

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN VĂN HUẤN

ĐỒ ÁN CHUYÊN NGÀNH

**Cấu hình mô hình mạng theo hướng tự động hoá trên nền
tảng EVE-NG và so sánh với cách cấu hình truyền thống
bằng lệnh CLI.**

**Config topology using network automation in EVE-NG platform
and compare with traditional configuration using Command Line
Interface.**

KỸ SƯ NGÀNH MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

TP. HỒ CHÍ MINH, 2024

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN VĂN HUẤN – 21522108

ĐỒ ÁN CHUYÊN NGÀNH

**Cấu hình mô hình mạng theo hướng tự động hoá trên nền
tảng EVE-NG và so sánh với cách cấu hình truyền thống
bằng lệnh CLI.**

**Config topology using network automation in EVE-NG platform
and compare with traditional configuration using Command Line
Interface.**

KỸ SƯ NGÀNH MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

GIẢNG VIÊN HƯỚNG DẪN

Th.S TRẦN THỊ DUNG

TP. HỒ CHÍ MINH, 2024

THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

LỜI CẢM ƠN

Lời đầu tiên em muốn gửi lời cảm ơn chân thành sâu sắc tới giảng viên hướng dẫn Th.S Trần Thị Dung đã tận tình hỗ trợ nhóm để hoàn thành đồ án chuyên ngành. Qua đồ án này, em đã tích lũy được nhiều kiến thức bổ ích về chuyên môn cũng như rèn luyện các kỹ năng cần thiết để phát triển sau này.

Mặc dù em đã nỗ lực hết mình để hoàn thành đồ án với tinh thần cao nhất, nhưng vì thiếu kinh nghiệm cũng như một số hạn chế khác, không tránh khỏi những sai sót và khuyết điểm. Em trân trọng và rất mong nhận được sự hiểu biết, chia sẻ và góp ý chân thành từ phía quý thầy cô cũng như từ các bạn sinh viên khác, để từ đó, em có thể tiếp tục hoàn thiện bản thân và cải thiện kết quả học tập của mình.

Em xin chân thành cảm ơn!

TP.HCM, ngày 28 tháng 6 năm 2024

MỤC LỤC

Chương 1. MỞ ĐẦU.....	2
1.1. Lí do chọn đề tài.....	2
1.2. Mục đích của đề tài	2
1.3. Đối tượng nghiên cứu.....	3
1.4. Phạm vi nghiên cứu.....	3
Chương 2. TỔNG QUAN.....	4
2.1. Các công cụ và môi trường	4
2.1.1. EVE-NG.....	4
2.1.2. Ansible	5
2.1.3. Netmiko Python	6
2.2. Mô hình mạng đề xuất.....	7
2.3. Cơ sở lý thuyết	8
2.3.1. Các giao thức mạng sử dụng.....	8
2.3.1.1. Network Address Translation (NAT).....	8
2.3.1.2. Open Shortest Path First (OSPF).....	9
2.3.1.3. Virtual Local Area Network (VLAN)	10
2.3.1.4. VLAN Trunking Protocol (VTP)	11
2.3.1.5. Rapid Spanning Tree Protocol (RSTP)	12
2.3.1.6. Hot Standby Router Protocol (HSRP).....	12
Chương 3. TRIỂN KHAI.....	14
3.1. Cài đặt EVE-NG và các image mô phỏng.....	14
3.2. Cài đặt thư viện Netmiko Python trên Linux Server.....	23
3.3. Cài đặt Ansible	24

3.4.	Viết playbook cho Ansible	24
Chương 4.	KIỂM THỬ MÔ HÌNH	29
4.1.	Viết các kịch bản của mô hình mạng	29
4.2.	Trường hợp 1: Switch Core 1 bị shutdown	38
4.3.	Trường hợp 2: Router Gateway 1 bị shutdown.....	41
Chương 5.	TỔNG KẾT	43
5.1.	So sánh giữa 2 hình thức cấu hình	43
5.2.	Kết luận	43
5.2.1.	Những công việc mà nhóm đã thực hiện được	43
5.2.2.	Phương hướng phát triển.....	44
5.3.	Tài liệu tham khảo	44
5.4.	Demo sản phẩm	44

DANH MỤC HÌNH

Hình 2-1: EVE-NG	4
Hình 2-2: Ansible	5
Hình 2-3: Netmiko	6
Hình 2-4: Mô hình mạng.....	7
Hình 3-1: Sau khi download file iso EVE-NG	14
Hình 3-2: Giao diện của VMWare Workstation Pro.....	14
Hình 3-3: File EVE-CE.ovf.....	15
Hình 3-4: Mục Settings của máy ảo	16
Hình 3-5: EVE-NG sau khi cài đặt	16
Hình 3-6: Màn hình đăng nhập của EVE-NG.....	17
Hình 3-7: Đăng nhập vào WinSCP	18
Hình 3-8: Giao diện của WinSCP sau khi thêm image.....	19
Hình 3-9: Thêm Router vào mô hình mạng	20
Hình 3-10: Thêm Switch Layer 2 và Switch Layer 3 (2 Switch có thể dùng 1 image)	21
Hình 3-11: Thêm VPCs.....	22
Hình 3-12: Giao diện của Linux Server (Ubuntu 18.04.06-desktop)	23
Hình 3-13: Kiểm tra phiên bản Netmiko	24
Hình 3-14: Kiểm tra phiên bản của Ansible	24
Hình 3-15: Các thư mục trong folder ansible	25
Hình 3-16: File hosts	25
Hình 3-17: File Core_config.yml.....	26
Hình 3-18: Sau khi chạy file yml	27
Hình 3-19: Kiểm tra các cấu hình sau khi chạy Ansible.....	28
Hình 4-1: Bật chế độ SSH trên Switch Core 1	30
Hình 4-2: Cấp địa chỉ IP cho Switch Core 1 để SSH đến.....	31
Hình 4-3: Các file kịch bản	32

Hình 4-4: File Core_configuration.py phần định nghĩa	33
Hình 4-5: File Core_configuration.py phần cấu hình	34
Hình 4-6: File Core_configuration.py phần đẩy các cấu hình vào thiết bị	35
Hình 4-7: Sau khi chạy lệnh 'python ./[tên file]' trên terminal.....	35
Hình 4-8: Kiểm tra cấu hình trên thiết bị	36
Hình 4-9: File log các cấu hình được đẩy vào thiết bị	37
Hình 4-10: Trước khi bị Shutdown, trạng thái của Core 1 là Active cho VLAN 10 và 20	38
Hình 4-11: Trên Core 2 sẽ là trạng thái Standby (chờ) cho 2 VLAN 10 và 20	39
Hình 4-12: Trace 8.8.8.8 trên VPC thuộc VLAN 10	39
Hình 4-13: Sau khi shutdown Core 1, Core 2 sẽ chiếm quyền	40
Hình 4-14: Thực hiện trace trên VPC 10	41
Hình 4-15: Trace 8.8.8.8 trên VPC10	41
Hình 4-16: Trace 8.8.8.8 trên VPC10	42

DANH MỤC BẢNG

Bảng 1: Bảng địa chỉ IP cho thiết bị	30
Bảng 2: Bảng so sánh 2 kiểu cấu hình	43

DANH MỤC TỪ VIẾT TẮT

TÓM TẮT ĐỒ ÁN

Đồ án này tập trung vào việc tự động hóa cấu hình mạng bằng cách sử dụng nền tảng mô phỏng EVE-NG, thư viện Netmiko và công cụ Ansible. EVE-NG được sử dụng để mô phỏng các thiết bị mạng ảo, Netmiko giúp kết nối và thực thi lệnh trên các thiết bị, trong khi Ansible quản lý và tự động hóa các tác vụ cấu hình. Mục tiêu là tạo ra một hệ thống tự động hóa cấu hình mạng hiệu quả, tiết kiệm thời gian, đảm bảo tính nhất quán và giảm thiểu sai sót do cấu hình thủ công. Đồ án cũng đề xuất các phương hướng phát triển như mở rộng kịch bản mô phỏng, tích hợp công nghệ mới, tăng cường bảo mật, và cải thiện hiệu suất và độ tin cậy của hệ thống.

Chương 1. MỞ ĐẦU

1.1. Lí do chọn đề tài

Trong quá trình học các môn học về mạng máy tính thì việc cấu hình các thiết bị mạng là việc bắt buộc phải làm để cho mô hình mạng được hoạt động hiệu quả. Việc cấu hình sẽ có nhiều công việc lặp đi lặp lại và nếu trong quá trình làm, chúng ta vô tình cấu hình sai một dòng lệnh hoặc bỏ sót một câu lệnh cũng có thể ảnh hưởng lớn đến việc hoạt động của hệ thống mạng.

Vì vậy, đề tài này sẽ giới thiệu cách cấu hình và quản lý mô hình mạng một cách tự động và đơn giản hoá việc tìm và khắc phục lỗi.

1.2. Mục đích của đề tài

Mục đích của đồ án này là đề xuất và triển khai một mô hình mạng ba lớp (Three-Layer Hierarchical Model in Cisco) trên môi trường mô phỏng EVE-NG, bao gồm lớp truy cập (Access Layer), lớp phân phối (Distribution Layer) và lớp lõi (Core Layer). Đồ án này sẽ sử dụng các công cụ tự động hoá như Netmiko Python để cấu hình các thiết bị mạng và Ansible để quản lý các tác vụ cấu hình. Cụ thể, đồ án nhằm đạt được các mục tiêu sau:

- **Thiết kế và mô phỏng một mô hình mạng ba lớp trên EVE-NG:** Xây dựng và cấu hình các thiết bị mạng tương ứng cho từng lớp.
- **Sử dụng Netmiko Python:** Tự động hoá việc cấu hình các thiết bị mạng thông qua script Python sử dụng thư viện Netmiko.
- **Quản lý cấu hình bằng Ansible:** Tích hợp Ansible để quản lý và thực thi các tác vụ cấu hình trên các thiết bị mạng một cách hiệu quả.
- **Đánh giá khả năng dự phòng của hệ thống mạng:** Thực hiện các kịch bản gặp lỗi để đánh giá khả năng dự phòng và tính ổn định của hệ thống mạng.

1.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu trong đề án này bao gồm:

- **Thiết bị mạng:** Các thiết bị như switch và router thuộc các lớp truy cập, phân phối và lõi trong mô hình mạng ba lớp.
- **Các công cụ tự động hoá:**
 - + **Netmiko:** Một thư viện của Python mã nguồn mở cho phép tự động hoá việc cấu hình thiết bị mạng thông qua giao thức SSH.
 - + **Ansible:** Một công cụ tự động hoá mã nguồn mở dùng để quản lý cấu hình và thực thi các tác vụ trên các thiết bị mạng.
- **Công nghệ và các giao thức mạng:** Các giao thức và công nghệ mạng như VLAN, STP, OSPF, HSRP, NAT sử dụng trong việc cấu hình mô hình mạng ba lớp.

1.4. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề án bao gồm các yếu tố sau:

- **Thiết kế và triển khai mô hình mạng ba lớp:** Bao gồm việc xây dựng và cấu hình các thiết bị mạng cho các lớp truy cập, phân phối, và lõi.
- **Sử dụng Netmiko để tự động hóa cấu hình:** Viết và thực thi các script Python để tự động hóa việc cấu hình thiết bị mạng.
- **Tích hợp Ansible để quản lý cấu hình:** Sử dụng Ansible để viết playbook và quản lý việc triển khai cấu hình trên các thiết bị mạng.
- **Kiểm tra và đánh giá:** Thực hiện kiểm tra tính dự phòng khi gặp sự cố và đánh giá tính sẵn sàng của mô hình mạng
- **Tài liệu hóa quy trình và kết quả:** Ghi lại toàn bộ quá trình từ thiết kế, triển khai đến đánh giá kết quả, cũng như những khó khăn gặp phải và cách giải quyết.

Chương 2. TỔNG QUAN

2.1. Các công cụ và môi trường

2.1.1. EVE-NG



Hình 2-1: EVE-NG

EVE-NG là một công cụ mô phỏng mạng cung cấp giao diện người dùng thông qua trình duyệt. Người dùng có thể tạo các node mạng từ một thư viện các template sẵn có, kết nối chúng lại với nhau và cấu hình chúng. Người dùng chuyên nghiệp hoặc quản trị viên có thể thêm các image phần mềm vào thư viện và tạo các mẫu các thiết bị mạng tùy chỉnh để hỗ trợ hầu hết các mô hình lab.

EVE-NG hỗ trợ cấu hình nhiều hypervisors trên một máy ảo. Nó chạy phần mềm thiết bị mạng thương mại trên Dynamips và IOU và chạy các thiết bị mạng khác, chẳng hạn như bộ định tuyến router mã nguồn mở, trên QEMU.

EVE-NG là một dự án nguồn mở và mã nguồn EVE-NG được đăng trên GitLab.

Ưu điểm của EVE-NG:

- Hỗ trợ đa nền tảng: Chạy trên các hệ điều hành như Windows, macOS, Linux.
- Hỗ trợ cài đặt trên các nền tảng ảo hóa như VMware, Hyper-V, và Proxmox.
- Hỗ trợ các thiết bị ảo từ nhiều nhà cung cấp khác nhau như Cisco, Juniper, Palo Alto, Fortinet, Huawei, v.v.
- Hỗ trợ cả các thiết bị ảo hóa mã nguồn mở như VyOS, pfSense, OpenWRT.
- Giao diện đồ họa thân thiện, cung cấp giao diện web trực quan, dễ sử dụng.

- Cho phép kéo thả để thiết kế và cấu hình topology mạng.

Nhược điểm của EVE-NG:

- Yêu cầu tài nguyên cao: Cần nhiều tài nguyên phần cứng (CPU, RAM, ổ cứng) để chạy các mô phỏng lớn.
- Cài đặt phức tạp: Việc cài đặt và cấu hình ban đầu có thể phức tạp đối với người dùng mới.
- Giới hạn tính năng miễn phí: Một số tính năng nâng cao yêu cầu phiên bản thương mại. Một số image cần phải trả phí để sử dụng.

2.1.2. Ansible



Hình 2-2: Ansible

Ansible là một công cụ tự động hóa mã nguồn mở do Red Hat phát triển, giúp quản lý cấu hình, triển khai ứng dụng và tự động hóa các tác vụ mạng. Ansible sử dụng file YAML để viết các playbook, cho phép mô tả các tác vụ một cách dễ hiểu và dễ bảo trì.

