

Instalación de Netmiko


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\info>pip install netmiko
Requirement already satisfied: netmiko in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (4.1.2)
Requirement already satisfied: paramiko>=2.7.2 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (2.11.0)
Requirement already satisfied: tenacity in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (8.1.0)
Requirement already satisfied: ntc-templates>=2.0.0 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (3.1.0)
Requirement already satisfied: setuptools>=38.4.0 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (63.2.0)
Requirement already satisfied: scp>=0.13.3 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (0.14.4)
Requirement already satisfied: textfsm==1.1.2 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (1.1.2)
Requirement already satisfied: pyserial in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (3.5)
Requirement already satisfied: pyyaml>=5.3 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from netmiko) (6.0)
Requirement already satisfied: future in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from textfsm==1.1.2->netmiko) (0.18.2)
Requirement already satisfied: six in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from textfsm==1.1.2->netmiko) (1.16.0)
Requirement already satisfied: pynacl>=1.0.1 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=2.7.2->netmiko) (1.5.0)
Requirement already satisfied: bcrypt>=3.1.3 in c:\users\info\appdata\local\programs\python\python310\lib\site-packages (from paramiko>=2.7.2->netmiko) (4.0.1)
```

Importar Netmiko

```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import netmiko
>>> |
```

Conéctate al servicio SSH de IOS XE usando netmiko.

 ssh.py - E:/UTNG/TSU/Cuarto Cuatrimestre/barron/Unidad III/ssh.py (3.10.7)

File Edit Format Run Options Window Help

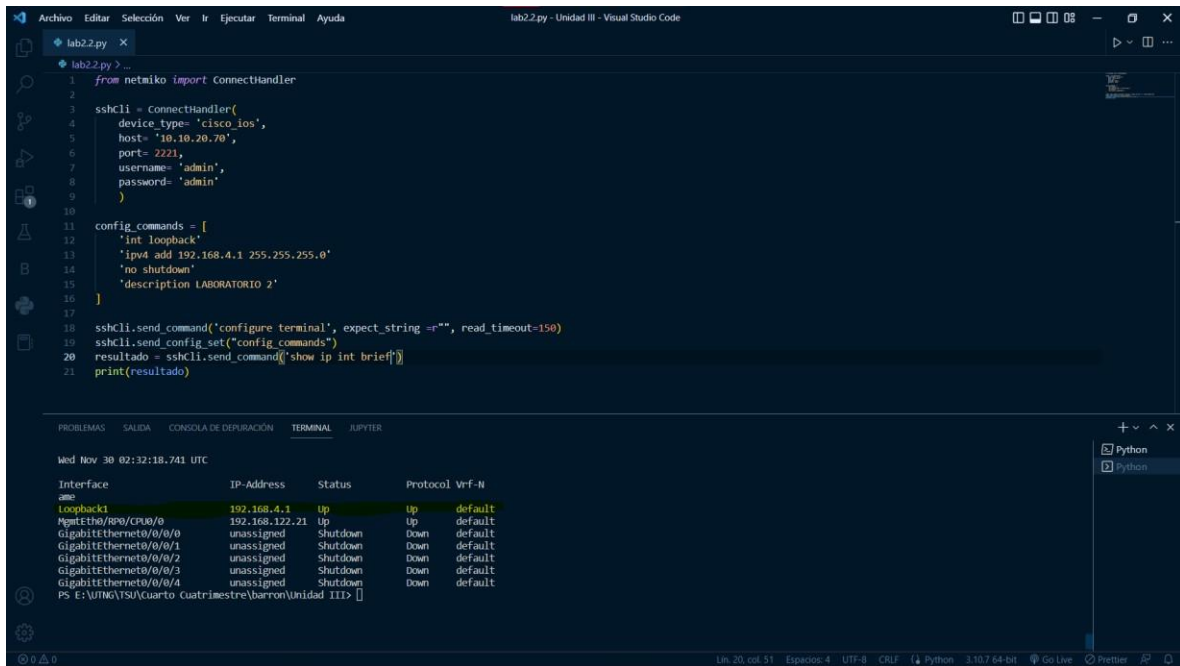
```
from netmiko import ConnectHandler

sshCli = ConnectHandler(
    device_type= 'cisco_ios',|
    host= '10.10.20.70',
    port= 2221,
    username= 'admin',
    password= 'admin'
)
output = sshCli.send_command("show ip int brief")
#print("show ip int brief:\n{}\n".format(output))
print(output)
```

Salida:

```
>>> ===== RESTART: E:/UTNG/TSU/Cuarto Cuatrimestre/barron/Unidad III/ssh.py =====
Thu Nov 24 15:55:46.504 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
>>>
```



```
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
11 config_commands = [
12     'int loopback',
13     'ip add 192.168.4.1 255.255.255.0',
14     'no shutdown',
15     'description LABORATORIO 2'
16 ]
17
18 sshCli.send_command('configure terminal', expect_string='', read_timeout=150)
19 sshCli.send_config_set("config_commands")
20 resultado = sshCli.send_command("show ip int brief")
21 print(resultado)
```

Wed Nov 30 02:32:18.741 UTC

Interface	IP-Address	Status	Protocol	Vrf-N
Loopback1	192.168.4.1	Up	Up	default
Management0/0/0	192.168.122.21	Up	Up	default
GigabitEthernet0/0/0	unassigned	Shutdown	Down	default
GigabitEthernet0/0/1	unassigned	Shutdown	Down	default
GigabitEthernet0/0/2	unassigned	Shutdown	Down	default
GigabitEthernet0/0/3	unassigned	Shutdown	Down	default
GigabitEthernet0/0/4	unassigned	Shutdown	Down	default

PS E:\WING\TSU\Cuarto Cuatrimestre\harron\Unidad III>

Conclusiones:

Netmiko es una libreria de redes multivendedores basada en Paramiko, que es una libreria estandar para las conexiones ssh Python. Con Netmiko como base, tienen la posibilidad de hacer programas y scripts que posibiliten y mejoren la administracion de los grupos de redes. Netmiko incluye como opcion la utilización de TextFSM para parsear la salida de los comandos y facilitar su uso subsiguiente.

Los propósitos de esta biblioteca son los siguientes:

- °Establezca con éxito una conexión SSH al dispositivo.
- °Simplifique la ejecución, recuperación y formato de los comandos de presentación.
- °Simplifique la ejecución de los comandos de configuración.
- °Resuma gran parte de la mecánica de bajo nivel de interacción con dispositivos.
- °Proporcione una API uniforme (relativamente) para interactuar con dispositivos.
- °Haga lo anterior en un amplio conjunto de proveedores y plataformas de redes.

Esto requiere que pasemos en ciertos argumentos, a saber, nuestro:

- tipo de dispositivo
- host (hostname o IP)
- nombre de usuario
- contraseña

Otros comandos que se pueden enviar con Netmiko:

Ejecución del comando show:

```
command = "show ip int brief"
```

```
command = "show ip arp"
```

```
command = "show ip int brief"
```

```
command = "show run | s interfaces"
```

Ajuste del factor retardo:

```
command = "copy flash:c880data-universalk9-mz.155-3.M8.bin flash:test1.bin"
```

Manejo de comandos que provocan tiempo ():

```
command = "del flash:/test3.txt"
```

Cambios de configuración:

```
commands = ["logging buffered 100000"]
```

La configuración cambia desde un archivo:

```
cfg_file = "config_changes.txt"
with ConnectHandler(**device1) as net_connect:
    output = net_connect.send_config_from_file(cfg_file)
    output += net_connect.save_config()
```