

Discovery 14: Manage Commit Queue

Introduction

In this activity, you will learn how to manage a commit queue.

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

- Enable the commit queue.
- Manage the commit queue and commit queue items.
- Temporarily disable configuration push to a specific device by using the commit queue.

Job Aid

The following job aid is available to help you complete the lab activities:

- This Lab Guide

The following table contains passwords that you might need.

Device	Username	Password
student-vm	student	1234QWer
nso-server	student	1234QWer

Required Resources

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

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

Command List

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

Linux Shell:

Command	Comment
source /opt/ncs/ncs-6.1/ncsrc	Source NSO environmental variable in Docker container.
ls ll	Display contents of the current directory.
cd	Move directly to user home directory.
cd ..	Exit out of current directory.
cd test	Move into the "test" folder which is a subfolder of the current directory.

Command	Comment
cd /home/student	Move into the "nso300" folder by specifying the direct path to it starting from the root of the directory system.
ncs_cli -C	Log in to NSO CLI directly from local server.

NSO CLI:

Command	Comment
switch cli	Change CLI style.
show ?	Display all command options for current mode.
configure	Enter configuration mode.
commit	Commit new configuration (configuration mode only command).
show configuration	Display new configuration that has not yet been committed (configuration mode only command).

Makefile commands for Docker environment:

Command	Comment
make build	Builds the main NSO Docker image.
make testenv-start	Starts the NSO Docker environment.
make testenv-stop	Stops the NSO Docker environment.
make testenv-build	Recompiles and reloads the NSO packages.
make testenv-cli	Enters the NSO CLI of the NSO Docker container.
make testenv-shell	Enters the Linux shell of the NSO Docker container.
make dev-shell	Enters the Linux shell of the NSO Docker development container.

Command Syntax Reference

This lab guide uses the following conventions for **command syntax**:

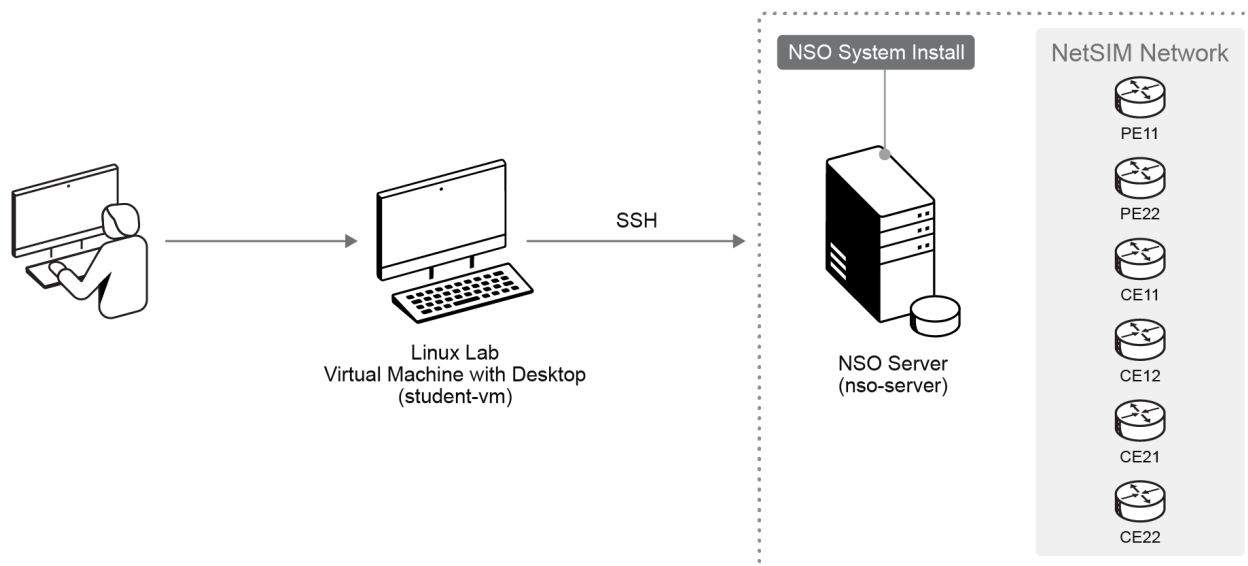
Formatting	Description and Examples
show running config	Commands in steps use this formatting.
<i>Example</i>	Type show running config
<i>Example</i>	Use the name command.
<div>show running config</div>	Commands in CLI outputs and configurations use this formatting.

