**Lab 3: Cấu hình SDN trên Aruba**

**1. Mục đích thực tập**

* Thực thi mạng SDN trên Switch Aruba
* Bài thực hành giúp sinh viên trong việc cài đặt, cấu hình hạ tầng mô hình thí nghiệm SDN trong phòng thí nghiệm bộ môn Hệ Thống Viễn Thông.
* Yêu cầu đầu ra: Xây dựng hệ thống SDN với pox controller để trích xuất dữ liệu và xử lý dữ liệu

**2. Thiết bị và dụng cụ thí nghiệm**

* Máy tính PC có sẵn card mạng, cài đăt sẵn một số phần mềm Mininet
* Switch Aruba tại Bộ môn Hệ thống Viễn thông

**3. Nội dung thực tập**

*3.1.Tìm hiểu lý thuyết*

- Cài đặt cấu hình để kích hoạt chế độ Openflow trên switch Aruba

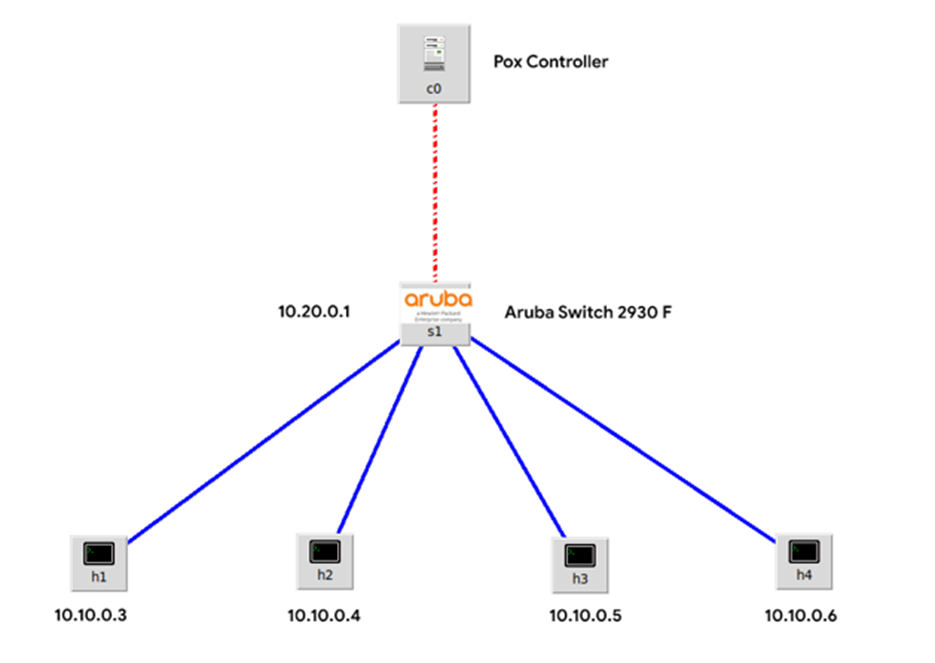
- Kết nối switch với POX Controller

- Điều kiện cần

* Kiến thức nền SDN, ảo hóa, L2 Network, L3 Network (Advance).
* Làm quen với từng thành phần: Controller, Switch, Hosts.
* Kiến thức Vlan: tagged (trunk), untagged (access).
* Kiến thức Python, Bash Scripting, Linux.
* Sử dụng wireshark để bắt và phân tích gói tin

# *Thực hành*

1. **Mô hình hệ thống mạng:**



Hệ thống mạng bao gồm:

4 host ubuntu: Sử dụng ubuntu sẽ thuận tiện trong việc cài đặt các gói tin, các gói cài đặt thêm và đặc biệt là các công cụ tấn công. Các host có địa chỉ IP như trên

1 Switch Aruba 2930F (s1): Switch là một trong những thành phần quan trọng trong việc luân chuyển gói tin, bóc tách gói tin và quản lý các VLAN thành phần. Với thiết bị 2930F, Hãng đã cung cấp một bản hướng dẫn sử dụng rất chi tiết về cách cấu hình với giao thức OpenFlow.

