
…
…   }
}
[ok]
```

c. Verify plan-notifications.

```
admin@ncs% run show configuration services plan-notifications
subscription ietf-te-notif {
    service-type /te:te/te:tunnels/te:tunnel;
}
[ok]
admin@ncs% run show configuration plan-path-for-notification

plan-path-for-notification /te:te/tunnels/tunnel-plan {
    service-path         /te:te/tunnels/tunnel;
    service-key-elements [ name ];
}
[ok]
```

2. (*IETF-TE-IOSXE CLI only*) Verify the dynamic-mapping for the IOSXE CLI NEDs installed. Following is an example for IOSXE CLI 6.86 NED.

   ```
   unhide tsdn
   admin@ncs% show cisco-rsvp-te-fp:cfp-configurations
   dynamic-device-mapping cisco-ios-cli-6.86:cisco-ios-cli-6.86 {
       python-impl-class-name rsvp_te_multi_vendors.IosXE;
   }
   ```

Installing Performance Measurement Services

This section discusses the procedure to copy the packages to install and verify the Performance Measurement service on SR-TE CFP-IOSXR CLI / SR-TE CFP-IOSXE CLI. The Performance Measurement service additionally requires L2NM and L3NM services to be installed.

For more information about how to install these services, see **Installing Core Function Packs on a Single NSO Instance.**

Note: To install PM-IOSXE CLI service, you must first install PM-IOSXR CLI service.

To install Performance Measurement service:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

2. Copy and link the following packages to install SR PM-IOSXR CLI service. If the service is already installed, skip to step 4 to install SR PM-IOSXE CLI service.

   ```
   cp ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz /opt/ncs/packages/
   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz
   cp ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz /opt/ncs/packages/
   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz
   ```

3. To install SR PM-IOSXE CLI service continue with step 4, else skip to step 5 to complete the SR PM-IOSXR CLI service installation.

4. Copy and link the following packages to install L3NM-IOSXE CLI:

   ```
   sudo cp ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz /opt/ncs/packages/
   sudo ln -s /opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz
   sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/
   ```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/var/opt/ncs/packages/ncs-xxx-cisco-ios-<version>.tar.gz
```

5. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                         [  OK  ]
   ```

   The SR PM-IOSXR CLI installation is now complete.

6. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

7. Perform the post installation tasks.

Performing Post Installation Tasks for PM Services

Do the following after installing SR PM service:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. Do the following post installation tasks for SR PM-IOSXR CLI installation. If the tasks are already performed for the installation, skip to step 3 to perform the post installation tasks for SR PM-IOSXE CLI installation.

   a. Load-merge the following kickers xml files to configure dynamic-mapping.

   ```
   $ ncs_cli -u admin
   admin@ncs> configure
   unhide debug
   admin@ncs% load merge 1_PM-cfp-configuration-kicker.xml
   admin@ncs% load merge PM-internal-plan-kicker-example.xml
   admin@ncs% load merge PM-profiles-kicker-example.xml
   admin@ncs% commit
   ```

   b. Load-merge the following status codes files to activate status codes for the service.

   ```
   admin@ncs% ncs_cli -u admin
   admin@ncs% configure
   admin@ncs% unhide debug
   admin@ncs% load merge PM-status-codes-example.xml
   admin@ncs% commit
   ```

   c. Load-merge the following plan notifications to activate notifications for the service.

```
admin@ncs% ncs_cli -u admin
admin@ncs% configure
admin@ncs% load merge PM-plan-notification-settings-example.xml
admin@ncs% commit
```

3. (*SR PM-IOSXE CLI only*) Load-merge the following file to add dynamic-mapping.

```
$ ncs_cli -u admin
admin@ncs% configure
unhide debug
load merge 2_PM-multi-vendor-iosxe-cli-example.xml
commit
```

Verifying the Post Installation Tasks for PM Services

Do the following to verify the PM services:

1. Verify kickers configuration.

```
unhide debug
admin@ncs% show kickers
data-kicker plan-notification-kicker-/cisco-pm-fp:pm/pm-plan {
    monitor     /cisco-pm-fp:pm/pm-plan;
    kick-node   /action;
    action-name generate-plan-notifications;
}
data-kicker pm-cfp-configuration-kicker {
    monitor     /cisco-pm-fp:cfp-configurations;
    kick-node   /cisco-pm-fp:pm/pm-actions;
    action-name update-internal-cfp-configurations;
}
data-kicker pm-internal-plan-kicker {
    monitor     /cisco-pm-fp-internal:pm-internal/pm-internal-plan;
    kick-node   /cisco-pm-fp:pm/pm-actions;
    action-name internal-plan-change-handler;
}
data-kicker pm-internal-profiles-change-kicker {
    monitor     /cisco-pm-fp:pm/cisco-pm-fp:pm-profiles;
    kick-node   /cisco-pm-fp:pm/cisco-pm-fp:pm-profiles/pm-profiles-actions;
    action-name profiles-change-handler;
}
```

2. Verify status-codes configuration.

```
admin@ncs% show status-codes
core-function-pack PM {
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and perform 
recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and perform 
recovery steps.";
    }
}
```

3. Verify plan-notifications configuration.

```
admin@ncs% run show configuration services plan-notifications
subscription pm-notif {
    service-type /cisco-pm-fp:pm/cisco-pm-fp:svc-profiles;
}
admin@ncs% run show configuration plan-path-for-notification
plan-path-for-notification /cisco-pm-fp:pm/pm-plan {
    service-path          /cisco-pm-fp:pm/svc-profiles;
    service-key-elements [ name ];
}
```

4. (*SR PM-IOSXE CLI only*) Verify dynamic-mapping.

```
admin@ncs% show cisco-pm-fp:cfp-configurations
dynamic-device-mapping cisco-ios-cli-6.86:cisco-ios-cli-6.86 {
python-impl-class-name pm_multi_vendors.IosXE;
}
```

Installing IETF Network Slice Services – IOSXR CLI

This section discusses the procedure to copy the packages to install and verify the IETF NSS-IOSXR CLI service on SR-TE CFP IOSXR CLI.

IETF NSS-IOSXR CLI additionally requires L2NM-IOSXR CLI, L3NM-IOSXR CLI, and Performance Management services to be installed.

For more information about installing these services, see **Installing and Uninstalling Cisco NSO T-SDN Function Pack Bundle on a Single NSO Instance**.

To install and verify IETF NSS - IOSXR CLI:

1. Go to the packages directory, and copy and link the required packages.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

   ```
   cp ncs-<version>-ietf-network-slice-service-EXAMPLE-<version>.tar.gz
   /opt/ncs/packages/
   ```

   ```
   sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-network-slice-service-
   EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-network-
   slice-service-EXAMPLE-<version>.tar.gz
   ```

   ```
   cp ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz /opt/ncs/packages/
   ```

   ```
   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-
   <version>.tar.gz
   ```

   ```
   cp ncs-<version>.cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz
   /opt/ncs/packages/
   ```

   ```
   sudo ln -s /opt/ncs/packages//ncs-<version>.cisco-pm-fp-internal-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages//ncs-<version>.cisco-pm-fp-internal-
   EXAMPLE-<version>.tar.gz
   ```

2. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                         [  OK  ]
   ```

   The IETF NSS-IOSXR CLI installation is now complete.

3. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

4. Perform the post installation tasks.

Performing Post Installation Tasks for IETF NSS-IOSXR CLI

Do the following after installing IETF NSS-IOSXR CLI:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. Load-merge the following kickers xml files to configure dynamic-mapping.

```
$ ncs_cli -u admin
admin@ncs> configure
unhide debug
admin@ncs% load merge IETF-NSS-internal-plan-kicker-example.xml
admin@ncs% load merge 1_PM-cfp-configuration-kicker-example.xml
admin@ncs% load merge PM-internal-plan-kicker-example.xml
admin@ncs% load merge PM-profiles-kicker-example.xml
admin@ncs% commit
```

3. Load-merge the following status codes files to activate status codes for the service.

```
admin@ncs% ncs_cli -u admin
admin@ncs% configure
admin@ncs% unhide debug
admin@ncs% load merge IETF-NSS-status-codes-example.xml
admin@ncs% load merge PM-status-codes-example.xml
admin@ncs% commit
```

4. Load-merge the following plan notifications to activate notifications for the service.

```
admin@ncs% ncs_cli -u admin
admin@ncs% configure
admin@ncs% load merge IETF-NSS-plan-notification-settings-example.xml
admin@ncs% load merge PM-plan-notification-settings-example.xml
admin@ncs% commit
```

5. Load-merge the following xml file to configure resource pool.

```
admin@ncs% ncs_cli -u admin
admin@ncs% configure
admin@ncs% load merge IETF-NSS-resource-pools-example.xml
admin@ncs% commit
```

Verifying the Post Installation Tasks for IETF NSS-IOSXR CLI

Verify the post installation tasks as follows:

1. Verify kickers configuration.

```
unhide debug
admin@ncs% show kickers
ietf-nss-internal-l2nm-plan-kicker {
    monitor     /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-ntw-
cisco:vpn-service-plan;
    kick-node   /ietf-nss:network-slice-services/ietf-nss-cisco:actions;
    action-name internal-plan-change-handler;
}
data-kicker ietf-nss-internal-l3nm-plan-kicker {
    monitor     /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services/l3vpn-ntw:vpn-
service-plan;
    kick-node   /ietf-nss:network-slice-services/ietf-nss-cisco:actions;
    action-name internal-plan-change-handler;
}
data-kicker plan-notification-kicker-/ietf-nss:network-slice-services/ietf-
nss-cisco:slice-service-plan {
    monitor     /ietf-nss:network-slice-services/ietf-nss-cisco:slice-
service-plan;
    kick-node   /action;
    action-name generate-plan-notifications;
}
```

2. Verify status-codes configuration.

```
admin@ncs% show status-codes
core-function-pack IETF-NSS {
    status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/ietf_nss_status_codes;
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and perform
recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and perform
recovery steps.";
    }
```

3. Verify plan-notifications configuration.

```
admin@ncs% run show configuration services plan-notifications
subscription ietf-nss-notif {
    service-type /ietf-nss:network-slice-services/ietf-nss:slice-service;
}
admin@ncs% run show configuration plan-path-for-notification
plan-path-for-notification /ietf-nss:network-slice-services/ietf-nss-
cisco:slice-service-plan {
    service-path         /ietf-nss:network-slice-services/slice-service;
    service-key-elements [ service-id ];
}
```

4. Verify the resource pools configuration.

```
admin@ncs% show resource-pools id-pool
id-pool mep-id-pool {
    range {
        start 1;
        end   8191;
    }
}
admin@ncs% show network-slice-services cfp-configurations mep-id-pool-name
mep-id-pool-name mep-id-pool;
```

Uninstalling on a Single NSO Instance

To uninstall Cisco NSO T-SDN FP Bundle, you must first remove the associated services and any associated devices from the system. Make sure no zombie services are running for the services and all the devices are removed from the device tree.

Note: Do not remove the T-SDN FP Bundle common packages or the CLI NEDs if you continue to use the Example Function Packs (services) as standalone flavors after uninstalling T-SDN FP Bundle.

If you uninstall a service (flavor) installed on SR-TE CFP-IOSXR CLI, the system continues to render the SR-TE CFP-IOSXR CLI services. However, if you uninstall SR-TE CFP-IOSXR CLI without uninstalling the flavor, only the services rendered by the flavor are available. In such cases, the flavor functions as a standalone service only if the common packages and the required packages for the services are available.

This section discusses the procedure to uninstall the Example Function Packs (flavors) and the Core Function Packs.

Reverting Changes to the NCS Configuration File on a Single NSO Instance

Revert the changes to the **ncs.config** file before you uninstall the flavors and the Cisco NSO T-SDN FP Bundle.

To revert the NCS configuration file:

1. Stop NCS.
2. Edit the **/etc/ncs/ncs.conf** as follows:
    a. Remove **service-state-changes** information under **<stream>**.

```
<notifications>
```

```
<event-streams>
    <stream>
        <name>service-state-changes</name>
        <description>Plan state transitions according to
  tailf-ncs-plan.yang</description>
        <replay-support>false</replay-support>
        <builtin-replay-store>
            <enabled>false</enabled>
            <dir>./state</dir>
            <max-size>S10M</max-size>
            <max-files>50</max-files>
        </builtin-replay-store>
    </stream>
```

b. Remove **<hide-group>** section at the end of the file.

```
<hide-group>
    <name>debug</name>
</hide-group>
<hide-group>
    <name>tsdn</name>
</hide-group>
<hide-group>
    <name>fastmap-private</name>
</hide-group>
<hide-group>
    <name>lsa</name>
</hide-group>
```

c. If you enabled the SSH port configuration for the CLI, Webui, and the northbound notifications, it is optional to revert the configuration.

Uninstalling Example Function Packs

Uninstalling a flavor installed on SR-TE CFP-IOSXR CLI continues to render the SR-TE CFP-IOSXR CLI services on the system. Delete all the services for the flavor before uninstalling the flavor.

Uninstalling IETF-TE Services

Use the information in this section to uninstall IETE-TF services. Uninstalling the IETF-TE service reverts the system to SR-TE CFP-IOSXR CLI. For more information on how to uninstall SR-TE CFP-IOSXR CLI, see Error! Reference source not found..

Note: Do not remove the T-SDN FP Bundle common packages or the CLI NEDs if you continue to use the Example Function Packs (services) as standalone flavors after uninstalling T-SDN FP Bundle.

To uninstall IETF-TE service:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.
2. To uninstall IETF-TE-IOSXR CLI, do the following. To uninstall IETF-TE-IOSXR CLI, skip to step 4.
    a. Delete plan-notifications.

    ```
    $ /opt/ncs/current/bin/ncs_cli -u admin
    admin@ncs> configure
    admin@ncs% delete services plan-notifications subscription ietf-te-notif
    admin@ncs% delete plan-path-for-notification /te:te/tunnels/tunnel-plan
    admin@ncs% commit
    ```

    b. Delete status-codes for IETF-TE.

    ```
    $ /opt/ncs/current/bin/ncs_cli -u admin
    admin@ncs> configure
    admin@ncs% unhide debug
    admin@ncs% delete status-codes core-function-pack IETF-TE
    admin@ncs% delete status-code-cfp IETF-TE
    admin@ncs% delete status-codes core-function-pack RSVP-TE
    admin@ncs% delete status-code-cfp RSVP-TE
    admin@ncs% commit
    ```

    c. Delete kickers.

    ```
    $ /opt/ncs/current/bin/ncs_cli -u admin
    $ ncs_cli -u admin
    admin@ncs> configure
    admin@ncs% unhide debug
    admin@ncs% delete kickers data-kicker ietf-te-fp-configuration-kicker
    admin@ncs% delete kickers data-kicker ietf-te-internal-plan-kicker
    admin@ncs% commit
    ```

    b. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

    ```
    sudo rm -f /var/opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
    <version>.tar.gz /opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-
    <version>.tar.gz

    sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-
    <version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-
    EXAMPLE-<version>.tar.gz
    ```

3. Continue with step 4 to uninstall IETF-TE-IOSXE CLI service, else skip to step 6 to complete uninstalling IETF-TE-IOSXR CLI service.

