Dùng Python lấy danh sách thông tin các thiết bị trong fabric SD-WAN của Cisco

1. **Mô tả:**

- Học viên thực hiện kết nối đến Sandbox SD-WAN (vManage), 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.

2. **Yêu cầu kĩ thuật:**

- Cài đặt thư viện requests, tabulate, click trên máy tính.
- Kết nối đến Sandbox SD-WAN (https://sandboxsdwan.cisco.com:8443/)
- Viết code bằng Python thực hiện yêu cầu:
  - Đăng nhập và xác thực
  - GET requests, POST requests
  - Lấy danh sách thông tin các thiết bị trong Controller

3. **Các bước thực hiện:**

Bước 1: Cài đặt thư viện

- Bấm tổ hợp phím Win+R để chạy cmd

- Trong màn hình cmd gõ : python –m pip install requests tabulate click –user

```
C:\Users\TuanHoanh>py -m pip install requests tabulate click --user
Collecting requests
  Using cached https://files.pythonhosted.org/packages/1a/70/1935c770cb3be6e3a8b78ced23d7e0f3b187f5cbfab4749523ed65d7c9b
1/requests-2.23.0-py2.py3-none-any.whl
Collecting tabulate
  Using cached https://files.pythonhosted.org/packages/c4/f4/770ae9385990f5a19a91431163d262182d3203662ea2b5739d0fcfc080f
1/tabulate-0.8.7-py3-none-any.whl
Collecting click
  Using cached https://files.pythonhosted.org/packages/dd/c0/4d8f43a9b16e289f36478422031b8a63b54b6ac3b1ba605d602f10dd54d
6/click-7.1.1-py2.py3-none-any.whl
Requirement already satisfied: certifi>=2017.4.17 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (f
rom requests) (2019.9.11)
Requirement already satisfied: idna<3,>=2.5 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from re
quests) (2.8)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (fr
om requests) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\tuanhoanh\appdata\roaming\python\pyth
on37\site-packages (from requests) (1.25.6)
Installing collected packages: requests, tabulate, click
  WARNING: The script tabulate.exe is installed in 'C:\Users\TuanHoanh\AppData\Roaming\Python\Python37\Scripts' which is
 not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed click-7.1.1 requests-2.23.0 tabulate-0.8.7
```

Bước 2: Viết code đăng nhập và xác thực

Import các thư viện cần thiết và tắt cảnh báo

```
import requests
import sys
import json
import click
from tabulate import tabulate
```

```
import SD_WAN_INFO

requests.packages.urllib3.disable_warnings()
```

Tạo file SD_WAN_INFO.py chứa thông tin kết nối và thông tin đăng nhập, ghi các thông tin như hình dưới, sau đó lưu lại và đóng file

```
IP = "sandboxsdwan.cisco.com"
USERNAME = "devnetuser"
PASSWORD = "Cisco123!"
```

Trở về file trước, ta khai báo các thông tin của Sandbox SD-WAN, các biến SDWAN_IP,USERNAME,PASSWORD sẽ lấy giá trị từ file SD_WAN.py đã tạo ở trên.

```
SDWAN_IP = SD_WAN_INFO.IP
SDWAN_USERNAME = SD_WAN_INFO.USERNAME
SDWAN_PASSWORD = SD_WAN_INFO.PASSWORD
```

Tiếp theo chúng ta sẽ tạo một class rest_api_lib và tạo contructor __init__ của class (lưu ý ngoài các tham số truyền vào chúng ta luôn phải thêm self đại diện cho instance của class và với nó chúng ta có thể kết nối đến các thuộc tính và phương thức của class đó.

```
class rest_api_lib:
    def __init__(self, vmanage_ip, username, password):
        self.vmanage_ip = vmanage_ip
        self.session = {}
        self.login(self.vmanage_ip, username, password)
```

Định nghĩa phương thức login và khai báo login_url

```
    def login(self, vmanage_ip, username, password):
        """Login to vmanage"""
        base_url_str = 'https://%s:8443/'%vmanage_ip
        login_action = '/j_security_check'
        login_url = base_url_str + login_action
```

Khai báo login_data chứa username và password để gửi lên xác thực

```
        login_data = {'j_username' : username, 'j_password' : password}
```

Chúng ta sẽ dùng phương thức session từ thư viện request để tạo một phiên làm việc mới, trong phiên làm việc vừa tạo đó gửi yêu cầu post để đưa thông tin đến login_url.

```
        sess = requests.session()
        login_response = sess.post(url=login_url, data=login_data, verify=False)
```

Để đảm bảo xác thực thành công chúng ta sẽ kiểm tra nội dung trả về và nếu nội dung trả về có tag <html/> thì nghĩa là đăng nhập thất bại. Nếu muốn xem đăng nhập thất bại thì nội dung trả về sẽ như thì sửa lại mật khẩu và bỏ dấu # đầu dòng của đoạn code dưới này.

```
        if b'<html>' in login_response.content:
```

```
        print ("Login Failed")
        #print(login_response.content)
        sys.exit(0)
```

Gắn session sess vào làm giá trị của thuộc tính session của class

```
        self.session[vmanage_ip] = sess
```

Bước 3: Viết code GET requests

Định nghĩa phương thức get_request, trong đây chúng ta sẽ phải tạo một url mới, tham số api truyền vào tùy mục đích sử dụng nên chúng ta sẽ dùng %s để url có thể thay đổi dễ dàng. Ta sẽ dùng tiếp session vừa được xác thực thành công để gọi các API tiếp theo mà không cần phải gửi kèm theo username, password; get để gửi yêu cầu lên server và nhận lời đáp lại, sau đó gắn thông tin từ lời đáp lại vào biến response. Để lấy dữ liệu, ta sẽ dùng response.content gắn vào biến data và đây cũng là giá trị trả về của phương thức get_request này.

```
    def get_request(self, api):
        url = "https://%s:8443/dataservice/%s"%(self.vmanage_ip, api)

        response = self.session[self.vmanage_ip].get(url, verify=False)
        data = response.content
        return data
```

Bước 4: Viết code POST requests

Phương thức post_request này tương tự với phương thức get_request ở trên nhưng khác ở chỗ post dùng để gửi yêu cầu tạo tài nguyên mới trên server. Và có thêm tham số đầu vào là payload và headers. Payload là nơi sẽ chứa đựng các thông tin gửi kèm theo khi gửi yêu cầu post. Headers khai báo kiểu nội dung là application/json.

```
    def post_request(self, api, payload, headers={'Content-Type': 'application/json'}):
        url = "https://%s:8443/dataservice/%s"%(self.vmanage_ip, api)
        payload = json.dumps(payload)
        print(payload)

        response = self.session[self.vmanage_ip].post(url=url, data=payload, headers=headers,
verify=False)
        data = response.json()
        return data
```

Bước 5: Viết code lấy danh sách thông tin các thiết bị

Tạo instance của class là sdwanp và truyền các tham số SDWAN_IP, SDWAN_USERNAME, SDWAN_PASSWORD

```
sdwanp = rest_api_lib(SDWAN_IP, SDWAN_USERNAME, SDWAN_PASSWORD)
```

Khi muốn cấu hình nhiều decorator @click.command() thì phải sử dụng group() để tạo nhiều decorator trong cùng một đoạn script.

```
@click.group()
def cli():
    pass
```

Tạo decorator @click.command() cấu hình Click để làm việc với hàm Python ngay sau decorator. Click.echo() dùng để in ra màn hình. Json.loads() dùng để giải mã đối tượng json thành đối tượng trong python.

```
@click.command()
def device_list():
    """Retrieve and return network devices list."""
    click.echo("Retrieving the devices.")

    response = json.loads(sdwanp.get_request('device'))
    items = response['data']
```

