

## Lab – CLI Automation with Python using netmiko

### Part 1: Instala el netmiko Módulo Python

En esta parte, instalará netmiko módulo en su entorno de Python. Netmiko es un módulo de python que simplifica ssh Conexión CLI a dispositivos de red. Tiene una funcionalidad integrada para identificar la ejecución de comandos en "modo exec", así como para aplicar nuevos comandos en la configuración en ejecución.

Explorar el netmiko módulo en el repositorio del proyecto GitHub: <https://github.com/ktbyers/netmiko>

#### Step 1: Usa pip para instalar netmiko .

- Inicio un nuevo símbolo del sistema de Windows ( cmd ).
- Instalar netmiko usando pip en el símbolo del sistema de Windows:

pip install netmiko

```
C:\Users\usuario>pip install netmiko
Collecting netmiko
  Downloading netmiko-4.1.2-py3-none-any.whl (196 kB)
----- 196.8/196.8 kB 1.5 MB/s eta 0:00:00
Requirement already satisfied: setuptools>=38.4.0 in c:\users\usuario\appdata\local\programs\python\python311\lib\site-packages (from netmiko) (65.5.0)
Collecting paramiko>=2.7.2
  Downloading paramiko-2.12.0-py2.py3-none-any.whl (213 kB)
----- 213.1/213.1 kB 4.3 MB/s eta 0:00:00
Collecting scp>=0.13.3
  Downloading scp-0.14.4-py2.py3-none-any.whl (8.6 kB)
Collecting tenacity
  Downloading tenacity-8.1.0-py3-none-any.whl (23 kB)
Collecting pyyaml>=5.3
  Downloading PyYAML-6.0-cp311-cp311-win_amd64.whl (143 kB)
----- 143.2/143.2 kB ? eta 0:00:00
Collecting textfsm==1.1.2
  Downloading textfsm-1.1.2-py2.py3-none-any.whl (44 kB)
----- 44.7/44.7 kB ? eta 0:00:00
Collecting ntc-templates>=2.0.0
  Downloading ntc_templates-3.1.0-py3-none-any.whl (332 kB)
----- 332.4/332.4 kB ? eta 0:00:00
Collecting pyserial
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
----- 90.6/90.6 kB ? eta 0:00:00
```

- Comprueba eso netmiko ha sido instalado exitosamente. Inicie Python IDLE y en el shell interactivo intente importar el netmiko módulo:

**import** netmiko

```
C:\Users\usuario>python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import netmiko
>>> _
```

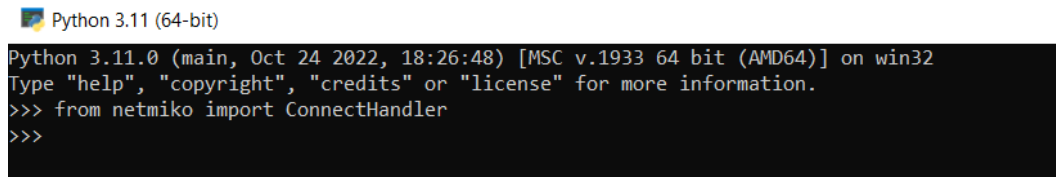
### Part 2: Conéctese al servicio SSH de IOS XE usando netmiko

#### Conéctese al servicio SSH de IOS XE usando netmiko .

los netmiko módulo proporciona un "ConnectHandler()" función para configurar el control remoto ssh conexión. Después de una conexión exitosa, el objeto devuelto representa la conexión ssh cli al dispositivo remoto.

- En Python IDLE, cree un nuevo archivo de secuencia de comandos de Python:
- En el nuevo editor de archivos de secuencias de comandos de Python, import "ConnectHandler()" función de la netmiko módulo:

**from** netmiko **import** ConnectHandler



```
Python 3.11 (64-bit)
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from netmiko import ConnectHandler
>>>
```

- Configurar un sshCli objeto de conexión usando "ConnectHandler()" función al dispositivo IOS XE.

```
sshCli = ConnectHandler(
    device_type='cisco_ios',
    host='192.168.56.101',
    port=22,
    username='cisco',
    password='cisco123!'
)
```

### Part 3: Usar netmiko para recopilar información del dispositivo

- Ejecute el archivo de script de Python para ver los resultados.