4. (*IETF-TE-IOSXE CLI only*) Unlink the following package in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-
EXAMPLE-<version>.tar.gz
```

5. Remove the XE CLI NED if it is not used in other services.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

6. Stop NSO.

```
### Make sure user delete all services and devices from TSDN

### Make sure there are no zombie services by running the command: show
zombies

sudo /etc/init.d/ncs stop
```

7. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart

Restarting ncs (via systemctl):

               [   OK   ]
```

Uninstalling IETF Network-Slice-Services – IOSXR CLI

Use the information in this section to uninstall the IETE-NSS service with IOSXR CLI. Be sure to delete all the IETF NSS services with CLI NED before uninstalling IETF NSS.

To uninstall IETF-TE-IOSXE-CLI:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.

2. Delete plan-notifications.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% delete services plan-notifications subscription ietf-nss-notif

admin@ncs% delete plan-path-for-notification /ietf-nss:network-slice-
services/ietf-nss-cisco:slice-service-plan

admin@ncs% commit
```

3. Delete status-codes for IETF-TE.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% unhide debug

admin@ncs% delete status-codes core-function-pack IETF-NSS
```

```
admin@ncs% delete status-code-cfp IETF-NSS
admin@ncs% commit
```

4. Delete kickers.

```
$ /opt/ncs/current/bin/ncs_cli -u admin
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% delete kickers data-kicker ietf-nss-internal-l2nm-plan-kicker
admin@ncs% delete kickers data-kicker ietf-nss-internal-l3nm-plan-kicker
admin@ncs% commit
```

5. Unlink the following package in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

```
sudo rm -f /opt/ncs/packages/ncs-<version>-ietf-network-slice-service-
EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-network-
slice-service-EXAMPLE-<version>.tar.gz
```

6. Remove XE CLI NED if it is not used in other services.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

7. Stop NSO.

```
### Make sure user delete all services and devices from TSDN
### Make sure there are no zombie services by running the command: show
zombies
sudo /etc/init.d/ncs stop
```

8. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
               [  OK  ]
```

Uninstalling Performance Measurement Service

Uninstall the PM packages to uninstall PM services with IOSXR CLI/IOSXE CLI NED. Removing PM service reverts the system to L2NM/L3NM/SR-TE CFP flavors with the corresponding NEDs. Use the information in this section to uninstall the Performance Measurement service.

Note: Do not remove the CLI NEDs if you continue to use the Core Function Packs (services) after uninstalling the PM service.

To uninstall the Performance Measurement service:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.

2. To uninstall SR PM (IOSXR CLI/IOSXE CLI) service, do the following.

   a. Delete plan-notifications.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% delete services plan-notifications subscription pm-notif
   admin@ncs% delete plan-path-for-notification /cisco-pm-fp:pm/pm-plan
   admin@ncs% commit
   ```

   b. Delete status-codes for PM.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% delete status-codes core-function-pack PM
   admin@ncs% delete status-code-cfp PM
   admin@ncs% commit
   ```

   c. Delete kickers.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   $ ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% unhide debug
   admin@ncs% delete kickers data-kicker pm-cfp-configuration-kicker
   admin@ncs% delete kickers data-kicker pm-internal-plan-kicker
   admin@ncs% data-kicker pm-internal-profiles-change-kicker
   admin@ncs% commit
   ```

   d. Unlink the following package in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

   ```
   sudo rm -f /opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-
   EXAMPLE-<version>.tar.gz
   ```

   ```
   sudo rm -f /opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-
   EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-
   fp-internal-EXAMPLE-<version>.tar.gz
   ```

3. To complete uninstalling PM-IOSXR CLI only, skip to step 5; else continue with step 4.

4. (SR PM-IOSXE CLI service only) Remove dynamic-mapping for the PM-IOSXE CLI service.

   ```
   ncs_cli -u admin
   configure
   ```

```
delete cisco-pm-fp:cfp-configurations dynamic-device-mapping cisco-ios-
cli-   <version>:cisco-ios-cli-<version>

commit
```

5. Stop NSO.

   ```
   ### Make sure user delete all services and devices from TSDN

   ### Make sure there are no zombie services by running the command: show
   zombies

   sudo /etc/init.d/ncs stop
   ```

6. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
              [   OK   ]
   ```

Uninstalling Core Function Packs

Use the information in this section to uninstall the SR-TE CFPs with IOSXE CLI/IOSXR-CLI. Uninstalling SR-TE CFP IOSXR CLI uninstalls the Cisco T-SDN FP Bundle.

Note: Do not remove the T-SDN FP Bundle common packages or the CLI NEDs if you continue to use the Example Function Packs (services) as standalone flavors after uninstalling T-SDN FP Bundle.

Uninstalling IETF-L2VPN-NM Services

Use the information in this section to uninstall L2NM services with IOSXR CLI/IOSXE CLI. Before you uninstall L2NM service, be sure to delete all the related services and the devices from the device tree.

Uninstalling L2NM-IOSXR CLI service reverts the system to SR-TE CFP-IOSXR CLI flavor and uninstalling the L2NM-IOSXE CLI service reverts the system to SR-TE CFP IOSXE-CLI flavor. For more information on how to uninstall CFP services, see **Uninstalling Core Function Packs**.

To uninstall L2NM service:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File**.

2. Do the following to uninstall L2NM-IOSXR CLI installation. (To uninstall L2NM-IOSXE CLI installation, skip to step 3).

   a. Delete plan-notifications.

   ```
   $ ncs_cli -u admin

   admin@ncs> configure

   admin@ncs% delete services plan-notifications subscription l2nm-notif
   ```

```
admin@ncs% delete plan-path-for-notification /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service-plan

admin@ncs% commit
```

b. Delete status-codes.

```
$ ncs_cli -u admin

admin@ncs> configure

admin@ncs% unhide debug

admin@ncs% delete status-codes core-function-pack IETF-L2NM

admin@ncs% delete status-code-cfp IETF-L2NM

admin@ncs% commit
```

c. Delete kickers.

```
$ ncs_cli -u admin

admin@ncs> configure

admin@ncs% unhide debug

admin@ncs% delete kickers data-kicker flat-L2vpn-internal-local-site-plan-kicker

admin@ncs% delete kickers data-kicker flat-L2vpn-internal-remote-site-plan-kicker

admin@ncs% delete kickers data-kicker l2nm-route-policy-kicker

admin@ncs% delete kickers data-kicker l2nm-defined-set-kicker

admin@ncs% delete kickers data-kicker flat-L2vpn-internal-site-plan-kicker

admin@ncs% delete kickers data-kicker l2nm-cfp-configuration-kicker
admin@ncs% commit
```

d. If AA is installed, delete notifications for the AA module.

```
unhide debug

/opt/ncs/current/bin/ncs_cli -u admin

configure

delete service-path-for-subscription /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service

commit
```

e. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

```
sudo rm -f /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz

sudo rm -f /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
sudo rm -f /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz
```

```
sudo rm -f /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
```

3. Do the following to uninstall L2NM-IOSXE CLI flavor, else skip to step 4 to complete uninstalling L2NM-IOSXR CLI flavor.

   a. Delete all the dynamic-mappings configured for the IOSXE CLI NED for the flavor.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% delete cisco-flat-L2vpn-fp:cfp-configurations dynamic-device-mapping cisco-ios-cli-<version>:cisco-ios-cli-<version>
   admin@ncs% commit
   ```

   b. Unlink the following packages in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

   ```
   sudo rm -f /var/opt/ncs/packages/ncs-<version>-flat-l2vpn-multi-vendors-<version>.tar.gz /opt/ncs/packages/ncs-<version>-flat-l2vpn-multi-vendors-<version>.tar.gz
   sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
   ```

4. Stop NSO.

   ```
   ### Make sure user delete all services and devices from TSDN
   ### Make sure there are no zombie services by running the command: show zombies
   sudo /etc/init.d/ncs stop
   ```

5. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                  [  OK  ]
   ```

Uninstalling IETF-L3VPN-NM Services

Use the information in this section to uninstall L3NM services with IOSXR CLI/IOSXE CLI NEDs. Before you uninstall L3NM service, be sure to delete all the related services and the devices from the device tree.

Uninstalling L3NM-IOSXR CLI flavor reverts the system to SR-TE CFP-IOSXR CLI and uninstalling L3NM-IOSXE CLI flavor reverts the system to L3NM IOSXR-CLI. For more information on how to uninstall SR-TE CFP-IOSXR CLI, see **Uninstalling Core Function Packs**.

To uninstall L3NM service:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.
2. Do the following to uninstall L3NM-IOSXR CLI installation. (To uninstall L3NM-IOSXE CLI installation, skip to step 3).
   a. Delete plan-notifications.

      ```
      $ ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% delete services plan-notifications subscription l3nm-notif
      admin@ncs% delete plan-path-for-notification /l3vpn-ntw:l3vpn-ntw/vpn-
      services/vpn-service-plan
      admin@ncs% commit
      ```

   b. Delete status-codes.

      ```
      $ ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% unhide debug
      admin@ncs% delete status-codes core-function-pack IETF-L3NM
      admin@ncs% delete status-code-cfp IETF-L3NM
      admin@ncs% commit
      ```

   c. Delete kickers.

      ```
      $ ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% unhide debug
      admin@ncs% delete kickers data-kicker service-assurance-subsciption-
      kicker-/l3nm:l3vpn-ntw/vpn-services/vpn-service
      admin@ncs% delete kickers data-kicker l3nm-internal-plan-kicker
      admin@ncs% delete kickers data-kicker ietf-l3nm-cfp-configuration-kicker
      admin@ncs% delete kickers data-kicker l3nm-defined-set-kicker
      admin@ncs% delete kickers data-kicker l3nm-policy-definition-kicker
      admin@ncs% commit
      ```

   d. If AA is installed, delete notifications for the AA module.

      ```
      unhide debug
      /opt/ncs/current/bin/ncs_cli -u admin
      configure
      delete service-path-for-subscription /l3vpn-ntw:l3vpn-ntw/vpn-
      services/vpn-service
      commit
      ```

e. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

```
sudo rm -f /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz

sudo rm -f /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz

sudo rm -f /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz
```

3. Do the following to uninstall L3NM-IOSXE CLI flavor, else skip to step 4 to complete uninstalling L3NM-IOSXR CLI flavor.

a. Delete all the dynamic-mappings configured for the IOSXE CLI NED for the flavor.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% delete cisco-flat-L3vpn-fp:cfp-configurations dynamic-device-mapping  cisco-ios-cli-<version>:cisco-ios-cli-<version>

admin@ncs% commit
```

b. Unlink the following packages in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

```
sudo rm -f /opt/ncs/packages/ncs-<version>-flat-l3vpn-multi-vendors-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-flat-l3vpn-multi-vendors-<version>.tar.gz

sudo rm -f /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

4. Stop NSO.

```
### Make sure user delete all services and devices from TSDN

### Make sure there are no zombie services by running the command: show zombies

sudo /etc/init.d/ncs stop
```

5. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart

Restarting ncs (via systemctl):
               [  OK  ]
```

Uninstalling SR-TE CFP Services

Before uninstalling the SR-TE CFP services, you must first remove any associated services and devices from the system. Make sure no zombie services are running for the services.

For more information on how to remove the services, see chapter ***Deleting Services*** in the ***Cisco NSO T-SDN FP Bundle User Guide***.

Uninstalling SR-TE CFP-IOSXE CLI reverts the system to SR-TE CFP-IOSXR CLI.SR-TE CFP-IOSXR CLI is the main component in Cisco NSO T-SDN FP Bundle. Uninstalling SR-TE CFP-IOSXR CLI and the associated packages from the system removes the Cisco NSO T-SDN FP Bundle. Only a user who has **sudo** privileges and is part of **ncsadmin** user group can perform this uninstallation.

To uninstall SR-TE CFP services:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.
2. The following procedure uninstalls SR-TE CFP-IOSXE CLI service.
   a. Delete dynamic-mapping for the installed IOSXR CLI NED versions.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% delete cisco-sr-te-cfp:cfp-configurations dynamic-device-mapping cisco-ios-cli-<version>:cisco-ios-cli-<version>

admin@ncs% commit
```

   b. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory. Remove the IOSXE CLI NEDs installed with the multi-vendor package.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-<version>.tar.gz /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

   c. Delete all the services and devices.
3. The following procedure uninstalls SR-TE CFP-IOSXR CLI. Uninstalling SR-TE CFP-IOSXR CLI and the associated packages from the system removes the Cisco NSO T-SDN FP Bundle.
   a. Delete plan notifications.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% delete services plan-notifications subscription cs-sr-te-notif

admin@ncs% delete services plan-notifications subscription sr-policy-notif

admin@ncs% delete plan-path-for-notification /cisco-cs-sr-te-cfp:cs-sr-te-plan
```

```
admin@ncs% delete services plan-notifications subscription sr-odn-notif

admin@ncs% delete plan-path-for-notification /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-odn:odn/odn-template-plan

admin@ncs% delete plan-path-for-notification /cisco-sr-te-cfp:sr-te/cisco-sr-te-cfp-sr-policies:policies/policy-plan

admin@ncs% commit
```

b. Delete status-codes.

```
$ /opt/ncs/current/bin/ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% delete status-code-cfp CS-SR
admin@ncs% delete status-codes core-function-pack CS-SR
admin@ncs% delete status-code-cfp SR
admin@ncs% delete status-codes core-function-pack SR
admin@ncs% commit
```

c. Delete kickers.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% delete kickers data-kicker cs-sr-te-internal-plan-kicker
admin@ncs% delete kickers data-kicker sr-te-cfp-configuration-kicker
admin@ncs% delete kickers data-kicker sr-te-odn-internal-plan-kicker
admin@ncs% delete kickers data-kicker sr-te-policy-internal-plan-kicker
admin@ncs% commit
```

d. Unlink the packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-custom-template-utils-<version>.tar.gz /opt/ncs/packages/ncs-<version>-custom-template-utils-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-<version>.tar.gz
sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-iosxr-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-iosxr-
<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-
common-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-
fp-common-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-lsa-utils-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-lsa-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-lsa-utils-
<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz
```

4. Stop NSO.

```
### Make sure user delete all services and devices from TSDN
### Make sure there are no zombie services by running the command: show
zombies
sudo /etc/init.d/ncs stop
```

5. Restart NSO with package reload.

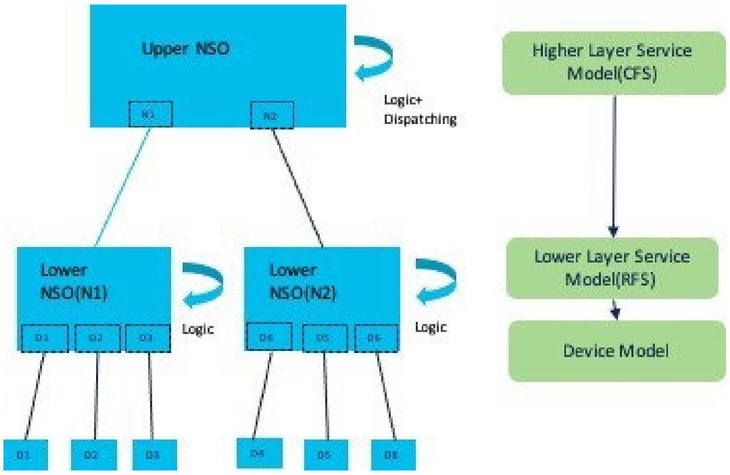
```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
               [  OK  ]
```

Installing and Uninstalling T-SDN FP Bundle in the LSA Model

The LSA model splits the T-SDN FP Bundle into two parts – one upper-level customer-facing service (CFS) part and one or more lower-level resource-facing service (RFS) parts.