Khai báo headers để hiển thị trong bảng và khai báo list table. Tạo vòng lặp cứ mỗi item trong biến items chứa dữ liệu này thì sẽ được ghi thêm vào list table. Tiếp theo chúng ta sẽ dùng hàm tabulate để xuất ra màn hình bảng danh sách thông tin các thiết bị trong fabric của SD-WAN.

```
    headers = ["Host-Name", "Device Type", "Device ID", "System IP", "Site ID", "Version",
"Device Model"]
    table = list()

    for item in items:
        tr = [item['host-name'], item['device-type'], item['uuid'], item['system-ip'], item['site-id'],
item['version'], item['device-model']]
        table.append(tr)
    try:
        click.echo(tabulate(table, headers, tablefmt="fancy_grid"))
    except UnicodeEncodeError:
        click.echo(tabulate(table, headers, tablefmt="grid"))
```

Cuối cùng là thêm command device_list vào cli() và viết hàm main

```
cli.add_command(device_list)

if __name__ == "__main__":
    cli()
```

Để chạy chương trình này, chúng ta sẽ vào cmd, đi đến thư mục đặt file python, chạy chương trình sdwan.py

```
C:\Users\TuanHoanh\Desktop>cd Device_list

C:\Users\TuanHoanh\Desktop\Device_list>sdwan.py
```

Kết quả:

```
C:\Users\TuanHoanh\Desktop\Device_list>sdwan.py
Usage: sdwan.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  device-list  Retrieve and return network devices list.
```

Để ý phần Usage có hướng dẫn cách sử dụng click, hiện tại commands hiện có là device-list nên trong cmd chúng ta sẽ gõ: >sdwan.py device-list

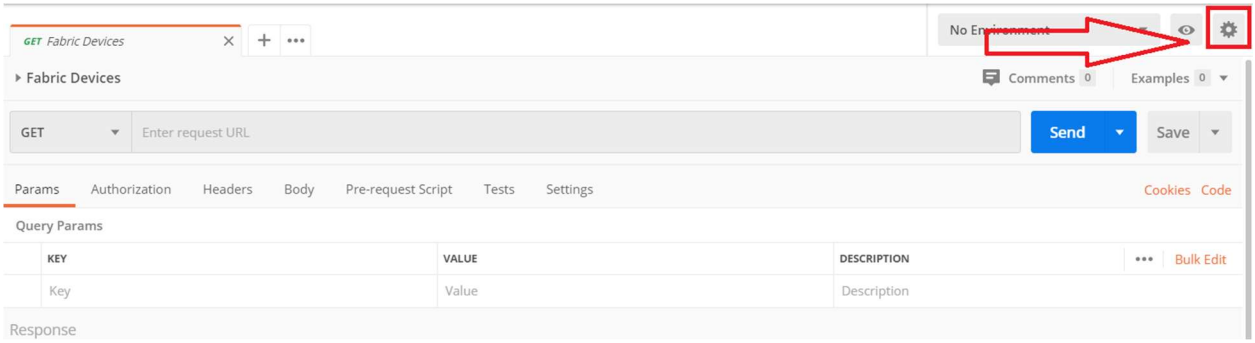
Kết quả là chúng ta lấy được danh sách các thiết bị hiện có trong mạng fabric SD-WAN.

```
C:\Users\TuanHoanh\Desktop\Device_list>sdwan.py device-list
Retrieving the devices.
```

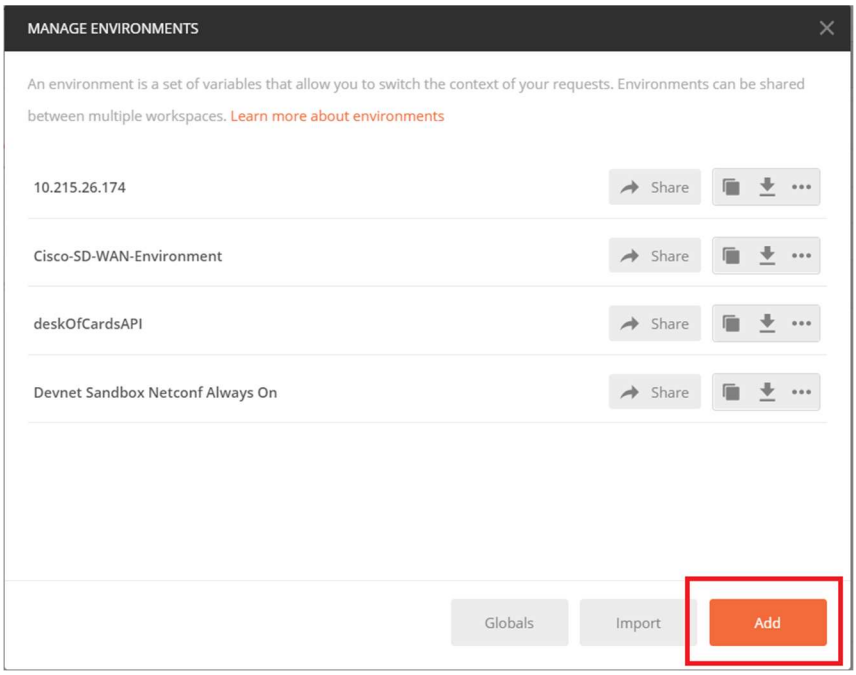
Host-Name	Device Type	Device ID	System IP	Site ID	Version	Device Model
vmanage	vmanage	4854266f-a8ad-4068-9651-d4e834384f51	4.4.4.90	100	18.3.1.1	vmanage
vsmart	vsmart	da6c566f-eb5f-4731-a89a-ff745661027c	4.4.4.70	100	18.3.0	vsmart
vbond	vbond	455407de-9327-467e-a0d2-d3444659dbb2	4.4.4.80	100	18.3.1	vedge-cloud
vedge01	vedge	4af9e049-0052-47e9-83af-81a5825f7ffe	4.4.4.60	200	18.3.1	vedge-cloud
vedge03.cisco.com	vedge	100faff9-8b36-4312-bf97-743b26bd0211	4.4.4.62	555	18.3.1	vedge-cloud
vedge22	vedge	f3d4159b-4172-462c-9c8d-8db76c31521d	4.4.4.61	300	18.3.1	vedge-cloud
vedge44	vedge	46c18a49-f6f3-4588-a49a-0b1cc387f179	4.4.4.63	555	18.3.1	vedge-cloud

