

Discovery 12: Replace a Device

Introduction

In this activity, you will simulate a device failure, perform a device replacement with the help of NSO and restore the device configuration to the new device.

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

- Replace a netsim device.
- Restore device configuration using NSO.

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 folder "test" which is a subfolder of the current directory.

Command	Comment
cd /home/student	Move into folder "nso300" 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**:

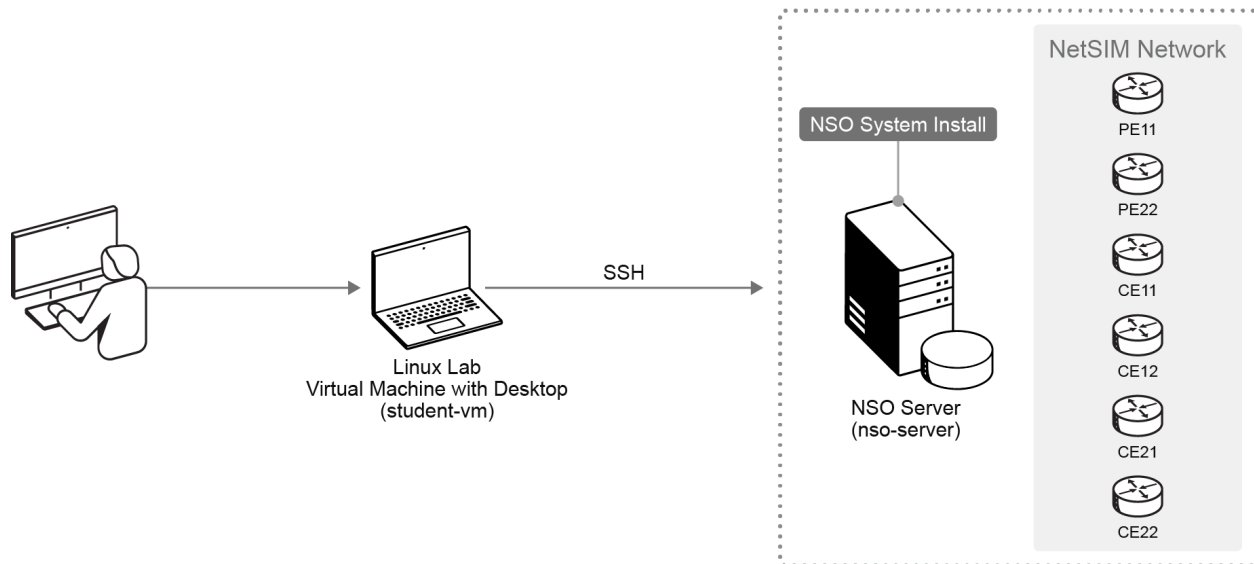
Formatting	Description and Examples
show running config	Commands in steps use this formatting.
<i>Example</i>	Type show running config
<i>Example</i>	Use the name command.
<div>show running config</div>	Commands in CLI outputs and configurations use this formatting.

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.

