

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA AN TOÀN THÔNG TIN**



**BÁO CÁO BTL: TRIỂN KHAI NSM CHO OPENSTACK  
HỌC PHẦN: KỸ THUẬT THEO DÕI VÀ GIÁM SÁT AN TOÀN  
MẠNG  
MÃ HỌC PHẦN: INT1429M**

**NHÓM LỚP: D21CQAT01 - B**

Sinh viên thực hiện:

B21DCAT193 Mai Đức Trung

B21DCATxxx Nguyễn Đức Hùng

B21DCATxxx Nguyễn Đức Trọng

B21DCATxxx Vương Đức Anh

Tên nhóm: 9

Tên lớp: 2

Giảng viên: Ninh Thị Thu Trang

**HÀ NỘI 2025**

## PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN

TT	Công việc / Nhiệm vụ	SV thực hiện	Thời hạn hoàn thành
1	Tìm hiểu về OpenStack, các service thường dùng	Vương Đức Anh	20/3
2	Tìm hiểu khái quát hệ thống NSM triển khai trên OpenStack	Mai Đức Trung	20/3
3	Tìm hiểu các công cụ NSM có thể triển khai trên kiến trúc mạng OpenStack	Nguyễn Đức Trọng	23/3
4	Cấu hình demo hệ thống OpenStack áp dụng trên hệ thống mạng vật lý thật của máy chủ vật lý và triển khai công cụ NSM (Snort) trên đó	Nguyễn Đức Hùng	30/3
5	Cấu hình demo hệ thống OpenStack triển khai cho nhiều cụm cloud riêng biệt và triển khai công cụ NSM (Snort) trên đó	Mai Đức Trung	30/3

## TÓM TẮT NỘI DUNG CÁC CUỘC HỌP

- **Buổi 1 (07/03):** Phân công cả nhóm tìm hiểu qua về hệ thống OpenStack và đề xuất các giải pháp NSM có thể áp dụng vào hệ thống demo
  - Vương Đức Anh và Mai Đức Trung tìm hiểu qua về OpenStack và các giải pháp NSM có thể triển khai
  - Nguyễn Đức Trọng và Nguyễn Đức Hùng triển khai thử hệ thống OpenStack thực tế
- **Buổi 2 (21/03):** Demo thử hệ thống OpenStack do Nguyễn Đức Hùng triển khai
  - Nguyễn Đức Trọng chưa triển khai được hệ thống, chuyển sang tìm hiểu các giải pháp để triển khai trên OpenStack trong thực tiễn
  - Phân chia Nguyễn Đức Hùng và Mai Đức Trung triển khai 2 demo OpenStack riêng biệt trên mạng vật lý thật và trên mạng ảo OpenStack .
- **Buổi 3 (04/04):** Rút kinh nghiệm sau báo cáo tiến độ lần 1
  - Phân công Vương Đức Anh và Nguyễn Đức Trọng viết một số luật Snort kèm kịch bản demo để triển khai trên 2 hệ thống OpenStack đã tạo dựng được
  - Mai Đức Trung triển khai thêm nhiều cụm máy ảo trên hệ thống demo và thực hiện theo dõi trên từng cụm máy ảo qua Snort

### NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ

TT	SV thực hiện	Thái độ tham gia	Mức hoàn thành CV	Kỹ năng giao tiếp	Kỹ năng hợp tác	Kỹ năng lãnh đạo
1	Mai Đức Trung	5	5	4	4	4
2	Nguyễn Đức Hùng	5	5	4	4	
3	Nguyễn Đức Trọng	4	4	3	4	
4	Vương Đức Anh	4	4	4	3	

***Ghi chú:***

- Thái độ tham gia: Đánh giá điểm thái độ tham gia công việc chung của nhóm (từ 0: không tham gia, đến 5: chủ động, tích cực).
- Mức hoàn thành CV: Đánh giá điểm mức độ hoàn thành công việc được giao (từ 0: không hoàn thành, đến 5: hoàn thành xuất sắc).
- Kỹ năng giao tiếp: Đánh giá điểm khả năng tương tác, giao tiếp trong nhóm (từ 0: không hoặc giao tiếp rất yếu, đến 5: giao tiếp xuất sắc).
- Kỹ năng hợp tác: Đánh giá điểm khả năng hợp tác, hỗ trợ lẫn nhau, giải quyết mâu thuẫn, xung đột
- Kỹ năng lãnh đạo: Đánh giá điểm khả năng lãnh đạo (từ 0: không có khả năng lãnh đạo, đến 5: có khả năng lãnh đạo tốt, tổ chức và điều phối công việc trong nhóm hiệu quả).