Formatting	Description and Examples
highlight	CLI output that is important is highlighted.
Example	<pre>student@student-vm:~\$ ncs --version 6.1</pre>
Example	<p>Save your current configuration as the default startup config.</p> <pre>Router Name# copy running startup</pre>
brackets ([])	Indicates optional element. You can choose one of the options.
Example:	<pre>(config-if)# frame-relay lmi-type {ansi cisco q933a}</pre>
<i>italics font</i>	Arguments for which you supply values.
Example	Open file ip tcp window-size bytes
angle brackets (< >)	In contexts that do not allow italics, arguments for which you supply values are enclosed in angle brackets [<>]. Do not type the brackets when entering the command.
Example	If the command syntax is ping <ip_address> , you enter ping 10.0.0.102
string	A non-quoted set of characters. Type the characters as-is.
Example	(config)# hostname MyRouter
vertical line ()	Indicates that you enter one of the choices. The vertical line separates choices. Do not type the vertical line when entering the command.
Example	If the command syntax is show ip route arp , you enter either show ip route or show ip arp , but not both.

Lab Topology Information

Your lab session is your own personal sandbox. Whatever you do in your session will not be reflected in anyone else's session. Your lab environment is a Linux server (Student-VM) acting as a jumphost and a Linux server (NSO-server) acting as a NSO server. NSO server includes NetSIM routers. This will be the network that you will orchestrate with your NSO.

Topology



Task 1: Enable Commit Queue

In this task, you will enable the commit queue functionality for all devices in NSO.

Activity

Step 1

Connect to the Student-VM.

You can connect to the server either by choosing the **Student-VM** from the device list or by clicking the **Student-VM** icon in the topology map.

Step 2

Open the terminal window.

Open the terminal window by clicking the **Terminal** icon in the bottom bar.

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

Step 3

Connect to the **nso-server** NSO server.

Connect to the **nso-server** NSO server with the **student** user using the SSH client. The authentication is already preconfigured with the public key authentication, therefore the password is not needed. The prompt will change, stating that you are now connected to the nso-server.

```
student@student-vm:~$ ssh student@nso-server
Last login: Tue Oct 3 09:14:42 2023 from 10.0.0.102
student@nso-server:~$
```

Step 4

Connect to the NSO CLI.

Use the **ncs_cli -C** command to enter NSO CLI and **config** to enter the configuration mode.

```
student@nso-server:~$ ncs_cli -C

User student last logged in 2024-02-05T09:52:53.80762+00:00, to nso-
server, from 100.64.0.11 using cli-ssh
student connected from 100.64.0.11 using ssh on nso-server
student@ncs# config
Entering configuration mode terminal
student@ncs(config)#
```

Step 5

Enable the commit queue functionality for the devices in NSO.

To enable the commit queue, use the **devices global-settings commit-queue** command. You will also configure the time between retries to 30 seconds. You will set the rollback-on-error behavior to roll back commit queue items in case of errors. The commit queue will place a lock on the failed queue item, blocking other queue items with overlapping devices from being executed. The rollback action will then automatically be invoked when the failed queue item has finished its execution. The lock is removed as part of the rollback. The atomic option set to false will allow configuration to be pushed to devices that can accept the changes even though some devices that are part of the transaction are currently unavailable.

```
student@ncs(config)# devices global-settings commit-queue enabled-by-
default true
student@ncs(config)# devices global-settings commit-queue retry-timeout
30
student@ncs(config)# devices global-settings commit-queue atomic false
student@ncs(config)# devices global-settings commit-queue error-option
rollback-on-error
student@ncs(config)#
```

Step 6

Commit the changes and exit the configuration mode.