1 Bộ điều khiển ubuntu (c0): Đóng vai trò quan trọng trong việc điều khiển, giao tiếp với switch qua Northbound API. Địa chỉ của controller tùy ý, trong manual ta mặc định đặt địa chỉ của c0 là 10.20.0.3. Controller sử dụng phần mềm Pox phiên bản 0.6

Bộ điều khiển có thể là bất cứ thiết bị nào có hệ điều hành như laptop cá nhân, một chiếc PC hoặc thậm chí một chiếc Raspberry Pi.

1. **Thực hành:**

### **B1: Đặt địa chỉ cho các host h1 h2 h3 h4, c0. Đảm bảo các host đã tắt tường lửa**

Đối với ubuntu: *sudo ufw disable* =>tắt tường lửa 1 lần và từ đó về sau

Đối với window:tắt trong phần cài đặt chung

Như trong hình trên các host có ip tương ứng:

h1 - 10.10.0.3

h2 - 10.10.0.4

h3 - 10.10.0.5

h4 - 10.10.0.6

c0 - 10.20.0.3

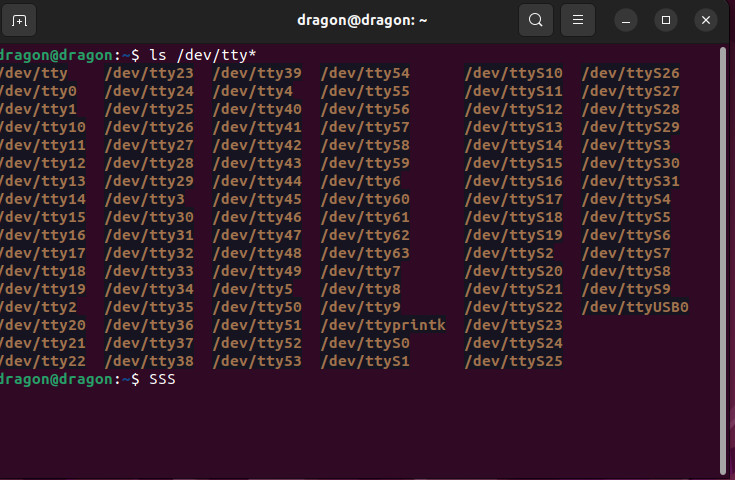
Việc cấu hình vlan 1 có interface vlan để vlan controller có default gateway

### **B2: Cấu hình vlan 2 gồm 4 host trên, vlan 1 là vlan controller như sau**

Tại máy sử dụng để cấu hình switch (trong trường hợp này đồng thời là controller)

Terminal:

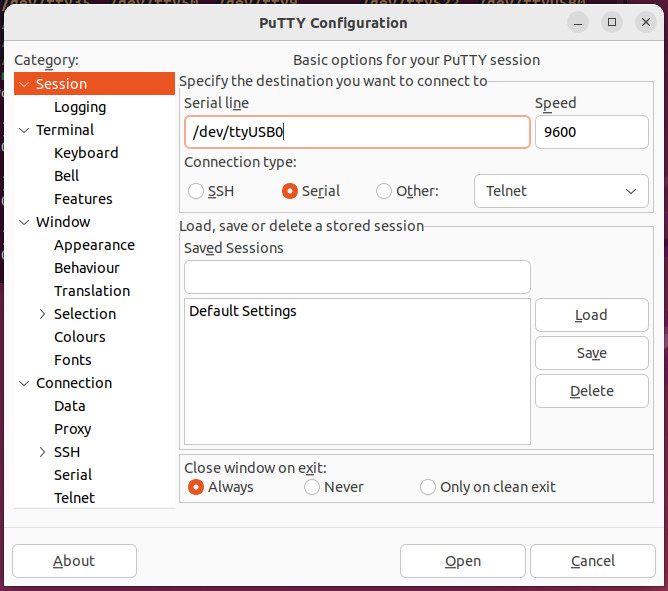
*ls /dev/tty\** => hiển thị tất cả các cổng serial ảo và thật, trong đó , cổng USB0 là cổng ta cần tìm (cổng thật)



