Discovery 1: Use Cisco NSO for the First Time

Introduction

In this lab, you will learn how to start the local instance of Cisco NSO. You will explore the operational and config modes of NSO CLI and perform simple device configuration transactional updates in the managed network.

Scenario

You have been hired as a junior network engineer at a company that uses Cisco NSO to automate device configuration management and network service activation tasks. Your manager requires that you get familiar with the environment by accomplishing simple activities of reading and writing device configurations in a lab with an already installed local NSO instance and a few virtual routers mimicking a small segment of the production network.

Activity Objective

To accomplish the task assigned to you by your manager, you will use a Linux-based virtual machine with the Cisco NSO software installed on it in your home directory.

After completing this activity, you will be able to meet the following objectives:

- Start the Cisco NSO local instance and launch the NSO CLI
- Examine the loaded packages and onboarded devices in NSO
- Inspect the currently active device configurations
- View the device configuration data in different formats
- Configure a device from Cisco NSO CLI

Job Aid

Device Information

Device	Description	IP Address	Credentials
Student DevBox	Linux Ubuntu VM	10.10.20.50	developer, C1sco12345
R1 (dist-rtr01)	Virtual Router, IOS XE - Amsterdam-17.3.4	10.10.20.175	cisco, cisco
R2 (dist-rtr02)	Virtual Router, IOS XE - Amsterdam-17.3.4	10.10.20.176	cisco, cisco
R3 (internet-rtr01)	Virtual Router, IOS XE - Amsterdam-17.3.4	172.21.1.181	cisco, cisco

Task 1: Access and Explore Cisco NSO CLI

In this task, you will start the local instance of Cisco NSO on your Student Workstation, explore the NSO CLI, and make sure that the device inventory is properly configured, and the managed devices are ready and reachable.

Activity

For this activity, complete the following steps:

Step 1 Connect to the Student DevBox.

On the topology diagram, click the **Student DevBox**.

Step 2 On the Student DevBox, from the NSO install directory, source the **ncsrc** script to update your Linux shell environment and get it ready to run the Cisco NSO software.

Open the Terminal application on the Student Workstation and run the following Linux shell commands:

```
developer@devbox:~$ cd /home/developer/nso
developer@devbox:~/nso$ source ncsrc
```

Note Updating the Linux shell environment by sourcing the **ncsrc** script only works for your current Terminal application window or tab. Make sure to continue the subsequent steps of the lab in the same Terminal window or tab. If you require to open an extra Terminal tab or window and run Cisco NSO-specific commands there, repeat this step in the new Terminal tab or window to update the shell environment accordingly. If you have accidentally closed the original Terminal tab or window, repeat this step after opening a new one.

Step 3 From the NSO runtime directory, start the **ncs** daemon process.

In the same Terminal application window or tab, run the following Linux shell commands:

```
developer@devbox:~$ cd /home/developer/nso-run
developer@devbox:~/nso-run$ ncs
```

Note It is important to start the **ncs** process from the appropriate Cisco NSO runtime instance directory. Make sure that in the Linux shell, your current working directory is **/home/developer/nso-run** before starting the daemon, otherwise it will *not* start but print out an error message instead.

Step 4 After the **ncs** daemon has started, launch the Cisco NSO CLI.

In the same Terminal application window or tab, in the Linux shell, launch the **ncs_cli** as the NSO local user **admin**:

```
developer@devbox:~/nso-run$ ncs_cli -u admin

User  admin  last  logged  in  2022-06-25T19:17:02.478784+00:00,  to
student-vm, from 100.65.0.11 using cli-ssh
admin connected from 100.65.0.11 using ssh on student-vm
admin@ncs>
```

Note Make sure that you do *not* mistype the username **admin**, otherwise, some data or actions in your NSO CLI session might become unavailable because of the RBAC restrictions.

Step 5 In the NSO CLI, switch from Juniper style to Cisco style.

