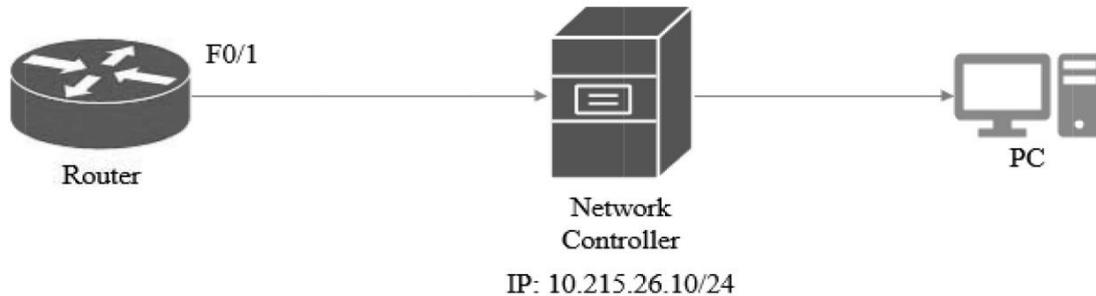


LAB - VIẾT CHƯƠNG TRÌNH LẤY DANH SÁCH CÁC THIẾT BỊ TRONG CONTROLLER (APIC-EM) CỦA CISCO

I. Sơ đồ



Hình 1: Sơ đồ bài Lab

II. Mô tả

Sơ đồ bài thực hành gồm một Router và một PC được đấu nối với nhau như hình 1. Trên sơ đồ này, học viên thực hiện kết nối đến Controller, thêm thiết bị vào Controller, viết code để lấy danh sách thông tin các thiết bị bằng ngôn ngữ Python. Máy PC phải đáp ứng yêu cầu đã cài đặt trạm làm việc cho developer.

III. Yêu cầu kỹ thuật

Học viên thực hiện đấu nối thiết bị, thực hiện một số cấu hình cơ bản trên Router như đặt hostname, password console, đặt địa chỉ IP như hình 1. Cài đặt thư viện requests, tabulate trên máy tính. Thực hiện cấu hình cho phép SSH, SNMP trên Router. Kết nối đến Controller, thêm thiết bị vào Controller. Get ticket và tạo hàm Get thông tin.

IV. Các bước thực hiện

Bước 1: Cấu hình IP, SSH, SNMP trên router

```
##Cấu hình IP ##
Router#configure terminal
Router(config)#interface f0/1
Router(config-if)#ip address dhcp
## cấu hình SSH ##
Router(config)#username admin password 123
```

```
Router(config)#ip domain-name 192.168.1.1
Router(config)#enable password vnpro
Router(config)#line vty 0 4
Router(config-line)#password cisco
Router(config-line)#login
Router(config-line)#exit
Router(config)#crypto key generate rsa
1024
Router(config)#ip ssh version 2
## Cấu hình SNMP ##
Router(config)#snmp-server community public rw
```

