

# Lab – CLI Automation with Python using netmiko

## Programación de Redes

UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

NOMBRE

JUAN PABLO PALMA APODERADO

CARRERA

TSU EN INFRAESTRUCTURA DE REDES DIGITALES

DOCENTE

BARRON RODRIGUEZ GABRIEL

NO. CONTROL

1221100259

FECHA

14 DE OCTUBRE DEL 2022

GRUPO

GIR0441

# Instructions

## Part 1: Install the netmiko Python module

In this part, you will install netmiko module into your Python environment. Netmiko is a python module that simplifies ssh CLI connection to network devices. It has built in functionality to identify to execute “exec mode” commands, as well as apply new commands in the running configuration.

### Step 1: Use pip to install netmiko.

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

```
pip install netmiko
```

```
C:\Users\pablo>pip install netmiko
Requirement already satisfied: netmiko in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (4.1.2)
Requirement already satisfied: ntc-templates>=2.0.0 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (3.1.0)
Requirement already satisfied: scp>=0.13.3 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (0.14.4)
Requirement already satisfied: tenacity in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (8.1.0)
Requirement already satisfied: pyserial in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (3.5)
Requirement already satisfied: paramiko>=2.7.2 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (2.12.0)
Requirement already satisfied: textfsm==1.1.2 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (1.1.2)
Requirement already satisfied: setuptools>=38.4.0 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (63.2.0)
Requirement already satisfied: pyyaml>=5.3 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (6.0)
Requirement already satisfied: six in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from textfsm==1.1.2->netmiko) (1.16.0)
Requirement already satisfied: future in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from textfsm==1.1.2->netmiko) (0.18.2)
Requirement already satisfied: cryptography>=2.5 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=2.7.2->netmiko) (38.0.3)
Requirement already satisfied: bcrypt>=3.1.3 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=2.7.2->netmiko) (4.0.1)
Requirement already satisfied: pynacl>=1.0.1 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=2.7.2->netmiko) (1.5.0)
Requirement already satisfied: cffi>=1.12 in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from cryptography>=2.5->paramiko>=2.7.2->netmiko) (1.15.1)
Requirement already satisfied: pycparser in c:\users\pablo\appdata\local\programs\python\python310\lib\site-packages (from cffi==1.12->cryptography>=2.5->paramiko>=2.7.2->netmiko) (2.21)
C:\Users\pablo>
```

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

```
import netmiko
```

## Part 2: Connect to IOS XE's SSH service using netmiko

```
Ejercicios > ssh2.2.py > ...
1  from netmiko import ConnectHandler
2
3  sshCli = ConnectHandler(
4      device_type= 'cisco_ios',
5      host= '10.10.20.70',
6      port= 2221,
7      username= 'admin',
8      password= 'admin'
9  )
10 output = sshCli.send_command("show ip int brief")
11 print(output)
12
```

Tue Dec 6 06:55:18.795 UTC

Interface	IP-Address	Status	Protocol	Vrf-Name
MgmtEth0/RP0/CPU0/0	192.168.122.21	Up	Up	default
GigabitEthernet0/0/0/0	unassigned	Shutdown	Down	default
GigabitEthernet0/0/0/1	unassigned	Shutdown	Down	default
GigabitEthernet0/0/0/2	unassigned	Shutdown	Down	default
GigabitEthernet0/0/0/3	unassigned	Shutdown	Down	default
GigabitEthernet0/0/0/4	unassigned	Shutdown	Down	default

PS C:\Users\pablo\Documents\UTNG Cuatrimestre 4\Programación de Redes\Unidad III\Ejercicios>

## Part 3: Use netmiko to gather information from the device

### Send show commands and display the output

- Using the `sshCli` object, returned by the `ConnectHandler()` function that represents the ssh cli remote session, send some “show” command and print the output. Use the `send_command()` function of the `sshCli` object with a string parameter that represents the command you wish to execute in the exec mode:

```
output = sshCli.send_command("show ip int brief")
print("show ip int brief:\n{}\n".format(output))
```

- Verify the data type of the “output” variable. How would you extract the IP address and the Interface Name into variables? What if there were multiple interfaces?