Ưu điểm của Ansible:

- Không cần agent: Không cần cài đặt phần mềm bổ sung trên các máy đích, chỉ cần SSH và Python.
- Dễ sử dụng: Sử dụng ngôn ngữ mô tả đơn giản, dễ tiếp cận ngay cả với người mới bắt đầu.
- Idempotency: Đảm bảo các tác vụ khi thực hiện nhiều lần vẫn cho kết quả như nhau.

- Quản lý tập trung: Quản lý và triển khai trên hàng ngàn máy chủ từ một nơi duy nhất.

Nhược điểm của Ansible:

- Hiệu suất: Khi quản lý nhiều máy cùng một lúc, hiệu suất của Ansible có thể bị ảnh hưởng do sử dụng kết nối SSH tuần tự. Dễ xảy ra timeout khi xử lý các task lớn.
- Hỗ trợ Windows hạn chế: Mặc dù có thể quản lý hệ thống Windows, nhưng Ansible hoạt động tốt nhất với các hệ thống Unix/Linux.
- Thiếu GUI: Ansible chủ yếu sử dụng dòng lệnh, thiếu một giao diện người dùng đồ họa tích hợp sẵn

2.1.3. Netmiko Python



Hình 2-3: Netmiko

Netmiko là một thư viện Python mã nguồn mở được phát triển bởi Kirk Byers. Thư viện này cung cấp một giao diện đơn giản và nhất quán để tự động hóa các tác vụ quản lý và cấu hình thiết bị mạng.

Ưu điểm của Netmiko:

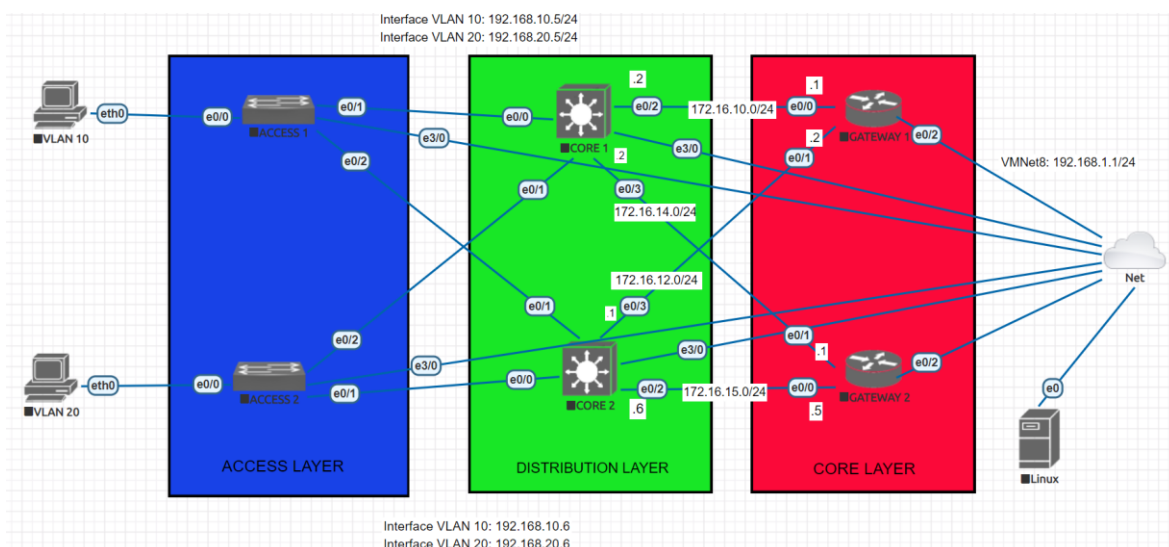
- Hỗ trợ một loạt các thiết bị từ nhiều nhà cung cấp như Cisco, Juniper, Arista, HP, Huawei, và nhiều hơn nữa.
- Giao diện đơn giản và dễ sử dụng

- Cung cấp các phương thức đơn giản để kết nối, gửi lệnh và nhận kết quả từ các thiết bị mạng.
- Sử dụng các phương thức Python quen thuộc, giúp dễ dàng tích hợp vào các script và ứng dụng hiện có.
- Netmiko tự động hóa việc kết nối SSH đến các thiết bị mạng, quản lý các phiên kết nối và xử lý các lỗi kết nối.

Nhược điểm của Netmiko:

- Bảo mật: Việc xử lý thông tin xác thực trong mã nguồn có thể gây ra các vấn đề bảo mật nếu không được quản lý đúng cách.
- Hiệu suất: Khi xử lý nhiều thiết bị cùng một lúc, hiệu suất của Netmiko có thể bị giới hạn do phụ thuộc vào kết nối SSH.
- Thiếu tính năng nâng cao: Netmiko chủ yếu tập trung vào việc gửi lệnh và thu thập đầu ra, thiếu các tính năng nâng cao như quản lý cấu hình phức tạp hoặc tự động hóa cấp cao.

2.2. Mô hình mạng đề xuất



Hình 2-4: Mô hình mạng

Mô hình mạng được thiết kế theo kiến trúc 3 lớp của Cisco bao gồm lớp kết nối (Access Layer), lớp phân phối (Distribution Layer) và lớp lõi (Core Layer) với các thiết bị mô phỏng như sau:

- **2 Router Gateway ở Core Layer:** chịu trách nhiệm vận chuyển dữ liệu tốc độ cao giữa các phần khác nhau của mạng. Đây là xương sống của mạng, cung cấp khả năng ra Internet của mô hình mạng. Trong mô hình, 2 router này sẽ chạy song song và khi 1 router bị sự cố, router còn lại sẽ làm backup cho router đã bị down.
- **2 Core Switch Layer 3 ở Distribution Layer:** kết nối các thiết bị giữa lớp Access và lớp Core. Thực hiện các chức năng định tuyến, phân phối IP đến các host trong LAN ảo (VLAN). Ở mô hình trên, Core 1 chịu trách nhiệm phân phối IP tự động (DHCP Pool) đến 2 VLAN 10 và 20 đồng thời cũng là thiết bị hoạt động chính để traffic có thể đến Core Layer. Core 2 là backup cho Core 1 khi Core 1 gặp sự cố.
- **2 Access Switch Layer 2 ở Access Layer:** cung cấp kết nối mạng cho các thiết bị như máy tính, điện thoại IP, và các thiết bị đầu cuối khác. Access 1 kết nối với VLAN 10 và Access 2 kết nối với VLAN 20.
- **1 máy ảo Linux:** triển khai Ansible trên máy ảo này để thực hiện các tác vụ cấu hình (Tasks) cho toàn bộ mô hình mạng.
- **Internet:** đóng vai trò là môi trường mạng public, mục đích để cho các thiết bị trong mạng LAN có thể giao tiếp được với bên ngoài

2.3. Cơ sở lý thuyết

2.3.1. Các giao thức mạng sử dụng

2.3.1.1. Network Address Translation (NAT)

NAT là một phương thức dùng để ánh xạ địa chỉ IP từ một không gian địa chỉ này sang một không gian địa chỉ khác. NAT thường được sử dụng để cho phép nhiều thiết bị trong một mạng cục bộ (LAN) sử dụng một địa chỉ IP công cộng duy nhất

để truy cập Internet. Điều này giúp tiết kiệm địa chỉ IP và bảo vệ mạng nội bộ khỏi các cuộc tấn công từ bên ngoài.

NAT có các chức năng sau:

- **Chuyển đổi địa chỉ IP:** Chuyển đổi địa chỉ IP riêng trong mạng nội bộ thành địa chỉ IP công cộng khi truy cập Internet.
- **Tiết kiệm địa chỉ IP:** Giảm nhu cầu sử dụng nhiều địa chỉ IP công cộng.
- **Bảo mật:** Che giấu cấu trúc và địa chỉ IP của mạng nội bộ khỏi các cuộc tấn công từ bên ngoài.

Nhược điểm:

- **Vấn đề về kết nối P2P:** NAT có thể gây khó khăn cho các kết nối peer-to-peer (P2P) vì nó ẩn địa chỉ IP thật của thiết bị.
- **Hiệu suất:** NAT có thể làm chậm tốc độ kết nối mạng do phải xử lý và chuyển đổi các gói tin.
- **Khả năng tương thích:** Một số ứng dụng và giao thức không hoạt động tốt qua NAT, đặc biệt là các ứng dụng yêu cầu kết nối liên tục hoặc giao thức sử dụng địa chỉ IP trong phần tải hữu ích (payload).

2.3.1.2. Open Shortest Path First (OSPF)

OSPF là một giao thức định tuyến liên miền theo trạng thái liên kết được sử dụng rộng rãi trong các mạng lớn và phức tạp. OSPF hoạt động bằng cách xây dựng một bản đồ topo của mạng và tính toán đường đi ngắn nhất đến từng mạng con bằng thuật toán Dijkstra. OSPF hỗ trợ cân bằng tải và cung cấp khả năng hội tụ nhanh khi có thay đổi trong cấu trúc mạng.

OSPF có các chức năng sau:

- **Định tuyến theo trạng thái liên kết:** Xây dựng bảng định tuyến dựa trên thông tin trạng thái liên kết trong mạng.
- **Cân bằng tải:** Phân phối lưu lượng mạng đồng đều trên nhiều đường dẫn.

- **Hội tụ nhanh:** Nhanh chóng cập nhật thông tin định tuyến khi có thay đổi trong mạng.

Nhược điểm:

- **Phức tạp trong cấu hình:** Cấu hình OSPF có thể phức tạp, đặc biệt là trong các mạng lớn với nhiều area.
- **Tài nguyên hệ thống:** OSPF yêu cầu nhiều tài nguyên hệ thống (CPU, RAM) để xử lý thông tin trạng thái liên kết và tính toán đường đi.
- **Khả năng mở rộng:** Mặc dù OSPF được thiết kế để mở rộng, việc quản lý nhiều area và các vùng không kết nối trực tiếp có thể gặp khó khăn.

2.3.1.3. Virtual Local Area Network (VLAN)

VLAN là một kỹ thuật chia mạng LAN vật lý thành nhiều mạng ảo độc lập. Mỗi VLAN hoạt động như một mạng LAN riêng biệt, giúp cải thiện hiệu suất mạng, tăng cường bảo mật và quản lý dễ dàng hơn. VLAN cho phép các thiết bị trên cùng một VLAN giao tiếp trực tiếp với nhau mà không cần đi qua router.

VLAN có các chức năng sau:

- **Phân chia mạng ảo:** Chia một mạng vật lý thành nhiều mạng ảo độc lập.
- **Tăng cường bảo mật:** Giới hạn truy cập giữa các VLAN, ngăn chặn các truy cập trái phép.
- **Cải thiện hiệu suất:** Giảm thiểu lưu lượng broadcast và cô lập các miền xung đột.

Nhược điểm:

- **Cấu hình phức tạp:** Việc cấu hình và quản lý VLAN đòi hỏi kiến thức kỹ thuật cao và có thể phức tạp.

- **Khả năng mở rộng hạn chế:** Số lượng VLAN trên một switch bị giới hạn bởi số lượng ID VLAN (4096 VLAN ID).
- **Quản lý bảo mật:** Nếu không được cấu hình đúng cách, VLAN có thể dễ bị tấn công từ các thiết bị nội bộ.

2.3.1.4. VLAN Trunking Protocol (VTP)

VTP là một giao thức quản lý VLAN được sử dụng để truyền thông tin cấu hình VLAN trong toàn bộ mạng chuyển mạch (switching network). VTP cho phép các switch tự động cập nhật cấu hình VLAN từ một switch trung tâm (server) đến các switch khác (client), giúp đơn giản hóa việc quản lý và triển khai VLAN trong một hệ thống lớn.

VTP có các chức năng sau:

- **Quản lý VLAN tập trung:** Cập nhật và phân phối thông tin VLAN cho tất cả các switch trong mạng.
- **Đồng bộ hóa VLAN:** Đảm bảo cấu hình VLAN nhất quán trên toàn bộ mạng.
- **Giảm thiểu sai sót:** Tự động hóa việc tạo và xóa VLAN, giảm thiểu lỗi cấu hình thủ công.

Nhược điểm:

- **Nguy cơ sai sót cấu hình:** Một lỗi cấu hình VTP trên một switch có thể lan rộng và ảnh hưởng đến toàn bộ mạng.
- **Bảo mật:** VTP có thể bị khai thác để thực hiện các cuộc tấn công, như gửi các thông điệp VTP giả mạo để thay đổi cấu hình VLAN.
- **Giới hạn trong môi trường hỗn hợp:** VTP hoạt động tốt nhất trong các mạng hoàn toàn sử dụng thiết bị của Cisco và có thể gặp khó khăn khi tích hợp với thiết bị của nhà cung cấp khác.

2.3.1.5. Rapid Spanning Tree Protocol (RSTP)

RSTP là một phiên bản cải tiến của giao thức Spanning Tree Protocol (STP), được thiết kế để cung cấp sự hội tụ nhanh hơn trong trường hợp có sự thay đổi trong topo mạng. RSTP giúp ngăn chặn các vòng lặp trong mạng chuyển mạch bằng cách tạo ra một cây tối thiểu kết nối tất cả các switch trong mạng. RSTP có khả năng phát hiện và khôi phục nhanh chóng khi có sự cố xảy ra, giảm thiểu thời gian downtime.

RSTP có các chức năng sau:

- **Phát hiện vòng lặp mạng:** Ngăn chặn và loại bỏ các vòng lặp trong mạng chuyển mạch.
- **Hội tụ nhanh:** Nhanh chóng chuyển đổi trạng thái các port khi có thay đổi trong topo mạng.
- **Cải thiện độ tin cậy:** Tăng cường khả năng phục hồi của mạng khi có sự cố xảy ra.

Nhược điểm:

- **Cấu hình và quản lý phức tạp:** Cấu hình và quản lý RSTP có thể phức tạp đối với những người mới làm quen.
- **Không hỗ trợ tất cả các tình huống:** RSTP không thể giải quyết tất cả các vấn đề về vòng lặp mạng, đặc biệt là trong các mạng lớn và phức tạp.
- **Khả năng tương thích:** Mặc dù RSTP tương thích ngược với STP, việc triển khai trong các mạng hỗn hợp có thể gặp khó khăn.