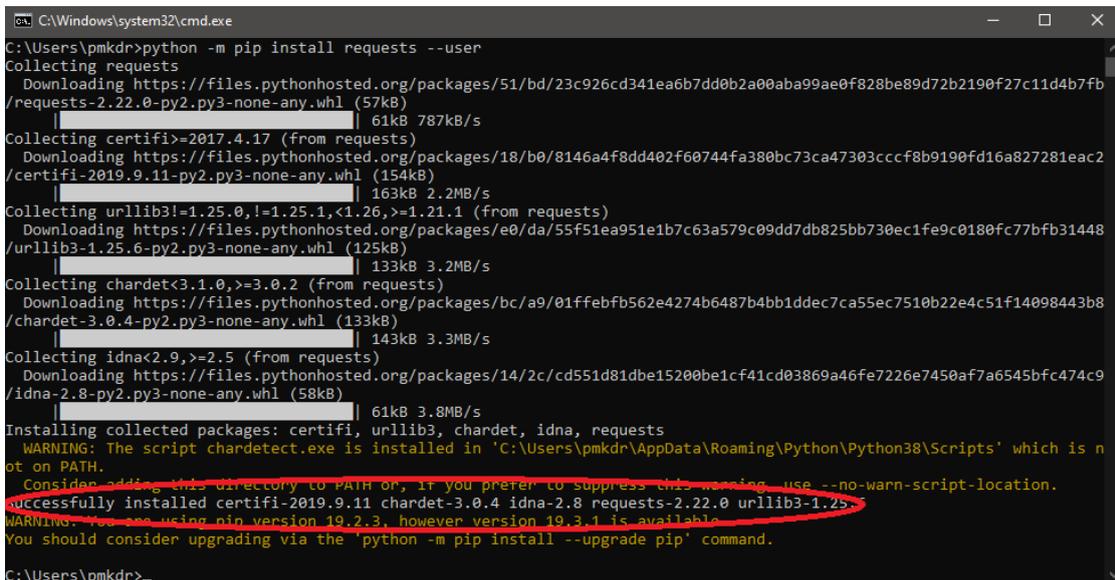
Bước 2: Thêm thiết bị vào Controller

Làm theo các công đoạn tại bước 2 của Lab 8.

Bước 3: Cài đặt thư viện requests, tabulate

Bấm tổ hợp phím Windows +R để mở Command Line (cmd). Để cài đặt thư viện request, trong màn hình cmd, dùng lệnh:

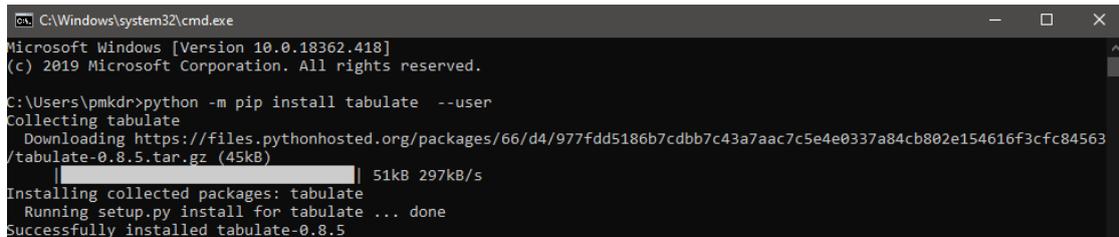
```
python -m pip install requests --user
```



```
C:\Windows\system32\cmd.exe
C:\Users\pkmkr>python -m pip install requests --user
Collecting requests
  Downloading https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0b2a00aba99ae0f828be89d72b2190f27c11d4b7fb
/requests-2.22.0-py2.py3-none-any.whl (57kB)
    |#####| 61kB 787kB/s
Collecting certifi<=2017.4.17 (from requests)
  Downloading https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744fa380bc73ca47303ccc8b9190fd16a827281eac2
/certifi-2019.9.11-py2.py3-none-any.whl (154kB)
    |#####| 163kB 2.2MB/s
Collecting urllib3!=1.25.0,!1.25.1,<1.26.>=1.21.1 (from requests)
  Downloading https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a579c09dd7db825bb730ec1fe9c0180fc77bfb31448
/urllib3-1.25.6-py2.py3-none-any.whl (125kB)
    |#####| 133kB 3.2MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8
/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    |#####| 143kB 3.3MB/s
Collecting idna<2.9,>=2.5 (from requests)
  Downloading https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9
/idna-2.8-py2.py3-none-any.whl (58kB)
    |#####| 61kB 3.8MB/s
Installing collected packages: certifi, urllib3, chardet, idna, requests
  WARNING: The script chardetect.exe is installed in 'C:\Users\pkmkr\AppData\Roaming\Python\Python38\Scripts' which is n
ot on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed certifi-2019.9.11 chardet-3.0.4 idna-2.8 requests-2.22.0 urllib3-1.25.0
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\pkmkr>
```

