# **Lab 9 - NETCONF : Network Configuration Protocol**

University of Colorado Boulder Dept. of Computer Science

Professor Levi Perigo, Ph.D.

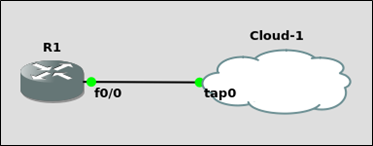
# **Summary**

The Network Configuration Protocol (NETCONF) provides mechanisms to install, manipulate, and delete the configuration of network devices. It is a standard defined by IETF RFC 6241. YANG is its data modeling language. NETCONF aims to help remove the time, cost and manual steps involved in network elements configuration. The objective of this lab is to provide an overview of NETCONF and its operations. Students are encouraged to expand on the topics for additional learning and experiments.

# **Objective 1: Getting started [65 Points]**

# **Problem Statement:**

1. Create the below topology in GNS3 on the NetMan VM.
2. Please use the router image c7200-15.2 image for this lab.



1. Refer this link to configure NETCONF over SSH on the router - <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cns/configuration/15-mt/cns-15-mt-book/netconf-sshv2.html>

Apart from SSH, which other secure transport methods are supported by NETCONF? [5 points]

Another transport method supported by NETCONF is TLS or transport layer security.

1. Establish a NETCONF session with the router. Paste screenshots of the commands – show netconf session and show netconf counters on the router indicating a successful NETCONF session. [20 points]

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

1. Using NETCONF message, get the current running configuration of the router. Paste screenshots of the messages. [5 points] [<https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cns/configuration/xe-3s/cns-xe-3s-book/cns-netconf.html> ]

A screenshot of a computer

Description automatically generated

1. Using NETCONF, get the name of the interfaces on the router. Paste screenshots of the messages. [5 points]

A computer screen with white text

Description automatically generated

1. Using NETCONF, get the running configuration of the interface FastEthernet1/0. Paste screenshots of the messages. [5 points]

A computer screen with white text

Description automatically generated

1. Using NETCONF-

a. Set the hostname of the router to – Lab9Router

b. Create a Loopback Interface 10, and

c. Assign the IP address 10.1.1.1/32 to the interface. Paste screenshots of the messages. [10 points]

A screenshot of a computer

Description automatically generatedA computer screen shot of a computer

Description automatically generated

1. Using NETCONF message, get the running configuration of the interface Loopback 10. Paste screenshots of the messages. [5 points]

A screen shot of a computer

Description automatically generated

1. Using NETCONF, copy the running-config to startup-config. Paste screenshots of the messages. [5 points]

A screen shot of a computer

Description automatically generated

1. How do you check if the router supports candidate configuration datastore? Does this router support it? [5 points]

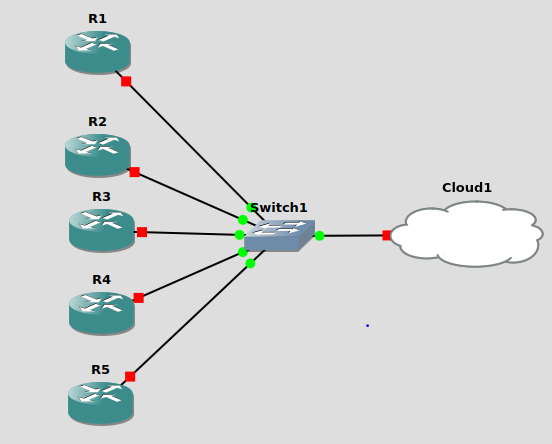


As soon as you ssh to the router you can see its listed capabilities. Datastore is not one of them.

# **Objective 2: NETCONF and Python [70 Points]**

[ncclient](https://pypi.org/project/ncclient/) is a Python library for NETCONF clients. It offers an intuitive API and makes writing network management scripts easier.

1. Create the below topology in GNS3-



1. The configuration to be implemented is present in the file 'lab9-obj2-conf.csv' (It is present in this directory).
2. The objective is to implement the configuration using a Python script and the ncclient library.
3. Only manual configurations that need to be done are configuring management IPs (198.51.100.11-198.51.100.15) and enabling netconf over SSH (username:lab, password:lab123, with privileged EXEC mode).
4. Implement this objective using the shortest (i.e. the most automated) method you can think of. You can use your own CSV file (with the similar configs) or the provided csv file itself. Please submit the CSV file if it is created by you. Please mention what you are using.
5. The final output should be a table printed out, as shown in the csv config file, with the details fetched from the routers.
6. Mention the steps you followed to complete this objective in the below cell.



How I achieved this, is I have a lab8-ob2.py that contains the functions that complete this objective. The cfg() function contains the config for all of the routers, the input for all the parameters of the function comes from the function: sendConfigs(). This function opens the provided CSV file and goes through every row. I exract the needed values with ipaddress library. I then call the cfg function to create the config and then send it to the router via Netconf and the ncclient library. Then to create the pretty table, I did a get-config in ncclient and used regex to extract the needed values in the table. The py file is provided in the assignment as well for future reference. Here is the pretty table:

