



UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

Tecnologías de la Información y Comunicación
Programa educativo:

TSU en Infraestructura de Redes Digitales
Área académica:

Programación de Redes
Asignatura:

Unidad III: Programación de Redes

Grupo: GIR0441

NETCONF w/Python: List Capabilities
Laboratorio 7:

Venado Soria German Emiliano
Alumno:

Gabriel Barrón Rodríguez
Docente:

Dolores Hidalgo, C.I.N., Gto., Miércoles 14 de Diciembre de 2022
Lugar y fecha:

Lab – NETCONF w/Python: List Capabilities

Objectives

Part 1: Install the ncclient Python module

Part 2: Connect to IOS XE's NETCONF service using ncclient

Part 3: List the IOS XE's capabilities – supported YANG models

Background / Scenario

Working with NETCONF does not require working with raw NETCONF RPC messages and XML. In this lab you will learn how to use the ncclient Python module to easily interact with network devices using NETCONF. You will learn how to identify which YANG models are supported by the device. This information is helpful when building a production network automation system, that requires specific YANG models to be supported by the given network device.

Required Resources

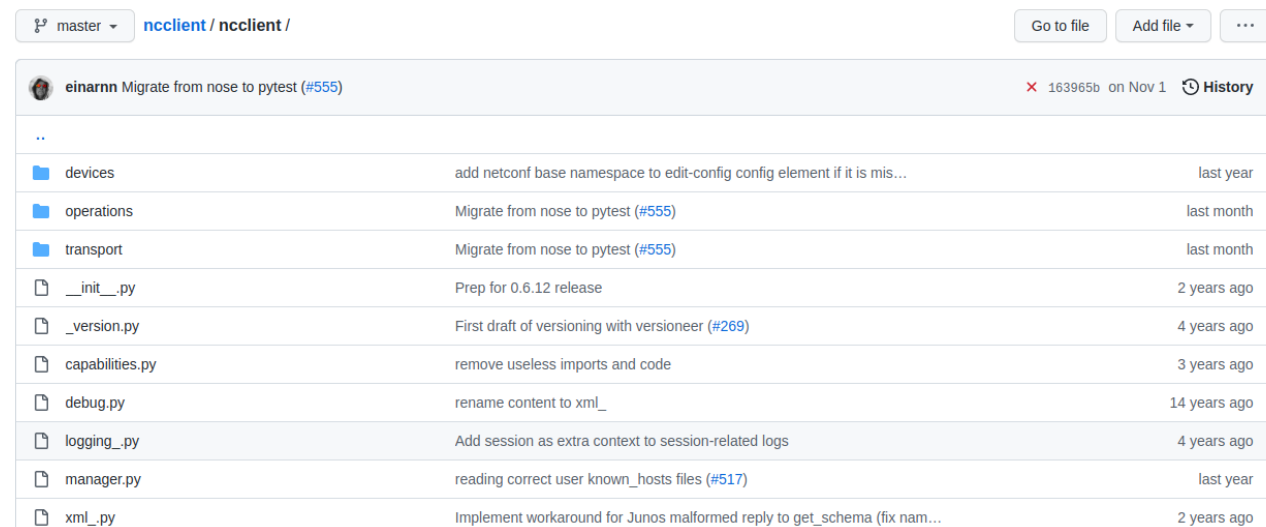
- Access to a router with the IOS XE operating system version 16.6 or higher
- Python 3.x environment

Instructions

Part 1: Install the ncclient Python module

In this part, you will install ncclient module into your Python environment. ncclient is a python module that simplifies NETCONF operations with built in functions that deal with the XML messages and RPC calls.

Explore the ncclient module on the project GitHub repository: <https://github.com/ncclient/ncclient>



The screenshot shows the GitHub repository for ncclient/ncclient. The repository is on the master branch. The commit history is displayed, showing a list of commits with their titles, descriptions, and dates. The commits are listed in descending order of time.

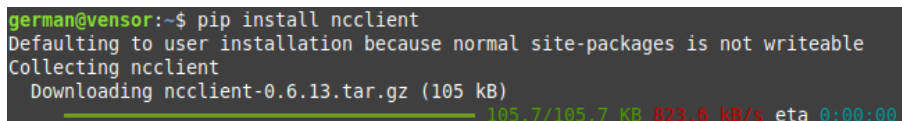
Commit	Description	Time
..		
devices	add netconf base namespace to edit-config config element if it is mis...	last year
operations	Migrate from nose to pytest (#555)	last month
transport	Migrate from nose to pytest (#555)	last month
__init__.py	Prep for 0.6.12 release	2 years ago
_version.py	First draft of versioning with versioneer (#269)	4 years ago
capabilities.py	remove useless imports and code	3 years ago
debug.py	rename content to xml_	14 years ago
logging.py	Add session as extra context to session-related logs	4 years ago
manager.py	reading correct user known_hosts files (#517)	last year
xml.py	Implement workaround for Junos malformed reply to get_schema (fix nam...	2 years ago

Primero, nos pide echar un vistazo al siguiente repositorio en Github.