2.3.1.6. Hot Standby Router Protocol (HSRP)

HSRP là một giao thức của Cisco được thiết kế để cung cấp khả năng chuyển đổi dự phòng (redundancy) cho các router trong mạng. HSRP cho phép một nhóm các router hoạt động cùng nhau để tạo ra một router ảo duy nhất. Nếu router chính (active) bị lỗi, một router dự phòng (standby) sẽ nhanh chóng tiếp quản vai trò của router chính, đảm bảo mạng vẫn hoạt động liên tục và không bị gián đoạn.

HSRP có các chức năng sau:

- **Chuyển đổi dự phòng:** Đảm bảo tính sẵn sàng cao bằng cách tự động chuyển đổi sang router dự phòng khi router chính bị lỗi.
- **Địa chỉ IP ảo:** Cung cấp một địa chỉ IP ảo để các thiết bị trong mạng sử dụng làm gateway mặc định.
- **Cân bằng tải:** Phân phối lưu lượng mạng giữa các router chính và dự phòng để tối ưu hóa hiệu suất.

Nhược điểm:

- **Tiêu tốn tài nguyên:** HSRP yêu cầu tài nguyên bổ sung cho việc duy trì trạng thái của các router dự phòng.
- **Khả năng mở rộng hạn chế:** HSRP có thể gặp khó khăn khi triển khai trong các mạng lớn với nhiều nhóm HSRP.
- **Độc quyền của Cisco:** HSRP là giao thức độc quyền của Cisco, điều này có thể gây khó khăn khi tích hợp với thiết bị từ các nhà cung cấp khác.

Chương 3. TRIỂN KHAI




3.1. Cài đặt EVE-NG và các image mô phỏng.

- ❖ Chuẩn bị: VMWare Workstation Pro, WinSCP

Quá trình cài đặt EVE-NG trên nền tảng ảo hoá VMWare Workstation Pro và các image mô phỏng:

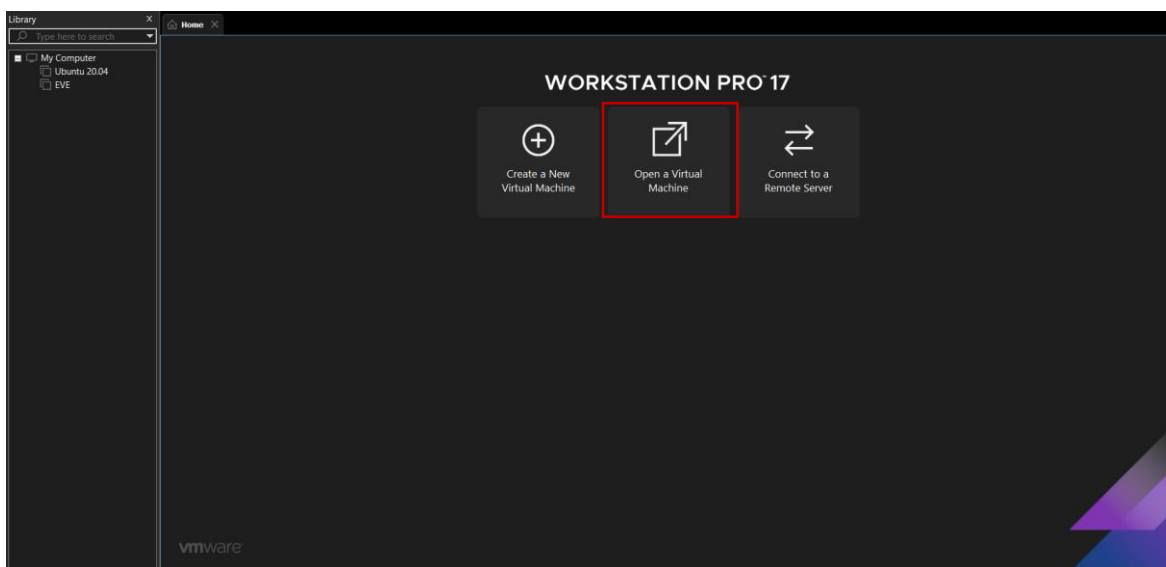
- Bước 1: Tải file iso của EVE-NG trên trang chủ của EVE-NG

Link: [EVE-NG Iso Download](#)

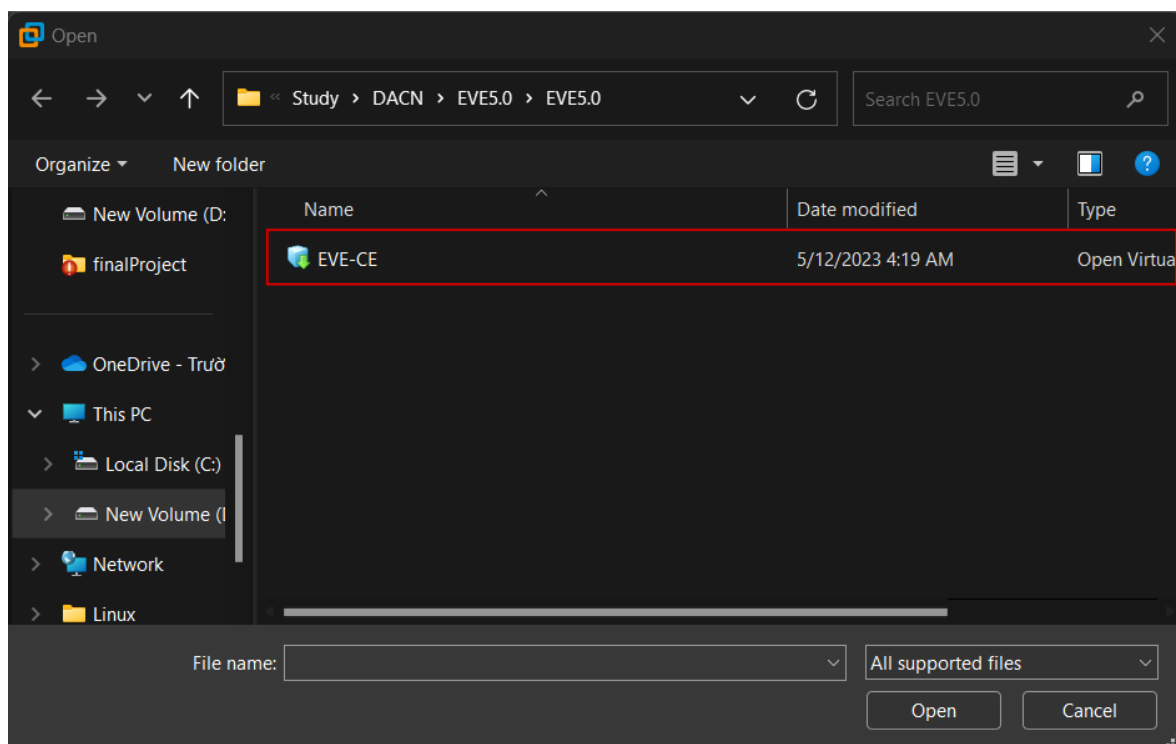
 EVE-CE.mf	5/12/2023 4:19 AM	MF File	1 KB
 EVE-CE	5/12/2023 4:19 AM	Open Virtualization Format Package	8 KB
 EVE-CE-disk1	5/12/2023 4:19 AM	Virtual Machine Disk Format	3,028,150 ...

Hình 3-1: Sau khi download file iso EVE-NG

- Bước 2: Cài đặt EVE-NG trên VMWare

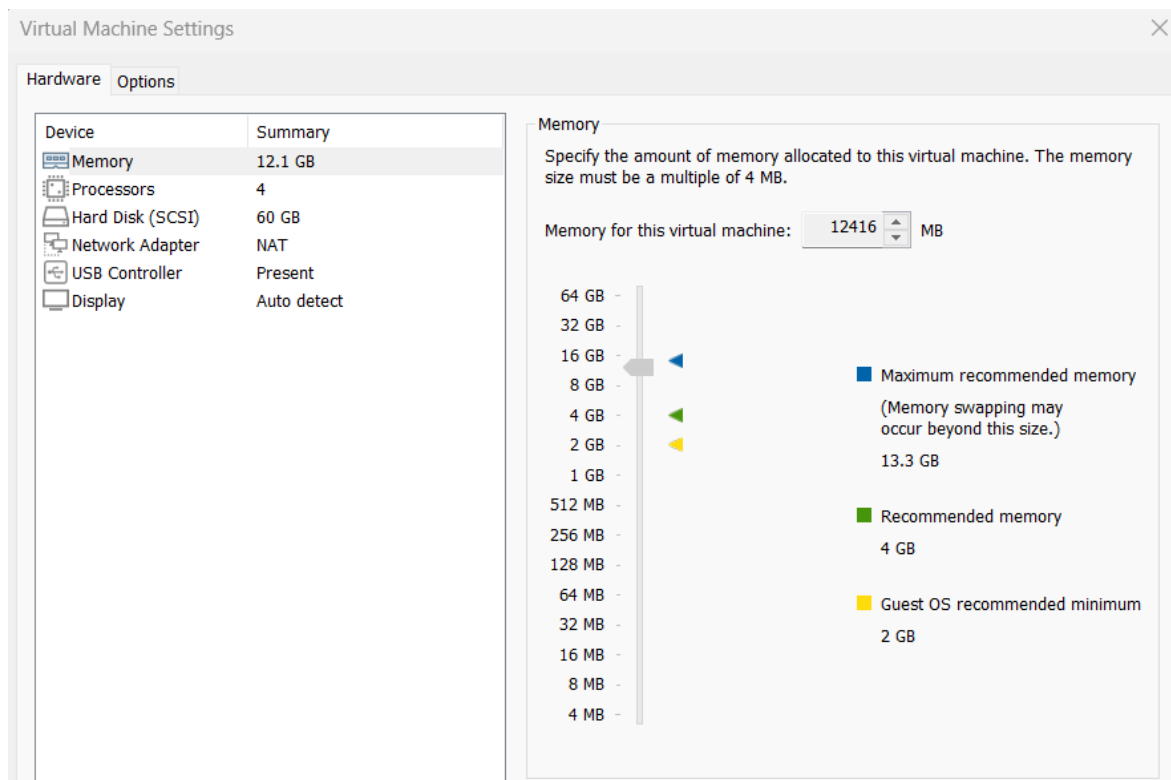


Hình 3-2: Giao diện của VMWare Workstation Pro



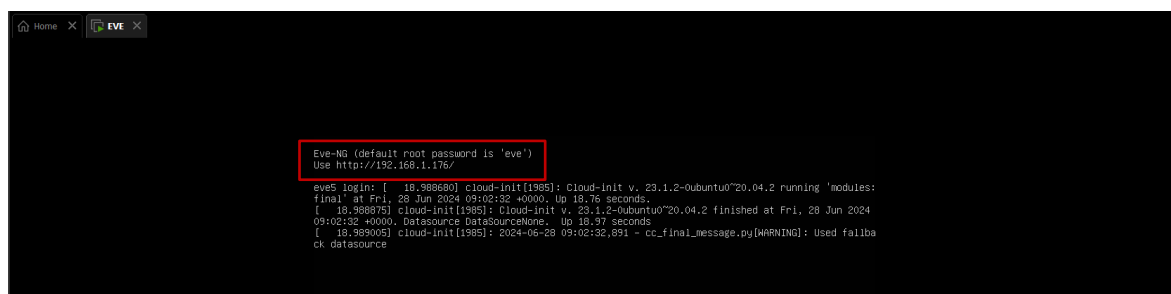
Hình 3-3: File EVE-CE.ovf

Tạo Virtual Machine mới bằng cách nhấn vào Open a Virtual Machine và chọn đến file EVE-CE.ovf. Sau đó nhập tên và đường dẫn lưu trữ của máy ảo.



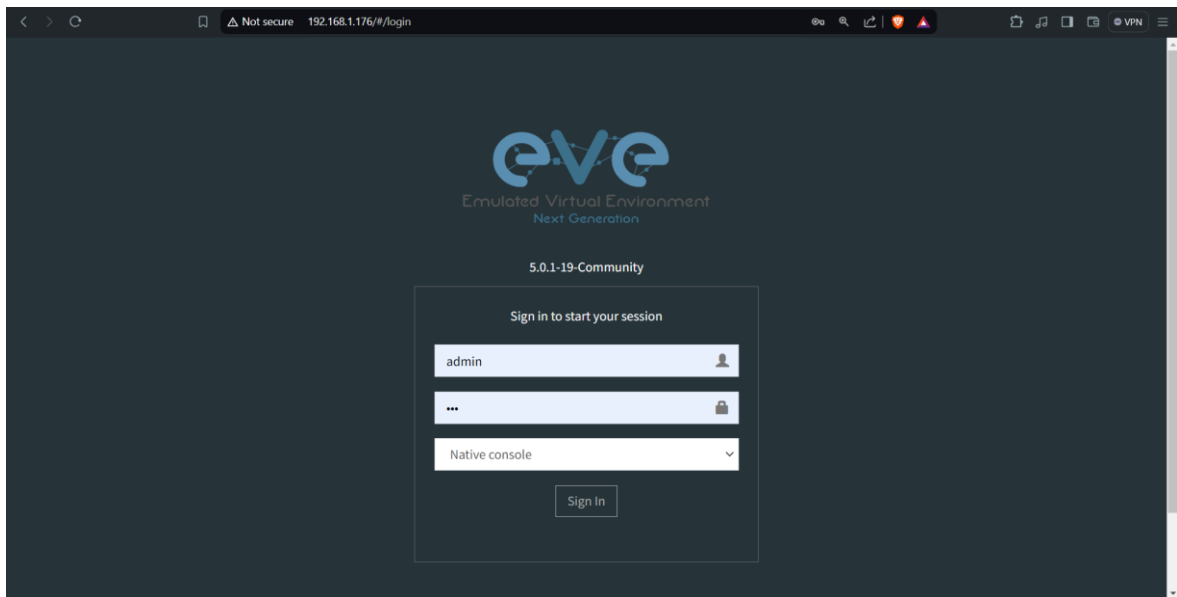
Hình 3-4: Mục Settings của máy ảo

Cấu hình đề nghị cho EVE-NG theo kinh nghiệm: Memory 12GB, Processors (CPU) 4, Hard Disk trên 40GB, Network Adapter là NAT và tick vào ô Virtualize Inter VT -x/EPT or AMD-V/RVI.