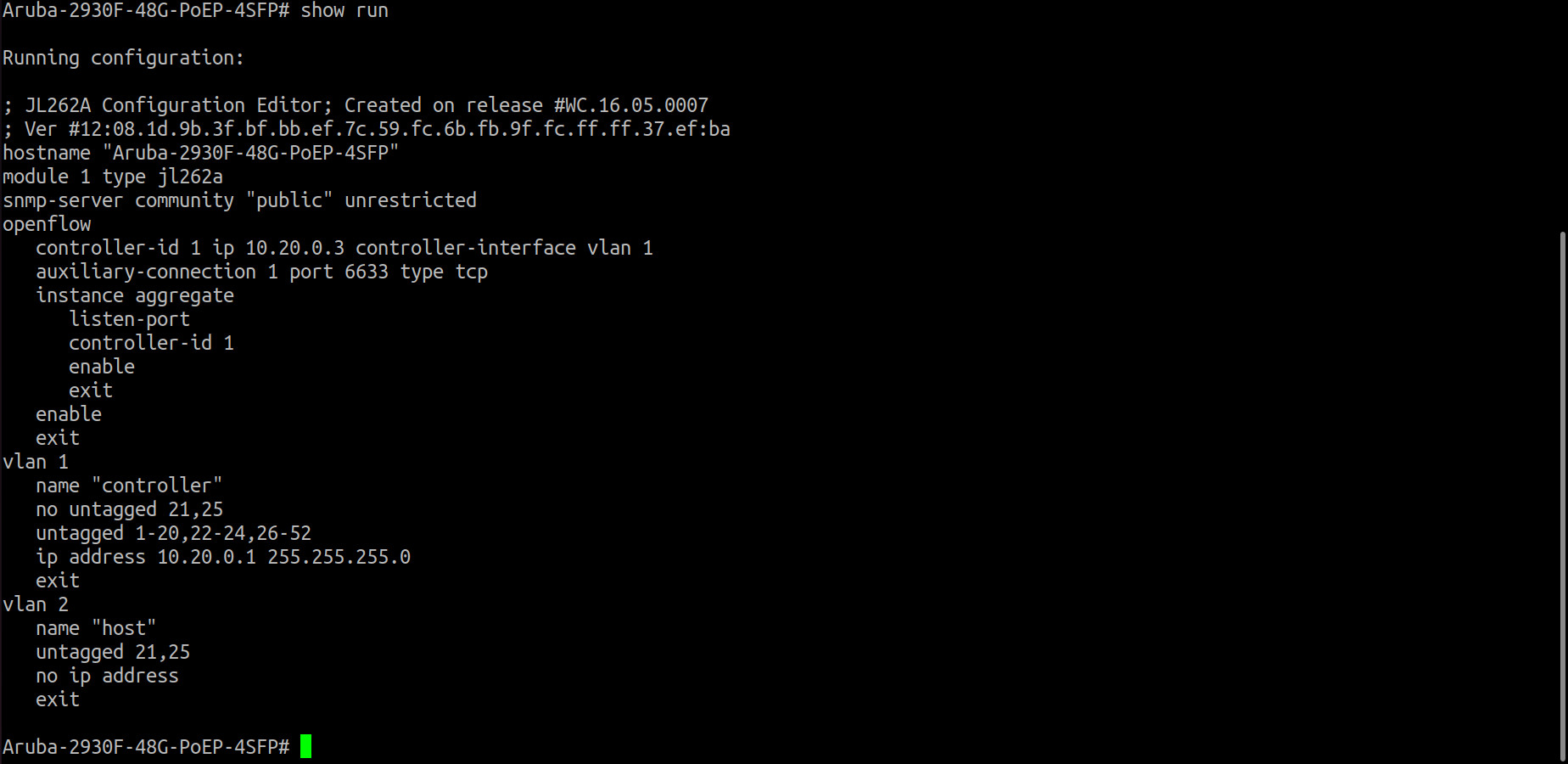
Đối với windows , ta mở device manager để check cổng serial (COM)