Sử dụng Postman tương tác với SD-WAN REST API

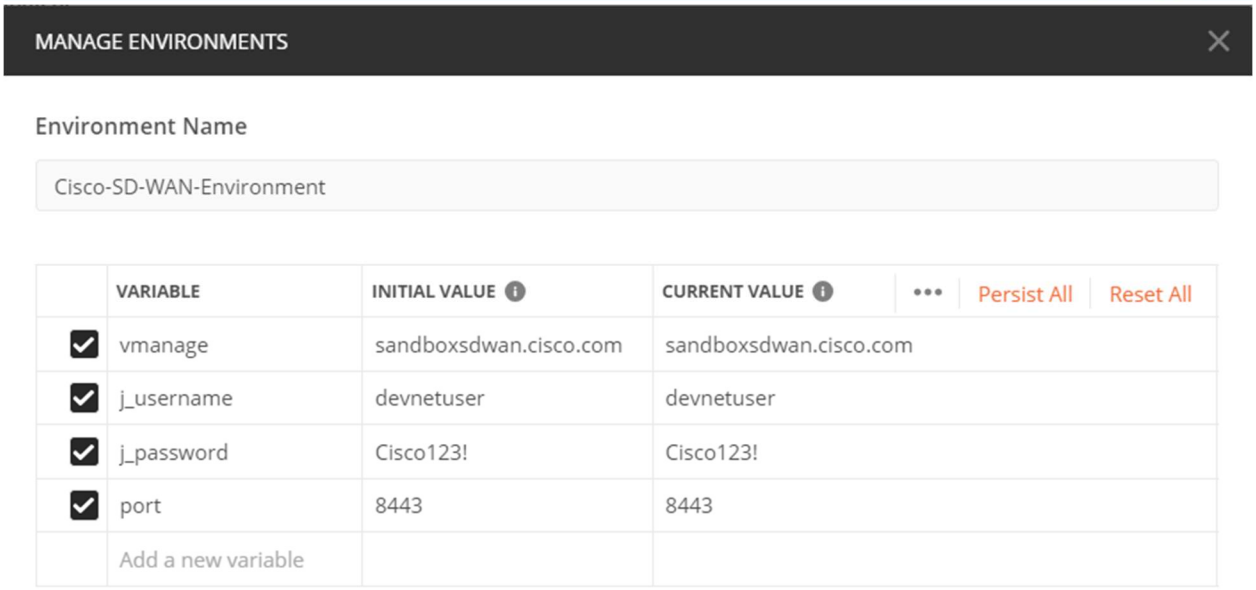
Đầu tiên chúng ta sẽ tạo Environment, nhấn vào nút hình bánh răng như hình vẽ.



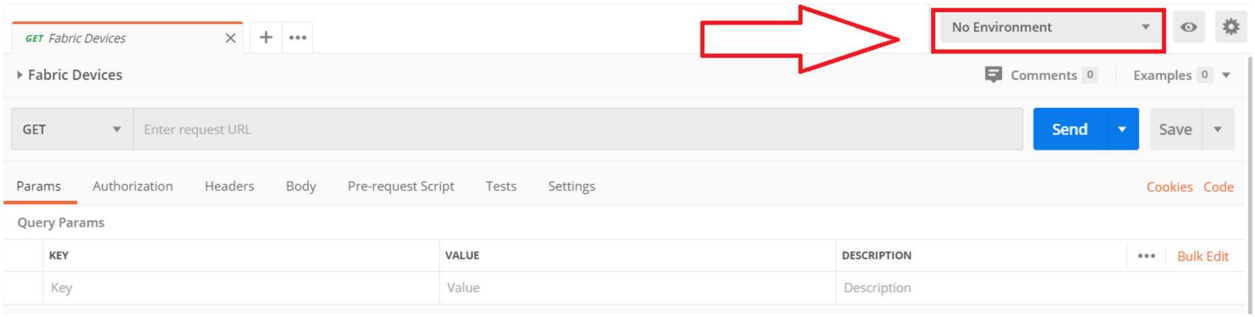
Nhấn nút Add



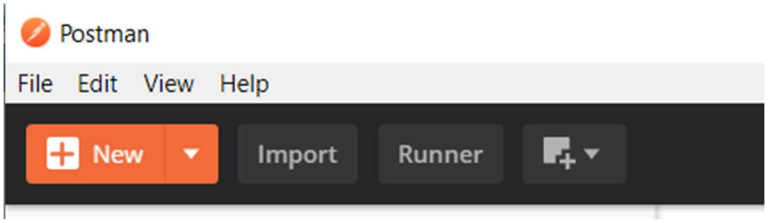
Sau đó gõ vào các thông tin như hình, trong đó vmanage có value là địa chỉ của con sandbox SD-WAN của Cisco, port là cổng kết nối, j_username và j_password là thông tin đăng nhập vào Sandbox SD-WAN. Sau khi điền đầy đủ thông tin nhấn nút Add để thêm và lưu lại.



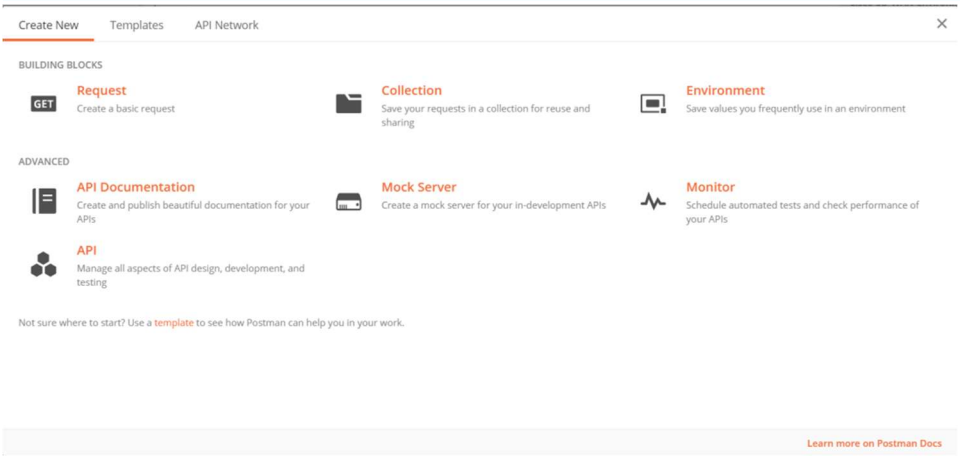
Bây giờ chúng ta sẽ trở về màn hình Request, click chuột vào ô như hình dưới, nó sẽ hiện lên danh sách các Environment đã tạo. Sau đó chọn Environment vừa tạo là Cisco-SD-WAN-Environment.



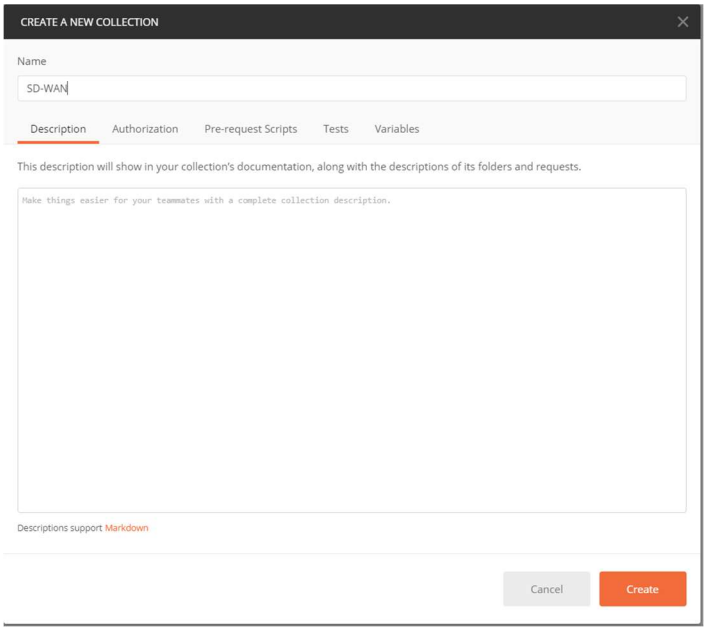
Nhấn nút New như hình dưới



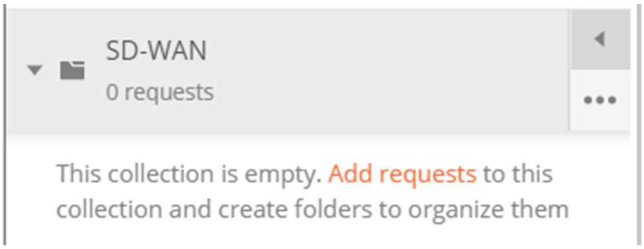
Chọn Collection để tạo Collection mới, mục đích của việc này là lưu lại và tập trung các Request vào 1 chỗ để dễ quản lý và sử dụng lại



