**ĐỀ THI FINAL LAB LỚP DEVNET**

**Câu 1 ( Git )** : Học viên thực hiện clone repository finallab trên github của vnpro ( <https://github.com/vnpro149> ) về máy mình và truy cập thư mục có tên của mình để tiến hành làm bài.

**Lưu ý:** sử dụng git command, không được thao tác trên Web.

Trong suốt quá trình làm bài, học viên push kết quả lên repository với tên folder tương ứng với bài đó ( ví dụ : playbook của ansible sẽ được push lên repository finallab/<tên học viên>/Ansible/<tên playbook> )

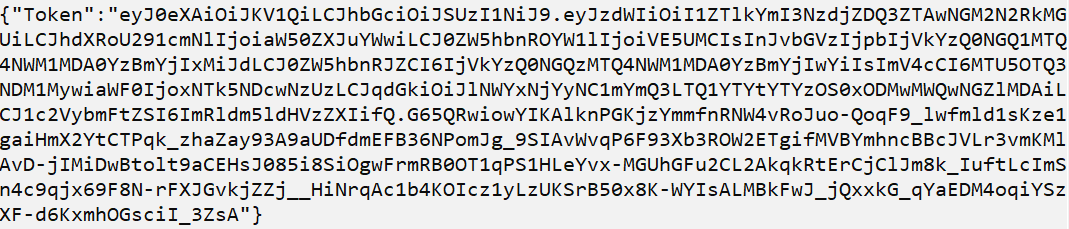
**Câu 2 ( cURL )** : Học viên dùng cURL để lấy token của DNA Center.

Thông tin server DNA Center bao gồm

Thông tin tài khoản là devnetuser/Cisco123! .

**Lưu ý** : username và password được mã hóa base64.

**Kết quả :**

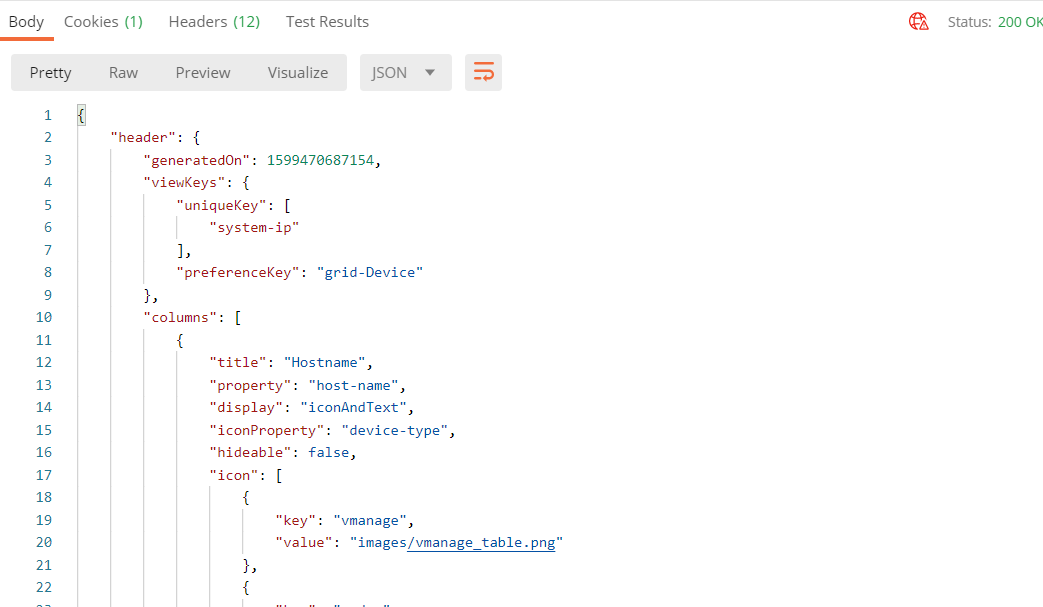


**Câu 3 ( POSTMAN )** : Học viên sử dụng Postman tương tác với SD-WAN vManage API để lấy thông tin các thiết bị.

Các thông tin cho trước :

* IP của vManage : 10.215.26.189:8443.
* Thông tin tài khoản : admin/admin.
* Vào API Document ở https://10.215.26.189:8443/apidocs để tìm kiếm thông tin thêm.

**Kết quả :**



**Câu 4 ( Python )** : Học viên thực hiện viết code Python tương tác với Meraki API thực hiện các yêu cầu sau: LEARN PYTHON THE HARD WAY

a) Hàm của phương thức get (phải có headers, url).

b) Hàm lấy các thiết bị trong organization inventory (organization id có thể sử dụng Postman để lấy và gán giá trị đó cho biến org\_id, chuyển về định dạng json có indent 4).

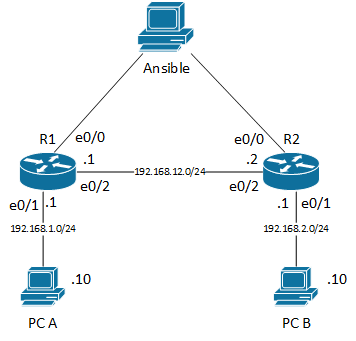
c) Hàm in danh sách các thiết bị có network id là null trong câu b (sử dụng vòng lặp để dò thiết bị phù hợp yêu cầu, sau đó gán các thiết bị vào một list và in ra màn hình list đó).