The lower-level node layer (RFS node) comprises the managed devices mounted on their /devices tree. The lower-level node pushes the configurations to the devices in the network. These lower-level nodes are added as devices in the upper-level node on their /devices tree. The upper-level node does not contain any devices, except the lower-level device nodes. The upper-level node and the lower-level node communicate with each other by using Netconf.

The following diagram shows the T-SDN FP Bundle installation by using the LSA deployment model.



The upper-level node comprises the CFS packages for the T-SDN FP Bundle, the RFS NEDs, and other common packages and corresponding NEDs (such as core-fp-common and core-fp-common-ned).

The lower-level node has the RFS package for the T-SDN FP Bundle, other common packages, and the required device NEDs.

In the LSA model, install the upper-level node and each lower-level node (for each CFP) on separate NSO instances. The upper-level node is one common node to both the lower-level nodes. It is recommended to install the lower- level node for the CFP first and then the upper-level node. This is because you must add the lower-level nodes as devices on the upper-level node.

Installing T-SDN FP Bundle on the Lower-Level Nodes

This section discusses the required packages and configurations required to install the FP bundle on the lower-level node(s) in the LSA model.

Package Categories and Packages – Lower-Level Nodes

The following table shows the package categories and the packages extracted on the lower-level nodes. The IOSXR CLI NED is the default NED and is packaged with the installation tar file. The IOSXR Netconf NEDs are downloadable from the Cisco website.

RFS Node Packages	
Package Category	**Packages**
T-SDN FP Bundle packages	ncs-6.1.4-cisco-sr-te-cfp-internal-6.0.0.tar.gz ncs-6.1.4-sr-te-multi-vendors-6.0.0.tar.gz ncs-6.1.4-cisco-L2vpn-fp-internal-6.0.0.tar.gz ncs-6.1.4-l2vpn-multi-vendors-6.0.0.tar.gz ncs-6.1.4-l3vpn-multi-vendors-6.0.0.tar.gz ncs-6.1.4-cisco-L3vpn-fp-internal-6.0.0.tar.gz
T-SDN FP Bundle common packages	ncs-6.1-core-fp-common-1.33.0.tar.gz ncs-6.1-custom-template-utils-2.0.13.tar.gz ncs-6.1-core-fp-delete-tag-service-1.0.6.tar.gz ncs-6.1-core-fp-plan-notif-generator-1.0.10.tar.gz ncs-6.1-lsa-utils-1.0.4.tar.gz ncs-6.1.4-cisco-tsdn-core-fp-common-6.0.0.tar.gz ncs-6.1.4-cw-device-auth-6.0.0.9.tar.gz
LSA NED packages	**IOSXR CLI NED:** ncs-6.1-cisco-iosxr-7.52.2.tar.gz ncs-6.1-cisco-iosxr-7.46.2.tar.gz **IOSXE CLI NED:** ncs-6.1-cisco-ios-6.86.6.tar.gz ncs-6.1-cisco-ios-6.100.tar.gz
CNC User Specific Packages	ncs-6.1.4-cisco-tm-tc-fp-6.0.0-9.tar.gz ncs-6.1.4-dlm-svc-6.0.0-22.tar.gz nso-6.1.4-cw-device-auth-6.0.0-9.tar.gz
Example packages	ncs-6.1.4-cisco-rsvp-te-fp-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-rsvp-te-multi-vendors-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-cisco-pm-fp-internal-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-pm-multi-vendors-EXAMPLE-6.0.0.tar.gz

Modifying the NCS Configuration File on the Lower-Level Nodes

Back up the **/etc/ncs/ncs.conf** file and then modify the file for each lower-level node of the CFPs. Use this backup file to restore the configurations while uninstalling the CFPs.

For more information about the ncs.conf file, see the **nso_man-<*version*>.pdf** documentation in **volume5**.

To modify the NCS configuration file on the lower-level nodes:

1. (*Optional*) Configure the SSH port for CLI, webui, and netconf northbound parameters. You can choose to either enable or disable the SSH port configuration, as required, for these parameters. By default, the SSH port configuration for these parameters is disabled. For more information on these parameters, see the ***NSO documentation***.

   To enable the SSH port configuration, if required, provide the port numbers as per your requirement.

   SSH port for CLI

   ```
   <cli>
     <enabled>true</enabled>
     <!-- Use the builtin SSH server -->
     <ssh>
            <enabled>true</enabled>
            <ip>0.0.0.0</ip>
            <port>${North_Bound_CLI_SSH_Port}</port>
            <extra-listen>
             <ip>::</ip>
             <port>${North_Bound_CLI_SSH_Port}</port>
              </extra-listen>
     </ssh>
   ```

   webui

   Enable webui either in TCP or SSL.

   ```
   <webui>
   <enabled>true</enabled>
     <transport>
       <tcp>
          <enabled>true</enabled>
           <ip>0.0.0.0</ip>
           <port>${North_Bound_Web_UI_Port}</port>
       </tcp>
   ```

```
       <ssl>
<enabled>true</enabled>
           <ip>0.0.0.0</ip>
           <port>${SSL_port}</port>
           <key-file>${NCS_CONFIG_DIR}/ssl/cert/host.key</key-file>
           <cert-file>${NCS_CONFIG_DIR}/ssl/cert/host.cert</cert-file>
       </ssl>
     </transport>
```

netconf northbound

```
<netconf-north-bound>
    <enabled>true</enabled>
        <transport>
<ssh>
<enabled>true</enabled>
<ip>0.0.0.0</ip>
<port>${Netconf_North_Bound_port}</port>
</ssh>
```

2. Add the following under **notifications/event-streams**.

Dispatch map

```
<stream>
   <name>dispatch-map-update</name>
    <description>Device addition/removal on RFS notified to
CFS</description>
     <replay-support>true</replay-support>
      <builtin-replay-store>
         <enabled>true</enabled>
         <dir>${NCS_RUN_DIR}/state</dir>
         <max-size>S10M</max-size>
         <max-files>50</max-files>
      </builtin-replay-store>
 </stream>
```

Custom-template events

```
<stream>
   <name>custom-template-events</name>
    <description>Custom Template updates on RFS notified to
CFS</description>
     <replay-support>true</replay-support>
      <builtin-replay-store>
```

```
            <enabled>true</enabled>
            <dir>${NCS_RUN_DIR}/state</dir>
            <max-size>S10M</max-size>
            <max-files>50</max-files>
            </builtin-replay-store>
      </stream>
```

3. Configure the following under **/logs**.

   netconf-trace-log

```
<netconf-trace-log>
 <enabled>true</enabled>
 <filename>${NCS_LOG_DIR}/netconf-north.trace</filename>
 <format>pretty</format>
 </netconf-trace-log>
```

   webui-browser-log

```
 <webui-browser-log>
   <enabled>true</enabled>
   <filename>${NCS_LOG_DIR}/webui-browser.log</filename>
 </webui-browser-log>
```

4. Append the **<hide-group>** information to the file.

```
<hide-group>
 <name>tsdn</name>
</hide-group>
<hide-group>
 <name>debug</name>
</hide-group>
<hide-group>
 <name>lsa</name>
</hide-group>
```

5. Add and set **<commit-message>** parameter to **false** under suppress-commit-message-context.

```
<commit-message>false</commit-message>
```

6. Configure Java-API parameters.

```
<japi>
<new-session-timeout>PT3600S</new-session-timeout>
<query-timeout>PT3600S</query-timeout>
<connect-timeout>PT3600S</connect-timeout>
<event-reply-timeout>PT3600S</event-reply-timeout>
</japi>
```

Installing FP Bundle on the Lower-Level Nodes

To install T-SDN FP Bundle on the lower-level nodes:

1. Make sure you have completed the tasks described in sections **Preparing the NSO Environment to Install the Cisco T-SDN FP Bundle** and **Modifying the NCS Configuration File on the Lower-Level Nodes**.
2. Log in to the host machine as a **sudo** user, who is also part of the **ncsadmin** user group.
3. Obtain and download the **tsdn-<*version*>-nso-<*version*>.signed.bin** file from Cisco website and copy it to the lower-level node.
4. Extract the content of the bin file to the current directory. If the folder already exists, back up the existing folder.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin
   ```

   This verifies the authenticity of the product. However, if you encounter any network connectivity issues, run the following command to skip this verification.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin --skip-verification
   ```

5. Untar the installer tar.gz file to the current directory to extract the T-SDN FP Bundle packages. If the folder already exists, be sure to create a backup of the existing folder.

   ```
   $tar -xf tsdn-<version>-nso-<version>.tar.gz
   ```

6. Go to the lower-level node packages directory and change the current directory as follows:

   ```
   $ cd tsdn-<version>-nso-<version>/LSA/RFS/packages
   ```

   Note: If you are installing DLM, TM-TC, and/or NCA user-specific packages, download the packages and create the symbolic links for these packages before proceeding to step 7.

7. Copy the following packages to the **/opt/ncs/packages/** directory and create symbolic links from **/var/opt/ncs/packages**.

   ```
   sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
   /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz

   sudo cp ncs-<version>-cisco-sr-te-cfp-internal-<version>.tar.gz
   /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-<version>.tar.gz

   sudo cp ncs-<version>-sr-te-multi-vendors-<version>.tar.gz
   /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-<version>.tar.gz

   sudo cp ncs-<version>-cisco-iosxr-<version>.tar.gz /opt/ncs/packages/ncs-
   <version>-cisco-iosxr-<version>.tar.gz

   sudo cp ncs-<version>-cisco-ios-<version>.tar.gz /opt/ncs/packages/ncs-
   <version>-cisco-ios-<version>.tar.gz

   sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
   /opt/ncs/packages/ncs-<version>-custom-template-utils-<version>.tar.gz
   ```

```
sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-core-fp-common-<version>.tar.gz
sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
sudo cp ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz
sudo cp ncs-<version>-lsa-utils-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-lsa-utils-<version>.tar.gz
sudo cp ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-l2vpn-fp-internal-<version>.tar.gz
sudo cp ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz
sudo cp ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz
sudo cp ncs-<version>-l3vpn-multi-vendors-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-<version>.tar.gz
sudo cp ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz
sudo cp ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
<version>.tar.gz
sudo cp ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz
sudo cp ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-
common-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-
internal-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-<version>.tar.gz
```

8. Restart NSO with package-reload.

   ```
   $ sudo /etc/init.d/ncs restart-with-package-reload
   ```

   T-SDN FP Bundle is now installed on the lower-level node.

Performing Post Installation Tasks on the Lower-Level Nodes

Do the following:

1. Load the following bootstrap data.

   ```
   cd tsdn-<version>-nso-<version>/LSA/RFS/bootstrap-data

   ncs_cli -u <user>

   configure
   ```

```
unhide debug
load merge bootstrap-autopopulate-dispatch.xml
load merge commit-queue-settings.xml
load merge L2VPN-internal-plan-monitor.xml
load merge L2VPN-status-codes.xml
load merge L3VPN-internal-plan-monitor.xml
load merge L3VPN-status-codes.xml
load merge PM-internal-plan-monitor-example.xml
load merge PM-status-codes-example.xml
load merge rfs-custom-template-settings.xml
load merge rfs-dispatch-map-settings.xml
load merge RSVP-TE-internal-plan-monitor-example.xml
load merge RSVP-TE-status-codes-example.xml
load merge SR-internal-plan-monitor.xml
load merge SR-status-codes.xml
commit
```

2. Set NACM rules. In the following example, the ncs-admin user ID is admin.

```
% set nacm groups group ncsadmin user-name admin
% commit
Commit complete.
```

3. Configure the local **ncsadmin** user as a **cfp-local-user** for the CFP to identify the user to push the configurations.

```
% configure
% set cfp-local-user admin
% commit
```

4. Set the public key for ssh algorithms.

```
# Global settings method
------------------------
% show devices global-settings ssh-algorithms public-key
public-key [ ssh-ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521 rsa-sha2-512 rsa-sha2-256 ];
% set devices global-settings ssh-algorithms public-key [ ssh-ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521 rsa-sha2-512 rsa-sha2-256 ssh-rsa ]
% commit

% show device global-settings ssh-algorithms public-key
public-key [ ssh-ed25519 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521 rsa-sha2-512 rsa-sha2-256 ssh-rsa ];
```

```
# Device-specific method
------------------------
% show devices device PE1 ssh-algorithms public-key
No entries found.

% set devices device <DEVICE_NAME> ssh-algorithms public-key [ ssh-rsa ]
% commit

% show device device <DEVICE_NAME> ssh-algorithms public-key
public-key [ ssh-rsa ];
```

5. Add the global settings for timeout.

```
configure
set devices global-settings connect-timeout 300
set devices global-settings read-timeout 300
set devices global-settings write-timeout 300
```

Verifying the Installation on the Lower-Level Nodes

Do the following:

1. Verify the packages are installed and their status is UP.

```
admin@ncs> show packages package oper-status | tab
```

Note: If no Netconf NEDs are loaded, the package oper-status displays warning messages. Ignore these messages.