Ô Name gõ vào SD-WAN, sau đó nhấn Create ở dưới



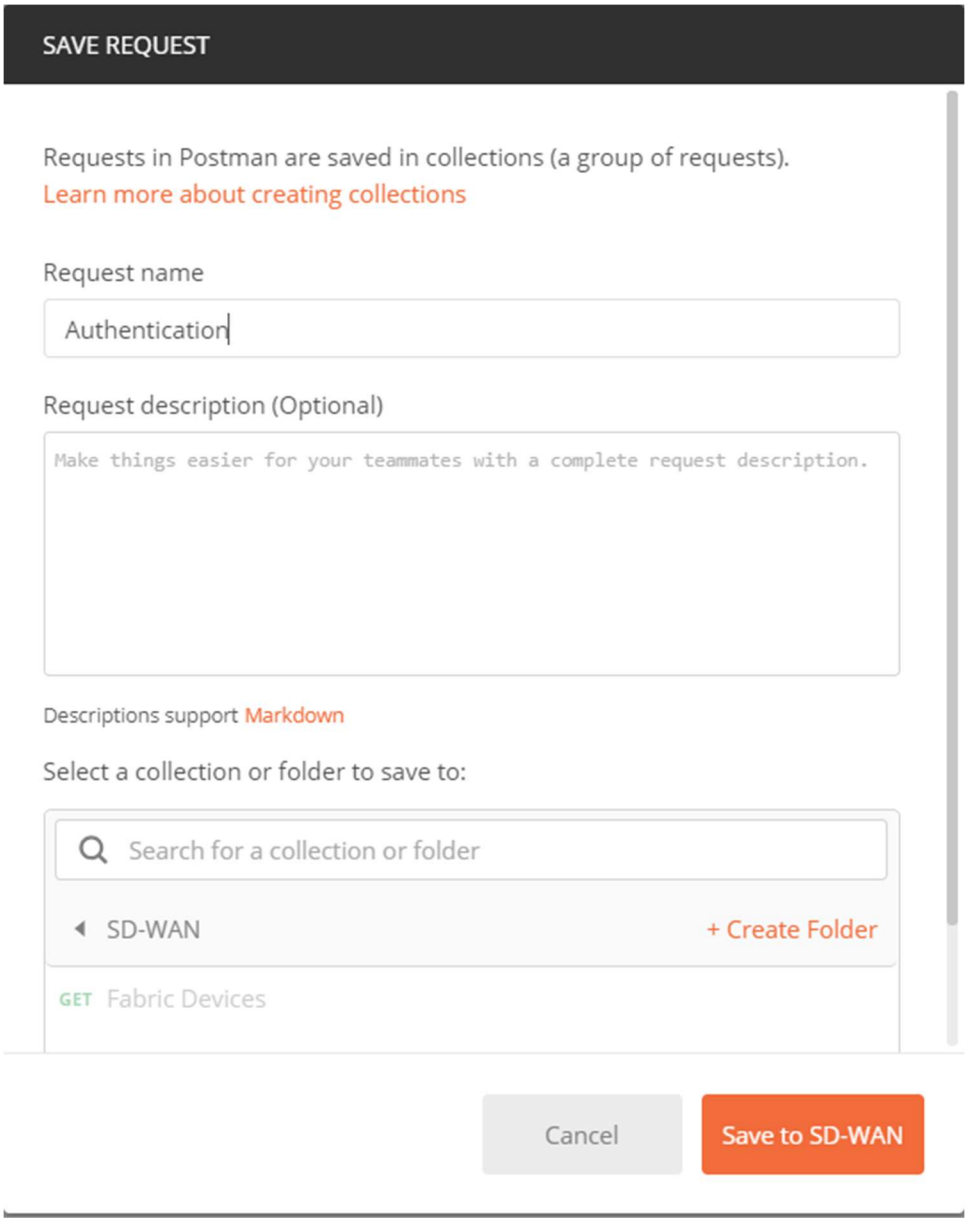
Sau khi tạo xong Collection, nhấn vào Collection vừa tạo thì nó sẽ hiện Collection đang trống, lưu ý dòng chữ màu cam Add Requests, nhấn vào chữ để thêm Requests



1. Post Authentication:

Phần này sẽ giúp chúng ta chắc chắn rằng chỉ những user được cấp phép mới được kết nối đến API

Tạo request mới, gõ vào Authentication và nhấn save



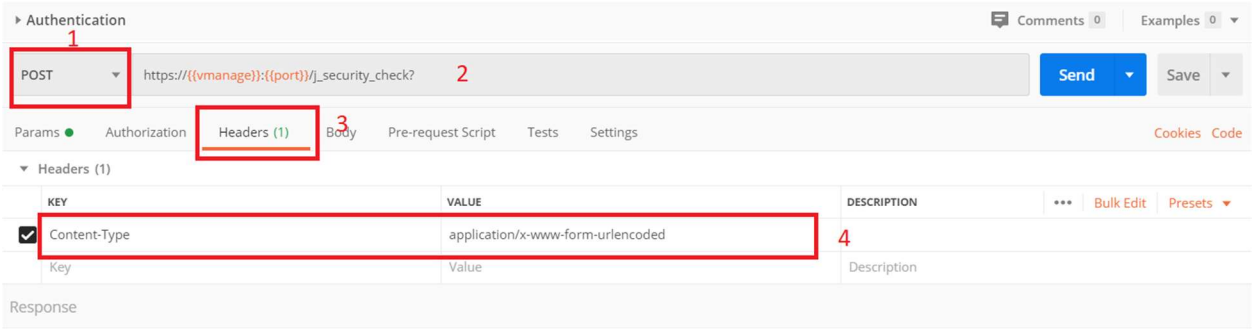
Tiếp theo chúng ta sẽ thực hiện bốn bước như hình dưới là:

Số 1: đổi get thành post

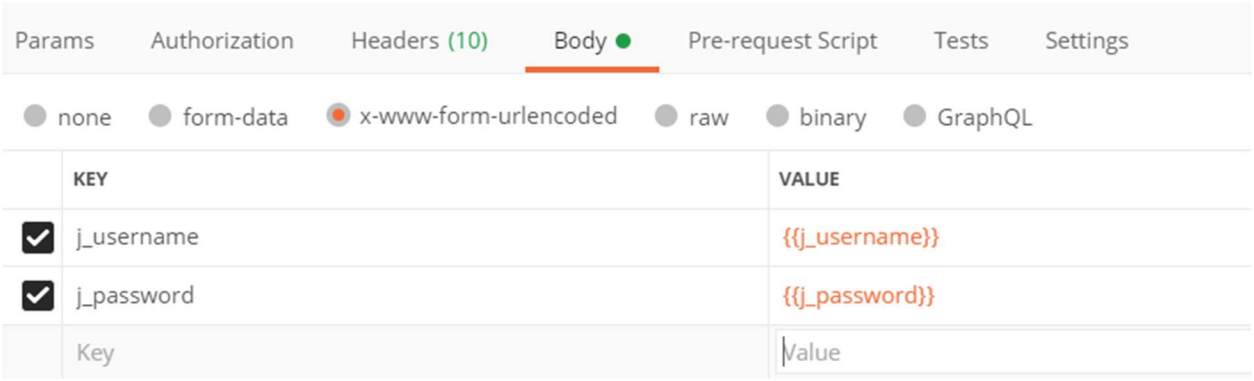
Số 2: nhập URL: https://{{vmanage}}:{{port}}/j_security_check , phần đặt trong 2 dấu ngoặc nhọn để biểu thị đó là biến, biến này sẽ lấy từ Environment mà chúng ta đã khai báo từ phần chuẩn bị

Số 3: Chuyển sang tab Headers

Số 4: Nhập Content-Type và application/x-www-form-urlencoded vào cặp key/value tương ứng



Chuyển sang tab Body kế bên Headers, chọn ô tròn x-www-form-urlencoded, điền các thông tin như j_username, j_password bên phần value thì đặt trong 2 dấu ngoặc nhọn để biểu thị đó là biến, biến này sẽ lấy từ Environment mà chúng ta đã khai báo từ phần chuẩn bị.