*sudo putty* => mở putty

Setup như hình, chọn open



*sh run* => ra kết quả cấu hình hiện tại



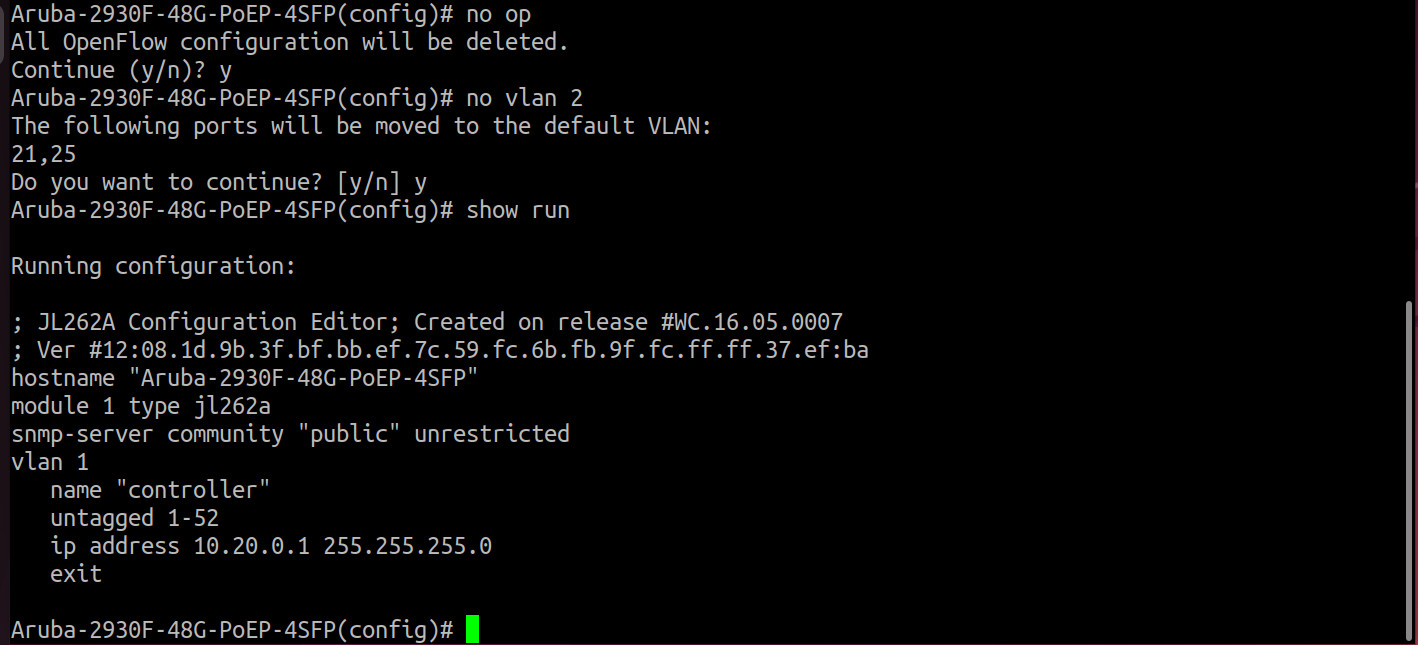
Tại đây ta có thể thấy những cấu hình cơ bản của aruba đã cấu hình openflow, ta cần xóa cấu hình đã có trước.

*config terminal =>* vào cấu hình config của switch

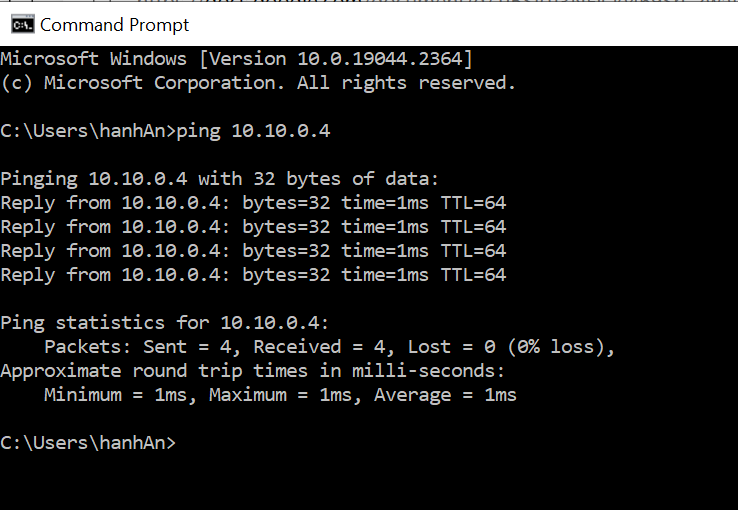
*no op* => tắt giao thức openflow, press y

xóa tất cả vlan cũ (ngoại trừ vlan 1 - vlan mặc định)