2. Verify the package information.

```
admin@ncs> show packages package package-version | select build-info ncs
version | select build-info file | select build-info package sha1 | select
oper-status error-info | select oper-status up | tab
```

3. Verify bootstrap configuration.

```
% show devices global-settings commit-queue
enabled-by-default false;
async;
atomic             false;
retry-attempts     0;
retry-timeout      30;
error-option       stop-on-error;
% show status-code-cfp
status-code-cfp L2VPN;
status-code-cfp L3VPN;
status-code-cfp RSVP-TE;
status-code-cfp SR;
```

```
% show rfs-monitor-path
rfs-monitor-path /cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-
internal-local-site/cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-plan;
rfs-monitor-path /cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-
internal-remote-site/cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-
plan;
rfs-monitor-path /cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-internal-
site/cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-plan;
rfs-monitor-path /cisco-flat-L3vpn-fp-internal:flat-L3vpn-internal/cisco-
flat-L3vpn-fp-internal:flat-L3vpn-plan;
rfs-monitor-path /cisco-rsvp-te-fp:rsvp-te/tunnel-te-plan;
rfs-monitor-path /cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-odn-
internal:odn/cisco-sr-te-cfp-sr-odn-internal:odn-template-plan;
rfs-monitor-path /cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-policies-
internal:policies/cisco-sr-te-cfp-sr-policies-internal:policy-plan;

% show auto-populate-dispatch-map
auto-populate-dispatch-map true;

% show ct-event-stream-enabled
ct-event-stream-enabled true;

% show status-codes | nomore
core-function-pack L2VPN {
    status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/flat_L2vpn_status_codes;
    status-code 301 {
        reason              "Device unreachable";
        category            device;
        severity            ERROR;
        recommended-actions "Check device connectivity from NSO and perform
recovery steps.";
    }
    status-code 302 {
        reason              "Device out of sync";
        category            device;
        severity            ERROR;
        recommended-actions "Check sync between device and NSO, and perform
recovery steps.";
    }
…
```

4. Verify the NACM rules.

```
% show nacm
read-default     deny;
write-default    deny;
exec-default     deny;
groups {
    group ncsadmin {
        user-name [ admin private ];
    }
    group ncsoper {
        user-name [ public ];
    }
}
...
```

5. Verify cfp-local-user.

```
% show cfp-local-user
cfp-local-user admin;
```

6. Verify lsa role is set for the lower-level node.

```
% show lsa role
role lower-layer;
```

Installing T-SDN FP Bundle on the Upper-Level Node

This section discusses the required packages and configurations required to install the FP bundle on the upper-level node in the LSA model.

Package Categories and Packages – Upper-Level Node

The following table shows the package categories and the packages extracted on the upper-level node.

Note: The cs-sr-te-cfp package in SR-TE CFP-IOSXR CLI is supported only on IOSXR CLI 7.46 NED, IOSXR CLI 7.52, IOSXR NC 7.8 (or later) NED.

CFS Node Packages	
Package Category	**Packages**
T-SDN FP Bundle packages	ncs-6.1.4-cisco-sr-te-cfp-6.0.0.tar.gz ncs-6.1.4-cisco-cs-sr-te-cfp-6.0.0.tar.gz ncs-6.1.4-ietf-l2vpn-nm-6.0.0.tar.gz ncs-6.1.4-ietf-l3vpn-nm-6.0.0.tar.gz ncs-6.1.3-cisco-cfp-jwt-auth-1.0.0.tar.gz

T-SDN FP Bundle common packages	ncs-6.1-core-fp-plan-notif-generator-1.0.10.tar.gz ncs-6.1-custom-template-utils-2.0.13.tar.gz ncs-6.1-core-fp-common-1.33.0.tar.gz ncs-6.1-lsa-utils-1.0.4.tar.gz ncs-6.1.4-cisco-tsdn-core-fp-common-6.0.0.tar.gz ncs-6.1.3-resource-manager-4.2.0.tar.gz
LSA NED packages	ncs-6.1.4-cisco-L2vpn-fp-internal-ned-6.0.0.tar.gz ncs-6.1.4-cisco-L3vpn-fp-internal-ned-6.0.0.tar.gz ncs-6.1.4-cisco-pm-fp-internal-ned-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-cisco-rsvp-te-fp-internal-ned-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-lsa-utils-ned-1.0.tar.gz ncs-6.1.4-custom-template-utils-ned-1.0.tar.gz ncs-6.1.4-core-fp-common-ned-1.0.tar.gz ncs-6.1.4-cisco-sr-te-cfp-internal-ned-6.0.0.tar.gz ncs-6.1.4-cisco-nso-nc-6.1.tar.gz
Example packages	ncs-6.1.4-cisco-aa-service-assurance-6.0.0.tar.gz ncs-6.1.4-cisco-pm-fp-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-ietf-network-slice-service-EXAMPLE-6.0.0.tar.gz ncs-6.1.4-ietf-te-fp-EXAMPLE-6.0.0.tar.gz

Modifying the NCS Configuration File on the Upper-Level Node

Back up the **/etc/ncs/ncs.conf** file and then modify the file for the upper-level node. Use the backup file to restore the configurations while uninstalling the CFPs.

For more information about the ncs.conf file, see the **nso_man-<*version*>.pdf** documentation in **volume5**.

To modify the NCS configuration file on the upper-level node:

1. It is optional to configure the SSH port for CLI, webui, and netconf northbound parameters. You can choose to either enable or disable the SSH port configuration, as required, for these parameters. By default, the SSH port configuration for these parameters is disabled. For more information on these parameters, see the ***NSO documentation***.

   The following shows how to enable the SSH port configuration, if required. Provide the port numbers as per your requirement.

   SSH port for CLI

   ```
   <cli>
     <enabled>true</enabled>
     <!-- Use the builtin SSH server -->
     <ssh>
            <enabled>true</enabled>
            <ip>0.0.0.0</ip>
            <port>${North_Bound_CLI_SSH_Port}</port>
   ```

```
          <extra-listen>
          <ip>::</ip>
          <port>${North_Bound_CLI_SSH_Port}</port>
          </extra-listen>

  </ssh>
```

webui

You can enable webui in either TCP or SSL.

```
<webui>
<enabled>true</enabled>
  <transport>
    <tcp>
       <enabled>true</enabled>
        <ip>0.0.0.0</ip>
        <port>${North_Bound_Web_UI_Port}</port>
    </tcp>

       <ssl>
              <enabled>true</enabled>
          <ip>0.0.0.0</ip>
          <port>${SSL_port}</port>
          <key-file>${NCS_CONFIG_DIR}/ssl/cert/host.key</key-file>
          <cert-file>${NCS_CONFIG_DIR}/ssl/cert/host.cert</cert-file>
       </ssl>
     </transport>
```

netconf northbound

```
<netconf-north-bound>
    <enabled>true</enabled>
        <transport>
<ssh>
<enabled>true</enabled>
<ip>0.0.0.0</ip>
<port>${Netconf_North_Bound_port}</port>
</ssh>
```

2. Add the stream service-state-changes as follows.

```
<notifications>
<event-streams>
  <stream>
    <name>service-state-changes</name>
    <description> Service state changes according to
     tailf-ncs-plan.yang and tailf-ncs-services.yang</description>
    <replay-support> true </replay-support >
    <builtin -replay-store>
      <enabled> true </enabled>
      <dir>${NCS_RUN_DIR}/state</dir>
      <max-size>S10M</max-size>
      <max-files>50</max-files>
    </builtin-replay-store>
  </stream>
```

3. If AA is installed, add AA notification streams to generate AA configuration change notifications.

```
<stream>
   <name>service-aa-changes</name>
   <description>Notifications relating to the service aa configuration
change</description>
   <replay-support> true </replay-support>
   <builtin-replay-store>
     <enabled> true </enabled>
     <dir>${NCS_RUN_DIR}/state</dir>
     <max-size>S10M</max-size >
     <max-files>50</max-files >
   </builtin-replay-store>
</stream>
```

4. If you have installed the jwt-auth package, enable the package authentication under <aaa>.

```
<aaa>
       <package-authentication>
         <enabled>true</enabled>
         <packages>
           <package>cisco-cfp-jwt-auth</package>
         </packages>
       </package-authentication>
     </aaa>
```

Note: If you do not want to use this authentication feature, set the package authentication to **false.**

5. Append the **<hide-group>** information to the file.

```
<hide-group>
    <name>tsdn</name>
</hide-group>
  <hide-group>
```

```
        <name>debug</name>
      </hide-group>
    <hide-group>
       <name>fastmap-private</name>
       </hide-group>
       <hide-group>
           <name>lsa</name>
       </hide-group>
```

6. Add or update **<start-timeout>** parameter under <python-vm>.

```
<python-vm>
   <start-timeout>PT300S</start-timeout>
</python-vm>
```

Installing FP Bundle on the Upper-Level Node

To install T-SDN FP Bundle on the upper-level node:

1. Be sure to have installed T-SDN FP Bundle on the lower-level node. This is because you must add the lower-level node devices as devices on the upper-level node, configure the lower-level node, and sync it to the upper-level node. For more information, see **Installing T-SDN FP Bundle on the Lower-Level Nodes**.

2. Make sure to have performed the tasks mentioned in section **Modifying the NCS Configuration File on the Upper-Level Node**.

3. Log in to the host machine as a sudo user, who is also part of the ncsadmin user group.

4. Obtain and download the required **tsdn-<*version*>-nso-<*version*>.signed.bin** file from Cisco website and copy it to the upper-level node.

5. Extract the content of the bin file to the current directory. If the folder already exists, back up the existing folder.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin
   ```

   This verifies the authenticity of the product. However, if you encounter any network connectivity issues, run the following command to skip this verification.

   ```
   $ sh tsdn-<version>-nso-<version>.signed.bin --skip-verification
   ```

6. Untar the installer tar.gz file to the current directory to extract the T-SDN FP Bundle packages. If the folder already exists, be sure to create a backup of the existing folder.

   ```
   $tar -xf tsdn-<version>-nso-<version>.tar.gz
   ```

7. Go to the upper-level node packages directory and change the current directory as follows:

   ```
   $ cd tsdn-<version>-nso-<version>/LSA/CFS/packages
   ```

Note: If you are installing DLM, TM-TC, and/or NCA user-specific packages, download the packages and create the symbolic links for these packages before proceeding to step 8.

8. Copy the T-SDN FP Bundle packages to the **/opt/ncs/packages/** directory and create symbolic links from **/var/opt/ncs/packages**.

```
sudo cp ncs-<version>-lsa-utils-ned-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-lsa-utils-ned-<version>.tar.gz

sudo cp ncs-<version>-custom-template-utils-ned-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-custom-template-utils-ned-<version>.tar.gz

sudo cp ncs-<version>-core-fp-common-ned-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-core-fp-common-ned-<version>.tar.gz

sudo cp ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz

sudo cp ncs-<version>-cisco-sr-te-cfp-internal-ned-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-ned-<version>.tar.gz

sudo cp ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-cisco-sr-te-cfp-<version>.tar.gz

sudo cp ncs-<version>-cisco-cs-sr-te-cfp-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-<version>.tar.gz

sudo cp ncs-<version>-cisco-nso-nc-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-cisco-nso-nc-<version>.tar.gz

sudo cp ncs-<version>-core-fp-common-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-core-fp-common-<version>.tar.gz

sudo cp ncs-<version>-lsa-utils-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-lsa-utils-<version>.tar.gz

sudo cp ncs-<version>-custom-template-utils-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-custom-template-utils-<version>.tar.gz

sudo cp ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz

sudo cp ncs-<version>-resource-manager-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz

sudo cp ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-ietf-l2vpn-nm-<version>.tar.gz

sudo cp ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /opt/ncs/packages/ncs-
<version>-ietf-l3vpn-nm-<version>.tar.gz

sudo cp ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz

sudo cp ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz

sudo cp ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz

sudo cp ncs-<version>-ietf-network-slice-service-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-ietf-network-slice-service-EXAMPLE-
<version>.tar.gz
```

```
sudo cp ncs-<version>-cisco-rsvp-te-fp-internal-ned-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-internal-ned-EXAMPLE-
<version>.tar.gz
sudo cp ncs-<version>-cisco-L3vpn-fp-internal-ned-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-ned-<version>.tar.gz
sudo cp ncs-<version>-cisco-L2vpn-fp-internal-ned-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-ned-EXAMPLE-
<version>.tar.gz


sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-ned-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-lsa-utils-ned-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-ned-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-utils-
ned-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-ned-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-common-ned-
<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-
common-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-ned-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-
internal-ned-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-
<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-nso-nc-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-cisco-nso-nc-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-core-fp-plan-notif-
generator-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz
```

```
sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-internal-ned-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-internal-ned-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-ned-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-flat-L3vpn-fp-internal-ned-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-ned-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-flat-L2vpn-fp-internal-ned-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz

sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-network-slice-service-EXAMPLE-<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-network-slice-service-EXAMPLE-<version>.tar.gz
```

9. Restart NSO with package-reload.

   ```
   $ sudo /etc/init.d/ncs restart-with-package-reload
   ```

   T-SDN FP Bundle is now installed on the upper-level node.

Performing Post Installation Tasks on the Upper-Level Node

Do the following:

1. Load the following bootstrap data. Make sure the LSA cluster is up and the Netconf notifications is running.

   ```
   cd tsdn-<version>-nso-<version>/LSA/CFS/bootstrap-data

   ncs_cli -u <user>

   configure

   unhide tsdn

   load merge IETF-L2NM-AA-notification-settings.xml

   load merge IETF-L3NM-AA-notification-settings.xml

   commit

   unhide debug

   load merge CS-SR-status-codes.xml

   load merge SR-status-codes.xml
   ```

```
load merge RSVP-TE-status-codes-example.xml
load merge L3VPN-status-codes.xml
load merge L2VPN-status-codes.xml
load merge IETF-TE-status-codes-example.xml
load merge IETF-L3NM-status-codes.xml
load merge IETF-L2NM-status-codes.xml
load merge IETF-NSS-status-codes-example.xml
load merge PM-status-codes-example.xml
commit

load merge CS-SR-internal-plan-kicker.xml
load merge 1_SR-cfp-configuration-kicker.xml
load merge 1_IETF-TE-cfp-configuration-kicker-example.xml
load merge 1_IETF-L2NM-cfp-configuration-kicker.xml
load merge IETF-L2NM-route-policy-kicker.xml
load merge 1_IETF-L3NM-cfp-configuration-kicker.xml
load merge IETF-L3NM-route-policy-kicker.xml
load merge IETF-NSS-internal-plan-kicker-example.xml
load merge 1_PM-cfp-configuration-kicker-example.xml
load merge PM-profiles-kicker-example.xml
load merge cfs-dispatch-map-settings.xml
commit

load merge CS-SR-plan-notification-settings.xml
load merge SR-plan-notification-settings.xml
load merge IETF-TE-plan-notification-settings-example.xml
load merge IETF-L3NM-plan-notification-settings.xml
load merge IETF-L2NM-plan-notification-settings.xml
load merge IETF-NSS-plan-notification-settings-example.xml
load merge PM-plan-notification-settings-example.xml
commit

load merge 2_SR-multi-vendor-iosxe-cli.xml
load merge 2_IETF-TE-multi-vendor-iosxe-cli-example.xml
load merge 2_IETF-L2NM-multi-vendor-iosxe-cli.xml
load merge 2_IETF-L3NM-multi-vendor-iosxe-cli.xml
load merge 2_PM-multi-vendor-iosxe-cli-example.xml
commit
```

```
load merge commit-queue-settings.xml
load merge IETF-L3NM-resource-pools.xml
load merge xr-color-resource-pool.xml
load merge xr-bidirectional-association-id-resource-pool.xml
load merge xr-disjoint-group-id-resource-pool.xml
load merge xr-sman-id-resource-pool.xml
load merge IETF-NSS-resource-pools-example.xml
commit
## Required for UI interface mapping ##
=======================================
load merge interface-mapping-xe-cli-6.86.xml
load merge interface-mapping-xr-cli-7.49.xml
commit

## LSA netconf notification kicker ##

## For TSDN bundle image ##
load merge cfs-netconf-notification-kicker.xml
commit

## For TSDN CLI image. ##
cd <TSDN_build>/netconf_data
load merge cfs-netconf-notification-kicker.xml
commit
```

2. Set NACM rules. In the following example, the ncs-admin user ID is admin.

