

Prática - Automação com Netmiko/SSH

1 Netmiko

O Netmiko é uma biblioteca de código aberto, que busca simplificar a execução de comandos por meio de SSH, que consiste em uma alternativa mais segura em comparação com o uso de Telnet.

Como para realizar as configurações em um dispositivo, faz necessário executar muitas linhas de comando, que são repetitivas, a biblioteca Netmiko abstrai boa parte dos comandos necessários (ex: *enable*, *conf t* etc).

Outra vantagem é o suporte a vários fabricantes (lista ainda crescendo), de modo que pode-se escrever scripts para automatizar configurações em equipamentos de diferentes fabricantes, usando a mesma biblioteca.

Mais informações sobre a biblioteca podem ser encontradas em:

<https://pynet.twb-tech.com/blog/netmiko-python-library.html>

Github da biblioteca:

<https://github.com/ktbyers/netmiko>

Para instalar a biblioteca, use o comando a seguir no terminal:

```
1 pip3 install netmiko
```

2 Configuração do ambiente

Para a realização desse roteiro de prática, monte o cenário da figura a seguir no GNS3, utilizando as imagens do Switch e Router que foram passadas.

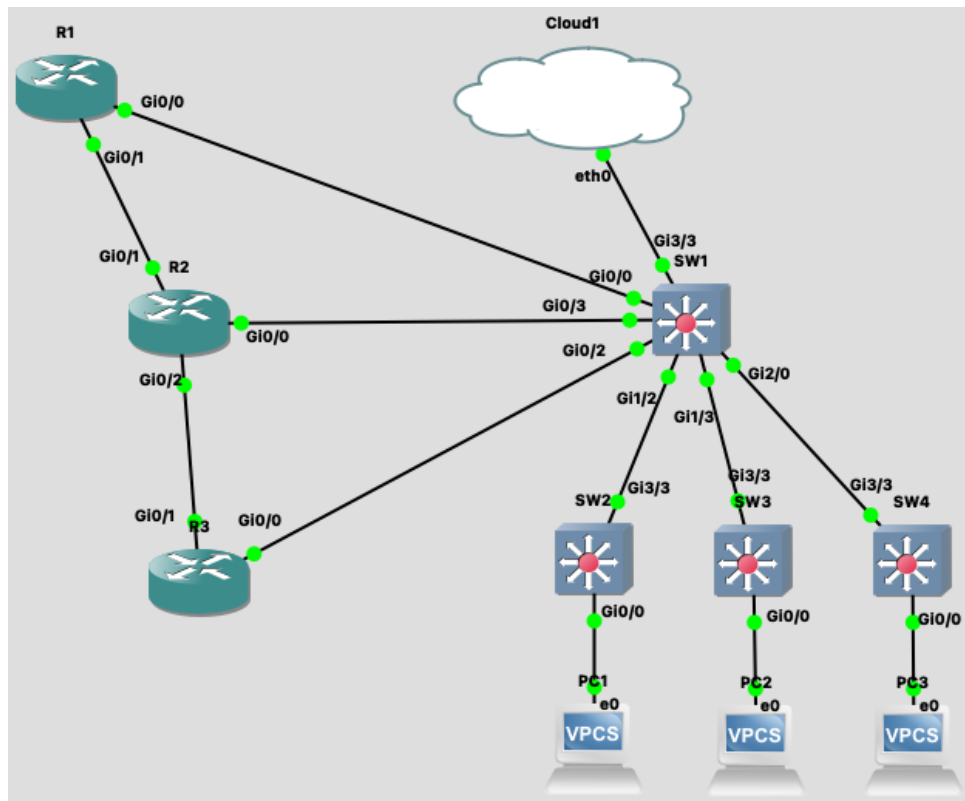
Configure os endereços IP nos dispositivos, seguindo o que foi feito na prática anterior.

2.1 Habilitação do SSH nos Dispositivos

Primeiro, deve-se colocar o dispositivo dentro de um domínio (“ifpb.edu.br” no exemplo). Depois, deve-se criar uma chave para uso do SSH (serão usados 1024 bits no exemplo). Além disso, deve-se fazer a configuração do login local nas interfaces de linha vty e a criação do usuário, como já feito na prática anterior.

```
1 enable
2 conf t
3 ip domain-name ifpb.edu.br
4 crypto key generate rsa
5 end
```

Após isso, tente realizar conexão SSH com um dos dispositivos para garantir que a configuração foi feita de maneira correta.



3 Exemplos para Teste do Netmiko

Execute os exemplos a seguir para testar e aprender a usar os principais métodos da biblioteca (sempre consulte a documentação caso precise de informação adicional ou para conhecer outros métodos oferecidos pela biblioteca).