## MỤC LỤC

PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN .....	2
TÓM TẮT NỘI DUNG CÁC CUỘC HỌP .....	2
MỤC LỤC .....	4
DANH MỤC CÁC HÌNH VẼ .....	5
DANH MỤC CÁC TỪ VIẾT TẮT.....	6
MỞ ĐẦU .....	7
<b>CHƯƠNG 1.</b> Tổng quan về đề tài.....	8
1.1 Giới thiệu đề tài .....	8
1.2 Hệ thống OpenStack .....	8
1.3 Các giải pháp NSM được triển khai trong thực tế.....	23
<b>CHƯƠNG 2.</b> Triển khai hệ thống OpenStack .....	24
2.1 Giới thiệu chương .....	24
2.2 Demo hệ thống triển khai mạng ảo trên OpenStack .....	24
<b>2.2.1</b> Mô hình hệ thống .....	24
<b>2.2.2</b> Quá trình cài đặt.....	24
<b>2.2.3</b> Triển khai hệ thống .....	30
2.3 Demo hệ thống OpenStack triển khai trên mạng vật lý thật trên LAN .....	34
<b>2.3.1</b> Cấu hình trước khi cài đặt .....	34
TÀI LIỆU THAM KHẢO .....	34

## DANH MỤC CÁC HÌNH VẼ

Hình 1 Sơ đồ mạng hệ thống.....	24
Hình 2 Cấu hình floating IP trên Pfsense .....	25
Hình 3 Cấu hình NAT forwarding trên Pfsense.....	25
Hình 4 Cấu hình rule WAN trong Pfsense .....	25
Hình 5 Rule Snort trên WAN Interface .....	26
Hình 6 Rule Snort trên LAN interface.....	26
Hình 7 Lấy mật khẩu admin OpenStack.....	27
Hình 8 Giao diện đăng nhập OpenStack.....	27
Hình 9 Các dải mạng được cấu hình .....	27
Hình 10 Các security group .....	28
Hình 11 Cấu hình rule trong security group .....	28
Hình 12 Cấu hình router OpenStack .....	28
Hình 13 Cấu hình interface cho các router để kết nối mạng internal với external.....	29
Hình 14 Các instance trong hệ thống .....	29
Hình 15 Cấu hình Port Forwarding trên máy chủ OpenStack .....	30
Hình 16 SSH từ WAN vào instance trên dải 192.168.222.0/24 .....	30
Hình 17 SSH từ WAN vào instance trên dải 20.20.0.0/24 .....	31
Hình 18 Alert trên Snort theo dõi các cố gắng SSH đến instance từ WAN .....	31
Hình 19 SSH từ LAN vào instance trên dải 20.20.0.0/24 .....	32
Hình 20 SSH từ LAN vào instance trên dải 192.168.222.0/24.....	32
Hình 21 Alert trên Snort theo dõi các cố gắng SSH đến instance từ LAN.....	32
Hình 22 Tiến hành tấn công DOS và quét cổng trên máy chủ OpenStack.....	33
Hình 23 Alert trên Snort theo dõi các cố gắng tấn công đến máy chủ OpenStack .....	33
Hình 24 Kết nối giữa 2 instance khác security group .....	34

## DANH MỤC CÁC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Thuật ngữ tiếng Anh/Giải thích</b>	<b>Thuật ngữ tiếng Việt/Giải thích</b>
LAN	Local Area Network	Mạng máy tính cục bộ, kết nối các thiết bị (máy tính, máy in, điện thoại, camera, v.v.) trong phạm vi nhỏ (phòng, tòa nhà, văn phòng, v.v.)
WAN	Wide Area Network	Mạng diện rộng, kết nối các mạng LAN lại với nhau trên một khu vực rộng lớn (như thành phố, quốc gia hoặc toàn cầu).
IP	Internet Protocol	Giao thức định danh thiết bị trong mạng – mỗi thiết bị có một địa chỉ IP để nhận và gửi dữ liệu
TCP	Transmission Control Protocol	TCP là giao thức truyền dữ liệu đảm bảo độ tin cậy.
NAT	Network Address Translation	Kỹ thuật chuyển đổi địa chỉ IP riêng (private) trong mạng LAN thành địa chỉ IP công cộng (public) để ra Internet.
SSH	Secure Shell	Giao thức mã hóa giúp quản trị từ xa một máy tính qua mạng.
NSM	Network Security Monitoring	Giám sát an ninh mạng
IaaS	Infrastructure as a Service	Cơ sở hạ tầng dưới dạng dịch vụ. Đây là một mô hình điện toán đám mây, trong đó nhà cung cấp dịch vụ (cloud provider) cung cấp các tài nguyên hạ tầng CNTT cơ bản