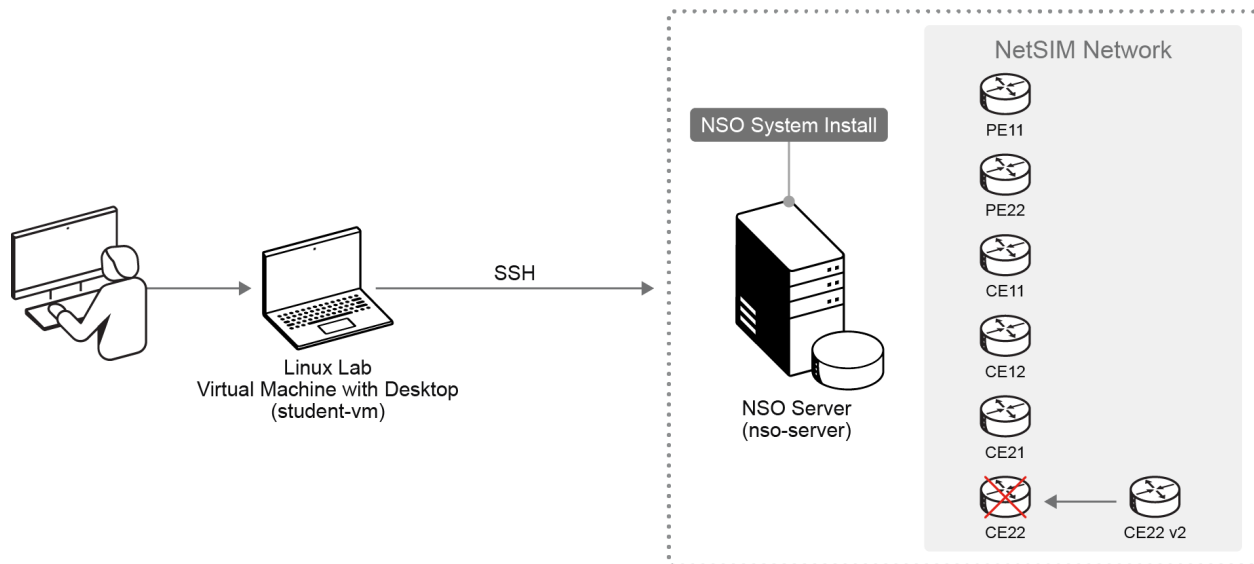
Lab Topology Information

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 a NetSIM router. This will be the network that you will orchestrate with your NSO.

Topology



Visual Objective



Task 1: Simulate Device Failure with NetSim

In this task, simulate device failure with NetSim and then create a new replacement device.



Device failure can also be simulated in a Docker environment, by stopping, removing, and recreating a NetSim device container.

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 on the **Student-VM** icon in the topology map.

Step 2

Open the terminal window.

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

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

Step 3

Connect to the NSO server *nso-server*.

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

Copy the **l3vpn-rfs** package from **packages** to **/var/opt/ncs/packages** and compile it.

Use the following commands:

```
student@nso-server:~$ cp -r packages/l3vpn-rfs /var/opt/ncs/packages
student@nso-server:~$ make -B -C /var/opt/ncs/packages/l3vpn-rfs/src/
make: Entering directory '/var/opt/ncs/packages/l3vpn-rfs/src'
mkdir -p ../load-dir
mkdir -p java/src//
/opt/ncs/current/bin/ncsc `ls l3vpn-rfs-ann.yang` > /dev/null 2>&1 &&
echo "-a l3vpn-rfs-ann.yang" ` \
    -c -o ../load-dir/l3vpn-rfs.fxs yang/l3vpn-rfs.yang
make: Leaving directory '/var/opt/ncs/packages/l3vpn-rfs/src'
student@nso-server:~$
```



The purpose of the **make** utility is to determine automatically which pieces of a large program need to be recompiled and issue the commands to recompile them. You can specify build details with additional parameters as in the preceeding example. Parameter **-B** will unconditionally make all targets specified in the Makefile. Parameter **-C *dir*** will change to directory *dir* before reading the makefiles or doing anything else.

Step 5

Connect to the NSO CLI and reload the packages.

Use the **ncs_cli -C** command to connect to the NSO CLI and the **packages reload**

command to reload the packages. Verify the l3vpn-rfs package is active by inspecting the reload result.

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

User student last logged in 2024-02-05T20:42:00.816531+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# packages reload

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has
completed.
>>> System upgrade has completed successfully.
reload-result {
  package cisco-ios-cli-6.85
  result true
}
reload-result {
  package cisco-iosxr-cli-7.41
  result true
}
reload-result {
  package l3vpn-rfs
  result true
}
student@ncs#
System message at 2024-02-05 20:45:26...
  Subsystem stopped: ncs-dp-1-cisco-ios-cli-6.85:IOSDp
student@ncs#
System message at 2024-02-05 20:45:26...
  Subsystem started: ncs-dp-2-cisco-ios-cli-6.85:IOSDp
student@ncs#
```

Step 6

Configure the CE22 VPN link using the l3vpn-rfs service.

Enter the config mode and configure the CE22 device as a part of the l3vpn-rfs service.

