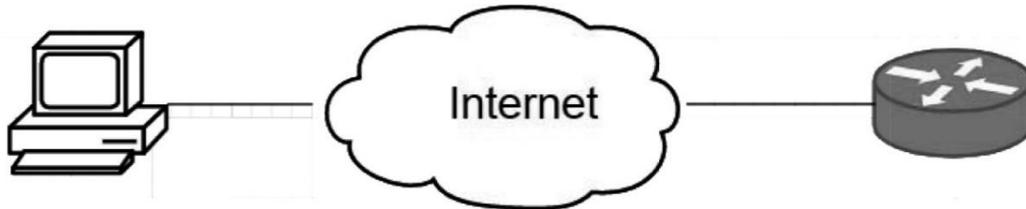


# LAB - VIẾT CHƯƠNG TRÌNH DÙNG NGÔN NGỮ PYTHON SỬ DỤNG RESTCONF ĐỂ THỰC HIỆN CHỈNH SỬA IP VÀ SUBNET MASK TRÊN THIẾT BỊ SANDBOX CISCO

## I. Sơ đồ



Hình 1: Sơ đồ bài Lab

## II. Mô tả

- Sơ đồ bài thực hành gồm 1 PC và 1 thiết bị chạy hệ điều hành IOS XE được đấu nối với nhau như hình 1.

- Trên sơ đồ này, học viên thực hiện kết nối đến thiết bị mạng, viết chương trình để thay đổi địa chỉ IP 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 kết nối máy tính đến Agent bằng mạng Internet.

- Cài đặt thư viện requests trên máy tính.

- Viết code bằng Python thực hiện yêu cầu:

+ Lấy danh sách các cổng.

+ Hỏi user chọn cổng muốn chỉnh sửa địa chỉ IP.

+ In ra thông tin của cổng đã chọn.

+ Yêu cầu nhập IP và Subnet mask.

+ Gửi cấu hình cổng lên server.

+ In kết quả cổng sau chỉnh sửa (có thể dùng lại hàm ở yêu cầu số 1).

## IV. Các bước thực hiện

### 4.1. Cài đặt thư viện

Thực hiện cài đặt thư viện như các công đoạn đã trình bày ở **Lab 13**.

### 4.2. Viết chương trình

- Import các thư viện và khai báo các thông tin cần để giao tiếp với thiết bị:

```
import json
import requests
import sys
requests.packages.urllib3.disable_warnings()

HOST = 'ios-xe-mgmt.cisco.com'
PORT = '9443'
USER = 'developer'
PASS = 'C1sco12345'
```

Host, port, user, pass là các thông tin lấy được ở phần C.

Xác định cổng Management (để người dùng không được chọn cổng đó để chỉnh sửa):

```
MANAGEMENT_INTERFACE = "GigabitEthernet1"
```

Tạo url\_base:

```
url_base = "https://{h}:{p}/restconf".format(h=HOST, p=PORT)
```

Khai báo headers:

```
headers = {'Content-Type': 'application/yang-data+json',
'Accept': 'application/yang-data+json'}
```

Hàm lấy danh sách các cổng:

```
def get_configured_interfaces():
url = url_base + "/data/ietf-interfaces:interfaces"
response = requests.get(url, auth=(USER, PASS), headers=headers,
verify=False)

return response.json()["ietf-interfaces:interfaces"]
["interface"]
```

- Tạo url để gửi bằng cách nối thêm phần cần thiết vào url\_base.
- Lấy thông tin bằng get bao gồm url, authentication(user và pass), headers, verify.

Cấu hình IP công:

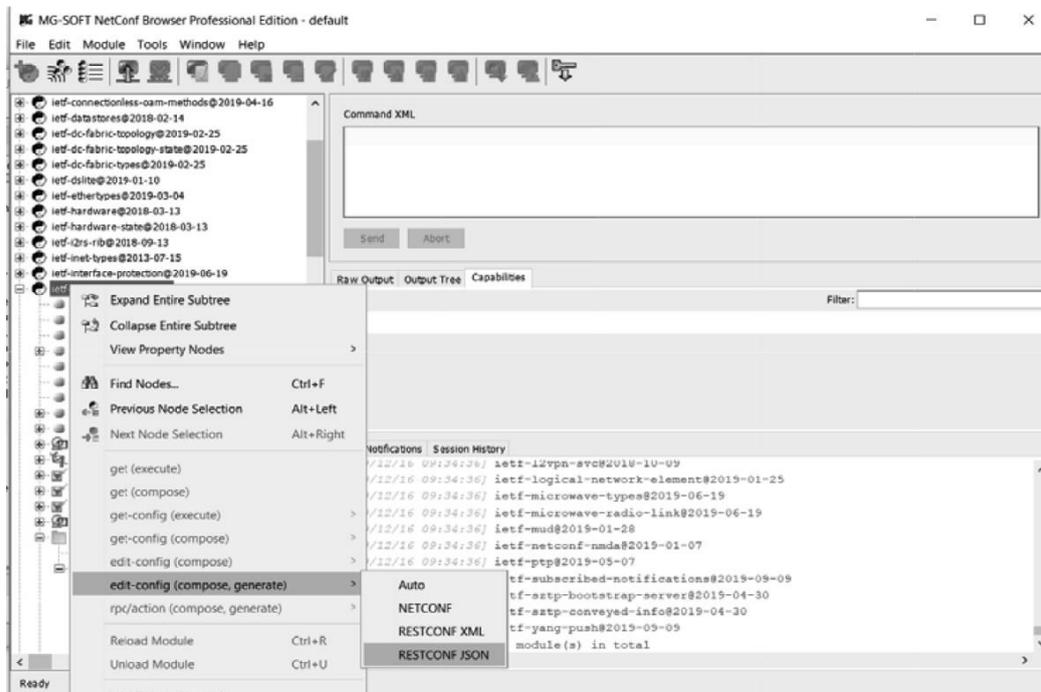
```
def configure_ip_address(interface, ip):
# RESTCONF URL của công
url = url_base + "/data/ietf-interfaces:interfaces/interface
    ={i}".format(i=interface)

#Tạo payload để sửa IP
data = {
"ietf-interfaces:interface":{
    "name": interface,
    "type": "iana-if-type:ethernetCsmacd",
    "ietf-ip:ipv4":{
    "address":{
        "ip": ip["address"],
    "netmask": ip["mask"]
        }
    }
    }
}

#Dùng PUT để chỉnh sửa
response = requests.put(url, auth=(USER, PASS), headers=headers,
    verify=False, json=data)

print(response.text)
```

Vào MG-Soft NetConf Browser để biết cách viết phần payload (chọn ietf-interfaces):



```
{
  "ietf-interfaces:interfaces":{
    "interface":[
      {
        "name":"x",
        "description":"x",
        "type":"iana-if-type:vdsl2",
        "enabled":false,
        "link-up-down-trap-enable":"disabled",
        "ietf-ip:ipv4":{
          "enabled":false,
          "forwarding":false,
          "mtu":68,
          "address":[
            {
              "ip":"136.122.166.54",
              "prefix-length":0
            }
          ]
        }
      }
    ],
  }
}
```

Lưu ý:

"type": "iana-if-type:ethernetCsmacd" do Router trên sandbox sử dụng loại cổng này.

"netmask": ip["mask"] do bài lab này yêu cầu nhập netmask.

Lấy thông tin chi tiết của cổng đã chọn:

```
def print_interface_details(interface):
    url = url_base + "/data/ietf-interfaces:interfaces/interface
        ={i}".format(i=interface)

    response = requests.get(url, auth=(USER, PASS),
                            headers=headers, verify=False)

    intf = response.json()["ietf-interfaces:interface"]
    print("Name: ", intf["name"])

    try:
        print("IP Address: ", intf["ietf-ip:ipv4"]["address"][0]
              ["ip"], "/", intf["ietf-ip:ipv4"]["address"][0]["netmask"])
    except KeyError:
        print("IP Address: UNCONFIGURED")

    print()

    return(intf)
```

Chỉ lấy phần IP và netmask nên ta sẽ vào tới phần [ip] và [netmask] của intf.

- Hỏi người dùng chọn cổng nào (Nếu chọn cổng Management thì phải chọn lại):

```
def interface_selection(interfaces):
    sel = input("Bạn muốn cấu hình với cổng nào? ")

    while sel == MANAGEMENT_INTERFACE or not sel in [intf["name"]
                                                       for intf in interfaces]:
        print("INVALID: Select an available interface.")
        print("          " + MANAGEMENT_INTERFACE + " is used for
              management.")
        print("          Choose another Interface")
        sel = input("Bạn muốn cấu hình với cổng nào? ")

    return(sel)
```

Ở đây ta sẽ sử dụng vòng lặp while để bắt người dùng nhập lại nếu chọn cổng management hoặc nhập sai chính tả tên cổng.

- Yêu cầu người dùng nhập IP và Mask:

```
def get_ip_info():  
    ip = {}  
    ip["address"] = input("Nhập IP address: ")  
    ip["mask"] = input("Nhập subnet mask: ")  
    return(ip)
```

- Hàm main:

```
def main():  
    #Lấy danh sách các cổng  
    interfaces = get_configured_interfaces()  
  
    print("Router có những cổng sau: \n")  
    for interface in interfaces:  
        print(" * {name}".format(name=interface["name"]))  
    print("")  
  
    #Hỏi user chọn cổng nào  
    selected_interface = interface_selection(interfaces)  
    print(selected_interface)  
  
    #In ra thông tin của cổng đã chọn  
    print("Thông tin cổng hiện tại:")  
    print_interface_details(selected_interface)  
  
    #Yêu cầu người dùng nhập IP và Mask  
    ip = get_ip_info()  
  
    #Gửi cấu hình cổng lên server  
    configure_ip_address(selected_interface, ip)  
    #In kết quả  
    print("Thông tin cổng sau cấu hình:")
```

```
print_interface_details(selected_interface)

if __name__ == '__main__':
    sys.exit(main())
```