Hình 3-5: EVE-NG sau khi cài đặt

Truy cập vào Web UI của EVE-NG bằng cách truy cập đến URL được hiển thị trên màn hình là <http://192.168.1.176/>



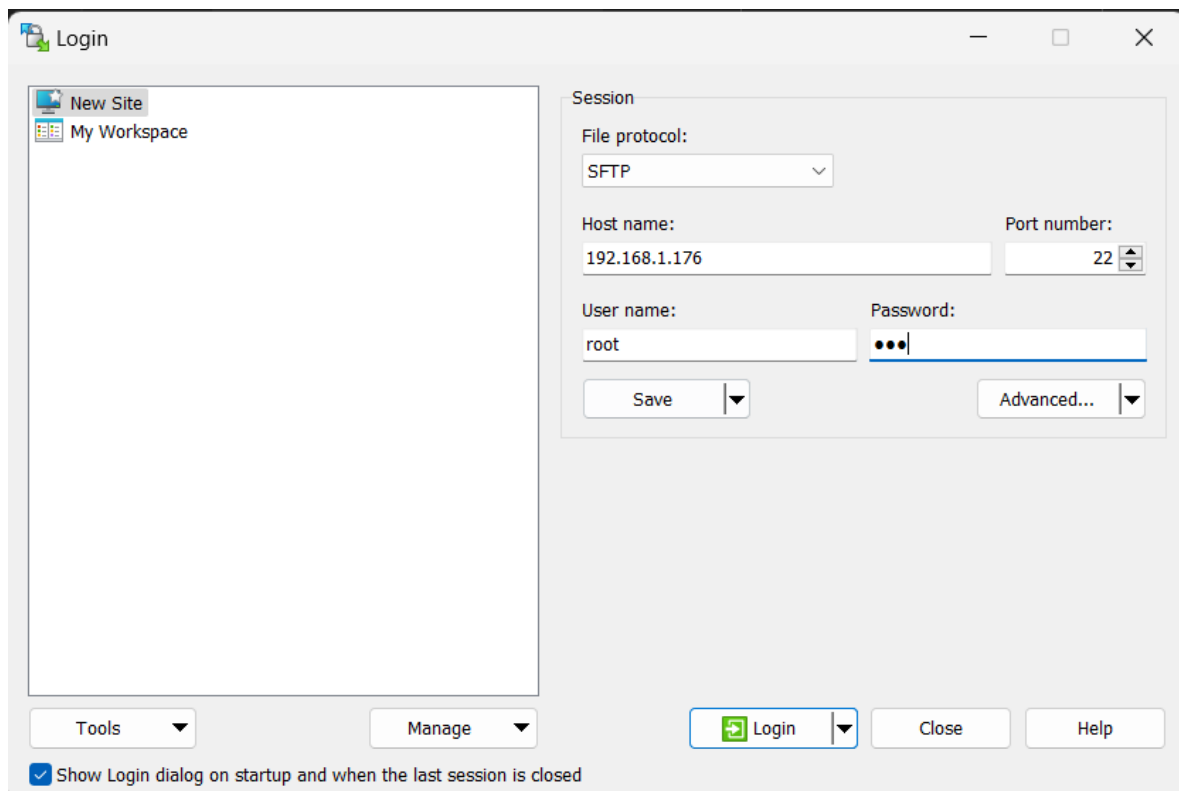
Hình 3-6: Màn hình đăng nhập của EVE-NG

Đăng nhập với username là **admin** và password là **eve**

- Bước 3: Thêm các image mô phỏng vào EVE-NG.

Tải các image để mô phỏng chức năng của các Router, Linux Server, Switch Layer 2, và Switch Layer 3. Lưu ý: các image mô phỏng chủ yếu được tải miễn phí từ cộng đồng của EVE-NG chia sẻ lại nên có thể nguồn tải sẽ không chính thống từ EVE-NG nhưng vẫn đảm bảo được các tính năng cần thiết.

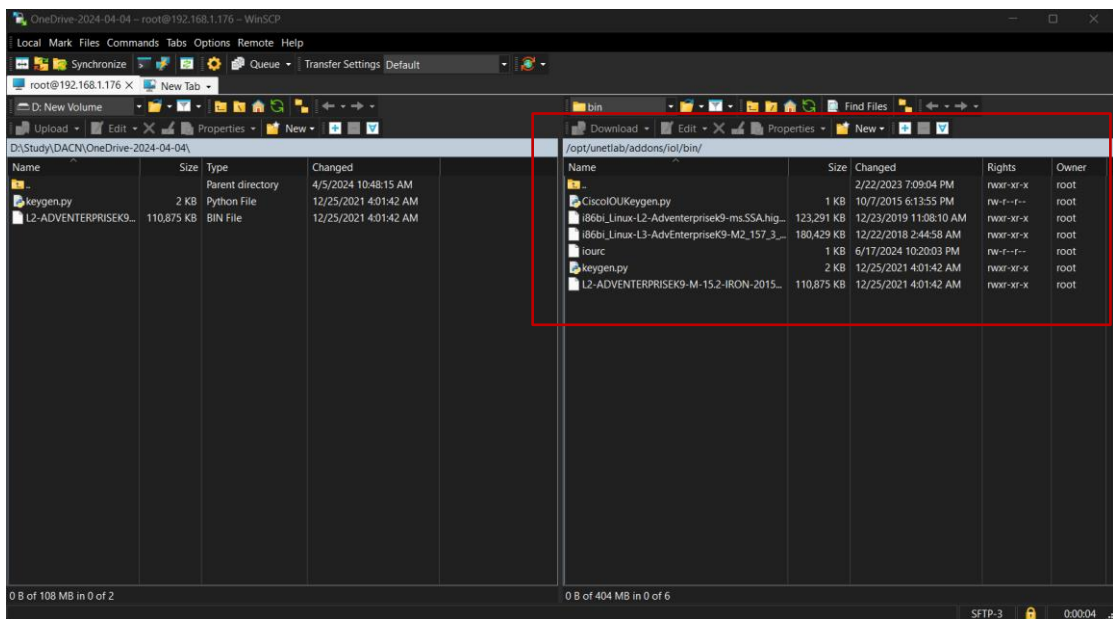
Link download cho các image tại đây [Image](#)



Hình 3-7: Đăng nhập vào WinSCP

Đăng nhập vào WinSCP với Hostname là 192.168.1.176, Port là 22, Username root và Password là eve

Tham khảo cách thêm các image từ tài liệu của EVE-NG ở đường dẫn [Add Image Router Switch](#) và [Add Linux Server](#). Sau khi thêm thành công sẽ được như hình 3-8



Hình 3-8: Giao diện của WinSCP sau khi thêm image

Sau khi thêm xong các image, truy cập vào Web UI của EVE-NG tiến hành thiết kế mô hình mạng như đã đề xuất.

ADD A NEW NODE



Template

Cisco IOS 7206VXR (Dynamips) ▼

Number of nodes to add

1


Image

c7200-adventerprisek9-mz.152-4.S6.image ▼

Name/prefix

7206VXR

Icon

 Router.png ▼

Idle PC

0x62f21000

NVRAM (KB)

128

RAM (MB)

256

Slot 1

Empty ▼

Slot 2

Empty ▼

Slot 3

Empty ▼

Slot 4

Empty ▼

Slot 5

Empty ▼

Slot 6

Empty ▼

Startup configuration

None ▼

Hình 3-9: Thêm Router vào mô hình mạng

ADD A NEW NODE

Template

Cisco IOL

Number of nodes to add

1

Image

i86bi_Linux-L2-Adventerprisek9-ms.SSA.hig

Name/prefix

R

Icon

Switch2.png

NVRAM (KB)

1024

RAM (MB)

1024

Ethernet portgroups (4 int each)

1

Serial portgroups (4 int each)

0

Startup configuration

None

Delay (s)

0

Left

684

Top

531

Save

Cancel

Hình 3-10: Thêm Switch Layer 2 và Switch Layer 3 (2 Switch có thể dùng 1 image)

ADD A NEW NODE

Template

Virtual PC (VPCS)


Number of nodes to add

1

Name/prefix

VPC

Icon

 Desktop.png

Startup configuration

None

Delay (s)

0

Left

684

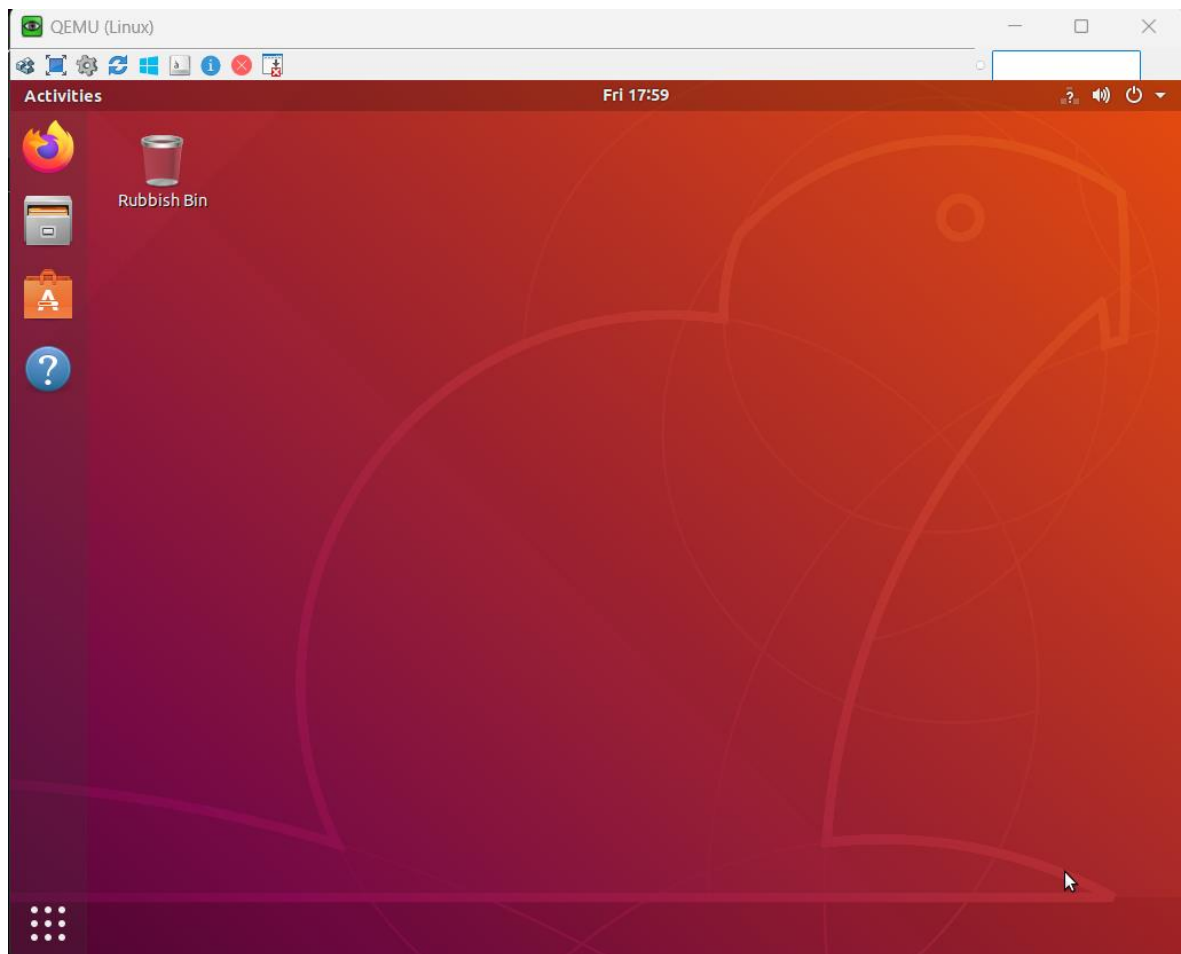
Top

531

Save

Cancel

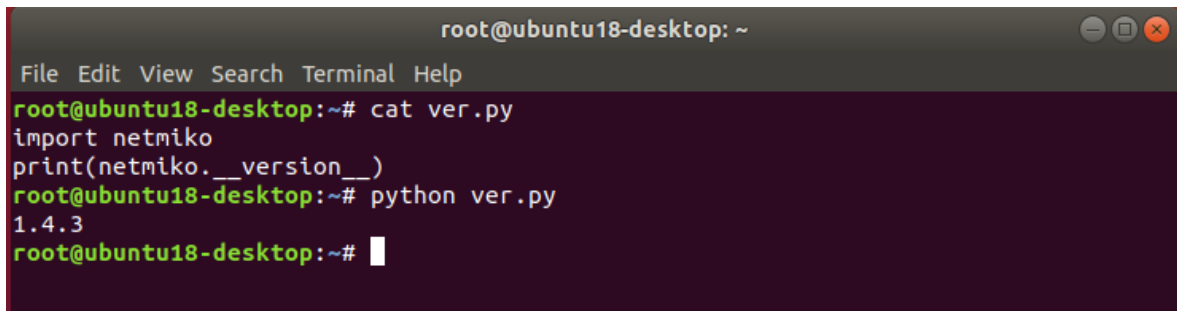
Hình 3-11: Thêm VPCs



Hình 3-12: Giao diện của Linux Server (Ubuntu 18.04.06-desktop)

3.2. Cài đặt thư viện Netmiko Python trên Linux Server

- Bước 1: Cập nhật hệ thống bằng câu lệnh:
`sudo apt update`
`sudo apt upgrade -y`
- Bước 2: Cài đặt Python và Pip
`sudo apt install python3 python3-pip -y`
- Bước 3: Cài đặt Netmiko
`pip3 install netmiko`
- Bước 4: Kiểm tra cài đặt bằng cách tạo 1 file python và import Netmiko vào, nếu chạy ra được như hình 3-13 tức là cài đặt thành công