*no vlan 2* => xóa vlan 2, giờ chỉ còn vlan 1



Thực hiện ping 2 host với nhau => thành công



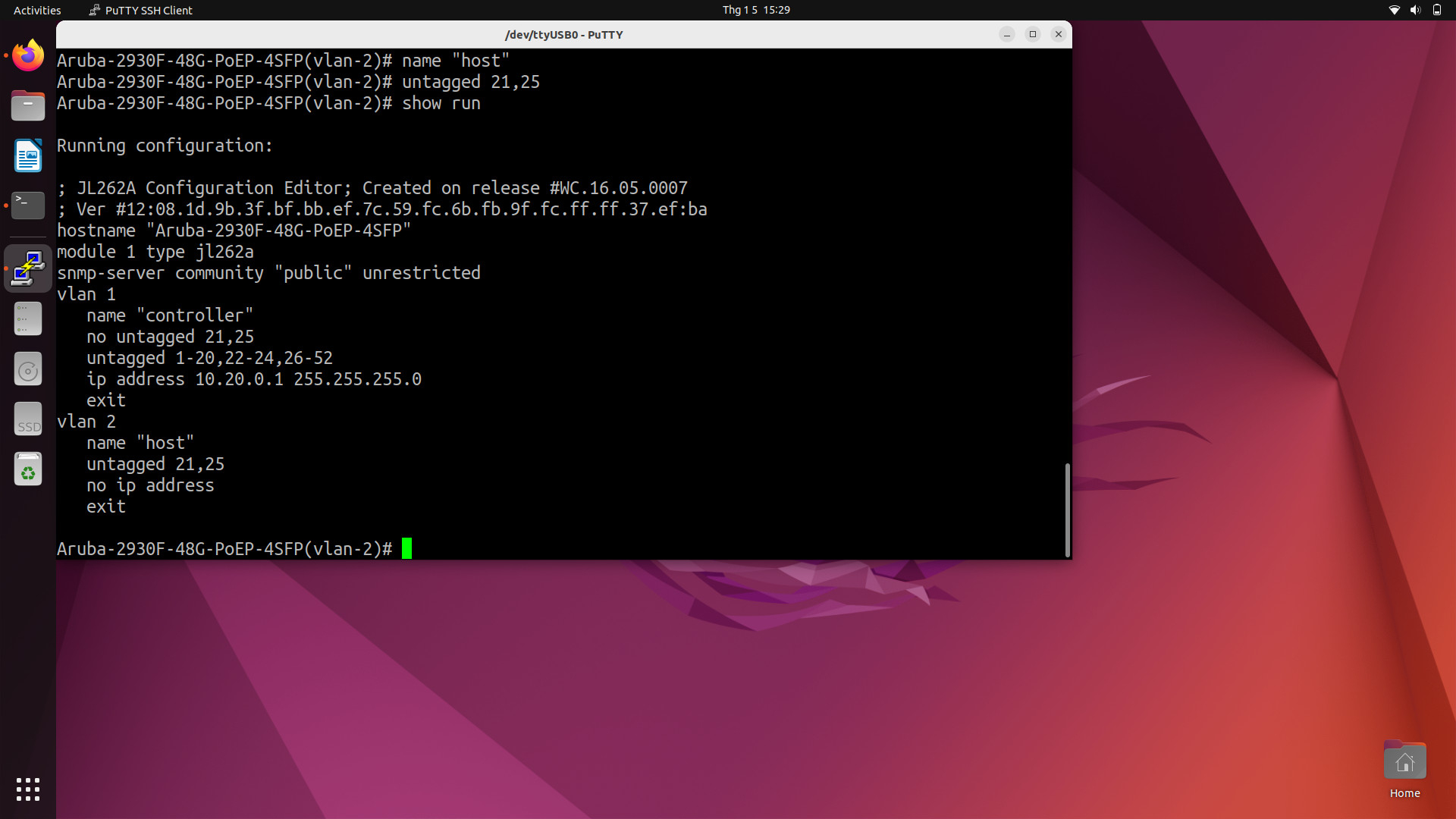
Cấu hình vlan 2 để chứa 4 host

*vlan 2*

*name “host”*

*untagged 21,25* (do 2 host sử dụng 2 port 21, 25 trên switch)

*sh run*

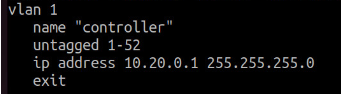


=> cấu hình vlan 2 thành công

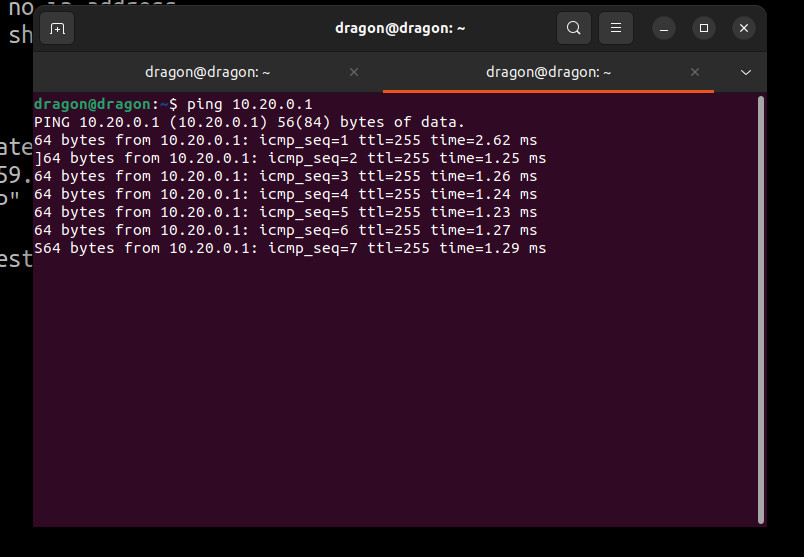
Vlan 1 chứa controller => đặt ip address cho interface vlan 1

*vlan 1*

*ip address 10.20.0.1 255.255.255.0*



ping controller đến 10.20.0.1



### **B3: Cấu hình openflow**

Thực hiện các cấu hình như hình:

*> openflow*

*> enable*