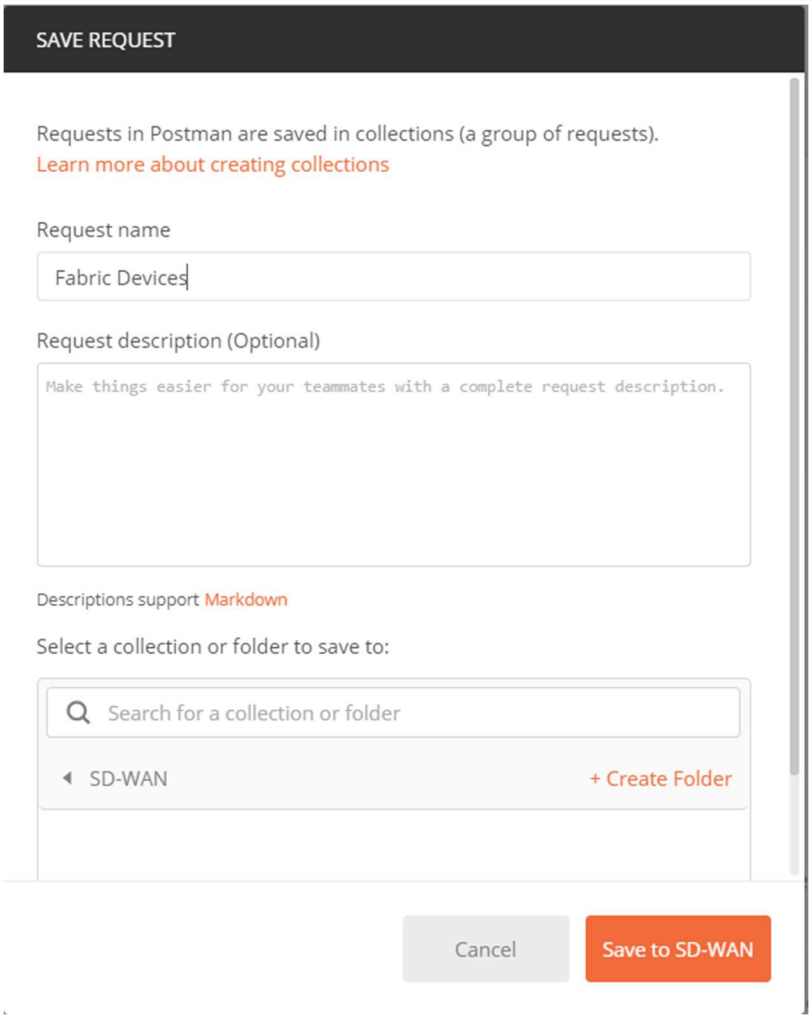


Công việc cuối cùng là nhấn nút Send và xem kết quả ở Cookies. Cookies tên JSESSIONID sẽ được sử dụng cho những lời gọi API tiếp theo trong bài lab này. Cookie này có thời hạn và lập tức đại diện cho việc xác thực của tài khoản devnetuser.

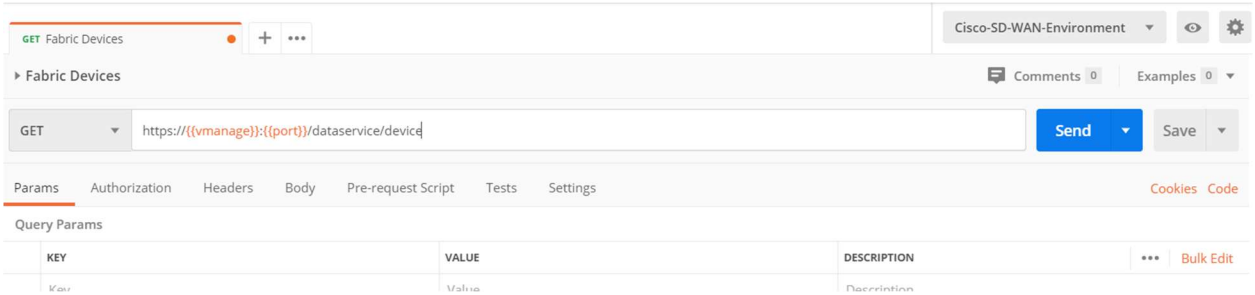
2. Get Fabric Devices:

Chúng ta sẽ tạo Request dùng phương thức GET và phần đuôi URL là dataservice/device để nhận danh sách tất cả các thiết bị hiện đang được kết nối trong SD-WAN fabric.

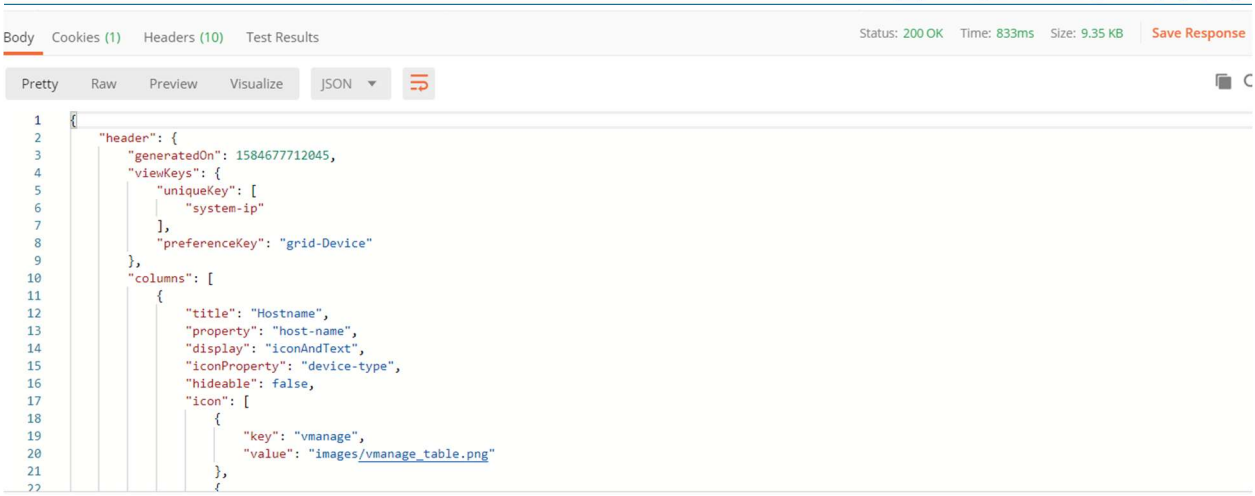
Tạo request mới, gõ vào Fabric Devices phần name và nhấn Save to SD-WAN



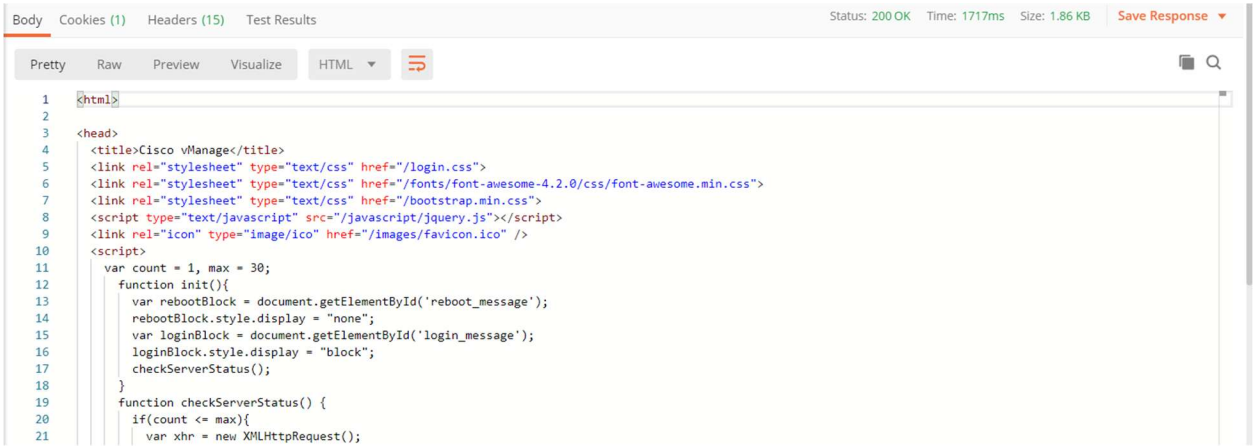
Tiếp theo ta sẽ nhập URL: https://{{vmanage}}:{{port}}/dataservice/device



Nhấn nút Send, và kết quả như hình dưới là chúng ta đã thành công lấy danh sách các thiết bị đang được kết nối trong SD-WAN fabric.