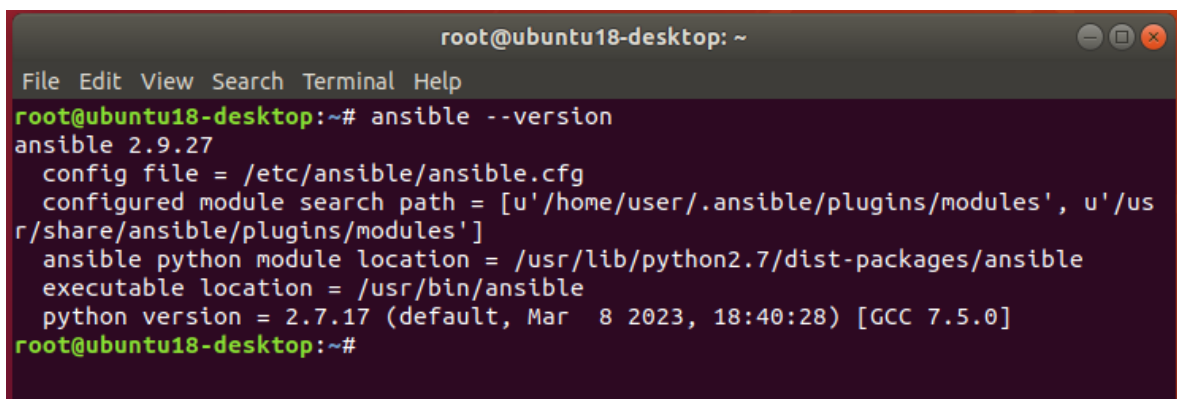
A terminal window titled 'root@ubuntu18-desktop: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
root@ubuntu18-desktop:~# cat ver.py
import netmiko
print(netmiko.__version__)
root@ubuntu18-desktop:~# python ver.py
1.4.3
root@ubuntu18-desktop:~#
```

Hình 3-13: Kiểm tra phiên bản Netmiko

3.3. Cài đặt Ansible

- Bước 1: Cập nhật hệ thống bằng câu lệnh:
sudo apt update
sudo apt upgrade -y
- Bước 2: Cài đặt Ansible bằng câu lệnh:
sudo apt install ansible -y
- Bước 3: Kiểm tra phiên bản của Ansible

A terminal window titled 'root@ubuntu18-desktop: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following command and output:

```
root@ubuntu18-desktop:~# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/user/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Mar  8 2023, 18:40:28) [GCC 7.5.0]
root@ubuntu18-desktop:~#
```

Hình 3-14: Kiểm tra phiên bản của Ansible

3.4. Viết playbook cho Ansible

Viết các file yml trong Linux Server để chạy các file Script Python

```
user@ubuntu18-desktop: /etc/ansible/playbook/script
File Edit View Search Terminal Help
user@ubuntu18-desktop:~$ cd /etc/ansible/
user@ubuntu18-desktop:/etc/ansible$ ls
ansible.cfg  hosts  playbook  roles
user@ubuntu18-desktop:/etc/ansible$ cd playbook/
user@ubuntu18-desktop:/etc/ansible/playbook$ ls
Access_config.yml  Core_config.yml  Gateway_config.yml  script  test.yml
user@ubuntu18-desktop:/etc/ansible/playbook$ cd script/
user@ubuntu18-desktop:/etc/ansible/playbook/script$ ls
Access_configuration.py  Gateway_configuration.py
Core_configuration.py    test.py
user@ubuntu18-desktop:/etc/ansible/playbook/script$
```

Hình 3-15: Các thư mục trong folder ansible

```
root@ubuntu18-desktop: /etc/ansible
File Edit View Search Terminal Help
GNU nano 2.9.3 hosts
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

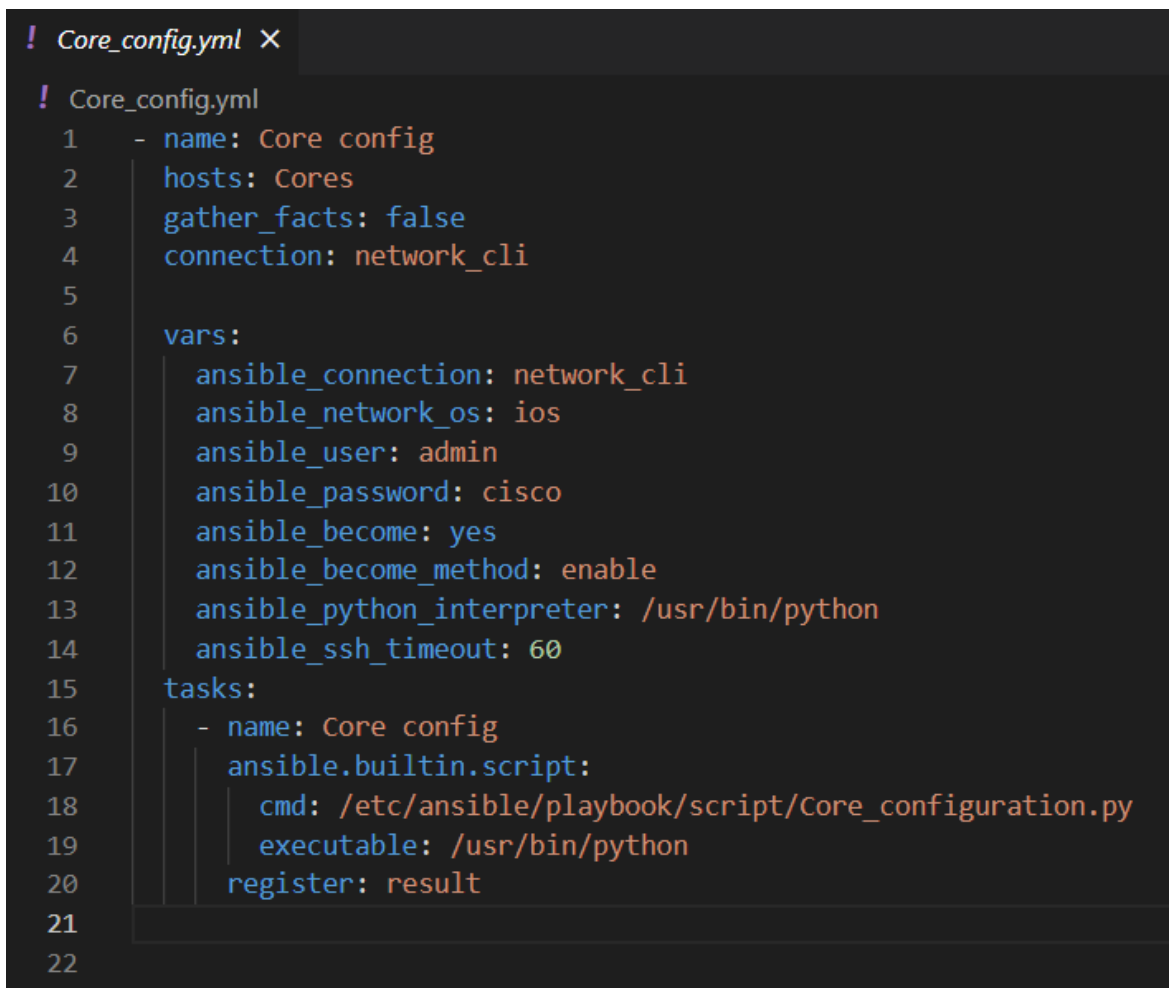
# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com

[Router]
R1 ansible_host=192.168.1.177 ansible_user=admin ansible_password=cisco
R2 ansible_host=192.168.1.178 ansible_user=admin ansible_password=cisco
[Cores]
C1 ansible_host=192.168.1.179 ansible_user=admin ansible_password=cisco
C2 ansible_host=192.168.1.180 ansible_user=admin ansible_password=cisco
[Accesses]
A1 ansible_host=192.168.1.183 ansible_user=admin ansible_password=cisco
A2 ansible_host=192.168.1.182 ansible_user=admin ansible_password=cisco
[all:vars]
ansible_connection=network_cli
ansible_network_os=ios
ansible_user=admin
ansible_password=cisco
```

Hình 3-16: File hosts

Để có thể dùng ansible để thực hiện các tác vụ lên thiết bị, phải định nghĩa các thông số cần thiết của thiết bị bao gồm địa chỉ IP, user, password. Các thiết bị có thể được gom nhóm lại như trong file hosts như là Routers, Cores, Accesses



```
! Core_config.yml X
! Core_config.yml
1  - name: Core config
2    hosts: Cores
3    gather_facts: false
4    connection: network_cli
5
6    vars:
7      ansible_connection: network_cli
8      ansible_network_os: ios
9      ansible_user: admin
10     ansible_password: cisco
11     ansible_become: yes
12     ansible_become_method: enable
13     ansible_python_interpreter: /usr/bin/python
14     ansible_ssh_timeout: 60
15   tasks:
16     - name: Core config
17       ansible.builtin.script:
18         cmd: /etc/ansible/playbook/script/Core_configuration.py
19         executable: /usr/bin/python
20         register: result
21
22
```

Hình 3-17: File Core_config.yml

Các thành phần của file yml:

- 'name': tên của playbook là Core config
- 'hosts': định nghĩa các đối tượng mà playbook được áp dụng các tác vụ, ở đây là nhóm "Cores" đã định nghĩa trong file hosts

- 'gather_facts': tắt việc thu thập các thông tin về hệ thống mặc định của Ansible. Do đây là thiết bị mạng nên không cần thiết.
- 'connection': chỉ định loại kết nối, ở đây là 'network_cli' để giao tiếp với thiết bị
- 'ansible_network_os': chỉ định hệ điều hành của thiết bị mạng, ở đây là ios
- 'ansible_user': username của thiết bị mạng là admin
- 'ansible_password': password của thiết bị mạng là cisco
- 'ansible_become': cho phép thực hiện các tác vụ nâng cao
- 'ansible_python_interpreter': đường dẫn đến trình thông dịch Python, ở đây có đường dẫn là '/usr/bin/python'
- 'ansible_ssh_timeout': thời gian chờ cho kết nối SSH là 60 giây
- 'ansible_builtin_script': module của ansible để chạy một file script
- 'cmd': đường dẫn đến file script, ở đây là '/etc/ansible/playbook/script/Core_configuration.py'
- 'executable': đường dẫn đến trình thông dịch Python
- 'register': Lưu kết quả của nhiệm vụ vào biến result

```

root@ubuntu18-desktop:/etc/ansible/playbook# ansible-playbook Core_config.yml

PLAY [Core config] *****

TASK [Core config] *****
changed: [C1]
changed: [C2]

PLAY RECAP *****
C1      : ok=1    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
C2      : ok=1    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

root@ubuntu18-desktop:/etc/ansible/playbook#

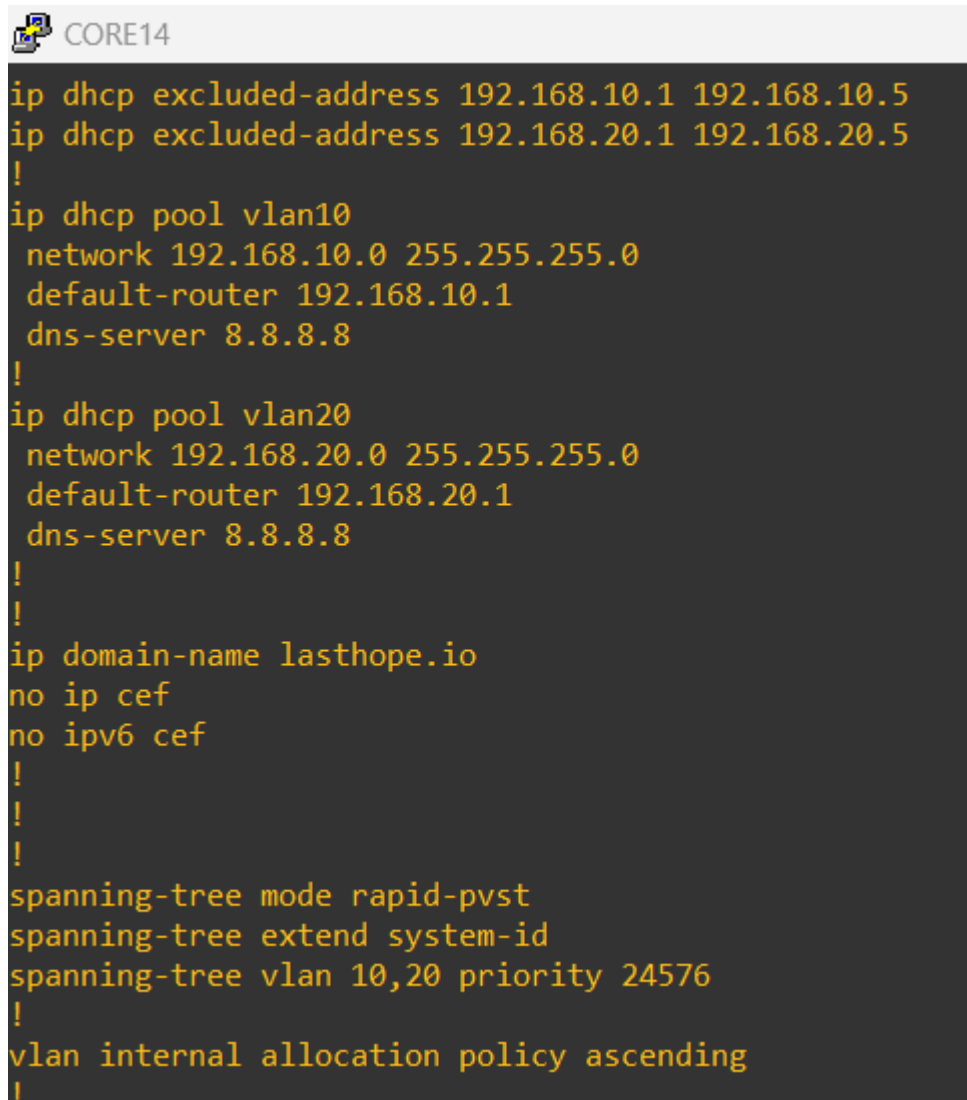
```

Hình 3-18: Sau khi chạy file yml

Chạy các file yml bằng lệnh 'ansible-playbook [tên file.yml]' dưới quyền root sẽ cho ra kết quả là các thiết bị đã có sự thay đổi trong cấu hình và các tasks được thực hiện thành công.

Tham khảo thêm các file yml khác tại đường dẫn này:

Link: <https://github.com/huannv2973/netmiko-configuration>



```
CORE14
ip dhcp excluded-address 192.168.10.1 192.168.10.5
ip dhcp excluded-address 192.168.20.1 192.168.20.5
!
ip dhcp pool vlan10
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 8.8.8.8
!
ip dhcp pool vlan20
 network 192.168.20.0 255.255.255.0
 default-router 192.168.20.1
 dns-server 8.8.8.8
!
!
ip domain-name lasthope.io
no ip cef
no ipv6 cef
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
spanning-tree vlan 10,20 priority 24576
!
vlan internal allocation policy ascending
!
```

Hình 3-19: Kiểm tra các cấu hình sau khi chạy Ansible

Chương 4. KIỂM THỬ MÔ HÌNH

4.1. Viết các kịch bản của mô hình mạng

Thiết bị	Interface	IP Address	Default Gateway
Router Gateway 1	E0/2	192.168.1.177/24	
	E0/1	172.16.12.2/24	
	E0/0	172.16.10.1/24	
Router Gateway 2	E0/2	192.168.1.178/24	
	E0/1	172.16.14.1/24	
	E0/0	172.16.15.5/24	
Switch Core 1	E3/0	192.168.1.189/24	
	VLAN 10	192.168.10.5/24	
	VLAN 20	192.168.20.5/24	
	E0/2	172.16.10.2/24	
	E0/3	172.16.14.2/24	
Switch Core 2	E3/0	192.168.1.190/24	
	VLAN 10	192.168.10.6/24	
	VLAN 20	192.168.20.6/24	
	E0/2	172.16.15.6/24	
	E0/3	172.16.12.1/24	
DHCP Pool cho các VLAN (Switch Core 1)	VLAN 10	192.168.10.0/24	
	VLAN 20	192.168.20.0/24	
Switch Access 1	E3/0	192.168.1.191/24	

Switch Access 2	E3/0	192.168.1.192/24	
VPC10		192.168.10.7/24	192.168.10.1/24
VPC20		192.168.20.7/24	192.168.20.1/24

Bảng 1: Bảng địa chỉ IP cho thiết bị

Để có thể dùng Netmiko để SSH và đẩy các lệnh cấu hình vào thiết bị thì cần phải bật chế độ SSH và cấp một địa chỉ IP động trên thiết bị.