## MỞ ĐẦU

Trong bối cảnh công nghệ điện toán đám mây ngày càng phát triển mạnh mẽ, nhu cầu triển khai các hệ thống ảo hóa linh hoạt, tiết kiệm chi phí và dễ mở rộng trở nên vô cùng cần thiết đối với các tổ chức và doanh nghiệp. OpenStack nổi lên như một nền tảng mã nguồn mở hàng đầu trong việc xây dựng và quản lý hạ tầng đám mây theo mô hình IaaS (Infrastructure as a Service). Với khả năng quản lý tài nguyên tính toán, lưu trữ và mạng một cách tập trung, OpenStack đang được ứng dụng rộng rãi trong cả môi trường nghiên cứu và thực tiễn triển khai. Tuy nhiên, cùng với sự gia tăng về quy mô và tính linh hoạt của hệ thống, các mối đe dọa về bảo mật mạng cũng ngày càng trở nên phức tạp hơn. Do đó, việc tích hợp các giải pháp bảo mật mạng hiện đại vào hạ tầng OpenStack là một yêu cầu cấp thiết. Trong đó, NSM (Network Security Monitoring) – giám sát an ninh mạng – đóng vai trò quan trọng trong việc phát hiện, phân tích và phản ứng với các mối nguy cơ tiềm ẩn trong mạng nội bộ.

Báo cáo này nhằm mục tiêu tìm hiểu kiến trúc tổng quan của hệ thống OpenStack, các thành phần chức năng chính, và nghiên cứu giải pháp triển khai NSM trên nền tảng OpenStack. Ngoài ra báo cáo này mô tả quá trình triển khai demo OpenStack trên hệ thống cụ thể và áp dụng công cụ Snort lên hệ thống nhằm mục đích ngăn chặn và giám sát. Qua đó, giúp hiểu rõ cách thức xây dựng môi trường giám sát an ninh mạng hiệu quả, đồng thời cung cấp kiến thức thực tiễn phục vụ cho việc triển khai, nghiên cứu và phát triển các hệ thống đám mây an toàn trong tương lai.

Nhóm cam kết rằng toàn bộ nội dung và hoạt động được trình bày trong báo cáo này đều được thực hiện trong môi trường lab cá nhân, cách ly hoàn toàn với mạng công cộng và các hệ thống không được phép truy cập. Mọi hành vi mô phỏng tấn công mạng, khai thác lỗ hổng, hoặc can thiệp vào hệ thống đều được thực hiện với mục đích nghiên cứu, học tập và kiểm thử khả năng giám sát an ninh mạng (NSM) trong khuôn khổ môn học.

Báo cáo bài tập lớn gồm 2 chương bao gồm các nội dung như sau:

- **Chương 1:** Chương này trình bày các khái niệm và kiến trúc cơ bản của hệ thống OpenStack. Nhóm sẽ mô tả khái quát các dịch vụ có trong dịch vụ OpenStack kèm kiến trúc hoạt động của dịch vụ mạng trong OpenStack. Ngoài ra chương này cũng sẽ giới thiệu các công cụ giám sát an ninh mạng có thể áp dụng và triển khai trong thực tiễn.
- **Chương 2:** Chương này mô tả quá trình xây dựng hệ thống OpenStack, triển khai hệ thống mạng, các cấu hình cũng như triển khai dịch vụ có trong OpenStack. Nhóm áp dụng giải pháp Snort dùng như một giải pháp giám sát, ngăn chặn trên hệ thống OpenStack và triển khai một số tấn công giả lập lên hệ thống để kiểm tra khả năng của giải pháp.

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Giới thiệu đề tài

Chương này trình bày hai nội dung trọng tâm nhằm xây dựng nền tảng lý thuyết cho việc triển khai và đánh giá giải pháp giám sát an ninh mạng trên môi trường OpenStack.