```
% set nacm groups group ncsadmin user-name admin
% commit
Commit complete.
```

3. Configure the local **ncsadmin** user as a **cfp-local-user** for the CFP to identify the user to push the configurations.

```
% configure
% set cfp-local-user admin
% commit
```

4. Configure the authgroups for the lower-level nodes. Provide the password for the user when prompted.

```
% configure
```

```
admin@ncs% set devices authgroups group cnc-rfs-auth default-map remote-name
admin remote-password

(<AES256 encrypted string>): *********

[ok]

admin@ncs% commit

Commit complete.
```

5. Onboard the lower-level node as a device to the upper-level node device tree.

```
% configure

set devices device rfs-1 address 10.10.10.10 port 2022 authgroup cnc-rfs-auth
out-of-sync-commit-behaviour accept

set devices device rfs-1 device-type netconf ned-id cisco-nso-nc-5.7

set devices device rfs-1 use-lsa

set devices device rfs-1 state admin-state unlocked

set devices device rfs-1 connect-timeout 300 read-timeout 300 write-timeout
300 connect-retries attempts 2 timeout 300

admin@ncs% commit

Commit complete
```

6. Run the following command to ensure receive notification setting.

```
set devices device rfs-1 netconf-notifications received-notifications max-
size 500000

admin@ncs% commit

Commit complete
```

Note: The timestamp on both the RFS node and the CFS node must be the same. It is recommended to use the same NTP server for time synchronization.

7. Configure the Netconf notification subscription for each lower-level node. You must append the lower-level node name for all notification streams. In the following example, the user admin is the local user, who is a part of the ncsadmin group.

```
set devices device rfs-1 netconf-notifications subscription rfs-cisco-custom-
template-events stream custom-template-events local-user admin store-in-cdb
true

set devices device rfs-1 netconf-notifications subscription rfs-dispatch-map-
update stream dispatch-map-update local-user admin store-in-cdb true

set devices device rfs-1 netconf-notifications subscription rfs-kicker-events
stream kicker-events local-user admin store-in-cdb false
```

Note: Ignore any notification alarms you may receive. The following steps help to resolve the issue.

8. Configure cluster. Be sure to use the same authgroup name that you used while onboarding the lower-level node. Provide the IP address and the port number of the lower-level node.

The lower-level node device names must be unique. Cluster fetch ssh host keys is also essential for the cluster to form and for its status to be up.

```
set cluster remote-node rfs-1 address 10.10.10.10 port 2022 authgroup cnc-rfs-auth username cisco

set cluster authgroup cnc-rfs-auth default-map remote-name <name> remote-password <password>
set cluster device-notifications disabled
set cluster commit-queue enabled
set cluster global-settings timeouts connect-timeout 300
commit

admin@ncs% request cluster remote-node rfs-1 ssh fetch-host-keys
result updated
fingerprint {
    algorithm ssh-ed25519
    value 83:a0:c2:62:85:dd:ee:bd:12:4f:a1:23:ae:47:d7:ca
}

admin@ncs% run show cluster
RECEIVED
NAME  NAME                  STATUS  LAST EVENT                        NOTIFICATIONS
-----------------------------------------------------------------------------------
rfs-1 device-notifications  up      0000-01-01T00:00:00-00:00         0
      ncs-events            up      2022-01-27T10:48:25.148393+00:00  630

REMOTE                                  LOCAL  REMOTE
NODE    ADDRESS         PORT  CHANNELS  USER   USER    STATUS  TRACE
-------------------------------------------------------------------------
rfs-1   10.10.10.10     2022  -         cisco  abc   up      disabled
```

9. Sync-from the lower-level device node and verify the device tree.

```
admin@ncs% request devices fetch-ssh-host-keys
fetch-result {
    device rfs-1
    result updated
    fingerprint {
        algorithm ssh-ed25519
        value ed:7b:1c:e4:77:80:ab:68:3b:17:40:69:68:9e:56:8d
    }
  }
  [ok]
```

```
admin@ncs% request devices sync-from
sync-result {
    device rfs-1
    result true
}
[ok]
admin@ncs% run show devices list

NAME   ADDRESS          DESCRIPTION  NED ID            ADMIN STATE
-----------------------------------------------------------------
rfs-1  10.10.10.10       -            cisco-nso-nc-6.1  unlocked
[ok]
```

10. Sync the dispatch-map and verify if the onboarded devices on the RFS node can be seen from the CFS node.

```
% request devices lsa dispatch-map sync
success true
detail Dispatch Map Synced Successfully

% show device lsa dispatch-map
device rfs-1 {
    ned-id cisco-nso-nc-6.1:cisco-nso-nc-6.1;
}
```

11. Verify the Netconf notification subscription is running for the added lower-level node device. This shows the upper-level and the lower-level nodes are connected, and communication is up and running.

```
admin@ncs% run show devices device rfs-1 netconf-notifications subscription

                                            FAILURE  ERROR
    NAME                             STATUS   REASON   INFO
    -----------------------------------------------------------
    rfs-cisco-custom-template-events running  -        -
    rfs-dispatch-map-update          running  -        -
    rfs-kicker-events                running  -        -
```

Once the installation is complete, onboard the devices only from the lower-level node.

Verifying the Installation on the Upper-Level Node

Do the following:

1. Verify the packages are installed and their status is UP.

```
admin@ncs> show packages package oper-status | tab
```

2. Verify the package information.

```
admin@ncs> show packages package package-version | select build-info ncs version
| select build-info file | select build-info package sha1 | select oper-status
error-info | select oper-status up | tab
```

3. Verify the NACM rules.

```
% show nacm
read-default     deny;
write-default    deny;
exec-default     deny;
groups {
    group ncsadmin {
        user-name [ admin private ];
    }
    group ncsoper {
        user-name [ public ];
    }
}
...
```

4. Verify cfp-local-user.

```
% show cfp-local-user
  cfp-local-user admin;
```

5. Verify the lsa role is set for the upper-level node.

```
% show lsa role
role upper-layer;
```

Uninstalling T-SDN FP Bundle in the LSA Model

This chapter explains the uninstallation procedure for the NSO T-SDN FP Bundle. To uninstall the T-SDN FP Bundle, you must first uninstall the Bundle from the upper-level node and then from the lower-level node since the lower-level node is added as a device to the upper-level node device tree.

Uninstalling T-SDN FP Bundle removes the CFP and the associated packages from the system. Only a user who has sudo privileges and is part of ncsadmin user group can perform the uninstallation process.

You must delete all the related services, devices from NSO, and any day-1 packages before performing the uninstallation procedure. If a cluster has only one lower-level node, delete the cluster.

Note: Do not delete any common packages and common kickers files if you have installed other CFPs.

Uninstalling T-SDN FP Bundle from the Upper-Level Node

To uninstall T-SDN FP Bundle from the upper-level node:

1. Delete kicker notifications.

```
% unhide debug
% delete kickers notification-kicker remote-dispatch-map-update-notifications
% delete kickers notification-kicker rfs-custom-template-change-notification
% delete kickers notification-kicker tsdn-kicker-events-notifications
% commit
Commit complete.
```

2. If you have other packages installed, skip to step 5.

3. Delete the cluster. If the cluster has only one lower-level node, delete the complete cluster.

```
% delete cluster remote-node rfs-1
% commit
Commit complete.
% delete cluster authgroup cluster-rfs-auth
% delete cluster device-notifications enabled
% commit
Commit complete.
```

4. Remove lower-node devices from the upper-level node device tree.

```
% delete devices device rfs-1
% commit
Commit complete.
```

5. Stop NCS by using the `ncs --stop` command.

6. Restore the backup of the ncs.conf file that you created during installation. For more information, see **Modifying the NCS Configuration File on the Upper-Level Node**.

7. Unlink and remove the packages in the **/var/opt/ncs/packages** directory and delete the packages from the **/opt/ncs/packages/** directory.

```
sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-aa-service-assurance-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-cs-sr-te-cfp-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-ned-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-ned-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-nso-nc-6.1.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-pm-fp-EXAMPLE-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-ned-EXAMPLE-<version>.tar.gz
sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-internal-ned-EXAMPLE-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-ned-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-common-ned-1.0.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-custom-template-utils-ned-1.0.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-ietf-network-slice-service-EXAMPLE-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-ietf-te-fp-EXAMPLE-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-lsa-utils-ned-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-custom-template-utils-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
```

8. Restart NSO with the package-reload option.

```
$ sudo /etc/init.d/ncs restart-with-package-reload
  Restarting ncs (via systemctl):
    [  OK  ]
```

T-SDN FP Bundle is now uninstalled from the upper-level node.

Uninstalling T-SDN FP Bundle from the Lower-Level Node

Delete all the related services. Perform the procedure described in this section for each lower-level node in the cluster.

To uninstall T-SDN FP Bundle from the lower-level node:

1. Delete devices from the lower-level node device tree.

```
% delete devices
% commit
Commit complete.
```

2. Stop NSO by using the `ncs --stop` command.

3. Restore the backup of the ncs.conf file that you created during installation. For more information, see **Modifying the NCS Configuration File on the Lower-Level Nodes**.

4. Unlink the packages from the **/var/opt/ncs/packages** directory and delete the packages from the **/opt/ncs/packages/** directory.

```
sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-rsvp-te-fp-EXAMPLE-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-sr-te-cfp-internal-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-tsdn-core-fp-common-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-l2vpn-multi-vendors-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-l3vpn-multi-vendors-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-L2vpn-fp-internal-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-L3vpn-fp-internal-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-pm-fp-internal-EXAMPLE-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-pm-multi-vendors-EXAMPLE-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-custom-template-utils-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-common-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-delete-tag-service-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-core-fp-plan-notif-generator-
<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-lsa-utils-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-iosxr-<version>.tar.gz

sudo rm -rf /opt/ncs/packages/ncs-<version>-cisco-ios-<version>.tar.gz
```

5. Restart NSO with the package-reload option.

```
$ sudo /etc/init.d/ncs restart-with-package-reload
  Restarting ncs (via systemctl):
          [  OK  ]
```

Upgrading the NSO T-SDN FP Bundle CFP

This section includes information about how to upgrade the T-SDN FP Bundle CFPs from version 5.x to version 6.0.0. It is recommended to back up your environment before performing the upgrade.

NSO must be up and running (on both the nodes in the LSA model) before beginning the upgrade procedure.

Note: T-SDN FP Bundle v6.0.0 supports two versions of IOSXR CLI and IOSXE CLI NEDs. The IOSXR CLI v7.46 and IOSXE CLI v6.86 NEDs support only T-SDN FP Bundle v5.0.0 features on T-SDN FP v6.0.0. You can choose to upgrade these NEDs to IOSXR CLI v7.52 and IOSXE CLI v6.100, as required, to avail the latest features. It is recommended to use the latest NED versions while onboarding new devices to NSO.

Before You Begin

Do the following before beginning the upgrade process. In the LSA model, perform these tasks on both the CFS and RFS nodes.

1. Obtain and place the **NSO 6.1.4 installer bin** file and the **tsdn-6.0.0-nso-6.1.4-signed.bin** file in **/home/user/** directory (replace user according to your setup).
2. Back up the current NCS packages and the current **etc/ncs/ncs.conf** file. For more information on how to back up and restore, see the ***NSO Administration Guide***.

   Note: Make sure to copy the backup tar file outside the **/var/opt/ncs** directory to prevent any loss of data.

   ```
   $ mkdir /home/admin/ncsBackup
   $ cp /opt/ncs/packages/* /home/admin/ncsBackup/
   $ echo "Backup current NCS"
   $ sudo /opt/ncs/current/bin/ncs-backup --non-interactive
   ```

3. Remove the old status-codes.

   ```
   admin@ncs% unhide debug
   admin@ncs% delete status-code-cfp
   admin@ncs% delete status-codes
   admin@ncs% commit
   ```

Upgrading NSO T-SDN FP Bundle CFP on a Single NSO Instance

Perform the following tasks to upgrade the T-SDN FP Bundle CFPs to v6.0.0.

Do the following:

1. Make sure to complete the tasks described in section **Before You Begin**.
2. Stop NSO.

   ```
   sudo /etc/init.d/ncs stop
   ```

3. Upgrade NSO to version 6.1.4 by performing system installation of NSO. For more information about how to upgrade NSO, see the ***NSO Installation Guide***.
4. Copy the configuration from **ncs.conf .install** file to the existing **etc/ncs/ncs.conf** file and edit the **ncs.conf** file to add/append configurations as described in section **Editing the NCS Configuration File on a Single NSO Instance**.
5. Remove the old packages in the **/opt/ncs/packages** directory. Do not remove any old NEDs you want to migrate to the new version.

   ```
   cd /opt/ncs/packages/
   sudo rm *.tar.gz
   ```

6. Remove the old symbolic links for the packages in **/var/opt/ncs/packages** directory.

   ```
   cd /var/opt/ncs/packages/
   sudo rm -f *
   ```

7. Change directory to **tsdn-6.0.0-nso-6.1.4**.
8. Extract the content of the T-SDN FP Bundle bin file to the current directory.

   ```
   $ sh tsdn-6.0.0-nso-6.1.4.signed.bin
   ```

   This verifies the authenticity of the product. However, if you encounter any network connectivity issues, run the following command to skip this verification.

   ```
   $ sh tsdn-6.0.0-nso-6.1.4.signed.bin --skip-verification
   ```

9. Untar the T-SDN FP Bundle **tar.gz** file to the current directory. If the folder already exists, be sure to create a backup of the existing folder.

   ```
   $ tar -xvf TSDN-6.0.0-nso-6.1.4.tar.gz
   ```

10. Copy the T-SDN FP Bundle 6.0.0 packages from the TSDN tar file to the **/var/opt/ncs/packages** directory.

    ```
    sudo cp *.tar.gz /opt/ncs/packages/
    ```

11. Create soft links for all the packages.

    ```
    cd /var/opt/ncs/packages
    sudo ln -s /opt/ncs/packages/*.tar.gz
    ```

12. Set the **ignore-initial-validation** flag and restart ncs with the package-reload option as follows. If you do not set this flag, the upgrade process fails with errors.

    a. Add the ignore-initial-validation flag in the start ( ) function.

    ```
    sudo vi /etc/init.d/ncs
    …
    …
    start() {
            echo -n $”Starting $prog: ‟
            . $ncsdir/ncsrc
            NCS_CONFIG_DIR=${confdir}
            NCS_RUN_DIR=${rundir}
            NCS_LOG_DIR=${logdir}
            export NCS_CONFIG_DIR NCS_RUN_DIR NCS_LOG_DIR
            $ncs -cd ${rundir}  ${heart} ${conf}
            RETVAL=$?
            Echo
            # [ $RETVAL = 0 ] && touch /var/lock/subsys/ncs
            return $RETVAL
    }
    …
    …
    Change it to
    …
    …
    start() {
            echo -n $”Starting $prog: ‟
            . $ncsdir/ncsrc
            NCS_CONFIG_DIR=${confdir}
            NCS_RUN_DIR=${rundir}
            NCS_LOG_DIR=${logdir}
            export NCS_CONFIG_DIR NCS_RUN_DIR NCS_LOG_DIR
            $ncs -cd ${rundir}  ${heart} ${conf} --ignore-initial-
    validation
            RETVAL=$?
            Echo
            # [ $RETVAL = 0 ] && touch /var/lock/subsys/ncs
            return $RETVAL
    }
    …
    …
    ```

    b. Reload systemd.

    ```
    sudo systemctl daemon-reload
    ```

    c. Restart ncs with package-reload option.

    ```
    sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs start
    ```

    d. Once NSO has started up, revert the start ( ) function script to its original content to remove the **--ignore-initial-validation** flag.

13. Reload systemd.