3.1 Exemplo Inicial com Netmiko

O exemplo a seguir realizar a conexão via SSH com o roteador, utilizando as informações que foram colocadas como parâmetro na criação do *ConnectHandler*. Após a conexão, pode-se enviar comandos individuais, utilizando o método *send_command*. Perceba que nesse exemplo, foi colocado apenas o comando “*show ip interface brief*”, de modo que os outros comandos necessários para entrar em modo privilegiado e depois em modo de configuração são abstraídos pela biblioteca.

```

1 from netmiko import ConnectHandler
2
3 R1 = {
4     "device_type": "cisco_ios",
5     "host": "192.168.1.101",
6     "username": "ruan",
7     "password": "gomes",
8 }
9
10 connect = ConnectHandler(**R1)
11
12 output = connect.send_command("show ip interface brief")
13 print(output)

```

3.2 Lista de Comandos com Netmiko

Muitas vezes, faz-se necessário enviar uma lista de comandos para o dispositivo. Para facilitar, a biblioteca oferece um método chamado *send_config_set*, que recebe como parâmetro uma lista de comandos a serem enviados e executados pelo dispositivo, como exemplificado no código a seguir, que realiza a configuração da lista de controle de acesso do dispositivo.

```
1 from netmiko import ConnectHandler
2
3 R1 = {
4     "device_type": "cisco_ios",
5     "host": "192.168.1.101",
6     "username": "ruan",
7     "password": "gomes",
8 }
9
10 connect = ConnectHandler(**R1)
11
12 cfg_list = [
13     "ip access-list extended TEST1",
14     "permit ip any host 1.1.1.1",
15     "permit ip any host 1.1.1.2",
16     "permit ip any host 1.1.1.3",
17     "permit ip any host 1.1.1.4",
18     "permit ip any host 1.1.1.5",
19 ]
20 cfg_output = connect.send_config_set(cfg_list)
21
22 print(cfg_output)
23
24 connect.save_config()
```

O método *save_config* salva o conteúdo de running-config em startup-config de modo a persistir as configurações no dispositivo.

3.3 Conectando com Múltiplos Dispositivos com Netmiko

Para realizar a configuração de vários dispositivos em um *loop*, deve-se criar um objeto *ConnectHandler* para cada um. Pode-se adicionar cada objeto criado em uma lista e iterar sobre a lista, para realizar a conexão e execução de comandos em cada dispositivo, como exemplificado no código a seguir, que coloca na tela os *hostnames* de cada roteador, usando o método *find_prompt*.

```
1 from netmiko import ConnectHandler
2
3 R1 = {
4     "device_type": "cisco_ios",
5     "host": "192.168.1.101",
6     "username": "ruan",
7     "password": "gomes",
8 }
9
10 R2 = {
11     "device_type": "cisco_ios",
12     "host": "192.168.2.102",
13     "username": "ruan",
14     "password": "gomes",
15 }
16
17 R3 = {
18     "device_type": "cisco_ios",
```

```

17     "host": "192.168.3.103",
18     "username": "ruan",
19     "password": "gomes",
20     }
21
22 for routers in (R1, R2, R3):
23     connect = ConnectHandler(**routers)
24     print(connect.find_prompt())
25     connect.disconnect()
26
27 print("Script finalizado")

```

4 Automação da Criação de VLANs com Netmiko

Deve-se criar um código Python, usando a biblioteca de Netmiko, para criar três VLANs nos Switches S1, S2 e S3, listadas a seguir:

- VLAN 2, de nome “Alunos”
- VLAN 3, de nome “Professores”
- VLAN 4, de nome “Técnicos”

No código, deve-se criar uma lista com os objetos *ConnectHandler* de cada *Switch* a ser configurado. Também deve-se criar uma lista com tuplas, contendo as informações das VLANs a serem criadas (ex: [(2, 'Alunos'), ...]). Deve-se criar todas as VLANs representadas nessa lista em todos os Switches representados na lista de Switches.

Utilize como base o código a seguir, para escrever o script de configuração das VLANs.

```

1     enable
2     conf t
3     vlan 2
4     name Alunos
5     exit

```

Execute *show vlan brief*, também usando a biblioteca Netmiko, para verificar se as VLANs foram de fato criadas e coloque na tela a saída desse comando. Por fim, salve as configurações no dispositivo usando o método *save_config*.

5 Script para Habilitação do SNMP em todos os Roteadores

Utilizando como base o código desenvolvido na última prática, crie agora um script, usando a biblioteca Netmiko, que possua uma lista com os objetos *ConnectHandler* de cada Roteador (nesse caso, serão os roteadores R1, R2 e R3) e uma lista de strings, representando os nomes de comunidade para cada um dos roteadores. Deve-se então habilitar o SNMP em todos os roteadores presentes na lista de roteadores, utilizando as strings de comunidade presentes na lista.

Após realizar a habilitação, deve-se criar um loop que fica consultando a quantidade de octetos que saíram e a quantidade de octetos que entraram na porta G0/0 em cada um dos roteadores, a cada 10 segundos. Esses valores devem ser imprimidos na tela, no formato a seguir:

```
1
2     R1 — in: 100 out: 120
3     R2 — in: 132 out: 133
4     R3 — in: 112 out: 111
5
```

Para realizar as consultas, use a biblioteca SNMP para Python. Olhar o exemplo para o SNMPv2 do link a seguir, para usar como base:

<https://pysnmp.readthedocs.io/en/latest/examples/hlapi/v3arch/asyncore/sync/manager/cmdgen/snmp-versions.html>

6 Entrega da Prática

Deve-se entregar os códigos python produzidos.