In the same Terminal application window or tab, in the NSO CLI, run the **switch cli** command:

```
admin@ncs> switch cli
admin@ncs#
```

Note When you change the CLI style, the CLI prompt changes accordingly. The prompt that ends with the hash character **#** indicates that you are using Cisco style CLI.

Step 6 In the NSO CLI, inspect the installed and loaded packages.

In the same Terminal application window or tab, in the NSO CLI, run the **show packages package package-version** command:

```
admin@ncs# show packages package package-version
                    PACKAGE
NAME                VERSION
-----------------------------
cisco-ios-cli-6.67  6.67.9
```

```
admin@ncs#
```

The command output displays the current list of Cisco NSO packages in use. In your local instance, you have a single NED package already installed and loaded—the **cisco-ios-cli-6.67** package—that allows you to onboard and manage CLI-based Cisco IOS-driven devices.

Step 7 In the NSO CLI, inspect the inventory of managed network devices.

In the same Terminal application window or tab, in the NSO CLI, run the **show devices list** command:

```
admin@ncs# show devices list
NAME  ADDRESS      DESCRIPTION  NED ID              ADMIN STATE
R1    10.10.20.175  -            cisco-ios-cli-6.67  unlocked
R2    10.10.20.176  -            cisco-ios-cli-6.67  unlocked
admin@ncs#
```

The command output shows that there are two network devices, R1 and R2 routers, onboarded and available for management from the local Cisco NSO instance using the NED cisco-ios-cli-6.67.

Step 8 In the NSO CLI, ensure that all the managed network devices are online and reachable from the orchestrator.

In the same Terminal application window or tab, in the NSO CLI, run the **devices connect** command:

```
admin@ncs# devices connect
connect-result {
    device R1
    result true
    info (admin) Connected to R1 - 10.10.20.175:23
}
connect-result {
    device R2
    result true
    info (admin) Connected to R1 - 10.10.20.176:23
}
```

The command output confirms that both R1 and R2 routers are responding to the incoming SSH connections. You are all set for managing your lab network from the local instance of Cisco NSO.

You have successfully started the local instance of Cisco NSO on your Student Workstation, explored the NSO CLI, and made sure that the device inventory is properly configured, and the managed network is ready for you to continue the lab exercises.

Task 2: Read and Write Device Configurations

In this task, you will learn how to read the device configurations from the Cisco NSO in different formats and how to perform simple device configuration transactional updates. Successful completion of the task will get you ready for the more advanced activities in network device management using NSO.

Activity

For this activity, complete the following steps:

Step 1 In the NSO CLI, display and inspect the CDB copy of the R1 router configuration.

In the same Terminal application window or tab, in the NSO CLI, run the **show running-config devices device R1 config** command:

```
admin@ncs# show running-config devices device R1 config
devices device R1
 config
  hostname  dist-rtr01
  tailfned police cirmode
  version   17.3
  no service pad
  service timestamps debug datetime msec
  service timestamps log datetime msec
  no service password-encryption
  service call-home
  login on-success log
  platform console virtual
  platform punt-keepalive disable-kernel-core
  platform qfp utilization monitor load 80
  enable secret 9
$9$wcK4O2eAONaS4U$By9v7mljUbfrzgDXzliJ1IiHGiTQyLbZP6.bWN3w9T.
  aaa new-model
  aaa authentication login default local
  aaa authorization exec default local
  aaa session-id common
  ...
  interface GigabitEthernet1
   description MGMT - 172.21.1.20/24
   no switchport
   negotiation auto
   no mop enabled
   no mop sysid
   ip address 10.10.20.175 255.255.255.0
   no shutdown
  exit
  interface GigabitEthernet2
   description VLAN 12 - R1 TO R2 P2P
   no switchport
   negotiation auto
   no mop enabled
   no mop sysid
   ip address 10.0.12.1 255.255.255.252
   no shutdown
  exit
  ...
admin@ncs#
```

Note Remember to use the **config** keyword as the final element of the command. That way you ensure that only the device configuration is displayed, without any administrative inventory data.