Step 1: Use pip to install ncclient.

- Start a new Windows command prompt (cmd).
- Install ncclient using pip in the Windows command prompt:

```
pip install ncclient
```

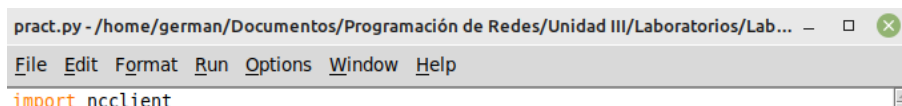


```
german@vensor:~$ pip install ncclient
Defaulting to user installation because normal site-packages is not writeable
Collecting ncclient
  Downloading ncclient-0.6.13.tar.gz (105 kB)
    105.7/105.7 KB 823.6 KB/s eta 0:00:00
```

Para este laboratorio se requiere de instalar el modulo

- Verify that ncclient has been successfully installed. Start Python IDLE and in the interactive shell try to import the ncclient module:

```
import ncclient
```



```
pract.py - /home/german/Documentos/Programación de Redes/Unidad III/Laboratorios/Lab...
File Edit Format Run Options Window Help
import ncclient
```

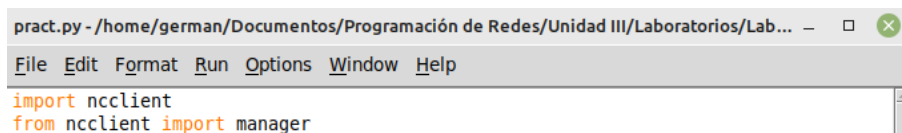
Part 2: Connect to IOS XE's NETCONF service using ncclient

Step 1: Connect to IOS XE's NETCONF service using ncclient.

The ncclient module provides a “manager” class with “connect()” function to setup the remote NETCONF connection. After a successful connection, the returned object represents the NETCONF connection to the remote device.

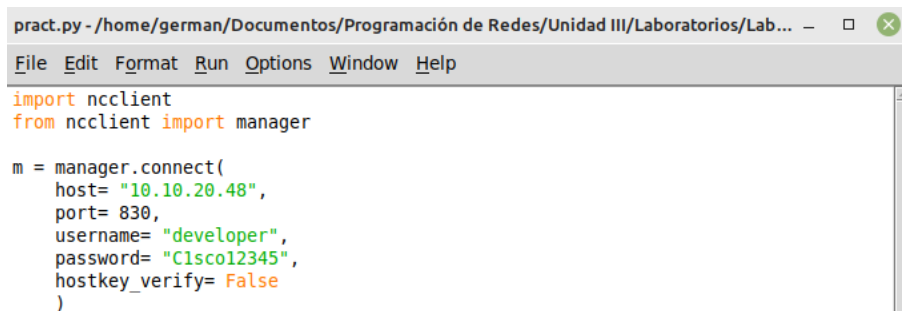
- In Python IDLE, create a new Python script file:
- In the new Python script file editor, import the “manager” class from the ncclient module:

```
from ncclient import manager
```



```
pract.py - /home/german/Documentos/Programación de Redes/Unidad III/Laboratorios/Lab...
File Edit Format Run Options Window Help
import ncclient
from ncclient import manager
```

- Setup an m connection object using the manager.connect() function to the IOS XE device.



```
pract.py - /home/german/Documentos/Programación de Redes/Unidad III/Laboratorios/Lab...
File Edit Format Run Options Window Help
import ncclient
from ncclient import manager

m = manager.connect(
    host= "10.10.20.48",
    port= 830,
    username= "developer",
    password= "Cisco12345",
    hostkey_verify= False
)
```

The parameters of the manager.connect() function are:

- host – the address (host or IP) of the remote device (adjust the IP address to match the router's current address)

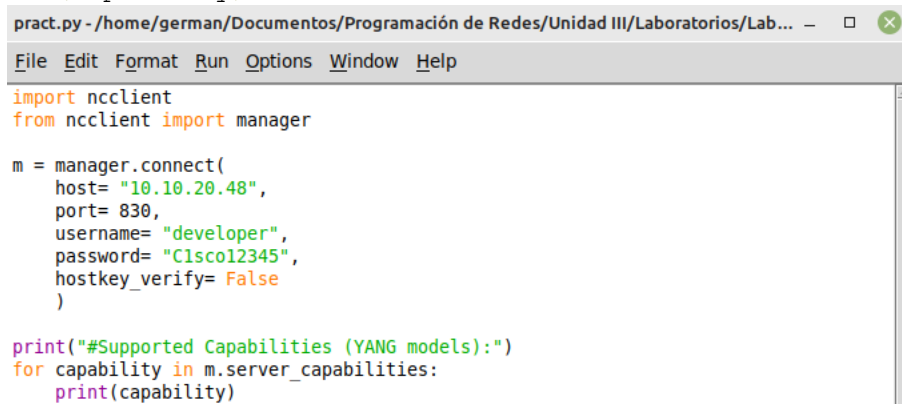
- `port` – the remote port of the NETCONF service
- `username` – remote ssh username (in this lab “cisco” for that was setup in the IOS XE VM)
- `password` – remote ssh password (in this lab “cisco123!” for that was setup in the IOS XE VM)
- `hostkey_verify` – whether to verify the ssh fingerprint (in lab it is safe to set to False, in production environments you should always verify the ssh fingerprints)

