

Lab - Viết chương trình dùng ngôn ngữ Python để thực hiện lấy ticket từ Network Controller.

1. Sơ đồ mạng:



Hình 1.1

2. Mô tả:

- Sơ đồ bài thực hành gồm 1 Server Network Controller và 1 PC được đấu nối với nhau như hình 1.1.
- Trên sơ đồ này, học viên thực hiện kết nối đến Controller, viết hàm get để lấy thông tin bằng ngôn ngữ Python.
- Máy PC phải đáp ứng yêu cầu đã cài đặt trạm làm việc cho developer.

3. Yêu cầu kĩ thuật:

- Học viên thực hiện thực hiện kết nối máy tính đến Server Network Controller
- Cài đặt thư viện requests trên máy tính.
- Viết code bằng Python thực hiện yêu cầu:
 - Lấy danh sách tất cả các thiết bị kết nối với Network Controller
 - Lấy số id của thiết bị cần xóa
 - Xóa thiết bị

4. Các bước thực hiện:

Bước 1: Cấu hình máy tính ping thành công đến Network Controller

- Máy PC để IP động là có thể ping được Network Controller

Bước 2: Cài đặt thư viện requests

- Bấm tổ hợp phím windows +R để mở Command Line (cmd)


```
id_seleted = item["id"]
return id_seleted
```

Viết hàm để lấy số id của thiết bị cần xóa: đầu tiên ta sẽ in ra màn hình tất cả các thiết bị kết nối với Network Controller (bài lab trước đã hướng dẫn), cho người dùng nhập số thứ tự của thiết bị cần xóa. Tiếp theo ta sẽ khai báo mảng device và gọi api để lấy danh sách các thiết bị, mục đích của việc này là để so sánh số thứ tự người dùng vừa nhập với số thứ tự trong danh sách, từ đó lấy ra số id tương ứng.

```
def delete_device():
    id = get_device_id()
    try:
        resp = pgpd.delete(api="network-device/"+id)
        status = resp.status_code
        response_json = resp.json()
        r = response_json["response"]
        print(r)
    except:
        print("Something wrong")
        sys.exit()
```

Sau khi có được số id của thiết bị cần xóa, ta gắn thêm số id vào phần api và gửi với phương thức là delete

```
def main():
    #Lay danh sach cac thiet bi
    result = get_network_device.network_device_list()
    print(tabulate(result,
        headers = ['number','hostname','ip','type','mac address','id'], tablefmt="rst"))
    result2 = delete_device()

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

Đây là hàm main, thực hiện các công việc như lấy danh sách các thiết bị và in ra màn hình, gọi hàm delete_device()

Bước 4: Chạy file python

Đây là danh sách thiết bị hiện có trong Network Controller

Device Name	IP Address	Reachability Status	Up Time	Last Updated Time	Last Inventory Collection Status
R1.R1vnpro	10.215.11.24	Reachable	0:21:13.14	2 minutes ago	Managed

Chạy file delete_device.py

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\Hoc\DevNet\Bao cao\lab3 (APIC-EM)\delete_device.py =====
{'token': 'ST-216-KTebWNaL4mW9zkkdVmy1-cas'}

Executing GET 'https://10.215.26.9/api/v1/network-device'
GET 'network-device' status 200

=====
number  hostname  ip          type          mac address   id
=====
1       R1.R1vnpro  10.215.11.24 Cisco 2851 Integrated Services Router 3a681ea6-26aa-40c5-8061-91c11898a0df 3a681ea6-26aa-40c5-8061-91c11898a0df
=====
Nhap so thu tu cua thiet bi can xoa:1

Executing GET 'https://10.215.26.9/api/v1/network-device'
GET 'network-device' status 200

Executing DELETE 'https://10.215.26.9/api/v1/network-device/3a681ea6-26aa-40c5-8061-91c11898a0df'
DELETE 'network-device/3a681ea6-26aa-40c5-8061-91c11898a0df' Status: 202

{'taskId': 'c0ba7065-alb3-40f4-bf4b-1f63140d5696', 'url': '/api/v1/task/c0ba7065-alb3-40f4-bf4b-1f63140d5696'}
>>> |
```

Vậy là chúng ta đã xóa thành công, kiểm tra lại trên Network Controller, nó sẽ báo no devices found!

Device Name	IP Address	Reachability Status	Up Time	Last Updated Time	Last Inventory Collection Status
No devices found! Please use the Discovery app to scan for devices.					