The command output displays the entire configuration of the R1 router, as it is stored in the NSO CDB. Because of the device data model definitions, the configuration displayed by NSO in Cisco style CLI looks almost identical to the config stored in the device itself.

Step 2 In the NSO CLI, inspect the R1 router configuration in an XML format.

In the same Terminal application window or tab, in the NSO CLI, run the **show running-config devices device R1 config | display xml** command:

```
admin@ncs# show running-config devices device R1 config | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
```

```
    <device>
      <name>dist-rtr01</name>
      <config>
        <hostname xmlns="urn:ios">R1</hostname>
        <tailfned xmlns="urn:ios">
          <police>cirmode</police>
        </tailfned>
        <version xmlns="urn:ios">17.3</version>
        <service xmlns="urn:ios">
          <conf>
            <pad>false</pad>
          </conf>
          <timestamps>
            <debug>
              <datetime>
                <msec/>
              </datetime>
            </debug>
            <log>
              <datetime>
                <msec/>
              </datetime>
            </log>
          </timestamps>
          <call-home/>
          ...
          ...
        <interface xmlns="urn:ios">
          <GigabitEthernet>
            <name>1</name>
            <description>MGMT - 172.21.1.20/24</description>
            <negotiation>
              <auto>true</auto>
            </negotiation>
            <mop>
              <xenabled>false</xenabled>
              <sysid>false</sysid>
            </mop>
            <ip>
              <address>
                <primary>
                  <address>10.10.20.175</address>
                  <mask>255.255.255.0</mask>
                </primary>
              </address>
            </ip>
          </GigabitEthernet>
          <GigabitEthernet>
            <name>2</name>
            <description>VLAN 12 - R1 TO R2 P2P</description>
            <negotiation>
              <auto>true</auto>
            </negotiation>
            <mop>
              <xenabled>false</xenabled>
              <sysid>false</sysid>
            </mop>
            <ip>
              <address>
                <primary>
                  <address>10.0.12.1</address>
                  <mask>255.255.255.252</mask>
                </primary>
              </address>
            </ip>
          </GigabitEthernet>
          ...
          ...
admin@ncs#
```

The command output displays the device configuration in the format closest to that of the CDB itself. You can think of that XML representation of the device config data as the actual data stored in the CDB.

Step 3 In the NSO CLI, switch to Juniper style and display the R1 router configuration as a sequence of **set** commands.

In the same Terminal application window or tab, in the NSO CLI, run the **switch cli** command to change to the alternative CLI style, and then enter **show configuration devices device R1 config | display set**:

```
admin@ncs# switch cli
[ok][2022-06-25 19:36:50]
admin@ncs>
admin@ncs>
admin@ncs> show configuration devices device R1 config | display set
set devices device R1 config hostname dist-rtr01
set devices device R1 config tailfned police cirmode
set devices device R1 config version 17.3
set devices device R1 config service conf pad false
set devices device R1 config service timestamps
set devices device R1 config service timestamps debug datetime
set devices device R1 config service timestamps debug datetime msec
set devices device R1 config service timestamps log datetime
set devices device R1 config service timestamps log datetime msec
set devices device R1 config service call-home
set devices device R1 config login on-success log
set devices device R1 config platform console virtual
set devices device R1 config platform punt-keepalive disable-kernel-
core true
set devices device R1 config platform qfp utilization monitor load
80
set devices device R1 config enable secret type 9
set devices device R1 config enable secret secret
$9$wcK4O2eAONaS4U$By9v7mljUbfrzgDXzliJ1IiHGiTQyLbZP6.bWN3w9T.
set devices device R1 config aaa new-model
set devices device R1 config aaa authentication login default local
set devices device R1 config aaa authorization exec default local
set devices device R1 config aaa session-id common
...
...
set devices device R1 config interface GigabitEthernet 1 description
"MGMT - 172.21.1.20/24"
set devices device R1 config interface GigabitEthernet 1 negotiation
auto true
set devices device R1 config interface GigabitEthernet 1 mop enabled
false
```

```
set devices device R1 config interface GigabitEthernet 1 mop sysid
false

set devices device R1 config interface GigabitEthernet 1 ip address
primary address 10.10.20.175

set devices device R1 config interface GigabitEthernet 1 ip address
primary mask 255.255.255.0

set devices device R1 config interface GigabitEthernet 2 description
"VLAN 12 - R1 TO R2 P2P"

set devices device R1 config interface GigabitEthernet 2 negotiation
auto true

set devices device R1 config interface GigabitEthernet 2 mop enabled
false

set devices device R1 config interface GigabitEthernet 2 mop sysid
false

set devices device R1 config interface GigabitEthernet 2 ip address
primary address 10.0.12.1

set devices device R1 config interface GigabitEthernet 2 ip address
primary mask 255.255.255.252

...

...

[ok][2022-06-25 19:38:43]

admin@ncs>
```