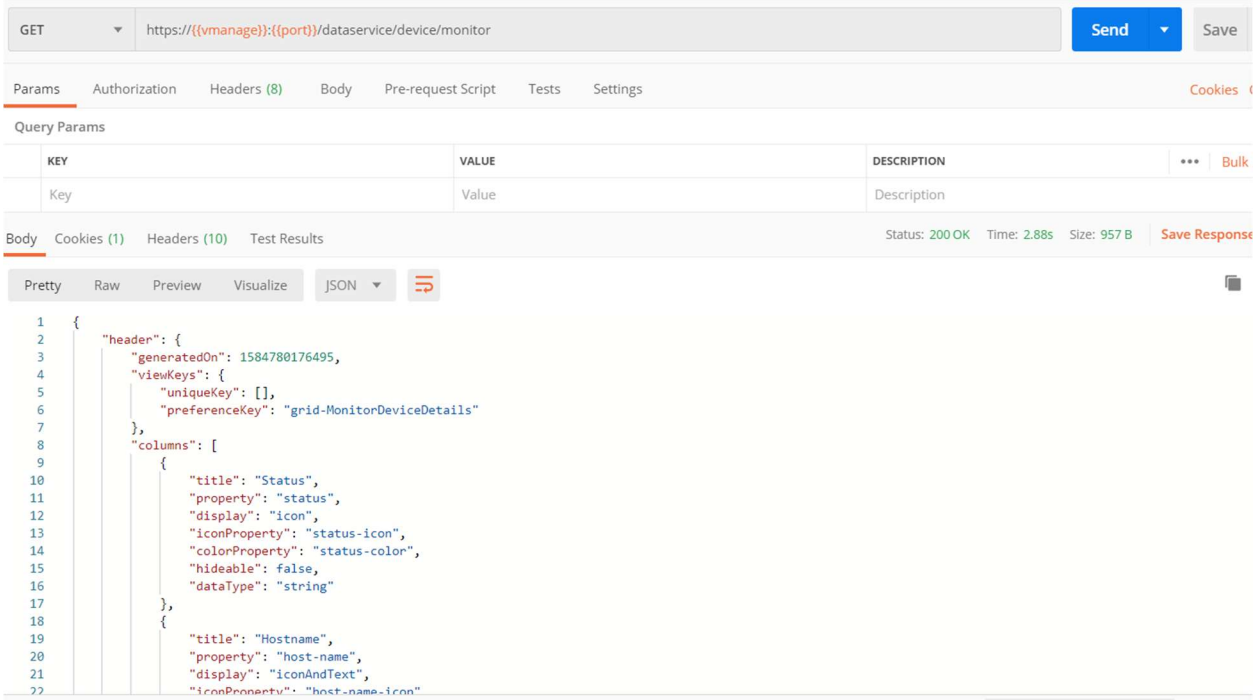


Lưu ý: nếu kết quả như hình dưới là do chúng ta chưa khai báo username, password. Do đó ta thực hiện request POST ở trên trước là để lấy cookie, sau khi đã có cookie thì trong thời hạn cookie còn hiệu lực thì ở những request tiếp theo chúng ta không cần phải gửi kèm username, password.



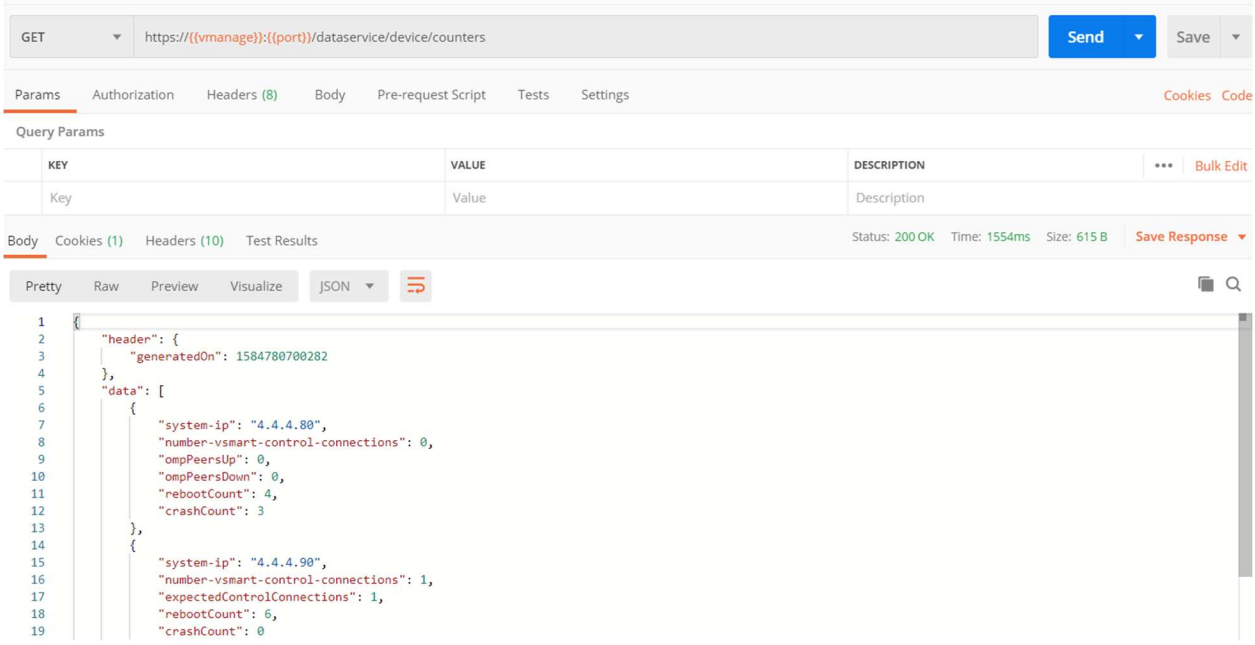
3. Get Device Status:

Chúng ta sẽ tạo Request dùng phương thức GET và phần đuôi URL là dataservice/device để nhận các thông tin đặc biệt mô tả trạng thái của các thiết bị trong SD-WAN fabric