*> controller 1 ip 10.20.0.3 controller-interface vlan 1* (config controller)

*> auxiliary-connection 1 port 6633 type tcp* (kết nối port default:6633 với giao thức TCP)

*> instance aggregate* (gom tất cả các vlan lại với nhau)

*> listen-port*

*> controller 1* (enable controller)

*> enable*

*> exit*

*> enable*

*> exit*



Sau khi cấu hình thành công, các bạn sẽ không ping được cho nhau, bước tiếp theo:

### **B4: Khởi động controller**

Tải phần mềm pox trên các nguồn đã cung cấp

*https://github.com/att/pox*

*cd pox* (vào thư mục chứa pox)

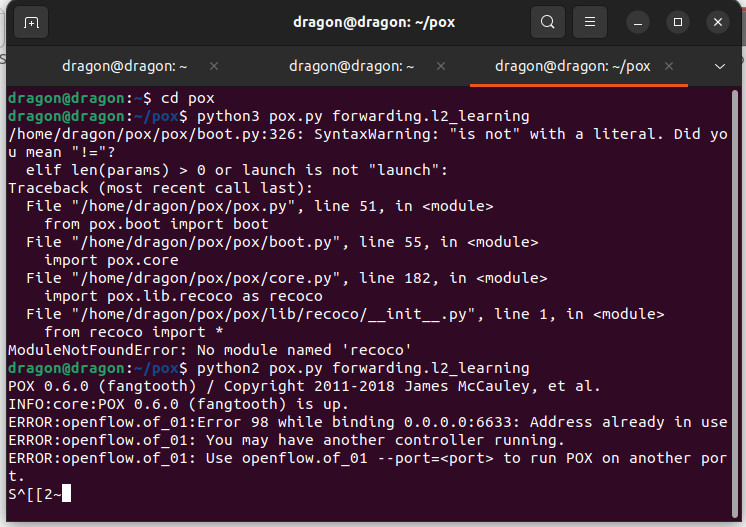
cài python2 bằng những câu lệnh :