In the Juniper-style CLI mode, the command output displays the configuration of a Cisco IOS-driven device using a syntax very similar to that of a Junos-based appliance. Keep in mind that this is only the *style* for the representation of the same config data copied from the device—the same XML config data stored in the CDB. The configuration structure, its hierarchy, the names of the items, and the syntax of the values remain the same across all the CLI styles and display modes, driven by the same device YANG data models loaded from the NED.

Note Because you switch between the alternative CLI styles, Cisco NSO does *not* convert the Cisco router configurations to Juniper router configurations or vice versa. It only changes the CLI look and feel to something that better matches your muscle memory.

Step 4 In the NSO CLI, switch *back* to Cisco style and enter the configuration mode.

In the same Terminal application window or tab, in the NSO CLI, run the **switch cli** command to change to the alternative CLI style, and then enter the **config** command:

```
admin@ncs> switch cli

admin@ncs#

admin@ncs# config
```

```
Entering configuration mode terminal
admin@ncs(config)#
```

Note When you enter the config mode in Cisco-style CLI, the CLI prompt changes to reflect that and starts showing your current configuration data context or mode, for example **(config)** for the context of the topmost data tree level.

Step 5 In the NSO CLI config mode, create a basic BGP configuration for the R1 router including a single BGP peer.

In the same Terminal application window or tab, in the NSO CLI config mode, enter the following commands line by line:

```
devices device R1
config
router bgp 65001
neighbor 10.0.12.2 remote-as 65002
```

```
admin@ncs(config)# devices device R1
admin@ncs(config-device-R1)# config
admin@ncs(config-config)# router bgp 65001
admin@ncs(config-router)# neighbor 10.0.12.2 remote-as 65002
admin@ncs(config-router)#
```

Note Your current configuration context or mode changes as you move deeper into the CDB data hierarchy, for example, when you enter the device config section or the BGP settings.

Step 6 In the NSO CLI config mode, display the current candidate configuration.

In the same Terminal application window or tab, in the NSO CLI config mode, run the **top** command, followed by the **show configuration** command:

```
admin@ncs(config-router)# top
admin@ncs(config)#
admin@ncs(config)# show configuration
devices device R1
 config
```

```
  router bgp 65001
   neighbor 10.0.12.2 remote-as 65002
  !
 !
!
admin@ncs(config)#
```

The combination of the Cisco NSO CLI **top** and **show configuration** commands allows you to see all the configuration commands that you have entered in the config session so far.

Note The output will *not* include the commands that you may have entered if these commands already exist in the CDB running-config.

Step 7 In the NSO CLI config mode, perform a transaction dry-run and inspect the minimal config diff calculated by the system.

In the same Terminal application window or tab, in the NSO CLI config mode, run the **commit dry-run** command:

```
admin@ncs(config)# commit dry-run
cli {
    local-node {
        data  devices {
                  device R1 {
                      config {
                          router {
             +                bgp 65001 {
             +                    neighbor 10.0.12.2 {
             +                        remote-as 65002;
             +                    }
             +                }
                          }
                      }
                  }
              }
    }
}
admin@ncs(config)#
```

The command output shows the items that will be added to the CDB copy of the device configuration and eventually to the device itself once you commit the candidate config in a regular transaction. The format of the output is a diff-like format, where each new item is marked with the plus sign (+) at the beginning of the line.