Part 3: List the IOS XE’s capabilities – supported YANG models

Step 1: Send show commands and display the output

- a. The `m` object, returned by the `manager.connect()` function that represents the NETCONF remote session. In every NETCONF session, the server first sends its list of capabilities – supported YANG models. With the `ncclient` module, the received list of capabilities is stored in the `m.server_capabilities` list.
- b. Use a for loop and a print function to print the device capabilities:

```
print("#Supported Capabilities (YANG models):")
for capability in m.server_capabilities:
    print(capability)
```



```
pract.py - /home/german/Documentos/Programación de Redes/Unidad III/Laboratorios/Lab... - □ ×
File Edit Format Run Options Window Help

import ncclient
from ncclient import manager

m = manager.connect(
    host= "10.10.20.48",
    port= 830,
    username= "developer",
    password= "Cisco12345",
    hostkey_verify= False
)

print("#Supported Capabilities (YANG models):")
for capability in m.server_capabilities:
    print(capability)
```

Con la ayuda del módulo `ncclient`, podremos obtener la lista de capacidades y almacenarla en la variable `m.server_capabilities`. Además, estaremos utilizando un bucle `for` con su respectiva función de impresión para mostrar las capacidades del dispositivo.

- c. Execute the Python script file to see the results.

```
Python 3.10.6 (main, Nov  2 2022, 18:53:38) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /home/german/Documentos/Programación de Redes/Unidad III/Laboratorios/Lab-7/pract.py
#Supported Capabilities (YANG models):
urn:ietf:params:netconf:base:1.0
urn:ietf:params:netconf:base:1.1
urn:ietf:params:netconf:capability:writable-running:1.0
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:rollback-on-error:1.0
urn:ietf:params:netconf:capability:notification:1.0
urn:ietf:params:netconf:capability:interleave:1.0
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit&also-supported=report-all-tagged
urn:ietf:params:netconf:capability:yang-library:1.0?revision=2016-06-21&module-set-id=730825758336af65af9606c071685c05
http://tail-f.com/ns/netconf/actions/1.0
http://tail-f.com/ns/netconf/extensions
http://cisco.com/ns/cisco-xe-ietf-ip-deviation?module=cisco-xe-ietf-ip-deviation&revision=2016-08-10
http://cisco.com/ns/cisco-xe-ietf-ipv4-unicast-routing-deviation?module=cisco-xe-ietf-ipv4-unicast-routing-deviation&revision=2015-09-11
http://cisco.com/ns/cisco-xe-ietf-ipv6-unicast-routing-deviation?module=cisco-xe-ietf-ipv6-unicast-routing-deviation&revision=2015-09-11
http://cisco.com/ns/cisco-xe-ietf-ospf-deviation?module=cisco-xe-ietf-ospf-deviation&revision=2018-02-09
http://cisco.com/ns/cisco-xe-ietf-routing-deviation?module=cisco-xe-ietf-routing-deviation&revision=2016-07-09
http://cisco.com/ns/cisco-xe-openconfig-acl-deviation?module=cisco-xe-openconfig-acl-deviation&revision=2017-08-25
http://cisco.com/ns/cisco-xe-openconfig-lldp-deviation?module=cisco-xe-openconfig-lldp-deviation&revision=2018-07-25
http://cisco.com/ns/mpls-static/devs?module=common-mpls-static-devs&revision=2015-09-11
```

d. Is the Cisco-IOS-XE-cdp YANG model supported by the device?

R= [Sí, si es compatible.](#)

Investigaciones

- **¿Qué es ncclient?**

ncclient (Netconf client) es una biblioteca de Python que facilita el desarrollo de aplicaciones y secuencias de comandos del lado del cliente en torno al protocolo NETCONF.

- **Características**

- Abstrae gran parte de la complejidad.
- Nos permite trabajar con mayor comodidad.

Conclusiones

Desde mi punto de vista el módulo ncclient, si es una de las librerías más ágiles para poder trabajar remotamente, pues basta solo con generar la sesión remota, sin tener que generar una URL ni indicar el tipo de autenticación, esto ahorra un poco de tiempo y espacio, y en base a sus funciones son fáciles de manejar y abstraer buenos resultados.