Use the **exit** command.

```
student@ncs(config)# commit
Commit complete.
student@ncs(config)# exit
student@ncs#
```

Activity Verification

You have completed this task when you attain these results:

- You enabled commit queue functionality for all devices.

Task 2: Manage Commit Queue Items

In this task, you will inspect and manage to commit queue items.

Activity

Step 1

Enter NSO configuration mode.

Use the **config** command.

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

Step 2

Configure the static-route service instance for CE21 device and commit the changes.

When committing changes, a commit queue will be used. A commit queue item ID is assigned to the transaction. In case the device is not available, the transaction for the CE21 device will wait in the queue and will be retried according to the retry interval.

```
student@ncs(config)# services static-route CE21
student@ncs(config-static-route-CE21)# route 10.200.0.0/24 next-hop
10.0.0.2
student@ncs(config-route-10.200.0.0/24)# top
student@ncs(config)# commit
commit-queue {
    id 1707320777797
    status async
}
Commit complete.
student@ncs(config)#
```

Step 3

Exit the configuration mode.

Use the **exit** command.

```
student@ncs(config)# exit
student@ncs#
```

Step 4

Review the status of the commit queue.

Use the **show devices commit-queue summary** command.

```
student@ncs# show devices commit-queue summary
ID              STATUS      DEVICES    COMPLETED  NAME
-----
170732077797    completed  [ CE21 ]   [ CE21 ]
student@ncs#
```

Step 5

Review the status of the commit queue in a more detailed way.

Use the **show devices commit-queue details** command. Take note of the ID of the completed queue item.

```
student@ncs# show devices commit-queue details
devices commit-queue completed queue-item 170732077797
when                2024-02-07T15:25:16.383517+00:00
status              completed
devices             [ CE21 ]
completed           [ CE21 ]
completed-services  [ /ncs:services/static-route:static-route[static-
route:device='CE21'] ]
student@ncs#
```

Step 6

Enter the configuration mode and configure the static-route service instance for two additional devices, CE11 and CE22. Commit the changes and exit the configuration mode.

If any of the devices in the transaction are unavailable, the configuration changes for those devices will wait in the queue while configuration for other devices will be applied. This is especially useful when any of the participating devices are slow to accept configuration changes or are occasionally offline.

In the following output, an alarm is raised for the device CE11. You will inspect the commit queue item and determine the status of the CE11 device.

```
student@ncs# config
Entering configuration mode terminal
student@ncs(config)# services static-route CE11
student@ncs(config-static-route-CE11)# route 10.200.0.0/24 next-hop
10.0.0.2
student@ncs(config-route-10.200.0.0/24)# top
student@ncs(config)# services static-route CE22
student@ncs(config-static-route-CE22)# route 10.200.0.0/24 next-hop
10.0.0.2
student@ncs(config-route-10.200.0.0/24)# top
student@ncs(config)# commit
commit-queue {
    id 1707320400687
```

```

        status async
    }
    Commit complete.
student@ncs(config)# *** ALARM connection-failure: Failed to connect to
device CE11: connection refused: NEDCOM CONNECT: Connection refused
(Connection refused) in new state
student@ncs(config)# exit
student@ncs#

```

Step 7

View the details of the commit queue.

Use the **show devices commit-queue summary** command. There are two items, one completed and one still executing. Take note of the ID of the executing queue item. You can see the device CE11 is in an error state.

```

student@ncs# show devices commit-queue summary

```

ID NAME	STATUS	DEVICES	WAITING FOR	TRANSIENT ERRORS	COMPLETED
1707320843058	executing	[CE11 CE22]	-	[CE11]	[CE22]


```

ID          STATUS    DEVICES    COMPLETED  NAME
-----
170732077797 completed  [ CE21 ]   [ CE21 ]

```

Step 8

Inspect the details of the queue item from the previous step.