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import netmiko
>>>
===== RESTART: C:\Users\jeoq1\Documents\2.2.py =====
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
>>>
```

### Part 4: Usar netmiko para modificar la configuración en el dispositivo

En los siguientes pasos, modificará la configuración del dispositivo mediante la creación de nuevas interfaces de loopback.

#### Crear una nueva interfaz de loopback

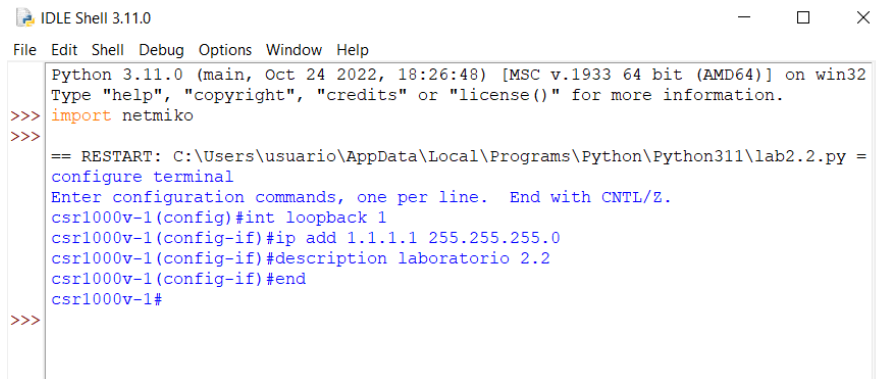
Utilizando el sshCli objeto, devuelto por el Controlador de conexión ( ) función que representa la ssh sesión remota cli, envíe algún comando de configuración e imprima la salida. Utilizar el enviar\_config\_establecer ( ) función de la sshCli objeto con un parámetro de lista que incluye los comandos de configuración como cadenas que desea ejecutar en el modo exec:

```
config_commands = [
    'int loopback 1',
    ' ip dirección 2.2.2.2 255.255.255.0',
]
```

```
'descripción LO QUE SEA'
    ]

producción = sshCli . enviar_config_set ( config_commands )
```

Ejecute el archivo de script de Python y verifique los resultados



```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import netmiko
>>>
== RESTART: C:\Users\usuario\AppData\Local\Programs\Python\Python311\lab2.2.py =
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
csr1000v-1(config)#int loopback 1
csr1000v-1(config-if)#ip add 1.1.1.1 255.255.255.0
csr1000v-1(config-if)#description laboratorio 2.2
csr1000v-1(config-if)#end
csr1000v-1#
>>>
```

Cómo ejecutar y mostrar la salida del "show ip int brief" después de que se crearon las interfaces de loopback?  
Para esto debemos colocar el código dado anteriormente para revisar las interfaces que se están utilizando.

### Conclusiones:

Que es netmiko?

Netmiko es una librería de redes multivendedores basada en Paramiko, que es una librería estándar para las conexiones ssh Python. Con Netmiko como base, se pueden realizar programas y scripts que faciliten y mejoren la administración de los equipos de redes. Netmiko incluye como opción el uso de TextFSM para parsear la salida de los comandos y facilitar su uso posterior. Para mayor información visitar la web de Netmiko.

Las ventajas de Netmiko sobre Paramiko son:

- Conéctese automáticamente a través de SSH a dispositivos de red.
- Proporciona una ejecución más sencilla de los comandos de shows y la salida de datos.
- Proporciona una funcionalidad más sencilla para los comandos de configuración, incluidas las acciones de confirmación.
- Soporte multidispositivo en plataformas y proveedores de dispositivos de red.

Para finalizar es de gran ayuda saber que tenemos este tipo de librerías para conectarnos bajo SSH a un dispositivo en específico y poder observar su configuración entre otras cosas del mismo dispositivo.

Investigando un poco más encontré algunos otros comandos que podrían funcionar para investigar más allá del dispositivo seleccionado.

Métodos de uso común de Netmiko:

- `net_connect.send_command ()`: esta función envía comandos por un canal de red y devuelve la salida según el patrón.
- `net_connect.send_command_timing ()`: devuelve la salida basada en el tiempo de un comando enviado por un canal de red.
- `net_connect.send_config_set ()`: aplica los ajustes de configuración a los dispositivos remotos.
- `net_connect.send_config_from_file ()`: aplica los ajustes de configuración de un archivo externo
- `net_connect.save_config ()`: exporta y guarda la configuración en ejecución como configuración de inicio.
- `net_connect.enable ()`: consulta el dispositivo para activar el modo de habilitación.
- `net_connect.find_prompt ()` - Devuelve el indicador actual del enrutador
- `net_connect.commit ()`: ejecuta comandos de confirmación en dispositivos como Juniper e IOS-XR
- `net_connect.disconnect ()` - Termina las sesiones
- `net_connect.write_channel ()` - Habilita escritura de bajo nivel
- `net_connect.read_channel ()`: habilita la lectura de bajo nivel.