## Part 4: Use netmiko to alter configuration on the device

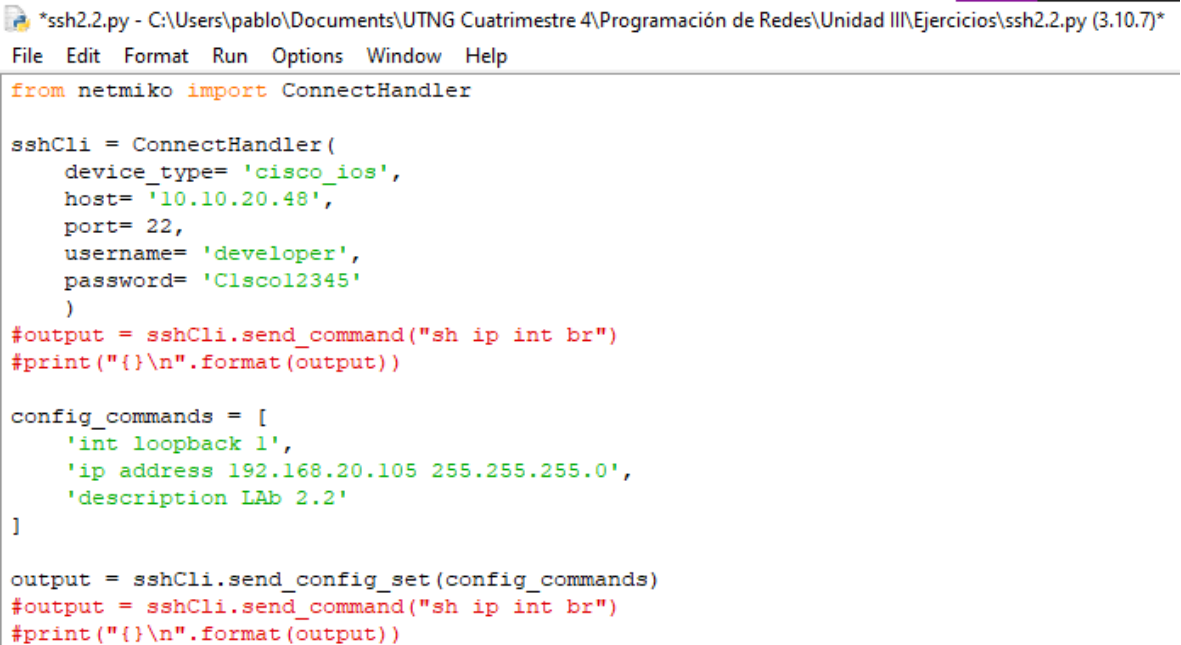
In the following steps, you will alter the configuration of the device by creating new loopback interfaces.

## Create a new loopback interface

Using the `sshCli` object, returned by the `ConnectHandler()` function that represents the ssh cli remote session, send some configuration command and print the output. Use the `send_config_set()` function of the `sshCli` object with a list parameter including the configuration commands as strings you wish to execute in the exec mode:

```
config_commands = [
    'int loopback 1',
    'ip address 2.2.2.2 255.255.255.0',
    'description WHATEVER'
]
```

```
output = sshCli.send_config_set(config_commands)
```



The screenshot shows a Python script editor window titled `*ssh2.2.py - C:\Users\pablo\Documents\UTNG Cuatrimestre 4\Programación de Redes\Unidad III\Ejercicios\ssh2.2.py (3.10.7)*`. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The script content is as follows:

```
from netmiko import ConnectHandler

sshCli = ConnectHandler(
    device_type= 'cisco_ios',
    host= '10.10.20.48',
    port= 22,
    username= 'developer',
    password= 'C1scol2345'
)

#output = sshCli.send_command("sh ip int br")
#print("{}\n".format(output))

config_commands = [
    'int loopback 1',
    'ip address 192.168.20.105 255.255.255.0',
    'description LAB 2.2'
]

output = sshCli.send_config_set(config_commands)
#output = sshCli.send_command("sh ip int br")
#print("{}\n".format(output))
```

*Ejecute el archivo de script de Python y verifique los resultados*

### Preguntas

*¿Por qué la salida de "show ip int brief" no incluye la interfaz "loopback1"?*

Por qué aún dentro del script después de configurar la interfaz lookback no se coloca la línea que muestra las interfaces creadas dentro del router.