Use the **show devices commit-queue queue-item <ID> details** command. Details such as the age of the item, devices participating, and pending changes not executed yet can be seen.

```

student@ncs# show devices commit-queue queue-item 1707320843058 details
devices commit-queue queue-item 1707320843058
age                639
status             executing
devices            [ CE11 CE22 ]
services           [ /ncs:services/static-route:static-route[static-
route:device='CE11'] /ncs:services/static-route:static-route[static-
route:device='CE22'] ]
transient-errors   [ CE11 ]
completed          [ CE22 ]
is-atomic          false
modification       CE11
data               <ip xmlns="urn:ios">
                   <route>
                     <ip-route-forwarding-list>
                       <prefix>10.200.0.0</prefix>
                       <mask>255.255.255.0</mask>
                       <forwarding-address>10.0.0.2</forwarding-address>

```



```
        </ip-route-forwarding-list>
    </route>
</ip>

local-user student
modification CE22
local-user student
student@ncs#
```

Step 9

Review if any alarms for the CE11 device exist.

Use the **show alarms alarm-list alarm CE11** command. There should be an alarm for the connection failure to the CE11 device. It seems the CE11 device is not online.

```
student@ncs# show alarms alarm-list alarm CE11
alarms alarm-list alarm CE11 connection-failure /devices/
device[name='CE11'] ""
  is-cleared                false
  last-status-change        2024-02-07T15:47:24.605741+00:00
  last-perceived-severity   major
  last-alarm-text           "Failed to connect to device CE11: connection
  refused: NEDCOM CONNECT: Connection refused (Connection refused) in new
  state"
  status-change             2024-02-07T15:47:24.605741+00:00
  received-time             2024-02-07T15:47:24.605741+00:00
  perceived-severity        major
  alarm-text                "Failed to connect to device CE11: connection
  refused: NEDCOM CONNECT: Connection refused (Connection refused) in new
  state"
student@ncs#
```

Step 10

Exit NSO CLI and verify the status of the NetSim device.

Use the **exit** command to exit the CLI, navigate to the **~/lab** folder, and review if the device CE11 is alive.

```
student@ncs# exit
student@nso-server:~$ cd lab
student@nso-server:~/lab$ ncs-netsim is-alive
DEVICE CE11 FAIL
DEVICE CE12 OK
DEVICE CE21 OK
DEVICE CE22 OK
DEVICE PE11 OK
DEVICE PE22 OK

student@nso-server:~/lab$ student@ncs#
```

Step 11

Start the CE11 NetSim device.

Use the **ncs-netsim start CE11** command.

```
student@nso-server:~/lab$ ncs-netsim start CE11
DEVICE CE11 OK STARTED
student@nso-server:~/lab$ student@ncs#
```

Step 12

Enter the NSO CLI and verify the status of the commit queue.

Use the **ncs_cli -C** command to enter the NSO CLI and then issue the **show devices commit-queue summary** command.

```
student@nso-server:~/lab$ ncs_cli -C

User student last logged in 2024-02-07T16:03:54.097516+00:00, to nso-
server, from 100.64.0.11 using cli-ssh
student connected from 100.64.0.11 using ssh on nso-server
student@ncs# show devices commit-queue summary
ID          STATUS    DEVICES          COMPLETED      NAME
170732077797 completed [ CE21 ]        [ CE21 ]
1707320843058 completed [ CE11 CE22 ]    [ CE11 CE22 ]

student@ncs#
```

Activity Verification

You have completed this task when you attain these results:

- You successfully configured devices through the commit queue functionality.
- In the end, all commit queue items are successfully processed with the status “completed.”

Task 3: Temporarily Block Configuration Delivery for a Device

In this task, you will block the configuration delivery to one of the devices by adding a lock to a device. Adding a lock will simulate, for example, the device maintenance window while still allowing the configuration changes to be accepted by NSO from the northbound interfaces and NSO CLI. You will then modify the device configuration several times, simulate multiple changes, inspect the waiting commit queue items, and finally release the waiting commit queue items so the device configuration can be delivered.

Activity

Step 1

Add a lock for the CE11 device.

Use the **devices commit-queue add-lock device <device>** command. A commit queue item ID is assigned to an empty transaction, locking the device.

```
student@ncs# devices commit-queue add-lock device CE11  
commit-queue-id 1707322341459  
student@ncs#
```

Step 2

Enter the NSO configuration mode.