Để cài đặt thư viện tabulate, hãy dùng lệnh:

```
python -m pip install tabulate --user
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pmkdr>python -m pip install tabulate --user
Collecting tabulate
  Downloading https://files.pythonhosted.org/packages/66/d4/977fdd5186b7cddb7c43a7aac7c5e4e0337a84cb802e154616f3cfc84563/tabulate-0.8.5.tar.gz (45kB)
    |#####| 51kB 297kB/s
Installing collected packages: tabulate
  Running setup.py install for tabulate ... done
Successfully installed tabulate-0.8.5
```

Bước 4: Viết hàm get để lấy thông tin

Tạo file getList.py trong C:/python (chung vị trí với file getTicket.py ở bài trước) với nội dung như sau:

```
import requests #Gửi các loại yêu cầu http đến server và nhận
response
import json #Là định dạng dữ liệu để truyền và nhận dữ liệu với
server
import sys # Để tương tác với hệ thống
from tabulate import tabulate #cung cấp các format bảng
import getTicket #Lấy các phương thức như get_auth_token, get
trong file getTicket.py đã viết ở bài lab trước
```

Khai báo một biến device dạng mảng array. Biến này dùng để chứa thông tin về các thiết bị đáp lại từ server.

```
device =[]
try:
    resp = getTicket.get(api="network-device") #Lấy thông tin
                                                về network-device
    status = resp.status_code #Lấy trạng thái của yêu cầu trên
    response_json = resp.json() #Lấy nội dung json đã mã hóa từ
                                lời đáp lại
    device = response_json["response"] #Gán thông tin từ lời
                                    đáp lại vào device
except:
    print("Something wrong,cannot get network device info")
    sys.exit()
```

Dòng `getTicket.get(api="network-device")` trong đó: `getTicket` dùng để liên kết file `getTicket.py`; `get` là hàm đã được định nghĩa trong file `getTicket.py`; `api="network-`

device” là gán giá trị “network-device” vào biến *api* trong hàm *get(ip, ver, uname, pword, api, params)*.

Nếu chương trình chạy đến phần “try” xảy ra lỗi thì lưu đề chương trình chuyển đến phần except.

```
# Nếu status khác 200 (nghĩa là không thành công) thì in lời đáp  
lại và thoát chương trình  
if status !=200:  
    print (resp.text)  
    sys.exit()
```

```
# Nếu kết quả là chuỗi trống thì in ra "No network Device Found"  
và thoát chương trình  
if device == []:  
    print("No network device found")  
    sys.exit()
```

Khai báo mảng là *device_list* để gán kết quả (thông tin các thiết bị):

```
device_list = []
```

Tạo một vòng lặp để dò từng item và thực hiện gán các giá trị vào *device_list*, *item["hostname"]* là dò tìm key hostname trong item và lấy ra value:

```
i=0  
for item in device:  
    i+=1  
    device_list.append([i,item["hostname"],item["managementIpAddress"],  
                        item["type"],item["instanceUuid"]])
```

In dữ liệu ra theo định dạng đã được xây dựng sẵn trong thư viện *tabulate*:

```
print(tabulate(device_list,  
               headers = ['number','hostname','ip','type',  
                           'mac address'], tablefmt="rst"))
```

Để chạy đoạn code vừa lập trình ta vào dấu nhắc cmd nhập lệnh:

```
Cd C://pythongetList.py
```

Kết quả (có thể thay đổi tùy thuộc vào số lượng thiết bị hiện có trong inventory):

```
{'token': 'ST-150-tXglLHSzCdFiUrfc7p37-cas'}  
Executing GET 'https://10.215.26.10/api/v1/network-device'  
GET 'network-device' status 200  
=====   
number hostname ip type mac address   
=====   
1 R1.vnpro 10.215.11.34 Cisco 2811VE Integrated Services Router dbe68fb6-807f-4a1c-b752-9bbce1d0bcc   
2 Sw.vnpro 10.215.11.35 Cisco Catalyst 3560V2-24TS Switch 2182566e-39c9-4641-bc4a-3f980857fd2   
3 Switch.192.168.1.1 10.215.14.58 Cisco 3750 Stackable Switches cd66c1b7-c9d8-4743-96e4-9cb3150bd90   
=====
```

Network Controller đã trả về các thông tin của các thiết bị có trong Inventory Devices.