*sudo service packagekit restart*

*sudo add-apt-repository universe*

*sudo apt install python2*

Chạy lệnh: *python2 pox.py forwarding.l2\_learning*

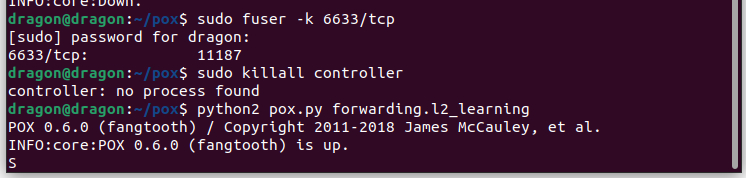


Nếu python3 lỗi => python2

Nếu bị lỗi pox “already in use” như trên hình

*sudo fuser -k 6633/tcp*

*sudo killall controller*



Nếu chưa thấy có sự kết nối, vào lại putty và sử dụng lệnh:

*op ins agg dis*

*op ins agg ena*

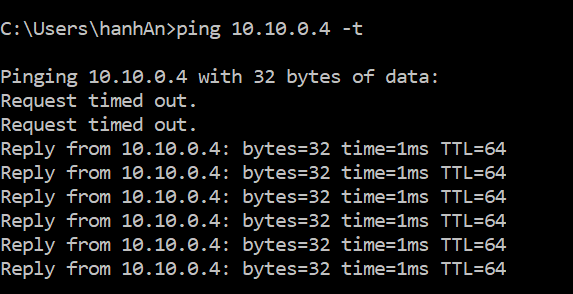
Quay trở lại putty sử dụng lệnh *> show op ins agg*



Nếu các bạn thấy bảng này , các bạn đã thành công trong việc kết nối Switch và controller giao tiếp bằng Northbound API

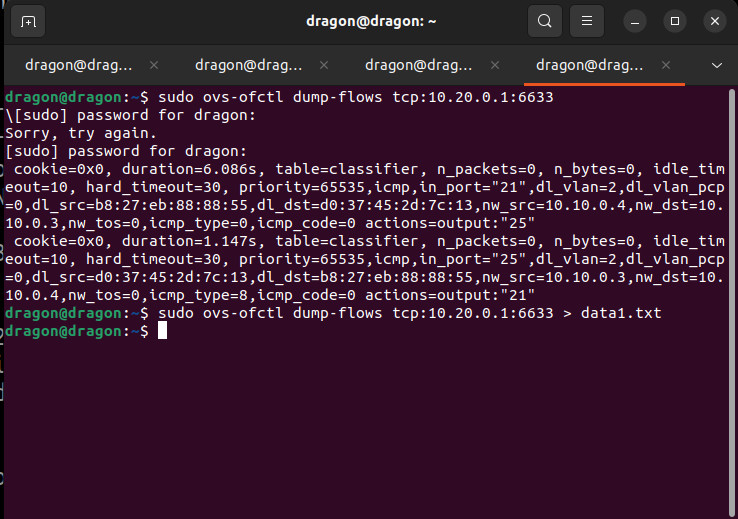
Connection Status: Connected => Switch đã kết nối được với controller và sẵn sàng trao đổi bản tin OpenFlow