```
student@ncs# config
Entering configuration mode terminal
student@ncs(config)# l3vpn-rfs vpn1 link CustomerA device CE22
interface 0/0 ip-address 10.100.0.1 mask 255.255.0.0
student@ncs(config-link-CustomerA)# commit
Commit complete.
student@ncs(config-link-CustomerA)#
```

Step 7

Exit the NSO CLI.

Use the **exit** command.

```
student@ncs (config-link-CustomerA) # top
student@ncs (config) # exit
student@ncs # exit
student@nso-server:~$
```

Step 8

Go to the NetSim lab devices folder.

Navigate to the **~/lab/netsim** directory, which is the home folder for the NetSim lab devices.

```
student@nso-server:~$ cd lab/netsim
student@nso-server:~/lab/netsim$
```

Step 9

List the NetSim devices.

List the devices with the **ncs-netsim list** command.

```
student@nso-server:~/lab/netsim$ ncs-netsim list
ncs-netsim list for /home/student/lab/netsim

name=CE11 netconf=12022 snmp=11022 ipc=5010 cli=10022 dir=/home/
student/lab/netsim/CE/CE0
name=CE12 netconf=12023 snmp=11023 ipc=5011 cli=10023 dir=/home/
student/lab/netsim/CE/CE1
name=CE21 netconf=12024 snmp=11024 ipc=5012 cli=10024 dir=/home/
student/lab/netsim/CE/CE2
name=CE22 netconf=12025 snmp=11025 ipc=5013 cli=10025 dir=/home/
student/lab/netsim/CE/CE3
name=PE11 netconf=12026 snmp=11026 ipc=5014 cli=10026 dir=/home/
student/lab/netsim/PE/PE0
name=PE22 netconf=12027 snmp=11027 ipc=5015 cli=10027 dir=/home/
student/lab/netsim/PE/PE1
student@nso-server:~/lab/netsim$
```

Step 10

Stop the CE22 device to simulate a critical device failure in a real-world environment.

Use the **ncs-netsim stop CE22** command.

```
student@nso-server:~/lab/netsim$ ncs-netsim stop CE22
DEVICE CE22 STOPPED
student@nso-server:~/lab/netsim$
```

Step 11

Enter the NSO CLI and try to connect to the devices using the **devices connect** command.

Connection to the CE22 device will fail, and a connection-failure alarm will be visible on the NSO CLI.

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

User student last logged in 2024-02-05T20:44:17.981669+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# devices connect
connect-result {
    device CE11
    result true
    info (student) Connected to CE11 - 127.0.0.1:10022
}
connect-result {
    device CE12
    result true
    info (student) Connected to CE12 - 127.0.0.1:10023
}
connect-result {
    device CE21
    result true
    info (student) Connected to CE21 - 127.0.0.1:10024
}
connect-result {
    device CE22
    result false
    info Failed to connect to device CE22: connection refused: NEDCOM
CONNECT: Connection refused (Connection refused) in new state
}
connect-result {
    device PE11
    result true
    info (student) Connected to PE11 - 127.0.0.1:10026
}
connect-result {
    device PE22
    result true
    info (student) Connected to PE22 - 127.0.0.1:10027
}
student@ncs# *** ALARM connection-failure: Failed to connect to device
CE22: connection refused: NEDCOM CONNECT: Connection refused
(Connection refused) in new state
student@ncs#
```

Step 12

Display the alarms of the CE22 device.

Retrieve the list of alarms using the **show alarms** command.

```
student@ncs# show alarms alarm-list alarm CE22
```



```
alarms alarm-list alarm CE22 connection-failure /devices/  
device[name='CE22'] ""  
  is-cleared                false  
  last-status-change        2024-02-05T20:50:03.40651+00:00  
  last-perceived-severity   major  
  last-alarm-text           "Failed to connect to device CE22: connection  
  refused: NEDCOM CONNECT: Connection refused (Connection refused) in new  
  state"  
  status-change 2024-02-05T20:50:03.40651+00:00  
    received-time      2024-02-05T20:50:03.40651+00:00  
    perceived-severity major  
    alarm-text         "Failed to connect to device CE22: connection  
    refused: NEDCOM CONNECT: Connection refused (Connection refused) in new  
    state"  
student@ncs#
```

Step 13

Display the configuration of the CE22 device.