```

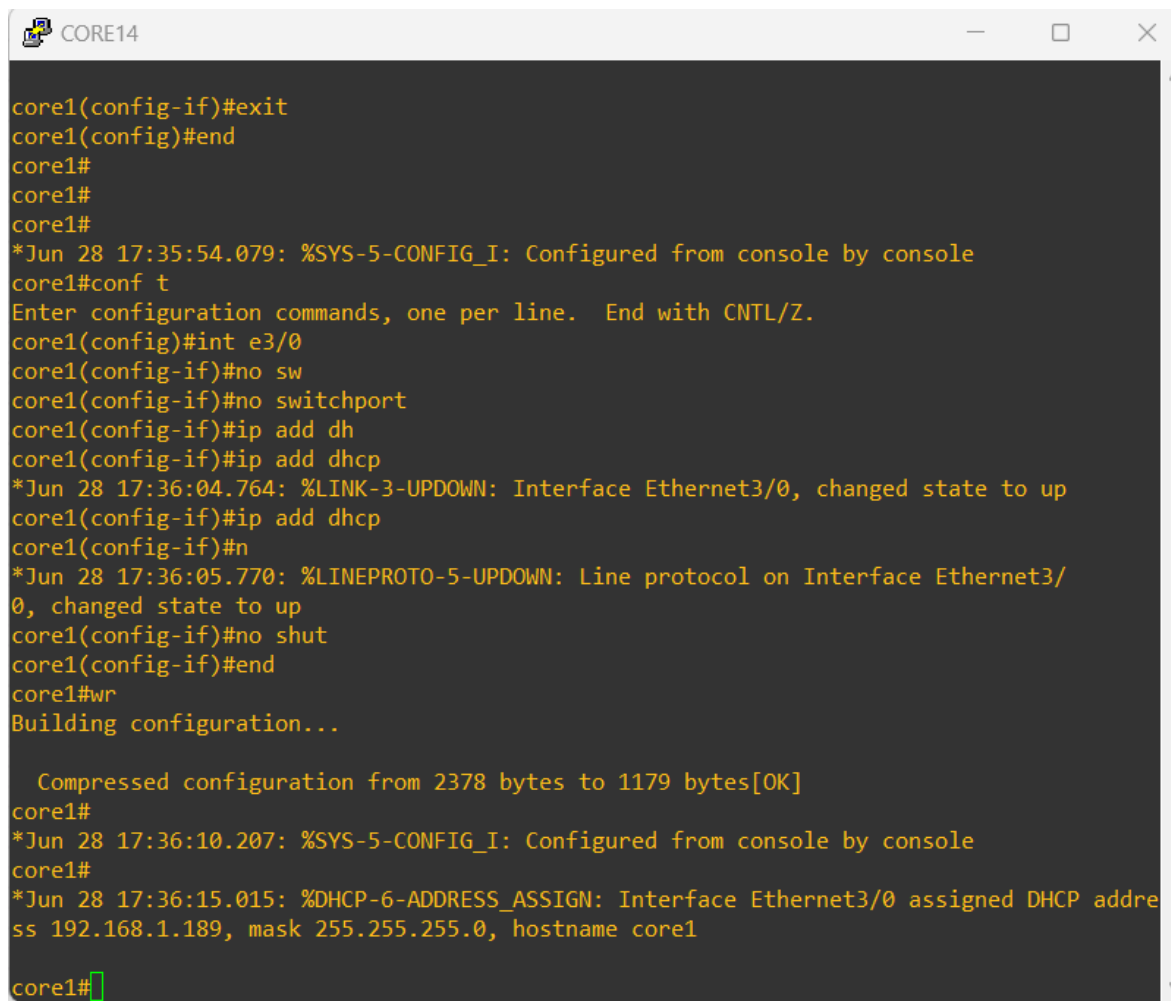
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname core1
core1(config)#ip domain-name lasthope.io
core1(config)#crypto key generate rsa
The name for the keys will be: core1.lasthope.io
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)

core1(config)#username admin secret cisco
core1(config)#username admin privilege 15
core1(config)#enable secret cisco
core1(config)#line vty 0 4
core1(config-line)#transport input ssh
core1(config-line)#login local
core1(config-line)#end
core1#wr
*Jun 28 17:34:09.864: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Jun 28 17:34:10.504: %SYS-5-CONFIG_I: Configured from console by console
core1#wr

```

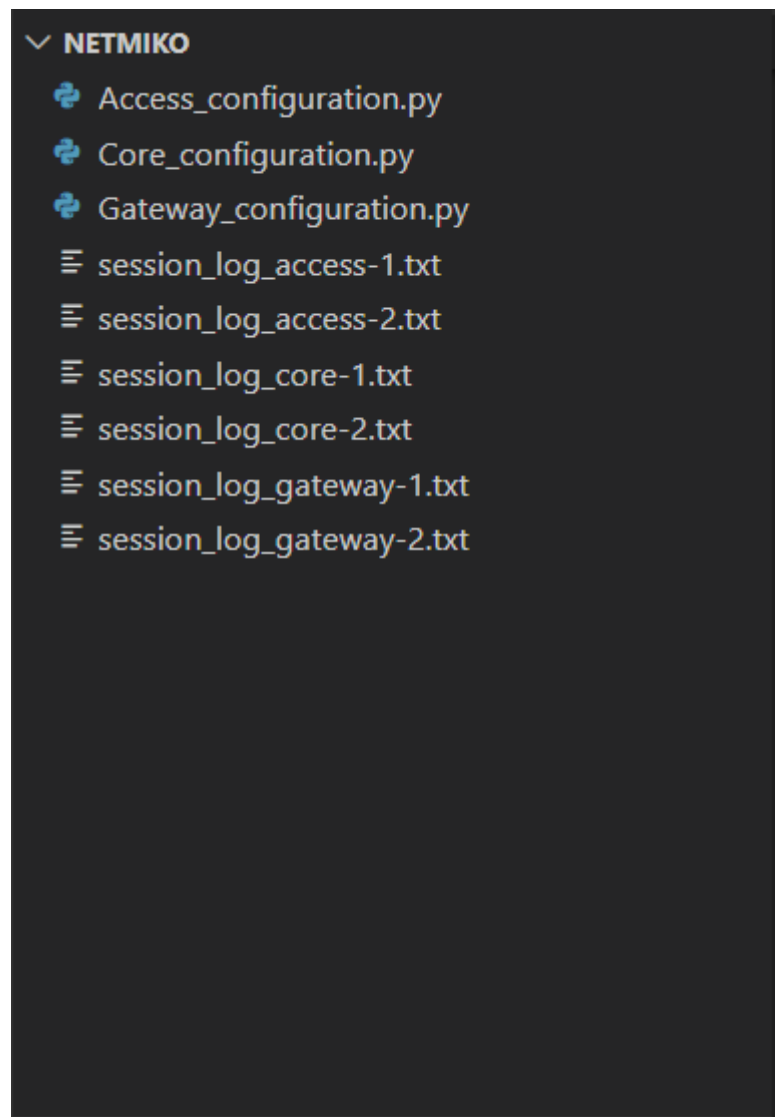
Hình 4-1: Bật chế độ SSH trên Switch Core 1



```
core1(config-if)#exit
core1(config)#end
core1#
core1#
core1#
*Jun 28 17:35:54.079: %SYS-5-CONFIG_I: Configured from console by console
core1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
core1(config)#int e3/0
core1(config-if)#no sw
core1(config-if)#no switchport
core1(config-if)#ip add dh
core1(config-if)#ip add dhcp
*Jun 28 17:36:04.764: %LINK-3-UPDOWN: Interface Ethernet3/0, changed state to up
core1(config-if)#ip add dhcp
core1(config-if)#n
*Jun 28 17:36:05.770: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet3/0, changed state to up
core1(config-if)#no shut
core1(config-if)#end
core1#wr
Building configuration...

  Compressed configuration from 2378 bytes to 1179 bytes[OK]
core1#
*Jun 28 17:36:10.207: %SYS-5-CONFIG_I: Configured from console by console
core1#
*Jun 28 17:36:15.015: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet3/0 assigned DHCP address 192.168.1.189, mask 255.255.255.0, hostname core1
core1#
```

Hình 4-2: Cấp địa chỉ IP cho Switch Core 1 để SSH đến



Hình 4-3: Các file kịch bản

Để dễ quản lý và cấu hình thì sẽ chia thành các file tương ứng cho từng lớp của mô hình mạng như là Gateway_configuration (lớp Core), Core_configuration (lớp Distribution) và Access_configuration (lớp Access).

```

Core_configuration.py > ...
1  from netmiko import ConnectHandler
2
3
4  core1 = {
5      "device_type": "cisco_ios",
6      "ip": "192.168.1.189",
7      "username": "admin",
8      "password": "cisco",
9      "fast_cli": True,
10     'session_log': 'session_log_core-1.txt',
11     'secret': 'cisco',
12     "conn_timeout": 10,
13 }
14
15 core2 = {
16     "device_type": "cisco_ios",
17     "ip": "192.168.1.190",
18     "username": "admin",
19     "password": "cisco",
20     "fast_cli": True,
21     'session_log': 'session_log_core-2.txt',
22     'secret': 'cisco',
23     "conn_timeout": 10,
24 }
25
26 cores = [core1, core2]
27

```

Hình 4-4: File Core_configuration.py phân định nghĩa

Định nghĩa các thiết bị sẽ SSH đến để thực hiện cấu hình bao gồm: device_type, IP, username, password và các tùy chọn khác để hỗ trợ cho việc chạy cấu hình. session_log để xuất ra các file log để theo dõi quá trình cấu hình.

```

25
26 cores = [core1, core2]
27
28 configurations = [
29     [
30         'hostname SW-CORE-1',
31         'vlan 10',
32         'name NhanVien',
33         'vlan 20',
34         'name IT', # create VLAN
35         'ip routing',
36         'no ip cef',
37         'interface range e0/0-1',
38         'sw trunk encap dot1q',
39         'sw mode trunk',
40         # 'sw trunk allow vlan 10,20', # config trunk
41         'spanning-tree mode rapid',
42         'spanning-tree vlan 10 root primary',
43         'spanning-tree vlan 20 root primary', # config RSTP (CORE 1 is primary)
44         'int e0/2',
45         'no switchport',
46         'ip add 172.16.10.2 255.255.255.0',
47         'no shut',
48         'int e0/3',
49         'no switchport',
50         'ip add 172.16.14.2 255.255.255.0',
51         'no shut',
52         'int vlan 10',
53         'ip add 192.168.10.5 255.255.255.0',
54         'no shut',
55         'int vlan 20',
56         'ip add 192.168.20.5 255.255.255.0',
57         'no shut' # config interface

```

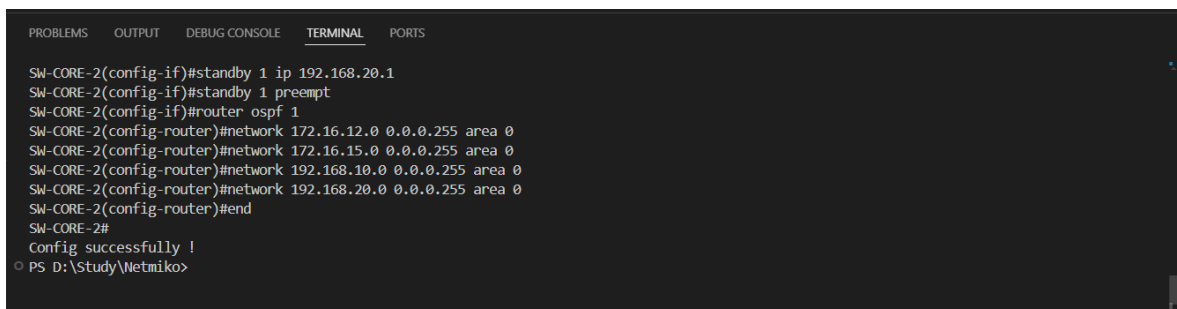
Hình 4-5: File Core_configuration.py phần cấu hình

```

133     def configure_device(device, commands):
134         net_connect = ConnectHandler(**device)
135         net_connect.enable()
136         output = net_connect.send_config_set(commands)
137         # save_config = net_connect.save_config()
138         # print('Saving configuration')
139         # print(save_config)
140         print(output)
141         net_connect.disconnect()
142
143     for core, config in zip(cores, configurations):
144         configure_device(core, config)
145
146     print ('Config successfully !')
147

```

Hình 4-6: File Core_configuration.py phân đẩy các cấu hình vào thiết bị