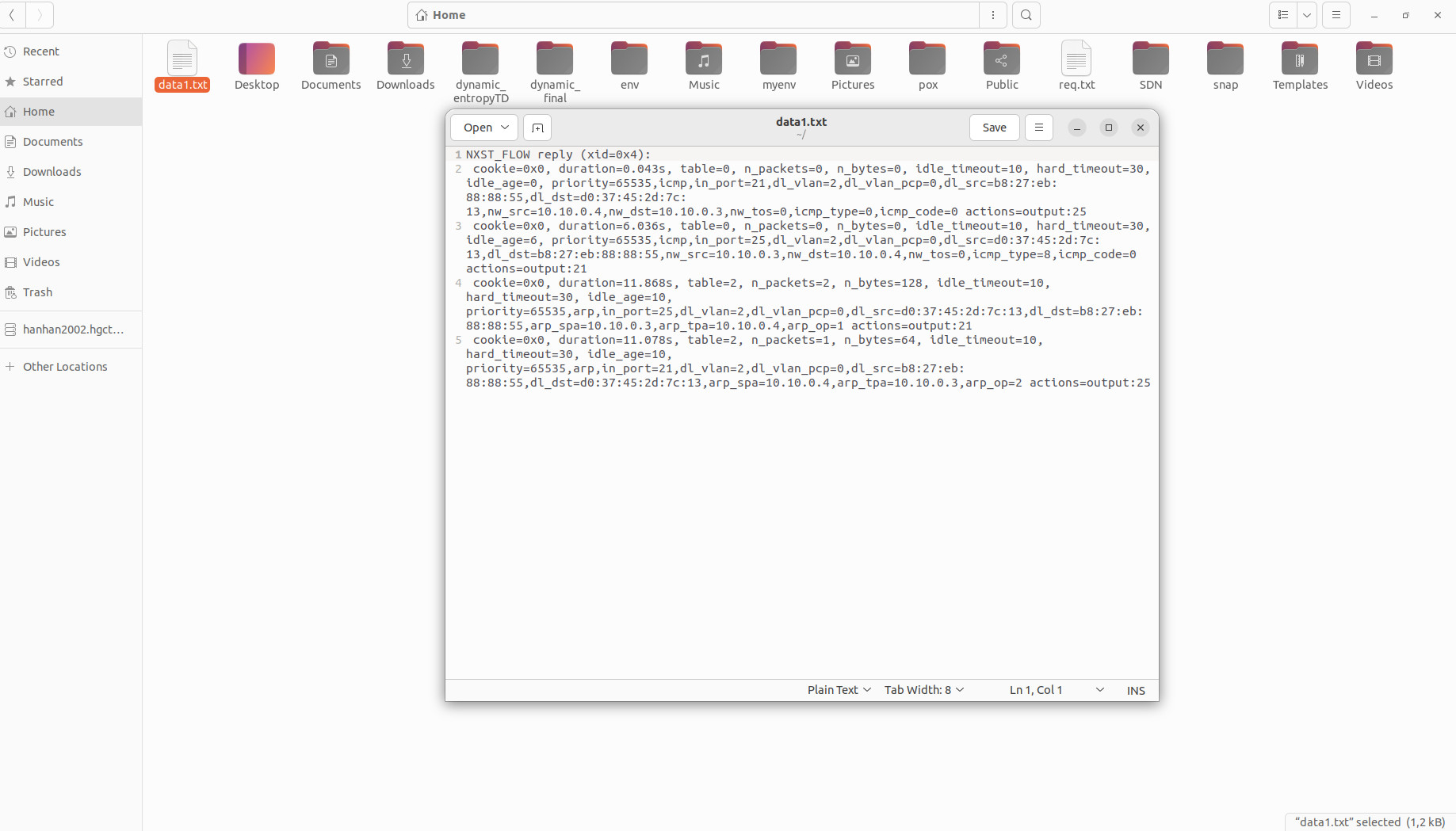
Kiểm tra kết nối, ping các host với nhau



Từ controller sử dụng lệnh:

*sudo ovs-ofctl dump-flows tcp:10.20.0.1:6633>flowTable.txt* (xuất flow entries ra file text)





**B5: Sử dụng Wireshark để bắt và phân tích gói tin OpenFlow**

**4. Nội dung báo cáo thực hành**

* 1. Các nội dung thực tập đã được tiến hành, các kết quả thí nghiệm
  2. Nhận xét đánh giá kết quả