*Escriba sus respuestas aquí*

*¿Cómo ejecutar y mostrar la salida del comando "show ip int brief" después de crear las interfaces de loopback?*

```
===== RESTART: C:\Users\pablo\Documents\UTNG Cuatrimestre 4\Programación
Interface      IP-Address      OK? Method Status      Protocol
GigabitEthernet1  10.10.20.48     YES NVRAM  up          up
GigabitEthernet2  unassigned      YES NVRAM  administratively down down
GigabitEthernet3  unassigned      YES NVRAM  administratively down down
Loopback1       192.168.20.105  YES manual up          up
```

*Escriba sus respuestas aquí*

Muestra de la interfaz creada pues es la dirección 192.168.20.105 que esta prendida.

*fin de preguntas*

*Fin de las preguntas*

*Agregue código para crear una nueva interfaz de bucle invertido (loopback2) con la misma dirección IP que en la interfaz de bucle invertido existente, solo que con una descripción diferente.*

*Ejecute el archivo de script de Python y verifique los resultados.*

*Preguntas*

*¿Se creó correctamente la nueva interfaz loopback2?*

*Escriba su respuesta aquí*

*¿Se aceptó, se aceptó parcialmente o se rechazó el nuevo cambio de configuración?*

## **¿Que es netmiko?**

Netmiko es una librería de redes multivendedores basada en Paramiko, que es una librería estandar para las conexiones ssh Python. Con Netmiko como base, se pueden realizar programas y scripts que faciliten y mejoren la administracion de los equipos de redes.

## **¿Para que sirve netimiko?**

Netmiko es una librería de Python open-source basada en otra librería que se llama Paramiko. Esta librería nos va a permitir administrar nuestros dispositivos fácilmente estableciendo conexiones SSH.

## **Netmiko multifuncional**

Netmiko al ser considerada una biblioteca multiplataforma tiene la versatilidad de poder operar dentro de una gran variedad de dispositivos de red.

- Arista vEOS
- Cisco ASA
- Cisco IOS
- Cisco IOS-XE

- Cisco IOS-XR
- Cisco NX-OS
- Cisco SG300
- HP ProCurve
- Juniper Junos
- Linux

### Comandos Más utilizados dentro de netmiko.

**1. net\_connect.send\_command() :**

Permite el envío de un comando a cualquier dispositivo de la red a través del canal.

Ejemplo: output = net\_connect.send\_command('show ip int brief')

**2. net\_connect.send\_config\_set():**

Permite el envío de los comandos de configuración a los dispositivos remotos.

**3. net\_connect.send\_config\_from\_file() :**

Permite el envío de los comandos de configuración a los dispositivos remotos, cargados desde un archivo.

**4. net\_connect.save\_config() :**

Permite guardar la configuración en ejecución en la configuración inicial.

**5. net\_connect.enable():**

Permite ingresar al modo de habilitación.

**6. net\_connect.disconnect():**

Realiza el cierre de la conexión.

### Ventajas de trabajar con netmiko

- Facilidad para lograr establecer exitosamente una conexión SSH con el dispositivo de red deseado.
- Ayuda a simplificar la recuperación de los datos salientes, así como también facilita la ejecución de los diversos comandos show,
- Permite simplificar la ejecución de los múltiples comandos de configuración existentes, incluyendo también las probables acciones de confirmación.
- Versatilidad ya que puede operar dentro de una gran variedad de plataformas y diversos proveedores de redes.
- Netmiko tiene también la gran virtud de poder admitir a más de conexiones SSH, conexiones Telnet, conexiones en serie, e inclusive Secure Copy.