Phần đầu tiên tập trung vào việc tìm hiểu tổng quan về hệ thống OpenStack, bao gồm kiến trúc tổng thể, các thành phần dịch vụ cốt lõi như Nova, Neutron, Glance, Keystone, v.v., cũng như cách các dịch vụ này phối hợp với nhau để cung cấp một hạ tầng điện toán đám mây hoàn chỉnh theo mô hình IaaS (Infrastructure as a Service). Nội dung này nhằm giúp người đọc hiểu rõ cách tổ chức và vận hành của một môi trường đám mây được xây dựng bằng OpenStack.

Phần thứ hai cung cấp cái nhìn tổng quát về các công cụ giám sát an ninh mạng (Network Security Monitoring - NSM) thường được triển khai trong thực tế. Nội dung sẽ đề cập đến vai trò của NSM trong việc phát hiện, phân tích và phản ứng với các mối đe dọa trong mạng, đồng thời giới thiệu một số công cụ phổ biến thường được sử dụng để xây dựng hệ thống NSM trong môi trường ảo hóa hoặc đám mây.

Thông qua chương này, người đọc sẽ có cái nhìn tổng thể về cả nền tảng OpenStack và các giải pháp NSM, làm cơ sở cho các chương tiếp theo liên quan đến thiết kế, triển khai và thử nghiệm hệ thống NSM trên môi trường OpenStack thực tế.

## 1.2 Hệ thống OpenStack

### 1.2.1 Khái niệm OpenStack

OpenStack là một nền tảng mã nguồn mở được sử dụng để triển khai và quản lý hạ tầng đám mây quy mô lớn. Được phát triển bởi Rackspace Hosting và NASA vào năm 2010, OpenStack đã trở thành một trong những giải pháp phổ biến nhất để xây dựng đám mây công cộng và đám mây riêng. Với kiến trúc mô-đun, OpenStack bao gồm nhiều dịch vụ hoạt động cùng nhau để quản lý tài nguyên tính toán, lưu trữ, mạng, và các chức năng hỗ trợ như điều phối và giám sát. Các dịch vụ này được tích hợp thông qua API, cho phép người dùng tùy chỉnh và mở rộng theo nhu cầu.





*Hình 1 Nền tảng OpenStack*

Các máy user từ ngoài mạng WAN muốn truy cập vào các Instance trong OpenStack sẽ đi qua route như sau:

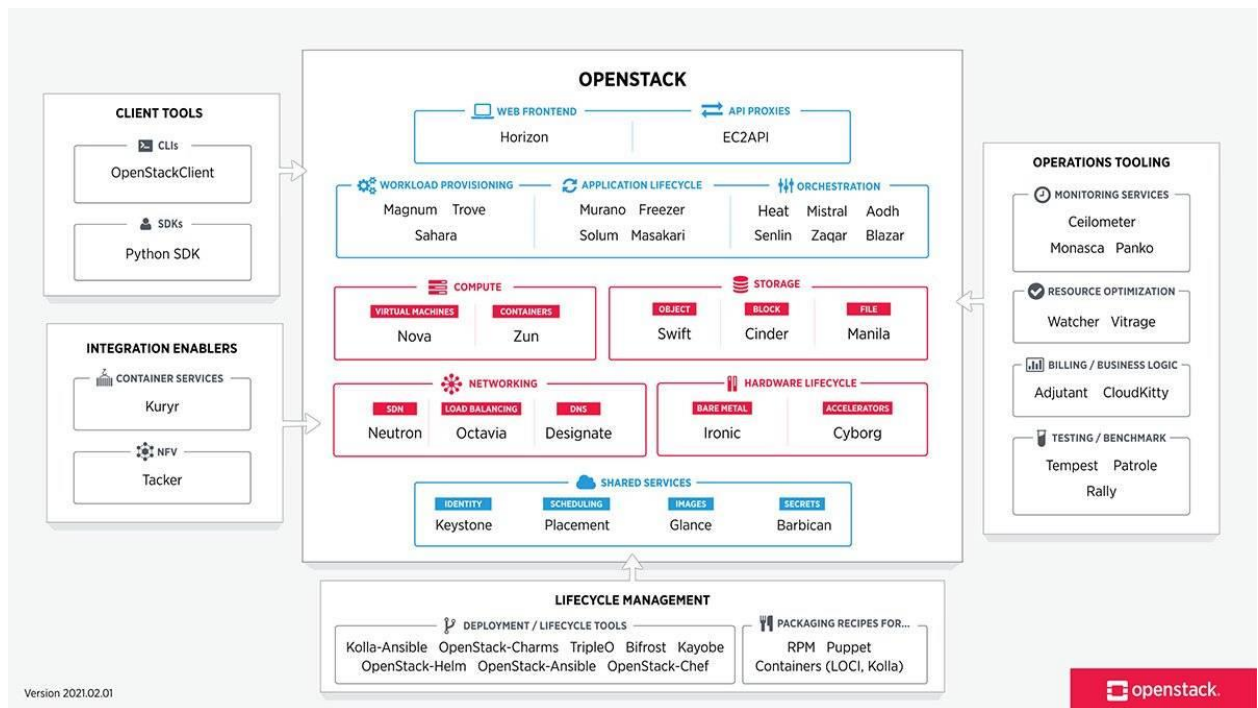
- Gói tin TCP gửi đến PfSense qua floating IP tạo bởi tường lửa
- Tường lửa chuyển tiếp gói tin TCP qua máy chủ OpenStack.
- Máy chủ OpenStack chuyển tiếp gói tin TCP đến Instance.