```
sudo systemctl daemon-reload
```

14. Verify the status of the packages.

```
admin@ncs> show packages package oper-status
```

15. Migrate the IOSXR CLI NED and IOSXE CLI NED.

```
admin@ncs% request devices device <Device_name> migrate new-ned-id cisco-
iosxr-cli-<new_NED_version> no-networking

admin@ncs% request devices device <Device_name> migrate new-ned-id cisco-ios-
cli-<new_NED_version> no-networking
```

16. Sync dispatch map to update with new NED ID.

```
admin@ncs% request devices lsa dispatch-map sync

success true

detail Dispatch Map Synced Successfully

[ok]
```

17. Sync the device to pull the new NED capabilities. For example, the IOSXE CLI NED 6.86 may have new capabilities over the earlier version of IOSXE CLI NED. These new capabilities may introduce new NSO device configuration from Day0 device configuration. Therefore, you must pull the new changes to bring the device back in-sync with NSO.

Note: When syncing the device configuration northbound to NSO, verify the new configuration is a Day0 configuration only with a dry-run.

```
admin@ncs% request devices device XECLI-0 sync-from dry-run
cli  config {
         interface {
             GigabitEthernet 1 {
                 ip {
                     dhcp {
                         client {
                             client-id {
     +                           ascii cisco-02bc.9833.b2f9-Gi1;
                             }
                         }
                     }
                 }
             }
         }
         router {
             isis-container {
                 isis 1 {
                     router-id {
     +                   Loopback 0;
                     }
                 }
             }
         }
     }
```

```
admin@ncs% request devices sync-from device [ XECLI-0 XECLI-1 ]
sync-result {
    device XECLI-0
    result true
}
sync-result {
    device XECLI-1
    result true
}
```

18. Clean up the old NED packages for the migrated device and reload the packages to remove the old NEDs from NSO.

```
$ rm /var/opt/ncs/packages/ncs-6.1-cisco-ios-6.86.3.tar.gz
$ rm /var/opt/ncs/packages/ncs-6.1-cisco-iosxr-7.46.3.tar.gz
```

19. Force reload NSO to remove the old NED packages from the NSO running instance.

```
admin@ncs> request packages reload force
```

20. Configure the bootstrap data for the new version, plan-notifications, status-codes, and kickers. For more information, see **Performing Post Installation Tasks for SR-TE CFP.**

21. Sync dispatch-map and verify the map is populated with the NED ID.

```
admin@ncs% request devices lsa dispatch-map sync
success true
detail Dispatch Map Synced Successfully
admin@ncs% show devices lsa dispatch-map
dispatch-map <Device_name> {
   ned-id cisco-ios-cli-6.86:cisco-ios-cli-6.86;
…
…
}
```

22. Verify the device configuration and backpointers are correct.

```
admin@ncs% show devices device <Device_name> config | display service-meta-
data
/* Refcount: 3 */
 /* Backpointer: [ /cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-odn-
internal:odn/cisco-sr-te-cfp-sr-odn-internal:odn-template-plan[cisco-sr-te-
cfp-sr-odn-internal:name='SR-ODN-XR-CLI'][cisco-sr-te-cfp-sr-odn-
internal:head-end='PIOSXR-0']/cisco-sr-te-cfp-sr-odn-internal:plan/cisco-sr-
te-cfp-sr-odn-internal:component[cisco-sr-te-cfp-sr-odn-
internal:type='ncs:self'][cisco-sr-te-cfp-sr-odn-
internal:name='self']/cisco-sr-te-cfp-sr-odn-internal:state[cisco-sr-te-cfp-
sr-odn-internal:name='cisco-sr-te-cfp-sr-odn-nano-services:config-apply']
/cisco-sr-te-cfp-internal:sr-te/cisco-sr-te-cfp-sr-policies-
internal:policies/cisco-sr-te-cfp-sr-policies-internal:policy-plan[cisco-sr-
te-cfp-sr-policies-internal:name='SR-POLICY-XR-CLI'][cisco-sr-te-cfp-sr-
policies-internal:head-end='PIOSXR-0']/cisco-sr-te-cfp-sr-policies-
internal:plan/cisco-sr-te-cfp-sr-policies-internal:component[cisco-sr-te-
cfp-sr-policies-internal:type='ncs:self'][cisco-sr-te-cfp-sr-policies-
internal:name='self']/cisco-sr-te-cfp-sr-policies-internal:state[cisco-sr-
te-cfp-sr-policies-internal:name='cisco-sr-te-cfp-sr-policies-nano-
services:config-apply'] ] */
segment-routing {
   global-block {
       lower-bound 16000;
       upper-bound 23999;
   }
```

23. Display and verify the external plans for the ODN service and policy service are in the **reached** state.

    ```
    admin@ncs% run show cisco-sr-te-cfp:sr-te odn odn-template-plan
    <ODN_Service_Name> plan
    admin@ncs% run show cisco-sr-te-cfp:sr-te policies policy-plan
    <Policy_service_Name> plan
    ```

The Cisco NSO T-SDN FP Bundle upgrade is now complete.

Upgrading NSO T-SDN FP Bundle in the LSA Model

Perform the following tasks to upgrade the CFPs in T-SDN FP Bundle to v6.0.0 in the LSA model.

Do the following:

1. Make sure to complete the tasks described in section **Before You Begin** .
2. Stop NSO on both the CFS and RFS nodes.

   ```
   sudo /etc/init.d/ncs stop
   ```

3. Upgrade CFS node.

   a. Upgrade NSO to version 6.1.4 on the CFS node by performing a system installation of NSO. For more information about how to upgrade NSO, see the ***NSO Installation Guide***.

   b. Copy the configuration from **ncs.conf .install** file to the existing **etc/ncs/ncs.conf** file and edit the **ncs.conf** file to add/append the configuration as described in section **Modifying the NCS Configuration File on the Upper-Level Node**.

   c. Remove the old packages in the **/opt/ncs/packages** directory.

   ```
   cd /opt/ncs/packages/
   sudo rm *.tar.gz
   ```

   d. Remove the old symbolic links for the packages in **/var/opt/ncs/packages** directory.

   ```
   cd /var/opt/ncs/packages/
   sudo rm -f *
   ```

   e. Change directory to **tsdn-6.0.0-nso-6.1.4**.

   f. Extract the content of the T-SDN FP Bundle bin file to the current directory.

   ```
   $ sh tsdn-6.0.0-nso-6.1.4.signed.bin
   ```

   This verifies the authenticity of the product. However, if you encounter any network connectivity issues, run the following command to skip this verification.

   ```
   $ sh tsdn-6.0.0-nso-6.1.4.signed.bin --skip-verification
   ```

   g. Untar the T-SDN FP Bundle **tar.gz** file to the current directory. If the folder already exists, be sure to create a backup of the existing folder.

   ```
   $ tar -xvf TSDN-6.0.0-nso-6.1.4.tar.gz
   ```

   h. Copy the T-SDN FP Bundle 6.0.0 packages for the CFS node from the TSDN tar file to the **/var/opt/ncs/packages** directory.

   ```
   sudo cp *.tar.gz /opt/ncs/packages/
   ```

   i. Recompile the NEDs in the **/opt/ncs/packages/upgrade-ned** directory (such as the IOSXE CLI NED and any other customized NEDs in any devices) with NSO v6.1.4.

   ```
   user$ ncs --version
   ```

   ```
   6.1.4
   ```

```
user$ cd /var/opt/ncs/packages
user$ tar -xzf <OLD_NED_TARBALL>
user$ cd <OLD_NED_PACKAGE>/src
user$ make clean all
```

j. Copy the new NSO pre-compiled Netconf NED from the NSO root directory to the **/var/opt/ncs/packages** directory.

```
user$ cp -r <NSO_6.1.4_ROOT>/packages/lsa/cisco-nso-nc-6.1
/var/opt/ncs/packages
```

4. Upgrade RFS node.

    a. Upgrade NSO to version 6.1.4 on the RFS node by performing a system installation of NSO. For more information about how to upgrade NSO, see the ***NSO Installation Guide***.

    b. Copy the configuration from **ncs.conf .install** file to the existing **etc/ncs/ncs.conf** file and edit the **ncs.conf** file to add/append the configuration as described in section **Modifying the NCS Configuration File on the Lower-Level Nodes**.

    c. Remove the old packages in the **/opt/ncs/packages** directory. Do not remove the old NEDs you to want to migrate to the new version.

    ```
    cd /opt/ncs/packages/
    sudo rm *.tar.gz
    ```

    d. Remove the old symbolic links for the packages in **/var/opt/ncs/packages** directory.

    ```
    cd /var/opt/ncs/packages/
    sudo rm -f *
    ```

    e. Change directory to **tsdn-6.0.0- nso-6.1.4**.

    f. Extract the content of the T-SDN FP Bundle bin file to the current directory.

    ```
    $ sh tsdn-6.0.0-nso-6.1.4.signed.bin
    ```

    This verifies the authenticity of the product. However, if you encounter any network connectivity issues, run the following command to skip this verification.

    ```
    $ sh tsdn-6.0.0-nso-6.1.4.signed.bin --skip-verification
    ```

    g. Untar the T-SDN FP Bundle **tar.gz** file to the current directory. If the folder already exists, be sure to create a backup of the existing folder.

    ```
    $ tar -xvf TSDN-6.0.0-nso-6.1.4.tar.gz
    ```

    h. Copy the T-SDN FP Bundle 6.0.0 packages for the RFS node from the TSDN tar file to the **/var/opt/ncs/packages** directory.

    ```
    sudo cp *.tar.gz /opt/ncs/packages/
    ```

5. Create soft links for all the packages.

    ```
    cd /var/opt/ncs/packages
    sudo ln -s /opt/ncs/packages/*.tar.gz
    ```

6. Set the **ignore-initial-validation** flag. If you do not set this flag, the upgrade process fails with errors.

   a. Add the ignore-initial-validation flag in the start ( ) function.

   ```
   sudo vi /etc/init.d/ncs
   …
   …
   start() {
           echo -n $”Starting $prog: “
           . $ncsdir/ncsrc
           NCS_CONFIG_DIR=${confdir}
           NCS_RUN_DIR=${rundir}
           NCS_LOG_DIR=${logdir}
           export NCS_CONFIG_DIR NCS_RUN_DIR NCS_LOG_DIR
           $ncs -cd ${rundir}  ${heart} ${conf}
           RETVAL=$?
           Echo
           # [ $RETVAL = 0 ] && touch /var/lock/subsys/ncs
           return $RETVAL
   }
   …
   …
   Change it to
   …
   …
   start() {
           echo -n $”Starting $prog: “
           . $ncsdir/ncsrc
           NCS_CONFIG_DIR=${confdir}
           NCS_RUN_DIR=${rundir}
           NCS_LOG_DIR=${logdir}
           export NCS_CONFIG_DIR NCS_RUN_DIR NCS_LOG_DIR
           $ncs -cd ${rundir}  ${heart} ${conf} -ignore-initial-validation
           RETVAL=$?
           Echo
           # [ $RETVAL = 0 ] && touch /var/lock/subsys/ncs
           return $RETVAL
   }
   …
   …
   ```

   b. Reload systemd.

   ```
   sudo systemctl daemon-reload
   ```

   c. Restart ncs with package-reload option on both the CFS node and the RFS node.

   ```
   sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs start
   ```

   d. Once NSO has started up, revert the start ( ) function script to its original content.

7. Reload systemd.

```
sudo systemctl daemon-reload
```

8. Verify the status of the packages.

```
admin@ncs> show packages package oper-status
```

9. (*On CFS only*) Update SSH host keys.

```
user@ncs% request devices fetch-ssh-host-keys
```

10. Migrate NEDs first on the CFS node and then on the RFS node.

```
user@ncs% request devices device rfs-node-1 migrate new-ned-id cisco-nso-nc-
6.1 no-networking
```

```
user@ncs% request devices device PIOSXR-0 migrate new-ned-id cisco-iosxr-
cli-7.46 no-networking
```

11. Sync the device to pull the new NED capabilities. For example, the IOSXE CLI NED 6.86 may have new capabilities over the earlier IOSXE CLI NED version. These new capabilities may introduce new NSO device configuration from Day0 device configuration. Therefore, you must pull the new changes to bring the device back in-sync with NSO.

Note: When syncing the device configuration northbound to NSO, verify the new configuration is a Day0 configuration only with a dry-run.

```
user@ncs% request devices device <XR-device> sync-from dry-run
cli  config {
          call-home {
              profile CiscoTAC-1 {
                  destination {
                      transport-method {
                          email-disable {
                              email {
     +                            disable;
                              }
                          }
                      }
                  }
              }
          }
      }
```

```
user@ncs> request devices device <XE-device> sync-from dry-run
cli config {
          service {
              conf {
     +            pad false;
              }
          }
      }
```

```
user@ncs% request devices sync-from device [ <XR-device> <XE-device> ]
sync-result {
    device <XR-device>
    result true
}
sync-result {
    device <XE-device>
    result true      }
```

12. Clean up the old NED packages for the migrated device and reload the packages to remove the old NEDs from NSO.

```
$ rm /var/opt/ncs/packages/ncs-6.1-cisco-ios-cli-6.77.9.tar.gz
$ rm /var/opt/ncs/packages/ncs-6.1-cisco-iosxr-cli-7.39.5.tar.gz
```

13. Configure the bootstrap data for the new version, plan-notifications, status-codes, and kickers as described in section **Performing Post Installation Tasks for SR-TE CFP.**

14. (*On CFS only*) Sync dispatch-map and verify the map is populated with the NED ID. When syncing the dispatch map, provide the RFS node for the sync to update the dispatch map.