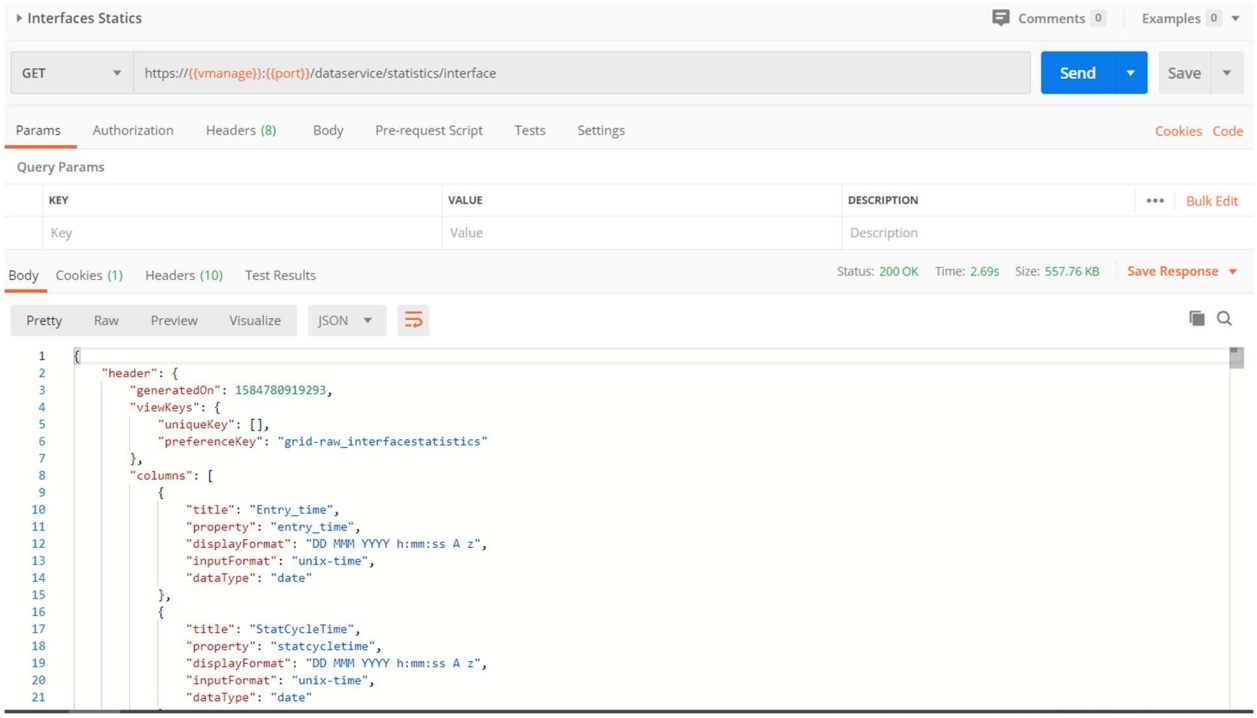
4. Get Device Counters:

Kế tiếp chúng ta thay phần đuôi URL là device/counters để xem các thiết bị đã khởi động lại bao nhiêu lần và nhấn nút Send.



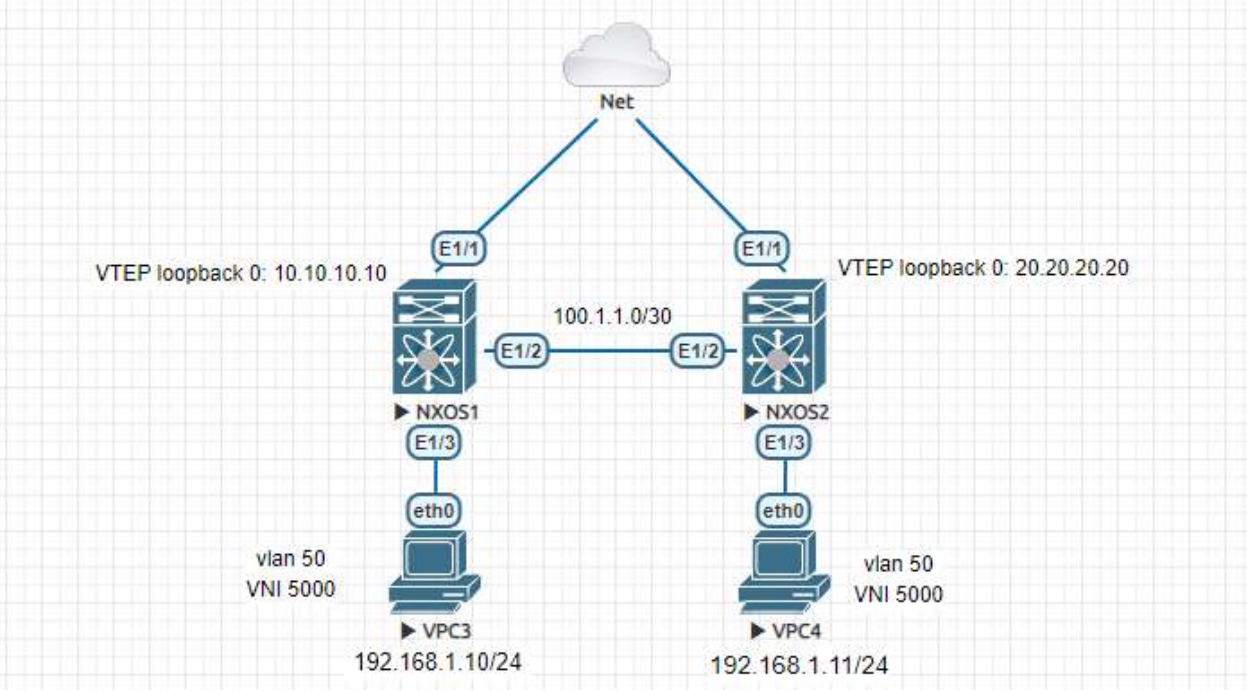
5. Get Interfaces Statistics:

Cuối cùng chúng ta thay phần đuôi URL là statistics/interface để nhận thống kê các cổng trên tất cả các thiết bị.



LAB: SỬ DỤNG ANSIBLE CẤU HÌNH TỰ ĐỘNG VXLAN

Sơ đồ mạng:



Mô tả:

- Sơ đồ gồm: 2 Switch Nexus 9000, 2 PC kết nối vào mạng Lan, và một PC sử dụng làm Linux Controler. Yêu cầu các thiết bị ping thông với Ansible Server.
- Máy tính của học viên có thể truy cập vào Ansible Server thông qua trình duyệt web.

Yêu cầu:

Thực hiện các công việc sau:

- Thêm thiết bị vào file host là file lưu địa chỉ IP của các thiết bị.
- Tạo playbook YAML để thực hiện cấu hình tự động cho Switch Nexus 9000.
- Chạy các playbook để cấu hình.

Các bước thực hiện:

- **Cấu hình SSH cho Switch Nexus 9000 thứ nhất.**

```
NXOS1#configure terminal

NXOS1(config)# feature SSH

NXOS1(config)# feature DHCP
```

```
NXOS1(config)# interface e1/1
NXOS1(config-if)# no switchport
NXOS1(config-if)# no shutdown
NXOS1(config-if)# ip address dhcp
```

- **Cấu hình SSH tương tự với NXOS2.**

```
NXOS2#configure terminal
NXOS2(config)# feature SSH
NXOS2(config)# feature DHCP
NXOS2(config)# interface e1/1
NXOS2(config-if)# no switchport
NXOS2(config-if)# no shutdown
NXOS2(config-if)# ip address dhcp
```

- **Cài đặt Ansible trên PC Linux Controller:**

Mở Terminal trên Ubuntu ta chạy các lệnh:

```
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible
```

Chuyển tới thư mục đã cài đặt Ansible và tiến hành sửa đổi file host:

```
Cd /etc/ansible
Sudo nano host
```

Ta thêm vào File host các thiết bị:

```
root@DESKTOP-KMNS09Q: /etc/ansible
  GNU nano 2.9.3

[nexus]
nxos1 ansible_host=10.215.26.154
nxos2 ansible_host=10.215.26.155

[all:vars]
ansible_user=admin
ansible_password=Vnpro123
ansible_connection= network_cli
ansible_network_os= nxos
```

Tham khảo link: https://github.com/vnpro149/Ansible/blob/master/vxlan-nxos/hosts

Giải thích:

- Ansible_host: địa chỉ ip của các thiết bị cần cấu hình thông qua SSH.
- Ansible_connection: network_cli là phương thức kết nối các thiết bị thông qua SSH.
- Ansible_network_os: ios ở đây dùng thiết bị nexus nên khai báo là nxos.
- Ansible_user, ansible_pass: là tài khoản để SSH thiết bị.