```

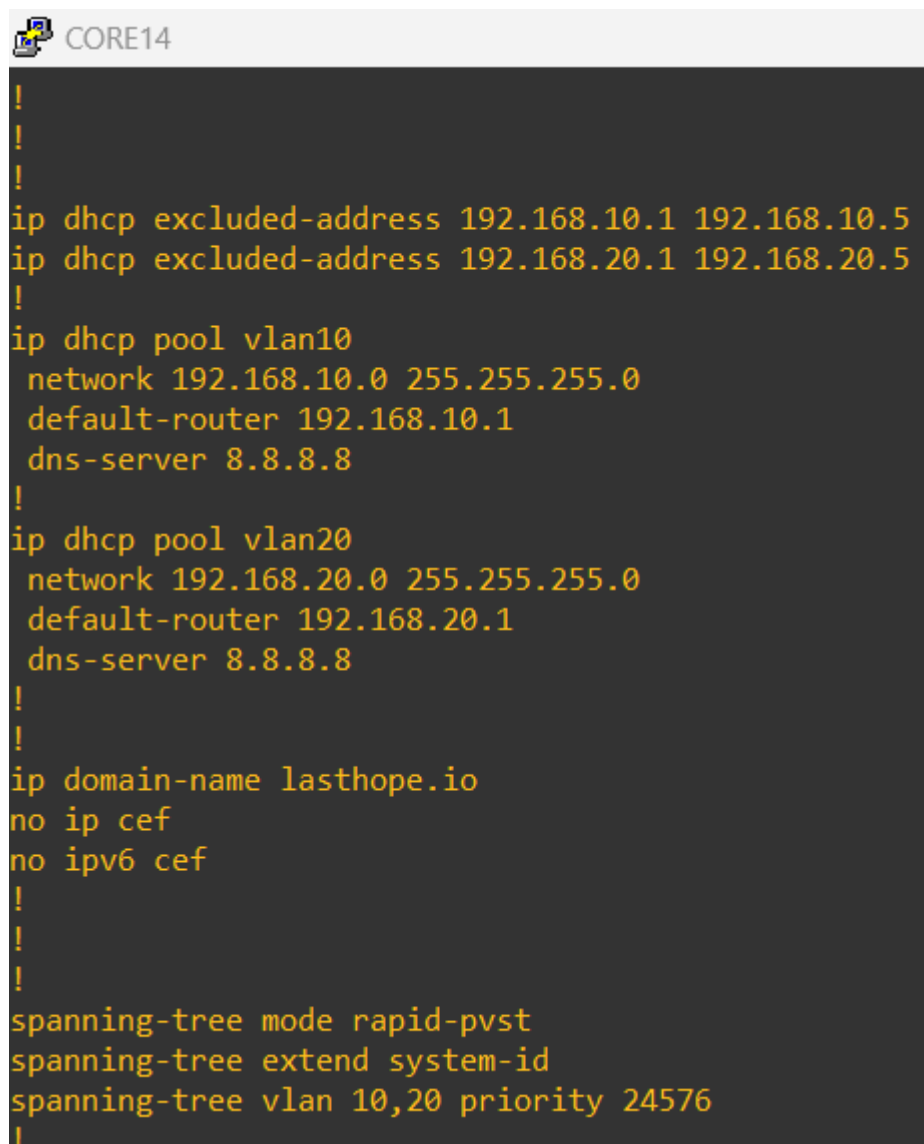
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

SW-CORE-2(config-if)#standby 1 ip 192.168.20.1
SW-CORE-2(config-if)#standby 1 preempt
SW-CORE-2(config-if)#router ospf 1
SW-CORE-2(config-router)#network 172.16.12.0 0.0.0.255 area 0
SW-CORE-2(config-router)#network 172.16.15.0 0.0.0.255 area 0
SW-CORE-2(config-router)#network 192.168.10.0 0.0.0.255 area 0
SW-CORE-2(config-router)#network 192.168.20.0 0.0.0.255 area 0
SW-CORE-2(config-router)#end
SW-CORE-2#
Config successfully !
PS D:\Study\Netmiko>

```

Hình 4-7: Sau khi chạy lệnh 'python ./[tên file]' trên terminal

Sau khi cấu hình xong sẽ xuất ra thông báo 'Config successfully !'



```
!
!
!
ip dhcp excluded-address 192.168.10.1 192.168.10.5
ip dhcp excluded-address 192.168.20.1 192.168.20.5
!
ip dhcp pool vlan10
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 8.8.8.8
!
ip dhcp pool vlan20
 network 192.168.20.0 255.255.255.0
 default-router 192.168.20.1
 dns-server 8.8.8.8
!
!
ip domain-name lasthope.io
no ip cef
no ipv6 cef
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
spanning-tree vlan 10,20 priority 24576
!
```

Hình 4-8: Kiểm tra cấu hình trên thiết bị

Chạy lệnh 'show run' trên thiết bị và thấy rằng các cấu hình đã được thêm vào từ file config bằng Netmiko.


```
≡ session_log_core-1.txt
1
2 SW-CORE-1#
3 SW-CORE-1#terminal width 511
4 SW-CORE-1#terminal length 0
5 SW-CORE-1#
6 SW-CORE-1#
7 SW-CORE-1#
8 SW-CORE-1#configure terminal
9 Enter configuration commands, one per line. End with CNTL/Z.
10 SW-CORE-1(config)#
11 SW-CORE-1(config)#hostname SW-CORE-1
12 SW-CORE-1(config)#vlan 10
13 SW-CORE-1(config-vlan)#name NhanVien
14 SW-CORE-1(config-vlan)#vlan 20
15 SW-CORE-1(config-vlan)#name IT
16 SW-CORE-1(config-vlan)#ip routing
17 SW-CORE-1(config)#no ip cef
18 SW-CORE-1(config)#interface range e0/0-1
19 SW-CORE-1(config-if-range)#sw trunk encap dot1q
20 SW-CORE-1(config-if-range)#sw mode trunk
21 SW-CORE-1(config-if-range)#spanning-tree mode rapid
22 SW-CORE-1(config)#spanning-tree vlan 10 root primary
23 SW-CORE-1(config)#spanning-tree vlan 20 root primary
24 SW-CORE-1(config)#int e0/2
25 SW-CORE-1(config-if)#no switchport
26 SW-CORE-1(config-if)#ip add 172.16.10.2 255.255.255.0
27 SW-CORE-1(config-if)#no shut
28 SW-CORE-1(config-if)#int e0/3
29 SW-CORE-1(config-if)#no switchport
30 SW-CORE-1(config-if)#ip add 172.16.14.2 255.255.255.0
31 SW-CORE-1(config-if)#no shut
32 SW-CORE-1(config-if)#int vlan 10
33 SW-CORE-1(config-if)#ip add 192.168.10.5 255.255.255.0
34 SW-CORE-1(config-if)#no shut
35 SW-CORE-1(config-if)#int vlan 20
36 SW-CORE-1(config-if)#ip add 192.168.20.5 255.255.255.0
37 SW-CORE-1(config-if)#no shut
```

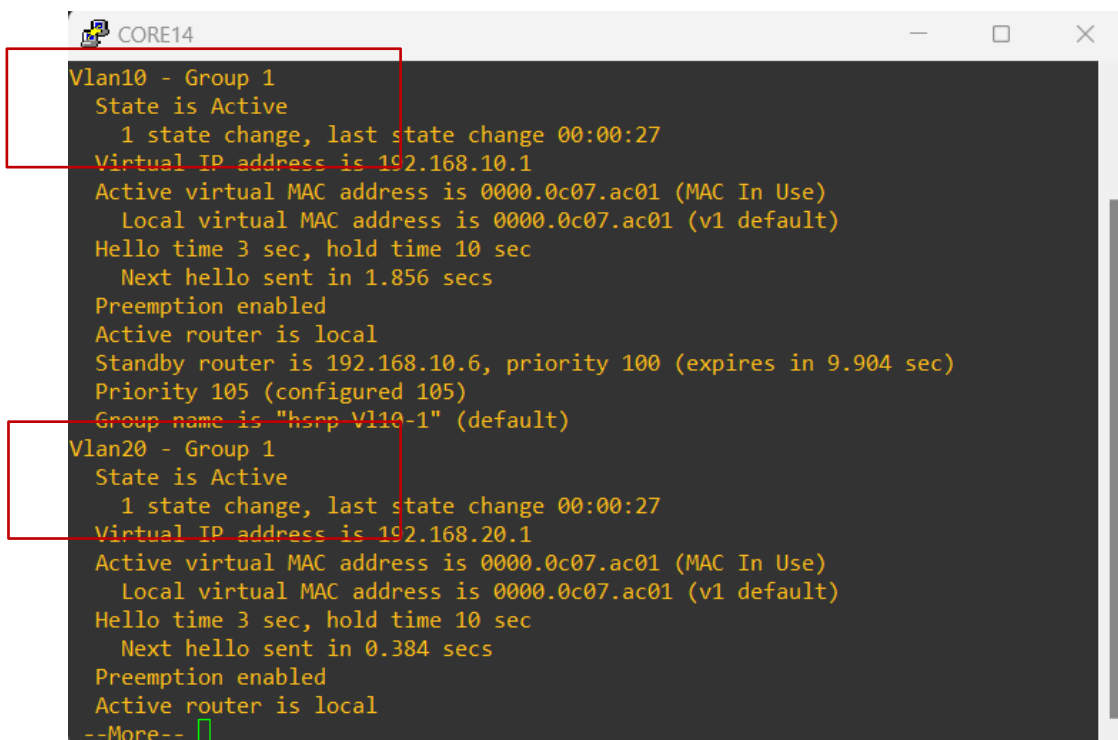
Hình 4-9: File log các cấu hình được đẩy vào thiết bị

Tham khảo thêm các file config các thiết bị còn lại tại đường dẫn này:

Link: <https://github.com/huannv2973/netmiko-configuration.git>

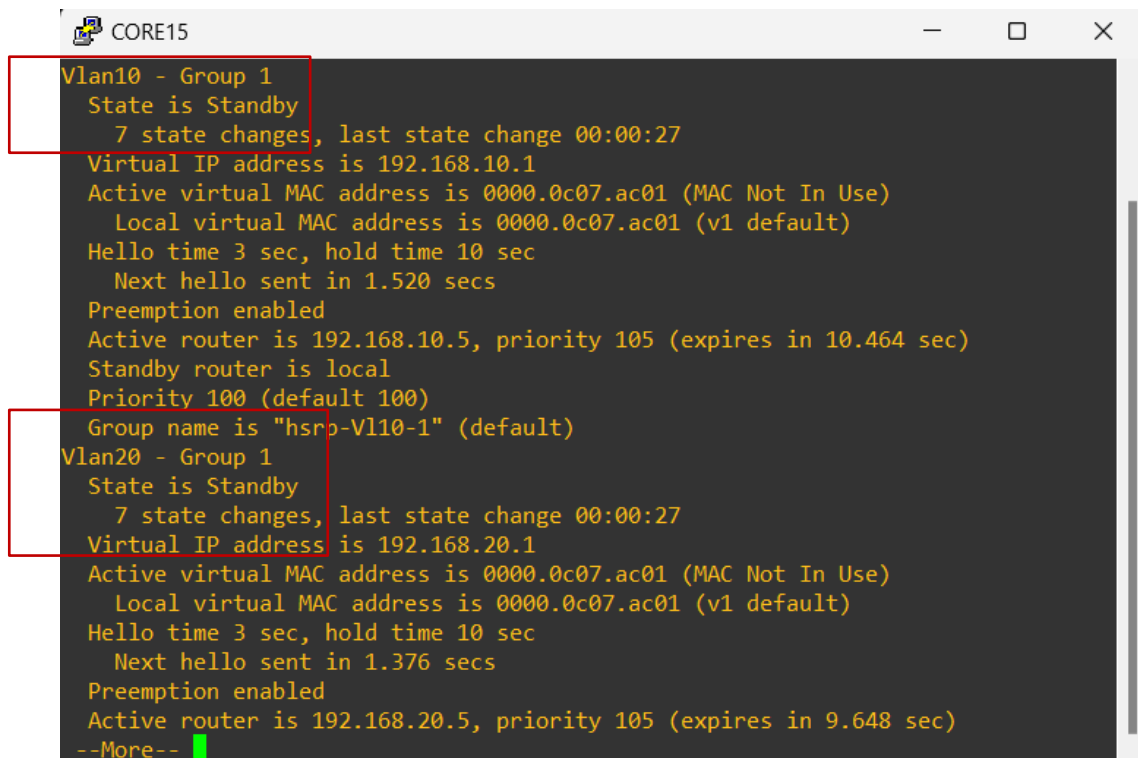
4.2. Trường hợp 1: Switch Core 1 bị shutdown

Trước khi bị SHUTDOWN:



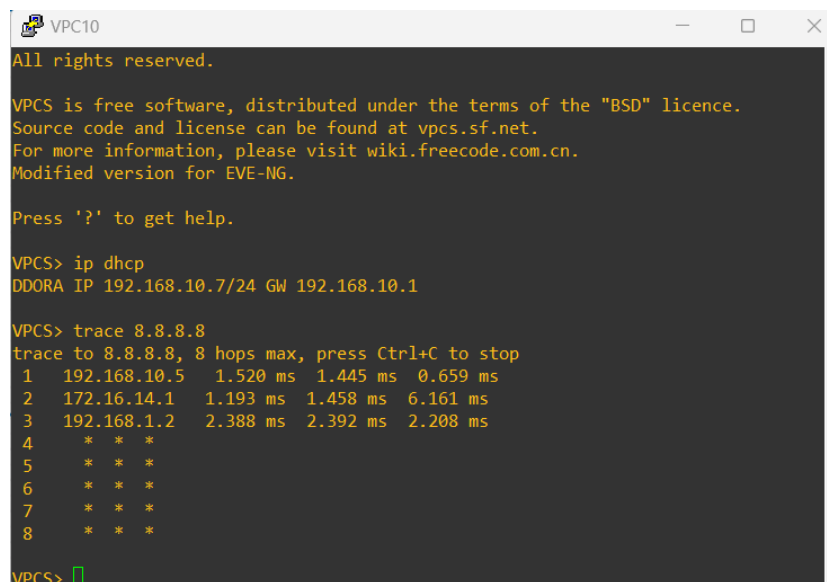
```
CORE14
Vlan10 - Group 1
  State is Active
    1 state change, last state change 00:00:27
  Virtual IP address is 192.168.10.1
  Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.856 secs
  Preemption enabled
  Active router is local
  Standby router is 192.168.10.6, priority 100 (expires in 9.904 sec)
  Priority 105 (configured 105)
  Group name is "hsrp-Vl10-1" (default)
Vlan20 - Group 1
  State is Active
    1 state change, last state change 00:00:27
  Virtual IP address is 192.168.20.1
  Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.384 secs
  Preemption enabled
  Active router is local
--More--
```

Hình 4-10: Trước khi bị Shutdown, trạng thái của Core 1 là Active cho VLAN 10 và 20



```
CORE15
Vlan10 - Group 1
State is Standby
  7 state changes, last state change 00:00:27
Virtual IP address is 192.168.10.1
Active virtual MAC address is 0000.0c07.ac01 (MAC Not In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
Hello time 3 sec, hold time 10 sec
  Next hello sent in 1.520 secs
Preemption enabled
Active router is 192.168.10.5, priority 105 (expires in 10.464 sec)
Standby router is local
Priority 100 (default 100)
  Group name is "hsrp-Vl10-1" (default)
Vlan20 - Group 1
State is Standby
  7 state changes, last state change 00:00:27
Virtual IP address is 192.168.20.1
Active virtual MAC address is 0000.0c07.ac01 (MAC Not In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
Hello time 3 sec, hold time 10 sec
  Next hello sent in 1.376 secs
Preemption enabled
Active router is 192.168.20.5, priority 105 (expires in 9.648 sec)
--More--
```

Hình 4-11: Trên Core 2 sẽ là trạng thái Standby (chờ) cho 2 VLAN 10 và 20



```
VPC10
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.
Modified version for EVE-NG.