```
admin@ncs% request devices lsa dispatch-map sync remote-nso <rfs_node>
success true
detail Dispatch Map Synced Successfully
```

15. Perform a service redeploy reconciliation from the CFS node to reconcile the RFS service configuration. This is because the RFS service configuration may sometimes lose its backpointers to the CFS service. A service redeploy reconciliation fixes this issue.

```
user@ncs% request sr-te odn odn-template <Service> re-deploy reconcile
```

Upgrading NSO T-SDN FP Bundle in the LSA High Availability Model

Perform the following tasks to upgrade the CFPs in T-SDN FP Bundle to v6.0.0 in the LSA HA model.

Do the following on the primary nodes, unless specified as secondary:

1. Make sure to complete the tasks described in section **Before You Begin** .

2. Disable and shutdown the secondary HA nodes for both the CFS and RFS nodes in the LSA model.

   Note: You must restore the secondary HA nodes after upgrading the primary HA nodes.

```
user@ncs> request high-availability disable
result NSO Built-in HA disabled
```

A message indicating a lost connection to the secondary is displayed. Ignore this message as this connection is re-established at the end of the upgrade procedure.

3. Remove the high-availability nominal role on both CFS and RFS nodes as this has backward incompatible changes in NSO 6.x.

```
user@ncs% set high-availability settings enable-failover false
user@ncs% delete high-availability ha-node <PRIMARY_HA_NODE> nominal-role
user@ncs% delete high-availability ha-node <SECONDARY_HA_NODE> nominal-role
user@ncs% commit
```

4. Stop NSO on both the CFS and RFS nodes.

```
sudo /etc/init.d/ncs stop
```

5. Perform the tasks described in section **Upgrading NSO T-SDN FP Bundle in the LSA Model** to upgrade the CFS node and RFS nodes.

6. Reconfigure nominal-role on both the CFS and RFS nodes.

```
 user@ncs% set high-availability ha-node <PRIMARY_HA_NODE> nominal-role
 primary
user@ncs% set high-availability ha-node <SECONDARY_HA_NODE> nominal-role
secondary
user@ncs% commit
```

7. Back up both the primary CFS and RFS nodes. Use these backup to replicate the secondary CFS and RFS nodes respectively.

```
user$ sudo /opt/ncs/current/bin/ncs-backup
INFO  Backup /var/opt/ncs/backups/<version>.backup.gz created successfully
```

8. Verify the high-availability status on the primary node.

```
user@ncs% run show high-availability
high-availability enabled
high-availability status mode primary
high-availability status current-id <PRIMARY_HA_NODE>
high-availability status assigned-role primary
high-availability status read-only-mode false
ID                 ADDRESS
------------------------------------
<SECONDARY_HA_NODE_ID> <IP_ADDRESS>
```

9. On the secondary node, do the following:

    a. Upgrade NSO on both the CFS and RFS nodes by performing a system installation of NSO. For more information on how to upgrade NSO, see the ***NSO documentation***.

    b. Export the primary HA nodes backup file from the primary CFS and RFS nodes into the **/var/opt/ncs/backups/** directories on secondary CFS and RFS nodes respectively.

    ```
    user$ sudo /opt/ncs/current/bin/ncs-backup --restore <backup.gz>
    Restore /etc/ncs from the backup (y/n)? y
    Restore /var/opt/ncs from the backup (y/n)? y
    INFO  Restore completed successfully
    ```

    c. Restart NSO on both the secondary CFS and RFS nodes.

    ```
    user$ sudo /etc/init.d/ncs restart-with-package-reload
    Stopping ncs: Starting ncs: .
    ```

    d. Verify the high-availability status on the secondary nodes.

    ```
    user$ ncs_cli
    user@ncs> show high-availability
    high-availability enabled
    high-availability status mode secondary
    high-availability status current-id <SECONDARY_HA_NODE>
    high-availability status assigned-role secondary
    high-availability status be-secondary-result initialized
    high-availability status primary-id <PRIMARY_HA_NODE>
    high-availability status read-only-mode false
    ```

Migrating NEDs by Using Phased Provisioning

Phased Provisioning is an add-on package in NSO to schedule provisioning tasks. You can use Cisco NSO Phased Provisioning to migrate a group of devices from one NED version to another. For more information on Cisco NSO Phased Provisioning, see ***NSO documentation***.

The following procedure describes how to create the provisioning payload and provision the devices in the TSDN FP Bundle network to migrate to the new NED version. The provisioning payload contains a task to describe the intent of the payload. The intent includes information such as the target devices, the action to be performed on the devices (such as the migrate action), and the policies that govern the action. The payload also includes the schedule information to run the task.

Before you begin migrating the NEDs, be sure to perform the following tasks:

1. Download the phased provisioning installer from Cisco website.
2. Unpack the installer.

   ```
   $ sh ncs-6.1.4-phased-provisioning-1.0.0.signed.bin
   ```

3. Untar the tar.gz file.

   ```
   $ tar -xvzf ncs-6.1.4-phased-provisioning-1.0.0.tar.gz
   ```

4. Copy the phased provisioning package to the T-SDN FP Bundle packages directory.

Do the following to provision the phased provisioning payloads to migrate the NEDs.

1. Create the phased provisioning payload to schedule the device migration to the new NED version. The following are sample payloads to migrate devices to the new IOSXR CLI version under different schedules. The <schedule> tag in the payloads indicate the run schedule.

Provisioning payload scheduled to run immediately

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <phased-provisioning xmlns="http://cisco.com/pkg/phased-provisioning">
    <task>
      <name>cli-ned-migration</name>
      <target>/devices/device</target>
      <filter>starts-with(name,'test')</filter>
      <action>
        <action-name>migrate</action-name>
        <variable>
          <name>new-ned-id</name>
          <expr>/packages/package[name='cisco-iosxr-cli-
7.52']/component[name='cisco-ios-xr']/ned/cli/ned-id</expr>
        </variable>
      </action>
      <policy>cli-ned-migration-policy</policy>
      <self-test>
        <test-expr>/devices/device[name='$DEVICE']/device-type/$TYPE/ned-
id[text()='cisco-iosxr-cli-7.52:cisco-iosxr-cli-7.52']</test-expr>
        <variable>
          <name>DEVICE</name>
          <expr>name</expr>
        </variable>
        <variable>
          <name>TYPE</name>
          <value>cli</value>
```

```
        </variable>
      </self-test>
    </task>
    <policies>
      <policy>
        <name>cli-ned-migration-policy</name>
        <schedule>
          <immediately/>
        </schedule>
        <batch>
          <size>2</size>
        </batch>
        <error-budget>1</error-budget>
      </policy>
    </policies>
  </phased-provisioning>
</config>
```

Provisioning payload scheduled to run at a future time

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <phased-provisioning xmlns="http://cisco.com/pkg/phased-provisioning">
    <task>
      <name>cli-ned-migration</name>
      <target>/devices/device</target>
      <filter>starts-with(name,'test')</filter>
      <action>
        <action-name>migrate</action-name>
        <variable>
          <name>new-ned-id</name>
          <expr>/packages/package[name='cisco-iosxr-cli-
7.52']/component[name='cisco-ios-xr']/ned/cli/ned-id</expr>
        </variable>
      </action>
      <policy>cli-ned-migration-policy</policy>
      <self-test>
        <test-expr>/devices/device[name='$DEVICE']/device-type/$TYPE/ned-
id[text()='cisco-iosxr-cli-7.52:cisco-iosxr-cli-7.52']</test-expr>
        <variable>
          <name>DEVICE</name>
          <expr>name</expr>
        </variable>
        <variable>
          <name>TYPE</name>
          <value>cli</value>
        </variable>
      </self-test>
    </task>
    <policies>
      <policy>
        <name>cli-ned-migration-policy</name>
        <schedule>
          <future>
            <time>0 6 * * 7</time>
            <window>
              <window-time>60</window-time>
              <unit>seconds</unit>
            </window>
          </future>
        </schedule>
        <batch>
          <size>2</size>
        </batch>
        <error-budget>1</error-budget>
      </policy>
    </policies>
  </phased-provisioning>
</config>
```

2. Load merge the payload to NSO and commit the transaction.
3. Run the phased provisioning task to migrate to the new NED version.

```
admin@ncs> request phased-provisioning task cli-ned-migration run
success true
detail Task successfully processed.
```

4. Verify the status of the provisioning task. If the task is successfully completed, there should be no failed nodes.

```
admin@ncs> show phased-provisioning task-status cli-ned-migration state
state completed
admin@ncs> show phased-provisioning task-status cli-ned-migration failed-
nodes
% No entries found.
```

Appendix A: Changing Python Startup Command Configuration

Use the information in this section only if you are unable to change the default Python to Python 3. Change the Python startup command configuration after extracting the T-SDN FP Bundle packages during the T-SDN FP Bundle installation. For more information on how to extract the T-SDN FP Bundle packages, see installation instructions in this documentation.

To change the Python startup command configuration:

1. Navigate to the extracted TSDN package directory.

   ```
   $ cd tsdn-<version>-nso-<version>
   ```

2. Copy the **start-vm** file.

   ```
   $ mkdir -p /opt/cisco/nso/tsdn
   ```

   ```
   $ cp init_data/scripts/ncs-start-python-vm-tsdn /opt/cisco/nso/tsdn/ncs-
   start-python-vm-tsdn
   ```

3. Configure the **<start-command>** in **/etc/ncs/ncs.conf** file as follows:

   ```
   </java-vm>
   <python-vm>
    <start-command>/opt/cisco/nso/tsdn/ncs-start-python-vm- tsdn</start-command>
    <run-in-terminal>
        <terminal-command>DEFAULT</terminal-command>
    </run-in-terminal>
    <logging>
        <log-file-prefix>${NCS_LOG_DIR}/ncs-python-vm</log-file-prefix>
    </logging>
   </python-vm>
   ```

   Note: If you make any changes to the ncs.conf file, restart ncs to apply the changes.

Appendix B: Passing the commit-queue async Flag

The **async** flag is an API constraint used in commit-queue. Set the **async** flag to commit a Create, Read, Update, Delete (CRUD) operation through commit-queue.

The following commands show how to set the **async** flag in different APIs.

NSO CLI

```
admin@ncs% load merge payload/IETF-TE.xml
[ok]
[edit]
admin@ncs% commit commit-queue async
commit-queue {
    id 1616809621834
    status async
}
Commit complete.
[ok]
```

JSON-RPC

JSON-RPC commit invocation with commit-queue async flag. For more information, see the ***NSO WebUI*** documentation.

```
{"jsonrpc":"2.0","id":497,"method":"validate_commit","params":{"th":3,"flags":[
"commit-queue=async"]}}
{jsonrpc: "2.0", id: 86, method: "commit", params: {th: 3, flags: ["commit-
queue=async"]}}
```

RESTCONF

With RESTCONF, the POST, PUT and DELETE calls can be sent with an additional parameter for commit-queue async. There is no change to GET calls.

```
http://<NSO-IP>:8080/restconf/data/sr-te?async-commit-queue=true
```

Python

Python API commit invocation with commit-queue async flag.

```
with ncs.maapi.single_write_trans(uinfo.username, "system", db=ncs.RUNNING) as
trans:
            root = ncs.maagic.get_root(trans)
            root.ncs__devices.device[input.device].config.asa__banner.login =
input.message
            commit_params = ncs.maapi.CommitParams()
            commit_params.commit_queue_async()
            res = trans.apply_params(False, commit_params)
```

Appendix C: Using Netconf NED in T-SDN FP Bundle

Installing and Uninstalling Core Function Packs-IOSXR NC

This section discusses the packages required to install and verify SR-TE CFP-IOSXR NC.

Note: The IOSXR CLI NED is the default NED and is bundled with T-SDN FP Bundle. The IOSXR NC NED is downloadable from the Cisco website. For information on supported Netconf NEDs, see **Cisco NSO and Cisco NED Requirements**.

Installing SR-TE CFP-IOSXR NC

SR-TE CFP-IOSXR NC requires SR-TE CFP-IOSXR CLI to be installed. For more information on how to install SR-TE CFP-IOSXR CLI, see **Installing Core Function Packs on a Single NSO Instance**.

To install SR-TE CFP-IOSXR NC:

1. Obtain and load the NETCONF NED into NCS.
2. Go to the packages directory.

```
$ cd tsdn-<version>-nso-<version>/standalone/packages
```

3. Copy and link the following packages to install SR-TE CFP-IOSXR NC.

```
sudo cp ncs-<version>-sr-te-multi-vendors-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
<version>.tar.gz

sudo cp ncs-<version>-cisco-iosxr_netconf-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr_netconf-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-iosxr_netconf-
<version>.tar.gz
```

4. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
                      [  OK  ]
```

5. Verify the installation. Make sure the packages are up and running and perform the post installation tasks.

```
admin@ncs% run show packages package package-version | select build-info ncs
version | select build-info file | select build-info package sha1 | select
oper-status error-info | select oper-status up | tab
```

Post Installation Tasks for SR-TE CFP- IOSXR NC

Do the following after installing SR-TE CFP-IOSXR NC:

1. Change the current directory to:

```
$ cd tsdn-<version>-nso-<version>/tsdn/bootstrap-data
```

2. Load-merge the **2_SR-multi-vendor-iosxr-netconf.xml** file to configure the dynamic-mapping.

```
$ /opt/ncs/current/bin/ncs_cli -u admin

admin@ncs> configure

admin@ncs% load merge 2_SR-multi-vendor-iosxr-netconf.xml

admin@ncs% commit
```

3. Verify the dynamic-mapping as follows:

```
admin@ncs% show cisco-sr-te-cfp:cfp-configurations
dynamic-device-mapping cisco-iosxr-nc-7.3:cisco-iosxr-nc-7.3 {
    python-impl-class-name sr_te_multi_vendors.NativeXR;
}
dynamic-device-mapping cisco-iosxr-nc-7.4:cisco-iosxr-nc-7.4 {
    python-impl-class-name sr_te_multi_vendors.NativeXR;
}
…
[ok]
```

Installing IETF-L2VPN-NM-IOSXR NC

L2NM picks up the standardized IETF version of L2VPN implementation. This section discusses the packages you must copy to install and verify the L2NM-IOSXR NC service.

L2NM-IOSXR NC installation requires L2VPN-NM-IOSXR CLI to be installed. For more information, see **Installing Core Function Packs on a Single NSO Instance**.