### ***1.2.2 Vai trò của OpenStack***

Các máy user từ ngoài mạng WAN muốn truy cập vào các Instance trong OpenStack sẽ đi qua route như sau:

- Tự động hóa quản lý tài nguyên: OpenStack cho phép tự động hóa việc phân bổ, giám sát, và tối ưu hóa tài nguyên (máy ảo, lưu trữ, mạng) thông qua các API và giao diện người dùng.
- Hỗ trợ triển khai ứng dụng: Cung cấp nền tảng để triển khai các ứng dụng phức tạp, đặc biệt trong môi trường giáo dục như, nơi cần mở rộng nhanh chóng trong các kỳ cao điểm (đăng ký môn học, thi cử).
- Tích hợp linh hoạt: Hỗ trợ tích hợp với các công nghệ hiện đại như container (Kubernetes qua Kuryr), mạng chức năng ảo (NFV qua Tacker), và các công cụ DevOps (Ansible, Chef).
- Tính mở và cộng đồng: Là mã nguồn mở, OpenStack cho phép tùy chỉnh và mở rộng theo nhu cầu, với tài liệu và hỗ trợ từ cộng đồng toàn cầu.

### ***1.2.3 Tổng quan về sơ đồ kiến trúc OpenStack***



Hình 2 Sơ đồ kiến trúc OpenStack (phiên bản 2021.02.01)

Sơ đồ kiến trúc OpenStack (phiên bản 2021.02.01) minh họa các thành phần chính, được chia thành các nhóm chức năng:

- Client Tools: Công cụ giao tiếp với OpenStack như CLI (OpenStackClient) và SDKs (Python SDK).
- Integration Enablers: Hỗ trợ tích hợp với container (Kuryr) và NFV (Tacker).
- OpenStack Core: Các dịch vụ lõi bao gồm Compute (Nova), Storage (Swift, Cinder, Manila), Networking (Neutron), Shared Services (Keystone, Glance), và Web Frontend (Horizon).
- Operations Tooling: Dịch vụ vận hành như giám sát (Ceilometer) và tối ưu hóa (Watcher).
- Lifecycle Management: Công cụ triển khai như Kolla-Ansible, TripleO.

Sơ đồ này giúp hiểu mối quan hệ giữa các dịch vụ, ví dụ: Nova sử dụng Glance để lấy hình ảnh máy ảo và Neutron để cấu hình mạng.

## 1.3 Các dịch vụ thường dung trong OpenStack

### 1.3.1 Swift (Object Storage)



*Hình 3 Logo Swift*

Swift là dịch vụ lưu trữ đối tượng thuộc hệ sinh thái OpenStack, được thiết kế để quản lý dữ liệu không có cấu trúc như hình ảnh, video, tài liệu, và bản sao lưu, với khả năng xử lý khối lượng dữ liệu lớn (quy mô petabyte). Đây là giải pháp lý tưởng cho các tổ chức cần hệ thống lưu trữ đáng tin cậy, hiệu quả trong môi trường phân tán.

Đặc điểm nổi bật:

- Khả năng mở rộng: Dễ dàng bổ sung node lưu trữ mới mà không gián đoạn hoạt động.
- Độ bền: Bảo vệ dữ liệu trước sự cố phần cứng qua sao chép hoặc mã hóa xóa.
- Tính khả dụng: Duy trì truy cập dữ liệu liên tục nhờ cơ chế phân tán thông minh.
- Hiệu quả chi phí: Phù hợp cho lưu trữ đám mây, nội dung đa phương tiện, hoặc sao lưu dữ liệu.