Press '?' to get help.

VPCS> ip dhcp
DDORA IP 192.168.10.7/24 GW 192.168.10.1

VPCS> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  192.168.10.5    1.520 ms  1.445 ms  0.659 ms
 2  172.16.14.1    1.193 ms  1.458 ms  6.161 ms
 3  192.168.1.2    2.388 ms  2.392 ms  2.208 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
```

Hình 4-12: Trace 8.8.8.8 trên VPC thuộc VLAN 10

- Trên VPC thuộc VLAN 10, thực hiện xin địa chỉ IP từ IP DHCP Pool được cấu hình trên Switch Core 1. Sau khi được cấp địa chỉ IP, VPC sẽ có địa chỉ là 192.168.10.7/24 và default-gateway là 192.168.10.1
- Sau đó, thực hiện lệnh trace đến DNS Server 8.8.8.8 của Google thì đường đi của VPC như sau: VPC10 -> Switch Core 1 -> Router Gateway 2 -> Internet

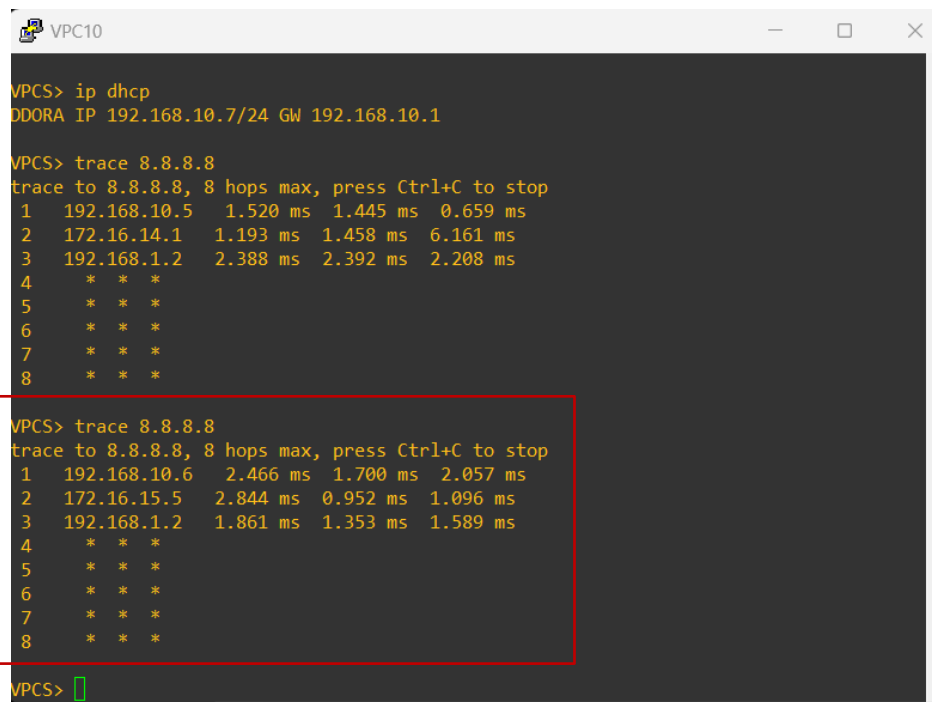
Sau khi bị SHUTDOWN:

```

CORE15
Vlan10 - Group 1
  State is Active
    8 state changes, last state change 00:00:05
  Virtual IP address is 192.168.10.1
  Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.912 secs
  Preemption enabled
  Active router is local
  Standby router is unknown
  Priority 100 (default 100)
  Group name is "hshp-Vl10-1" (default)
Vlan20 - Group 1
  State is Active
    8 state changes, last state change 00:00:03
  Virtual IP address is 192.168.20.1
  Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.576 secs
  Preemption enabled
  Active router is local
--More--

```

Hình 4-13: Sau khi shutdown Core 1, Core 2 sẽ chiếm quyền



```
VPCS> ip dhcp
DHORA IP 192.168.10.7/24 GW 192.168.10.1

VPCS> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  192.168.10.5    1.520 ms  1.445 ms  0.659 ms
 2  172.16.14.1    1.193 ms  1.458 ms  6.161 ms
 3  192.168.1.2    2.388 ms  2.392 ms  2.208 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *

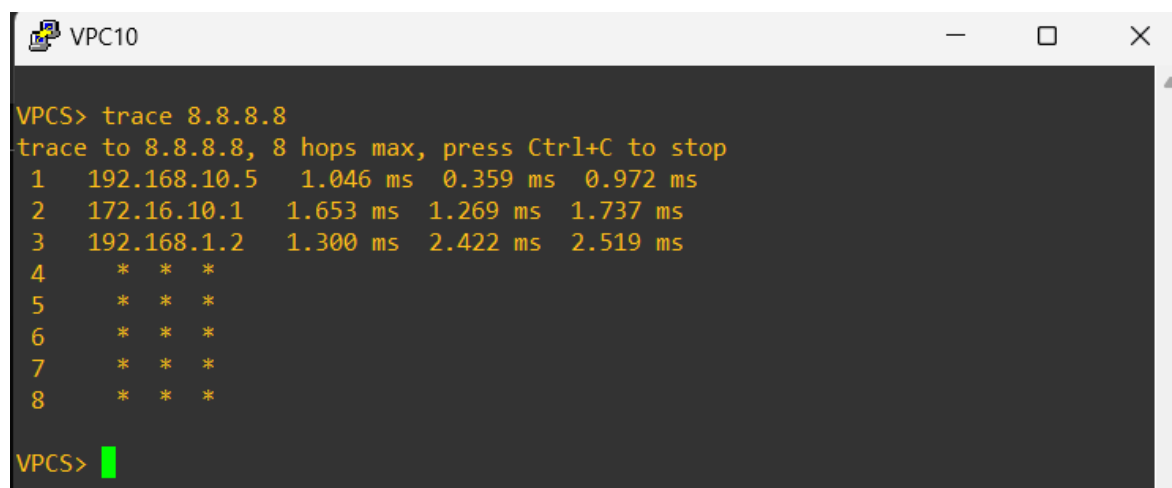
VPCS> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  192.168.10.6    2.466 ms  1.700 ms  2.057 ms
 2  172.16.15.5    2.844 ms  0.952 ms  1.096 ms
 3  192.168.1.2    1.861 ms  1.353 ms  1.589 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
```

Hình 4-14: Thực hiện trace trên VPC 10

Sau khi trace đến 8.8.8.8, đường đi của traffic đã có sự thay đổi, VPC -> Switch Core 2 -> Router Gateway 2 -> Internet

4.3. Trường hợp 2: Router Gateway 1 bị shutdown

Trước khi bị SHUTDOWN

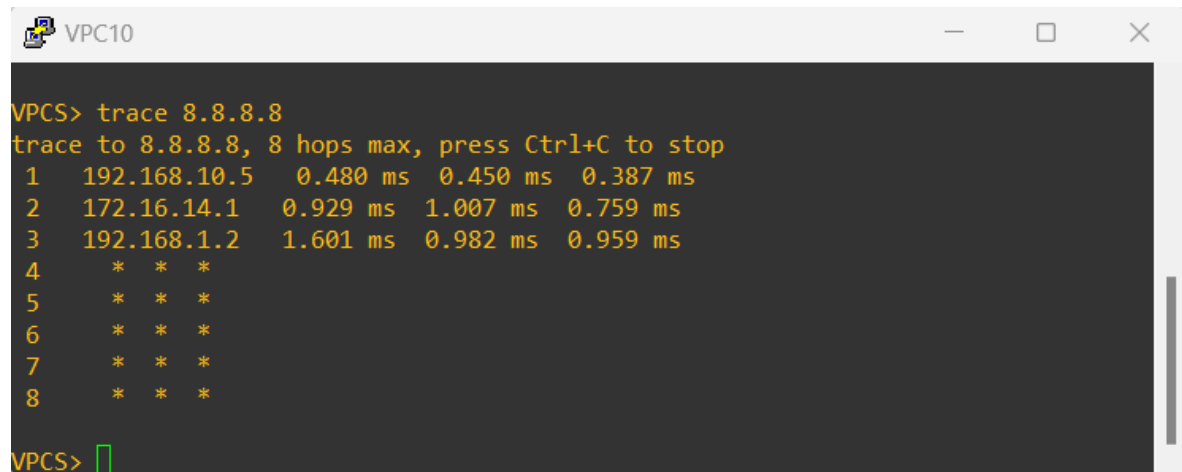


```
VPCS> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  192.168.10.5    1.046 ms  0.359 ms  0.972 ms
 2  172.16.10.1    1.653 ms  1.269 ms  1.737 ms
 3  192.168.1.2    1.300 ms  2.422 ms  2.519 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
```

Hình 4-15: Trace 8.8.8.8 trên VPC10

Đường đi của traffic sẽ là: VPC10 -> Switch Core 1 -> Router Gateway 1 -> Internet

Sau khi bị SHUTDOWN



```
VPC10
VPCS> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  192.168.10.5    0.480 ms  0.450 ms  0.387 ms
 2  172.16.14.1    0.929 ms  1.007 ms  0.759 ms
 3  192.168.1.2    1.601 ms  0.982 ms  0.959 ms
 4      * * *
 5      * * *
 6      * * *
 7      * * *
 8      * * *
VPCS> 
```

Hình 4-16: Trace 8.8.8.8 trên VPC10

Đường đi của traffic có thay đổi như sau: VPC10 -> Switch Core 1 -> Router Gateway 2 -> Internet

Chương 5. TỔNG KẾT

5.1. So sánh giữa 2 hình thức cấu hình

Kiểu cấu hình	CLI	Netmiko
Cách thức sử dụng	Thao tác thủ công, tiêu tốn thời gian, khả năng mắc lỗi cao	Tự động hoá, tiết kiệm thời gian và giảm sai sót
Độ linh hoạt và khả năng mở rộng	Linh hoạt, khó mở rộng	Linh hoạt và dễ mở rộng
Bảo mật	Kết nối trực tiếp, quản lý thủ công	Kết nối thông qua thư viện, dễ quản lý
Tính dễ dàng sử dụng	Đòi hỏi kiến thức sâu và học tập lâu	Dễ học và tái sử dụng nếu đã biết đến Python

Bảng 2: Bảng so sánh 2 kiểu cấu hình

5.2. Kết luận

5.2.1. Những công việc mà nhóm đã thực hiện được

- Thực hiện đề xuất và triển khai một mô hình mạng 3 lớp cơ bản
- Sử dụng Netmiko để cấu hình
- Sử dụng Ansible để quản lý các cấu hình
- Thực hiện các kịch bản lỗi của thiết bị
- So sánh và hiểu được 2 hình thức cấu hình cho mô hình mạng là truyền thống bằng lệnh CLI và tự động bằng Script Python

5.2.2. Phương hướng phát triển

- Mở rộng các kịch bản mô phỏng với quy mô lớn hơn và phức tạp hơn
- Nghiên cứu các xu hướng mới hiện nay như Network as Code (NaC) và Infrastructure as Code (IaC) trong lĩnh vực triển khai mô hình mạng và quản lý cấu hình

5.3. Tài liệu tham khảo

[1]

“Documentation.” <https://www.eve-ng.net/index.php/documentation/>

[2]

I. Ansible, “Ansible Documentation,” *Ansible.com*, 2019.

<https://docs.ansible.com/>

[3]

“Python for Network Engineers - Netmiko Library,” *pynet.twb-tech.com*.

<https://pynet.twb-tech.com/blog/netmiko-python-library.html>

[4]

Charles Judd, “Using Ansible Automation with EVE-NG,” *YouTube*, Jul. 29, 2021. <https://www.youtube.com/watch?v=H8yg7XnNfas> (accessed Jun. 29, 2024).

[5]

“Netmiko Tutorials For Network Engineers(12 Videos) - YouTube,” *www.youtube.com*.

<https://youtube.com/playlist?list=PL0ocymQm7YWbfFxETimO1VmgtVGRnEGyE&si=bQn3S0260ZSoU9a8> (accessed Jun. 29, 2024).

5.4. Demo sản phẩm

Trình tự của demo:

- Kích hoạt SSH trên các thiết bị, DHCP IP cho các thiết bị từ internet
- Chạy Ansible từ Linux Server
- Kiểm tra cấu hình trên thiết bị, chạy các kịch bản mô hình bị lỗi và traceroute các thiết bị ra ngoài Internet

Link demo: [DEMO](#)