Lưu ý: Thứ tự khai báo không ảnh hưởng tới kết quả.

Sau khi xong nhấn Ctrl + X, sau đó Y và enter để lưu file lại.

- **Tạo file playbook:**

Tạo file playbook: sudo nano nxos.yml (Lệnh: sudo nano <filename>.yml)

Gồm các công việc:

+ Bật các dịch vụ cần thiết cho Switch Nexus 9000: SSH, DHCP, OSPF, Vlan Segment

```
---
- name: NXOS1
  hosts: nxos1
  gather_facts: no

  tasks:
    - name: Enable services
      nxos_config:
        lines:
          - feature ospf
          - feature vn-segment-vlan-based
          - feature nv overlay
          - system jumbomtu 9216
```

```
- name: Config ospf
  nxos_config:
    lines:
      - router ospf 1
```

+ Cấu hình cho interface e1/2 (interface kết nối 2 Switch Nexus) và quảng bá ospf

```
- name: Set ospf for int e1/2
  nxos_config:
    parents:
      - int e1/2
    lines:
      - no switchport
      - no shutdown
      - ip address 100.1.1.1/30
      - ip router ospf 1 area 0
```

+ Tạo Vlan 50 và gắn cổng cho nó.

```
- name: Create vlan 50
  nxos_config:
    parents:
      - vlan 50
    lines:
      - vn-segment 5000

- name: Access vlan 50
  nxos_config:
    parents:
      - int e1/3
    lines:
      - switchport mode access
      - switchport access vlan 50
```

+ Cấu hình nve

```
- name: Creat NVE
  nxos_config:
    parents:
      - int nve 1
    lines:
      - no shut
      - source-interface loopback 0
      - member vni 5000
      - ingress-replication protocol static
      - peer-ip 20.20.20.20
```

+ Tạo looback và quảng bá ospf

```
  - name: Config ospf for looback
    nxos_config:
      parents:
        - int loopback 0
      lines:
        - ip address 10.10.10.10/32
        - ip router ospf 1 area 0
```

Tương tự NXOS1 ta cấu hình cho NXOS2:

+ Bật các dịch vụ cần thiết cho Switch Nexus 9000: SSH, DHCP, OSPF, Vlan Segment

```
- name: NXOS2
  hosts: nxos2
  gather_facts: no

  tasks:
    - name: Enable services
      nxos_config:
        lines:
          - feature ospf
          - feature vn-segment-vlan-based
          - feature nv overlay
          - system jumbomtu 9216

    - name: Config ospf
      nxos_config:
        lines:
          - router ospf 1
```

+ Cấu hình cho interface e1/2 (interface kết nối 2 Switch Nexus) và quảng bá ospf

```
  - name: Set ospf for int e1/2
    nxos_config:
      parents:
        - int e1/2
      lines:
        - no switchport
        - no shutdown
        - ip address 100.1.1.2/30
        - ip router ospf 1 area 0
```

+ Tạo Vlan 50 và gắn cổng kết nối là e1/3

```
- name: Create vlan 50
  nxos_config:
    parents:
      - vlan 50
    lines:
      - vn-segment 5000

- name: Access vlan 50
  nxos_config:
    parents:
      - int e1/3
    lines:
      - switchport mode access
      - switchport access vlan 50
```

+ Cấu hình nve

```
- name: Creat NVE
  nxos_config:
    parents:
      - int nve 1
    lines:
      - no shut
      - source-interface loopback 0
      - member vni 5000
      - ingress-replication protocol static
      - peer-ip 10.10.10.10
```

+ Tạo looback và quảng bá ospf

```
- name: Config ospf for looback
  nxos_config:
    parents:
      - int loopback 0
    lines:
      - ip address 20.20.20.20/32
      - ip router ospf 1 area 0
```

Tham khảo: https://github.com/vnpro149/Ansible/blob/master/vxlan-nxos/nxos.yml

Lab cấu hình VxLan căn bản trên Cisco Nexus: https://www.forum.vnpro.org/forum/ccnp-enterprise/encor/420749-lab-1-c%E1%BA%A5u-h%C3%ACnh-vxlan-c%C4%83n-b%E1%BA%A3n-tr%C3%AAn-cisco-nexus

Thực hiện chạy các Playbook bằng câu lệnh sau ansible-playbook nxos.yml

```
root@DESKTOP-KMNS09Q:/etc/ansible# ansible-playbook nxos.yml

PLAY [NXOS1] ***********************************************************************************************************************

TASK [Enable services] *************************************************************************************************************
ok: [nxos1]

TASK [Config ospf] *****************************************************************************************************************
ok: [nxos1]

TASK [Set ospf for int e1/2] *******************************************************************************************************
changed: [nxos1]

TASK [Create vlan 50] **************************************************************************************************************
ok: [nxos1]

TASK [Access vlan 50] **************************************************************************************************************
changed: [nxos1]

TASK [Creat NVE] *******************************************************************************************************************
changed: [nxos1]

TASK [Config ospf for looback] *****************************************************************************************************
changed: [nxos1]
```

Đã cấu hình trên NXOS1

```
PLAY [NXOS2] ***********************************************************************************************************************

TASK [Enable services] *************************************************************************************************************
ok: [nxos2]

TASK [Config ospf] *****************************************************************************************************************
ok: [nxos2]

TASK [Set ospf for int e1/2] *******************************************************************************************************
changed: [nxos2]

TASK [Create vlan 50] **************************************************************************************************************
ok: [nxos2]

TASK [Access vlan 50] **************************************************************************************************************
changed: [nxos2]

TASK [Creat NVE] *******************************************************************************************************************
changed: [nxos2]

TASK [Config ospf for looback] *****************************************************************************************************
changed: [nxos2]

PLAY RECAP *************************************************************************************************************************
nxos1                      : ok=7    changed=4    unreachable=0    failed=0
nxos2                      : ok=7    changed=4    unreachable=0    failed=0
```

Đã cấu hình trên NXOS2

Đã cấu hình hoàn tất.

Kiểm tra trên Nexus Switch

```
NXOS1# show ip ospf neighbors
 OSPF Process ID 1 VRF default
 Total number of neighbors: 1
 Neighbor ID     Pri State            Up Time  Address         Interface
 20.20.20.20       1 FULL/BDR         07:47:49 100.1.1.2       Eth1/2
```

Kiểm tra vxlan data

```
NXOS1# show nve vni data-plane
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       SU - Suppress Unknown Unicast
       Xconn - Crossconnect
       MS-IR - Multisite Ingress Replication

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      5000     UnicastStatic     Up    DP   L2 [50]
```

Kiểm tra cổng virtual network

```
NXOS1# show nve peers detail
Details of nve Peers:
----------------------------------------
Peer-Ip: 20.20.20.20
    NVE Interface       : nve1
    Peer State          : Up
    Peer Uptime         : 00:01:29
    Router-Mac          : n/a
    Peer First VNI      : 5000
    Time since Create   : 00:01:29
    Configured VNIs     : 5000
    Provision State     : peer-add-complete
    Learnt CP VNIs      : 5000
    vni assignment mode : SYMMETRIC
    Peer Location       : N/A
```

Kiểmt tra ping giữa 2 thiết bị trong cùng vlan 50