Even though the device is unreachable, the device configuration is still safely stored within the NSO CDB.

```
student@ncs# show running-config devices device CE22  
devices device CE22  
  address 127.0.0.1  
  port 10025  
  ...  
  ...  
  interface GigabitEthernet0/0  
    no switchport  
    vrf forwarding vpn1  
    ip address 10.100.0.1 255.255.0.0  
    no shutdown  
  exit  
  router bgp 65000  
    address-family ipv4 unicast vrf vpn1  
      neighbor 10.100.0.1 remote-as 65001  
    exit-address-family  
  !  
  !  
  !  
  !  
student@ncs#
```

Step 14

Exit the NSO CLI.

Use the **exit** command.

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

Step 15

Display the help page for the **ncs-netsim** command.

Review and think about how you can create and add an extra device to the NetSim network.

```

student@nso-server:~/lab/netsim$ ncs-netsim help
Usage ncs-netsim  [--dir <NetsimDir>]
                  create-network <NcsPackage> <NumDevices> <Prefix> |
                  create-device  <NcsPackage> <DeviceName>           |
                  add-to-network <NcsPackage> <NumDevices> <Prefix> |
                  add-device     <NcsPackage> <DeviceName> |
                  delete-network                                     |
                  [-a | --async] start [devname]                 |
                  [-a | --async] stop [devname]                  |
                  [-a | --async] reset [devname]                 |
                  [-a | --async] restart [devname]               |
                  list                                             |
                  is-alive [devname]                              |
                  status [devname]                                |
                  whichdir                                         |
                  ncs-xml-init [devname]                          |
                  ncs-xml-init-remote <RemoteNodeName> [devname] |
                  [--force-generic]                               |
                  packages                                         |
                  netconf-console devname [XPathFilter] |
                  [-w | --window] [cli | cli-c | cli-i] devname |
                  get-port devname [ipc | netconf | cli | snmp]

See manpage for ncs-netsim for more info. NetsimDir is optional
and defaults to ./netsim, any netsim directory above in the path,
or $NETSIM_DIR if set.

student@nso-server:~/lab/netsim$

```

Step 16

Add a new device CE22v2 that will replace the faulty CE22 one.

NetSim device names must be unique.

Before adding a device, you will need to give some permissions to the host keys, used by the NetSim devices, since it was set up by another non-student user. The RSA keys are located in the `/opt/ncs/current/netsim/confd/etc/confd/ssh/ssh_host_rsa_key` and `/opt/ncs/current/netsim/confd/etc/confd/ssh/ssh_host_ed25519_key`.

```

student@nso-server:~/lab/netsim$ sudo chmod 644 /opt/ncs/current/
netsim/confd/etc/confd/ssh/ssh_host_rsa_key
student@nso-server:~/lab/netsim$ sudo chmod 644 /opt/ncs/current/
netsim/confd/etc/confd/ssh/ssh_host_ed25519_key
student@nso-server:~/lab/netsim$ ncs-netsim add-device /var/opt/ncs/
packages/cisco-ios-cli-6.85 CE22v2

```

```
DEVICE CE22v2 CREATED
student@nso-server:~/lab/netsim$
```

Step 17

Start the newly created NetSim device.

Use the **ncs-netsim start CE22v2** command.

```
student@nso-server:~/lab/netsim$ ncs-netsim start CE22v2
DEVICE CE22v2 OK STARTED
student@nso-server:~/lab/netsim$
```

Step 18

List the NetSim devices again.

Since NetSim devices are running on the same host, you differentiate them through the port. Find the information about the CLI port that the old CE22 device was using. The new port number is 10028.