Step 8 In the NSO CLI config mode, display the native device commands that the system generates based on the calculated minimal config diff.

In the same Terminal application window or tab, in the NSO CLI, run the **commit dry-run outformat native** command:

```
admin@ncs(config)# commit dry-run outformat native
native {
    device {
        name R1
        data router bgp 65001
              neighbor 10.0.12.2 remote-as 65002
             !
    }
}
admin@ncs(config)#
```

The command output displays the native configuration commands that Cisco NSO will send to the device once you commit the candidate config in a regular transaction.

Step 9 In the NSO CLI config mode, finalize the transaction and show the details about its phases.

In the same Terminal application window or tab, in the NSO CLI, run the **commit | details** command:

```
admin@ncs(config)# commit | details
 2022-06-25T20:23:12.255 applying transaction...
entering validate phase for running usid=52 tid=1437
 2022-06-25T20:23:12.255 grabbing transaction lock... ok (0.000 s)
 2022-06-25T20:23:12.257 creating rollback file... ok (0.003 s)
 2022-06-25T20:23:12.260 run transforms and transaction hooks... ok
(0.002 s)
 2022-06-25T20:23:12.263 mark inactive... ok (0.000 s)
 2022-06-25T20:23:12.264 pre validate... ok (0.000 s)
 2022-06-25T20:23:12.265 run validation over the changeset... ok
(0.001 s)
```

```
 2022-06-25T20:23:12.266 run dependency-triggered validation... ok
(0.000 s)
 2022-06-25T20:23:12.266 check configuration policies... ok (0.000
s)
leaving validate phase for running usid=52 tid=1437 (0.011 s)
entering write-start phase for running usid=52 tid=1437
 2022-06-25T20:23:12.267 cdb: write-start
 2022-06-25T20:23:12.267 ncs-internal-service-mux: write-start
 2022-06-25T20:23:12.268 check data kickers... ok (0.000 s)
leaving write-start phase for running usid=52 tid=1437 (0.002 s)
entering prepare phase for running usid=52 tid=1437
 2022-06-25T20:23:12.269 cdb: prepare
 2022-06-25T20:23:12.272 ncs-internal-device-mgr: prepare
 2022-06-25T20:23:13.614 device R1: push configuration...
leaving prepare phase for running usid=52 tid=1437 (4.014 s)
entering commit phase for running usid=52 tid=1437
 2022-06-25T20:23:16.283 cdb: commit
 2022-06-25T20:23:16.306 ncs-internal-device-mgr: commit
 2022-06-25T20:23:16.307 device R1: push configuration: ok (2.693 s)
 2022-06-25T20:23:16.981 releasing transaction lock
leaving commit phase for running usid=52 tid=1437 (0.697 s)
 2022-06-25T20:23:16.981 applying transaction: ok (4.726 s)
Commit complete.
admin@ncs(config)#
```

The command updates the CDB with the requested device config changes and modifies the device itself accordingly. If the device activates the change successfully, it becomes persistent both on the device and in the CDB. The **| details** pipe command helps you understand the flow of transactions better and also shows the timings for the transaction phases through which the system proceeds.

You have successfully read the device configuration from NSO in different formats and performed a simple transactional update of a device configuration. Now you are ready to learn more about managing the network devices using Cisco NSO.

Review Questions

Question 1:

Which Cisco NSO CLI command switches the CLI look and feel between two alternative styles?

a) cli switch
b) style cisco
c) switch cli
d) switch style

Answers: The correct answer is **switch cli**.

Question 2:

Which Cisco NSO CLI command displays the configuration commands that Cisco NSO will send to a CLI-managed device once you commit its candidate config in a regular transaction, in the format supported by the device itself?

a) commit | details
b) commit | dry-run
c) commit dry-run outformat native
d) commit dry-run outformat xml

Answers: The correct answer is **commit dry-run outformat native**.