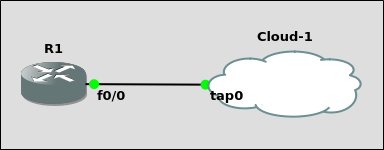
A screenshot of a computer screen

Description automatically generated

# **Objective 3: Troubleshooting [20 points]**

1. The router configuration file to be used in this objective is 'R1.cfg' (It is present in this directory).
2. Create a new project in GNS3 and create this exact topology – use router image c7200 and connect its FastEthernet0/0 interface to VM’s tap adapter.
3. Before starting the devices (i.e. pressing the green play button), import the .cfg file by right-clicking on the router and selecting Import config.
4. The objective is to make NETCONF work for this router. Write down the steps you followed to get it working. [20 points]
   * + 1. First I did a no sh on fa 0/0 because it was shutdown
       2. Next to be able to ping Virus in the first place, I set a static route from my virtual network to the router network. Then on the router I set a static route back to the virtual network
       3. I upped the privilege to 15 for SSH so that Netconf can make changes
       4. Then I removed the access list permissions for fa 0/0 as the ‘in’ prevents some Ips from contacting the router and the ‘out’ prevents ssh and packet filtering.
       5. Next I went into line vty 0 4 and did ‘login local’ and ‘transport input ssh’ which allows for those ssh connections
       6. I then did crypto key generate rsa for sshv2. I did 2048 bits and then did the command: ip ssh version 2. Now regular ssh is working. Time to enable Netconf
       7. Finally what I did to get NETCONF to work for this router is: ‘netconf ssh’. Now it works!





# **Objective 4 (Extra-Credit): NETCONF-YANG [25 Points]**

1. For this objective, you can use the Cisco IOS XR router or any other router that supports the YANG data-model.
2. You can add the router image to gns3 by: [<https://ipcisco.com/how-to-install-cisco-gns3-ios-images/> ]

Note that some router images such as IOS-XR may need KVM acceleration to work [VMware required]. If you do not have VMware setup, consider the following options instead.

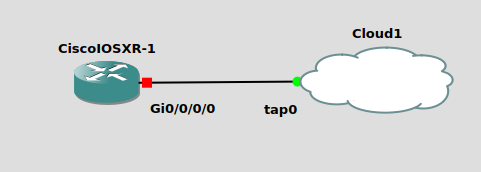
OR

You can also use the Cisco DevNet Sandbox which provides a virtual lab environment after setting up an account [<https://developer.cisco.com/site/sandbox/> ]

OR

Use the containerlab platform to achieve this objective [<https://containerlab.dev/> ]

1. Setup the below Topology with the router you have chosen:

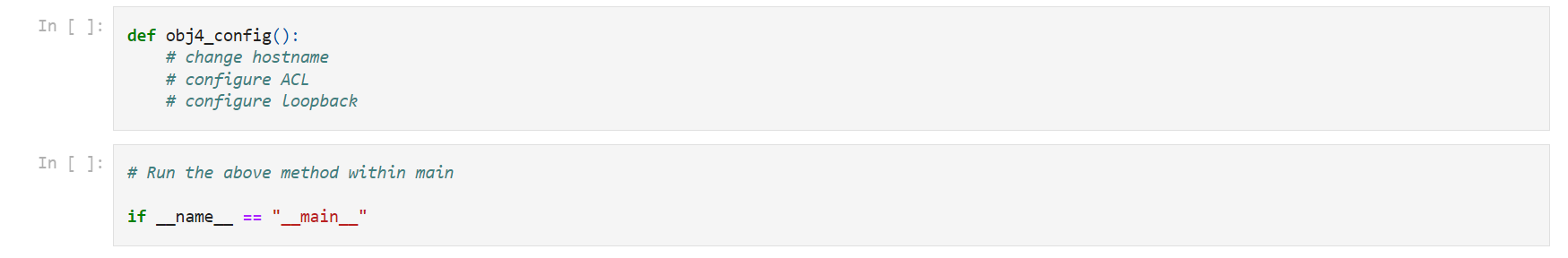


1. Enable NETCONF-YANG over SSH on the Cisco XR/router that supports YANG [<https://www.cisco.com/c/en/us/support/docs/storage-networking/management/200933-YANG-NETCONF-Configuration-Validation.html#anc9>].
2. Paste a screenshot of the output of ‘sh netconf-yang clients’ (for IOS-XR) indicating a successful NETCONF connection [NC version should not be unknown]. [10 Points]
3. Use ncclient to complete the below objectives [<https://ncclient.readthedocs.io/en/latest/> ]

a. Change hostname of router to ‘Lab9\_XR’

b. Configure an access list on the router’s Gi0/0/0 interface to allow inbound traffic only from the host VM’s IP, and

c. Create a loopback interface and assign it an IP of 10.11.12.13/32. Paste relevant screenshots of your script output. [15 Points]



# **Objective 5 – Report Question [5 Points]**

1. In your opinion, what are the advantages of using data models such as YANG or YAML in network automation?
   1. YANG and YAML allow for data to be processed using a certain type of structure. It can standardize the way we send information to the devices and can enable the process of automation.
2. Out of all the network automation tools you used (Netmiko, NAPALM, Ansible, NETCONF, etc.), which one is your favorite and why?
   1. I would say Netmiko is my favorite automation tool I have used because of its ease of use. I found that passing information to the routers via Netmiko took only basic ssh information and I could input any command I wanted via a string or .txt file. While it is not as versatile as NAPALM in terms of vendor, it is easy to pick up for a specific vendor. Ansible is a close second as there built in functions were very straight forward and the ability to structure everything in YAML made the deployment process a bit faster. I do enjoy the error checking capabilities of Ansible when you run a playbook. I found NETCONF to be the worst tool as it was extremely hard to set up and is very strict when it comes to syntax. The lack of ‘if exists’ errors makes it hard to determine whether the NETCONF command is finding what it wants or not.