Use the **config** command.

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

Step 3

Remove the static-route service instance for the device CE11 and commit the changes.

Use the **no services static-route <device>** command. Commit the changes. A new commit queue item is created.

```
student@ncs# no services static-route CE11  
student@ncs(config)# commit  
commit-queue {  
    id 1707322611688  
    status async  
}  
Commit complete.  
student@ncs(config)#
```

Step 4

Configure a GigabitEthernet interface on the device CE11.

Use the **devices device CE11 config** command. Commit the changes. A new commit queue item is created.

```
student@ncs(config)# devices device CE11 config interface  
GigabitEthernet 0/1  
student@ncs(config-if)# ip address 10.200.0.101 255.255.255.0  
student@ncs(config-if)# top  
student@ncs(config)# commit  
commit-queue {  
    id 1707322755231  
    status async  
}  
Commit complete.  
student@ncs(config)#
```

Step 5

Exit the configuration mode.

Use the **exit** command.

```
student@ncs(config)# exit
student@ncs#
```

Step 6

Verify the status of the commit queue.

Use the **show devices commit-queue summary** command. You can see that all the changes are pending but have not yet been pushed to the device.

```
student@ncs# show devices commit-queue summary
```

ID	STATUS	DEVICES	WAITING FOR	TRANSIENT ERRORS	COMPLETED	NAME
1707322341459	locked	[CE11]	-	-	-	
1707322611688	blocked	[CE11]	[CE11]	-	-	
1707322755231	blocked	[CE11]	[CE11]	-	-	

ID	STATUS	DEVICES	COMPLETED	NAME
1707320777797	completed	[CE21]	[CE21]	
1707320843058	completed	[CE11 CE22]	[CE11 CE22]	

```
student@ncs#
```

Step 7

Review the details of the locked commit queue item of the CE11 device.

Use the **show devices commit-queue queue-item <ID> details** command. There is no configuration, just an empty queue-item, representing a lock on the device.

Alternatively, you can use the **show devices commit-queue** command to show the details of all queue items.

```
student@ncs# show devices commit-queue queue-item 1707322341459 details
devices commit-queue queue-item 1707322341459
age          1019
status       locked
devices      [ CE11 ]
is-atomic    true
modification CE11
local-user   student
student@ncs#
```

Step 8

Unlock the queue item.

Use the **devices commit-queue queue-item <ID> unlock** command.

```
student@ncs# devices commit-queue queue-item 1707322341459 unlock
student@ncs#
```

Step 9

Review the status of the commit queue again.

Use the **show devices commit-queue summary** command. All the queue items were successfully processed and are now completed.

```
student@ncs# show devices commit-queue summary
ID                STATUS      DEVICES                COMPLETED            NAME
-----
1707320777797    completed  [ CE21 ]              [ CE21 ]
1707320843058    completed  [ CE11 CE22 ]        [ CE11 CE22 ]
1707322611688    completed  [ CE11 ]              [ CE11 ]
1707322755231    completed  [ CE11 ]              [ CE11 ]

student@ncs#
```

Step 10

Confirm the existence of the configuration of the GigabitEthernet interface on the CE11 device.

Use the **show running-config devices device CE11 config interface GigabitEthernet 0/1** command. You can observe that the queued configuration was successfully processed and is now part of the device configuration in the database and on the device.

```
student@ncs# show running-config devices device CE11 config interface
GigabitEthernet 0/1
devices device CE11
config
  interface GigabitEthernet0/1
    no switchport
    ip address 10.200.0.101 255.255.255.0
    no shutdown
  exit
!
!
student@ncs#
```

Activity Verification

You have completed this task when you attain these results:

- You blocked configuration delivery for a device.
- You modified the configuration of the locked device multiple times.

- You released the lock and verified the queue items were successfully processed.

Which command will show how many seconds queue items are waiting in a queue?

- ☐ **show devices commit-queue summary**
- ☐ **show devices commit-queue queue-item**
- ☐ **show devices commit-queue queue-item details**
- ☐ **show devices commit-queue details**