If you have created any additional devices, the port for the new CE device may differ on your machine. Note down the appropriate port.

```
student@nso-server:~/lab/netsim$ ncs-netsim list
ncs-netsim list for /home/student/lab/netsim

name=CE11 netconf=12022 snmp=11022 ipc=5010 cli=10022 dir=/home/
student/lab/netsim/CE/CE0
name=CE12 netconf=12023 snmp=11023 ipc=5011 cli=10023 dir=/home/
student/lab/netsim/CE/CE1
name=CE21 netconf=12024 snmp=11024 ipc=5012 cli=10024 dir=/home/
student/lab/netsim/CE/CE2
name=CE22 netconf=12025 snmp=11025 ipc=5013 cli=10025 dir=/home/
student/lab/netsim/CE/CE3
name=PE11 netconf=12026 snmp=11026 ipc=5014 cli=10026 dir=/home/
student/lab/netsim/PE/PE0
name=PE22 netconf=12027 snmp=11027 ipc=5015 cli=10027 dir=/home/
student/lab/netsim/PE/PE1
name=CE22v2 netconf=12028 snmp=11028 ipc=5016 cli=10028 dir=/home/
student/lab/netsim/CE22v2/CE22v2
student@nso-server:~/lab/netsim$
```

Activity Verification

You have completed this task when you attain these results:

- You have stopped the CE22 NetSim device and created a new NetSim device instance named CE22v2.

Task 2: Restore Device Configuration

In this task, you will restore the configuration of the faulty old device to the new replacement device.

Activity

Step 1

Connect to the NSO CLI and enter the configuration mode.

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

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

User student last logged in 2024-02-05T20:51:30.63003+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 2

Change the port of the CE22 device to the port used by the replacement device that you created in the previous task.

If no other NetSim devices were added, the port should be 10028. Change the port on the existing device and commit the changes.

```
student@ncs(config)# devices device CE22 port 10028
student@ncs(config-device-CE22)# top
student@ncs(config)# commit
Commit complete.
student@ncs(config)#
```

Step 3

Exit the configuration mode and fetch the SSH keys of the new device.

Use the **ssh fetch-host-keys** action on the CE22 device to update the SSH keys.

```
student@ncs(config)# exit
student@ncs# devices device CE22 ssh fetch-host-keys
result updated
fingerprint {
  algorithm ssh-rsa
  value 70:d0:17:8a:49:25:56:4e:29:a1:36:ec:c0:e8:e0:3c
}
student@ncs#
```

Step 4

Check the synchronization status of the CE22 device.

Since the device has been replaced, it is out-of-sync and an alarm is triggered.

```
student@ncs# devices device CE22 check-sync
result out-of-sync
info got: 8b4d5a2317f421fa0498afac619e0d0c expected:
b6e3bfa0b617ced3b6a54c32dd34aed9

student@ncs# *** ALARM out-of-sync: got:
8b4d5a2317f421fa0498afac619e0d0c expected:
b6e3bfa0b617ced3b6a54c32dd34aed9

student@ncs#
```

Step 5

Push the configuration stored in the NSO CDB to the replaced device.

Use the **sync-to** action on the CE22 devices to push the configuration.

```
student@ncs# devices device CE22 sync-to
result true
student@ncs#
```

Step 6

Exit the NSO CLI.

Use the **exit** command to exit the CLI.

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

Step 7

Connect to the replacement device by using **ncs-netsim cli-c** command.

You can use the **ncs-netsim cli-c <device-name>** command as an alternative to SSH. Because the connection is established by the ncs-netsim tool, there is no need to know the CLI port number and credentials.

```
student@nso-server:~/lab/netnsim$ ncs-netsim cli-c CE22v2

User admin last logged in 2024-02-05T21:02:56.154799+00:00, to nso-
server, from 127.0.0.1 using cli-ssh
admin connected from 100.64.0.11 using ssh on nso-server
CE22v2#
```

Step 8

Verify that the BGP configuration is present on the device.

The BGP configuration should resemble the configuration in the output below.

```
CE22v2# show running-config router
router bgp 65000
  address-family ipv4 unicast vrf vpn1
    neighbor 10.100.0.1 remote-as 65001
  exit-address-family
!
!
CE22v2#
```

Activity Verification

You have completed this task when you attain these results:

- You have restored the CE22 configuration to the replacement device.

Which command do you need to use to push the CE22 device configuration to the replaced device?

- ☐ **device device CE22 fetch-ssh-host-keys**
- ☐ **devices device CE22 check-sync**
- ☐ **devices device CE22 compare-config**
- ☐ **devices device CE22 sync-to**