**Lưu ý** :

* Các thông tin cho trước là một file python meraki\_info chứa base\_url, api key.
* Organization được sử dụng trong câu này có tên là Public API Lab.
* Truy cập API Document ở trang developer của Cisco phần Meraki để xem các thông tin cần thiết.

**Câu 5 ( Ansible )** : địa chỉ Ansible Server: 10.215.26.130, username: admin, password: vnpro@123

**Sơ đồ : (địa chỉ EVE chứa sơ đồ LAB Ansible: 10.215.26.151; LAB Ansible – DEVNET)**



**Yêu cầu :**

1, Cấu hình SSH cho thiết bị , đảm bảo máy chạy Ansible kết nối được đến với Router 1 và Router 2.

**Lưu ý** : cổng e0/0 của mỗi thiết bị cấu hình địa chỉ ip động ( dhcp ).

2, Tạo file inventory để kết nối đến thiết bị.

3, Học viên sử dụng host\_vars hoặc group\_vars để lưu trữ các giá trị như IP, netmask , …

4, Viết file playbook cấu hình các yêu cầu sau :

* Cấu hình IP các cổng như hình vẽ.
* Cấu hình static route để 2 PC A và PC B thấy nhau ( sử dụng module ios\_static\_route ).
* Ping kiểm tra kết nối thành công từ e0/1 của R2 đến R1 ( sử dụng module ios\_ping ).

5, Học viên đặt IP cho PC A và PC B , đảm bảo ping thành công từ PC A đến PC B.

**Câu 6 ( Docker ) :**

1, Học viên thực hiện viết Dockerfile để tạo một image chạy app python có sẵn theo các yêu cầu sau :

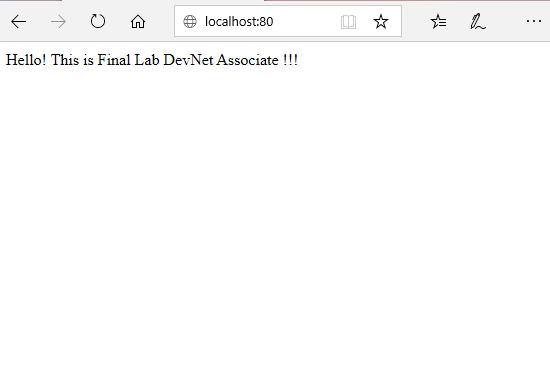
**Yêu cầu :**

* Sử dụng ubuntu official image
* Thực thi câu lệnh “apt-get update -y” và “apt-get install -y python-pip” trên image ubuntu distro
* Tạo thư mục /usr/src/app/ với câu lệnh “mkdir -p /usr/src/app/”
* Copy thư mục requirements.txt và app.py vào thư mục /usr/src/app/ vừa tạo ở trên
* Chạy câu lệnh “pip install --no-cache-dir -r /usr/src/app/requirements.txt để cài đặt các thư viện phụ thuộc.
* Cấu hình cho app listen port 5000
* Khởi chạy container với câu lệnh “python /usr/src/app/app.py”

2, Thực hiện build image từ Dockerfile vừa tạo bằng câu lệnh docker build.

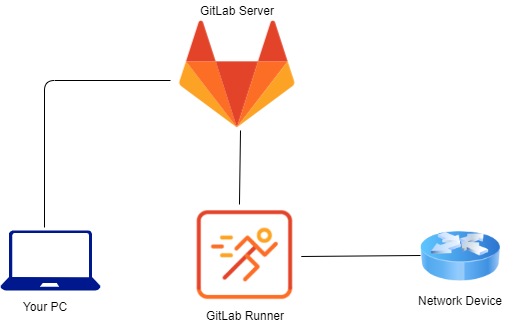
3, Tiến hành khởi chạy container bằng câu lệnh docker run với port là 80.

**Kết quả :**



**Câu 7 ( CI/CD ) :**

**Mô hình :**



**Yêu cầu :**

Học viên xây dựng pipeline để thực hiện quá trình CI/CD triển khai cấu hình của network device với gitlab server và gitlab runner sử dụng các Ansible playbook từ câu 5. ( Network Device tượng trưng cho sơ đồ của Câu 5 )

**Hướng dẫn thực hiện:**

1, Học viên thực hiện cài đặt và cấu hình cơ bản gitlab server.

2, Cài đặt và cấu hình gitlab runner.

3, Cấu hình SSH-Key với gitlab server.

4, Viết pipeline theo yêu cầu đề bài.

5, Push code từ local lên gitlab server.

6, Kiểm tra quá trình chạy của pipeline.

**Pipeline tham khảo :**

|  |
| --- |
| image: centos:latest  stages:  - validate  - deploy to test  validate:  stage: validate  script:  - echo "Validate stage"  deploy to test:  stage: deploy to test  tags:  - runner1  image: centos:latest  script:  - yum update -y  - yum install -y epel-release  - yum install ansible -y  - yum install python3-pip  - pip3 install paramiko  - cp -rf \* /etc/ansible/  - ssh-keyscan 10.215.26.211 >> ~/.ssh/known\_hosts  - cd /etc/ansible/host\_vars/  - ansible-playbook hostname.yml |