To install L2NM-IOSXR NC:

1. Go to the packages directory.

```
$ cd tsdn-<release>-nso-<version>/standalone/packages
```

2. Copy and link the following packages to install L2NM-IOSXR NC:

```
sudo cp ncs-<version>-ietf-l2vpn-nm-EXAMPLE-<version>.tar.gz
/opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-EXAMPLE-
<version>.tar.gz /var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-EXAMPLE-
<version>.tar.gz

sudo cp ncs-<version>-resource-manager-<version>.tar.gz /opt/ncs/packages/

sudo ln -s /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
/var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
```

3. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                         [  OK  ]
   ```

   The L2NM-IOSXR NC installation is now complete.

4. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

5. Perform the post installation tasks.

Post Installation Tasks for IETF-L2VPN-NM-IOSXR NC

Do the following after installing L2NM-IOSXR NC:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. Perform the following post-installation tasks:

   a. Load-merge the **IETF-L2NM-plan-notification-settings.xml** file to activate notifications.

      ```
      $ /opt/ncs/current/bin/ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% load merge IETF-L2NM-plan-notification-settings.xml
      admin@ncs% commit
      ```

   b. Load-merge the **IETF-L2NM-status-codes.xml** file to activate status-codes.

      ```
      $ /opt/ncs/current/bin/ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% unhide debug
      admin@ncs% load merge IETF-L2NM-status-codes.xml
      admin@ncs% commit
      ```

   c. Load-merge the **IETF-L2NM-internal-plan-kicker.xml** file to activate kicker settings.

      ```
      $ /opt/ncs/current/bin/ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% unhide debug
      admin@ncs% load merge IETF-L2NM-internal-plan-kicker.xml
      admin@ncs% commit
      admin@ncs% load merge IETF-L2NM-route-policy-kicker.xml
      ```

3. Verify the post-installation tasks as follows:

   a. Verify the kickers configuration.

```
unhide debug
data-kicker flat-L2vpn-internal-local-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-internal-local-site/cisco-flat-L2vpn-fp-internal-local-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker flat-L2vpn-internal-remote-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-internal-remote-site/cisco-flat-L2vpn-fp-internal-remote-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker flat-L2vpn-internal-site-plan-kicker {
    monitor     /cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-internal-site/cisco-flat-L2vpn-fp-internal-site:flat-L2vpn-plan;
    kick-node   /l2vpn-ntw:l2vpn-ntw/cisco-l2vpn-ntw:l2nm-actions;
    action-name internal-plan-change-handler;
}
data-kicker l2nm-defined-set-kicker {
    monitor     /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-l2vpn-routing-policy:defined-sets;
    kick-node   /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-l2vpn-routing-policy:policy-definitions;
    action-name internal-defined-sets-change-handler;
}
data-kicker l2nm-route-policy-kicker {
    monitor     /cisco-l2vpn-routing-policy:l2vpn-routing-policy/cisco-l2vpn-routing-policy:policy-definitions/cisco-l2vpn-routing-policy:policy-definition;
    kick-node   /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-ntw:vpn-service[vpn-nodes/vpn-node/te-service-mapping/te-mapping/odn/route-policy=current()/name];
    action-name reactive-re-deploy;
}
```

   b. Verify the status-codes.

```
admin@ncs% show status-codes
core-function-pack IETF-L2NM {
    status-code-enum-path cisco-tsdn-core-fp-
common/python/cisco_tsdn_core_fp_common/status_codes/ietf_l2vpn_nm_statu
s_codes;
    status-code 400 {
        reason              "Status code mapping has not been loaded for
function pack during install";
        category            user;
        severity            ERROR;
        recommended-actions "Bootstrap status code mapping";
    }
    status-code 404 {
        reason              "Input element's value is not supported";
        category            validation;
        severity            ERROR;
        recommended-actions "Verify that input element's value is
supported in the payload";
    }
:::
    }
}
[ok]
```

c. Verify the plan-notifications.

```
admin@ncs% run show configuration services plan-notifications
subscription l2nm-notif {
    service-type /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services/l2vpn-
ntw:vpn-service;
}
[ok]

admin@ncs% show plan-path-for-notification
plan-path-for-notification /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-
service-plan {
    service-path         /l2vpn-ntw:l2vpn-ntw/vpn-services/vpn-service;
    service-key-elements [ vpn-id ];
}
[ok]
```

Installing IETF-L3VPN-NM-IOSXR NC

L3NM picks up the standardized IETF version of L3VPN implementation. This section discusses the packages you must copy to install and verify the L3NM-IOSXR NC service.

Note: The IOSXR CLI NED is the default NED and is bundled with T-SDN FP Bundle. The IOSXR NC NED is downloadable from the Cisco website. For information on the supported Netconf NEDs, see **Cisco NSO and Cisco NED Requirements**.

L3VPN-NM-IOSXE CLI installation requires L3VPN-NM-IOSXR CLI to be installed. For more information, see **Installing Core Function Packs on a Single NSO Instance**.

To install L3VPN-NM-IOSXR NC:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/packages
   ```

2. Copy and link the following packages to install L3NM-IOSXR NC:

   ```
   sudo cp ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz /opt/ncs/packages/
   ```

   ```
   sudo ln -s /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz
   /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-<version>.tar.gz
   ```

3. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                         [  OK  ]
   ```

   The L3NM-IOSXR NC installation is now complete.

4. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

5. Perform the post installation tasks.

Post Installation Tasks for IETF-L3VPN-NM-IOSXR NC

Do the following after installing L3NM-IOSXR NC:

1. Change the current directory to:

   ```
   $ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
   ```

2. Perform the following post-installation tasks.

   a. Load-merge the **IETF-L3NM-plan-notification-settings.xml** file to activate notifications.

      ```
      $ ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% load merge IETF-L3NM-plan-notification-settings.xml
      admin@ncs% commit
      ```

   b. Load-merge the **IETF-L3NM-status-codes.xml** file to activate status-codes.

      ```
      $ ncs_cli -u admin
      admin@ncs> configure
      admin@ncs% unhide debug
      admin@ncs% load merge IETF-L3NM-status-codes.xml
      admin@ncs% commit
      ```

c. Load-merge the **IETF-L3NM-internal-plan-kicker.xml** file to activate the kickers.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% load merge IETF-L3NM-internal-plan-kicker.xml
admin@ncs% load merge IETF-L3NM-route-policy-kicker.xml
admin@ncs% commit
```

3. Verify the plan-notifications as follows:

```
admin@ncs% run show configuration services plan-notifications
subscription l3nm-notif {
    service-type /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services/l3vpn-ntw:vpn-
service;
}
[ok]

admin@ncs% run show configuration plan-path-for-notification
plan-path-for-notification /l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-service-plan
{
    service-path         /l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-service;
    service-key-elements [ vpn-id ];
}
[ok]
```

Uninstalling IETF-L2VPN-NM-IOSXR NC

To uninstall the L2NM-IOSXR NC flavor, you must first delete the L2NM services with NC NED. Uninstalling this flavor reverts the system to SR-TE CFP-IOSXR CLI flavor.

Before you uninstall L2NM-IOSXR NC flavor, be sure to delete all the related services and the devices from the device tree.

To uninstall L2NM-IOSXR NC:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File**.

2. Delete plan-notifications.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% delete services plan-notifications subscription l2nm-notif
admin@ncs% delete plan-path-for-notification /l2vpn-ntw:l2vpn-ntw/vpn-
services/vpn-service-plan
admin@ncs% commit
```

3. Delete status-codes for L2NM.

```
$ ncs_cli -u admin
```

```
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% delete status-codes core-function-pack IETF-L2NM
admin@ncs% delete status-code-cfp IETF-L2NM
admin@ncs% commit
```

4. Delete kickers.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% commit
```

5. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-EXAMPLE-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-ietf-l2vpn-nm-EXAMPLE-
<version>.tar.gz
```

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz /opt/ncs/packages/ncs-<version>-resource-manager-
<version>.tar.gz
```

6. Stop NSO.

```
### Make sure user delete all services and devices from TSDN
### Make sure there are no zombie services by running the command: show
zombies
sudo /etc/init.d/ncs stop
```

7. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
                [  OK  ]
```

Uninstalling IETF-L3VPN-NM-IOSXR NC

To uninstall the L3NM-IOSXR NC core function pack, you must first delete the L3NM services with NC NED. Uninstalling this core function pack reverts the system to SR-TE CFP-IOSXR NC flavor.

Before you uninstall L3NM-IOSXR NC flavor, be sure to delete all the related services and the devices from the device tree.

To uninstall L3NM-IOSXR NC:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File**.

2. Delete plan-notifications.

```
$ ncs_cli -u admin
```

```
admin@ncs> configure
admin@ncs% delete services plan-notifications subscription l3nm-notif
admin@ncs% delete plan-path-for-notification /l3vpn-ntw:l3vpn-ntw/vpn-services/vpn-service-plan
admin@ncs% commit
```

3. Delete status-codes for L3NM.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% delete status-codes core-function-pack IETF-L3NM
admin@ncs% delete status-code-cfp IETF-L3NM
admin@ncs% commit
```

4. Delete kickers.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% unhide debug
admin@ncs% commit
```

5. Unlink the following packages in **/var/opt/ncs/packages** and delete the packages from **/opt/ncs/packages/** directory.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-EXAMPLE-<version>.tar.gz /opt/ncs/packages/ncs-<version>-ietf-l3vpn-nm-EXAMPLE-<version>.tar.gz

sudo rm -f /var/opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz /opt/ncs/packages/ncs-<version>-resource-manager-<version>.tar.gz
```

6. Stop NSO.

```
### Make sure user delete all services and devices from TSDN
### Make sure there are no zombie services by running the command: show zombies
sudo /etc/init.d/ncs stop
```

7. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
Restarting ncs (via systemctl):
               [  OK  ]
```

Uninstalling SR-TE CFP-IOSXR NC

Uninstalling SR-TE CFP-IOSXR NC reverts the system to SR-TE CFP-IOSXR CLI.

To uninstall SR-TE CFP-IOSXR NC:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.
2. Delete dynamic-mapping for the flavor.

   ```
   $ /opt/ncs/current/bin/ncs_cli -u admin
   admin@ncs> configure
   admin@ncs% delete cisco-sr-te-cfp:cfp-configurations dynamic-device-mapping
   cisco-iosxr-nc-<version>:cisco-iosxr-nc-<version>
   admin@ncs% commit
   ```
3. Unlink the following package in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

   ```
   sudo rm -f /var/opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
   <version>.tar.gz /opt/ncs/packages/ncs-<version>-sr-te-multi-vendors-
   <version>.tar.gz
   ```
4. Remove the Netconf NED installed along with the multi-vendors package.

   ```
   sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-iosxr-nc-
   <version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-iosxr-nc-
   <version>.tar.gz
   ```
5. Stop NSO.

   ```
   ### Make sure user delete all services and devices from TSDN
   ### Make sure there are no zombie services by running the command: show
   zombies
   sudo /etc/init.d/ncs stop
   ```
6. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart
   Restarting ncs (via systemctl):
                    [  OK  ]
   ```

Installing and Uninstalling Example Function Packs-IOSXR NC

Installing IETF-TE-IOSXR NC

IETF-TE-IOSXR NC installation requires IETF-TE-IOSXR CLI to be installed. For more information on how to install IETF-TE-IOSXR CLI, see **Installing Example Function Packs on a Single NSO Instance**.

To install IETF TE-IOSXR NC:

1. Go to the packages directory.

   ```
   $ cd tsdn-<version>-nso-<version>/standalone
   ```

2. Copy and link the following packages to install IETF TE-IOSXR NC:

   ```
   sudo cp core-fp-packages/ncs-<version>-cisco-iosxr-netconf-<version>.tar.gz
   /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-cisco-iosxr-netconf-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-cisco-iosxr-netconf-
   <version>.tar.gz


   sudo cp example-packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
   <version>.tar.gz /opt/ncs/packages/

   sudo ln -s /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-
   <version>.tar.gz /var/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-
   EXAMPLE-<version>.tar.gz
   ```

3. Restart NSO with package reload.

   ```
   $ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart

   Restarting ncs (via systemctl):

                         [  OK  ]
   ```

   The IETF TE-IOSXR NC installation is now complete.

4. Verify the installation and make sure the packages are up and running.

   ```
   admin@ncs% run show packages package package-version | select build-info ncs
   version | select build-info file | select build-info package sha1 | select
   oper-status error-info | select oper-status up | tab
   ```

5. Perform the post installation tasks.

Post Installation Tasks for IETF TE-IOSXR NC

Do the following after installing IETF TE-IOSXR NC:

1. Change the current directory to:

```
$ cd tsdn-<version>-nso-<version>/standalone/bootstrap-data
```

2. Load-merge the **2_IETF-TE-multi-vendor-iosxr-netconf.xml** file to configure dynamic-mapping.

```
$ ncs_cli -u admin
configure
unhide debug
admin@ncs% load merge 2_IETF-TE-multi-vendor-iosxr-netconf.xml
admin@ncs% commit
```

3. Verify the dynamic-mapping as follows:

```
unhide tsdn
admin@ncs% show cisco-rsvp-te-fp:cfp-configurations
dynamic-device-mapping cisco-iosxr-nc-7.3:cisco-iosxr-nc-7.3 {
    python-impl-class-name rsvp_te_multi_vendors.NativeXR;
}
dynamic-device-mapping cisco-iosxr-nc-7.4:cisco-iosxr-nc-7.4 {
    python-impl-class-name rsvp_te_multi_vendors.NativeXR;
}
```

Uninstalling IETF-TE-IOSXR NC

Uninstalling this flavor reverts the system to IETF-TE IOSXR CLI. For information on how to uninstall IETF-TE IOSXR CLI, see **Uninstalling Example Function Packs**.

To uninstall IETF-TE-IOSXR NC:

1. Revert the **ncs.config** file. For more information, see **Reverting Changes to the NCS Configuration File on a Single NSO Instance**.

2. Delete dynamic-mapping for IETF-TE.

```
$ ncs_cli -u admin
admin@ncs> configure
admin@ncs% delete cisco-rsvp-te-fp:cfp-configurations dynamic-device-mapping
cisco-iosxr-nc-<version>:cisco-iosxr-nc-<version>
admin@ncs% delete te:cfp-configurations dynamic-device-mapping cisco-iosxr-
nc-<version>:cisco-iosxr-nc-<version>
admin@ncs% commit
```

3. Unlink the following package in **/var/opt/ncs/packages** and delete the package from **/opt/ncs/packages/** directory.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz /opt/ncs/packages/ncs-<version>-rsvp-te-multi-vendors-EXAMPLE-<version>.tar.gz
```

4. Remove the Netconf NED if they are not used in other services.

```
sudo rm -f /var/opt/ncs/packages/ncs-<version>-cisco-iosxr-nc-<version>.tar.gz /opt/ncs/packages/ncs-<version>-cisco-iosxr-nc-<version>.tar.gz
```

5. Stop NSO.

```
### Make sure user delete all services and devices from TSDN

### Make sure there are no zombie services by running the command: show zombies

sudo /etc/init.d/ncs stop
```

6. Restart NSO with package reload.

```
$ sudo NCS_RELOAD_PACKAGES=force /etc/init.d/ncs restart

Restarting ncs (via systemctl):

               [   OK  ]
```