Thành phần chính:

- Proxy Server: Tiếp nhận, xử lý, định tuyến yêu cầu API từ người dùng, cân bằng tải, quản lý giao dịch để tối ưu hiệu suất. Tường lửa chuyển tiếp gói tin TCP qua máy chủ OpenStack.
- Storage Nodes: Lưu trữ dữ liệu thực tế, gồm Account Server (quản lý thông tin tài khoản và danh sách container), Container Server (quản lý container chứa metadata của đối tượng), Object Server (lưu trữ nội dung đối tượng như tệp hình ảnh, video).
- Ring: Ánh xạ dữ liệu dùng thuật toán Consistent Hashing, phân bố đồng đều, hỗ trợ phục hồi khi có lỗi node.

Cơ chế bảo vệ dữ liệu:

- Sao chép: Mỗi đối tượng được sao chép (thường 3 bản) trên các node khác nhau, đảm bảo truy xuất khi có lỗi. Tường lửa chuyển tiếp gói tin TCP qua máy chủ OpenStack.
- Mã hóa xóa: Dữ liệu chia thành fragment kèm mã sửa lỗi, tiết kiệm dung lượng lưu trữ, phù hợp cho hệ thống ưu tiên tiết kiệm không gian.

- Giao thức truy cập: Sử dụng RESTful API (GET, PUT, POST, DELETE) để tích hợp dễ dàng vào ứng dụng web hoặc hệ thống quản lý dữ liệu đám mây.

Quy trình hoạt động:

- Dữ liệu được tổ chức theo cấu trúc: Accounts (đại diện người dùng/tổ chức, chứa container), Containers (thư mục logic để nhóm đối tượng), Objects (tệp dữ liệu như hình ảnh, tài liệu). Tường lửa chuyển tiếp gói tin TCP qua máy chủ OpenStack.
- Người dùng gửi yêu cầu (tải lên, tải xuống, xóa) qua RESTful API đến Proxy Server.
- Proxy Server dùng Ring xác định vị trí lưu trữ trên Storage Nodes.
- Dữ liệu lưu dưới dạng đối tượng, áp dụng sao chép hoặc mã hóa xóa tùy cấu hình.
- Hệ thống tự động kiểm tra, phục hồi dữ liệu khi phát hiện lỗi node qua cơ chế replicator, đảm bảo tính toàn vẹn và khả dụng.

### 1.3.2 Cinder (Block Storage)



Hình 4 Logo Cinder

Cinder là dịch vụ lưu trữ khối (Block Storage) thuộc hệ sinh thái OpenStack, cung cấp không gian lưu trữ bền vững cho các máy ảo, cơ sở dữ liệu, hoặc ứng dụng yêu cầu truy cập dữ liệu tốc độ cao. Cinder cho phép quản lý vòng đời của các khối lưu trữ (volume), phù hợp cho các ứng dụng cần hiệu suất cao và khả năng lưu trữ linh hoạt.

Đặc điểm nổi bật:

- Khả năng quản lý linh hoạt: Tạo, gắn, tháo, và xóa volume dễ dàng cho máy ảo hoặc ứng dụng.
- Độ bền: Đảm bảo dữ liệu được lưu trữ an toàn với các cơ chế sao lưu và snapshot.
- Hiệu suất cao: Hỗ trợ truy cập dữ liệu nhanh, phù hợp cho cơ sở dữ liệu hoặc hệ thống tệp.
- Tích hợp đa nền tảng: Hỗ trợ nhiều backend lưu trữ như LVM, NFS, Ceph, hoặc thiết bị lưu trữ thương mại.

Thành phần chính:

- API Server: Tiếp nhận và xử lý yêu cầu API từ người dùng, chuyển tiếp đến các dịch vụ khác.
- Scheduler: Phân bổ yêu cầu tạo volume đến backend lưu trữ phù hợp dựa trên cấu hình và tài nguyên.
- Volume Service: Quản lý vòng đời volume, thực hiện các thao tác như tạo, xóa, hoặc gắn volume vào máy ảo.
- Backend Storage: Hệ thống lưu trữ thực tế (LVM, Ceph, hoặc thiết bị lưu trữ doanh nghiệp) nơi dữ liệu volume được lưu.

Cơ chế bảo vệ dữ liệu:

- Snapshot: Tạo bản chụp nhanh của volume để sao lưu hoặc khôi phục dữ liệu tại một thời điểm.
- Backup: Lưu trữ bản sao volume vào hệ thống lưu trữ đối tượng (như Swift) để bảo vệ lâu dài.
- Replication: Sao chép volume giữa các backend để đảm bảo khả năng chịu lỗi và khôi phục sau thảm họa.

Giao thức truy cập: Sử dụng API RESTful (GET, POST, PUT, DELETE) để quản lý volume, tích hợp với các dịch vụ OpenStack như Nova (Compute) để gắn volume vào máy ảo.

Quy trình hoạt động:

- Dữ liệu được tổ chức dưới dạng volume (khối lưu trữ logic), snapshot (bản chụp nhanh), hoặc backup (bản sao lưu).
- Người dùng gửi yêu cầu (tạo, gắn, xóa volume) qua API đến API Server.
- Scheduler chọn backend lưu trữ phù hợp dựa trên yêu cầu (dung lượng, loại lưu trữ).
- Volume Service thực hiện thao tác trên backend, gắn volume vào máy ảo thông qua Nova.
- Hệ thống hỗ trợ sao lưu, snapshot, hoặc sao chép volume để đảm bảo an toàn và khôi phục dữ liệu khi cần.

### ***1.3.3 Nova - Dịch vụ Tính toán trong OpenStack***



## Hình 5 Logo Nova

Nova là dịch vụ tính toán thuộc hệ sinh thái OpenStack, chịu trách nhiệm quản lý và cung cấp tài nguyên tính toán để chạy các máy ảo (VM) hoặc container trong môi trường đám mây. Nova cho phép triển khai, quản lý và mở rộng các phiên bản tính toán một cách linh hoạt, phù hợp cho các ứng dụng từ nhỏ đến quy mô doanh nghiệp.

### Đặc điểm nổi bật:

- Quản lý linh hoạt: Tạo, khởi động, tạm dừng, hoặc xóa máy ảo một cách dễ dàng.
- Khả năng mở rộng: Hỗ trợ thêm node tính toán để đáp ứng nhu cầu tài nguyên tăng cao.
- Tích hợp chặt chẽ: Làm việc với các dịch vụ OpenStack như Cinder (lưu trữ khối), Neutron (mạng), và Glance (hình ảnh) để cung cấp môi trường đám mây hoàn chỉnh.
- Hiệu suất cao: Tối ưu hóa tài nguyên phần cứng để đảm bảo hiệu suất máy ảo.

### Thành phần chính:

- API Server: Tiếp nhận và xử lý yêu cầu API từ người dùng, chuyển tiếp đến các dịch vụ liên quan.
- Scheduler: Phân bổ yêu cầu tạo máy ảo đến các node tính toán dựa trên tài nguyên và chính sách.
- Compute Node: Quản lý tài nguyên phần cứng (CPU, RAM, disk) và chạy các máy ảo thông qua hypervisor (như KVM, VMware, hoặc Hyper-V).
- Conductor: Điều phối các tác vụ phức tạp, như cập nhật cơ sở dữ liệu, để giảm tải cho Compute Node.
- Database: Lưu trữ thông tin trạng thái và cấu hình của máy ảo.

### Cơ chế bảo vệ dữ liệu:

- Live Migration: Di chuyển máy ảo giữa các node tính toán mà không làm gián đoạn dịch vụ.
- Snapshot: Tạo bản chụp nhanh của máy ảo để sao lưu hoặc khôi phục trạng thái.
- High Availability: Tự động phát hiện và khôi phục máy ảo khi node tính toán gặp sự cố.

Giao thức truy cập: Sử dụng RESTful API (GET, POST, PUT, DELETE) để quản lý máy ảo, tích hợp với các công cụ quản lý đám mây hoặc giao diện người dùng.

### Quy trình hoạt động:

- Dữ liệu được tổ chức dưới dạng máy ảo (VM), sử dụng hình ảnh từ Glance, lưu trữ từ Cinder, và mạng từ Neutron.

- Người dùng gửi yêu cầu (tạo, khởi động, xóa VM) qua API đến API Server.
- Scheduler chọn node tính toán phù hợp dựa trên tài nguyên (CPU, RAM) và yêu cầu người dùng.
- Compute Node triển khai máy ảo thông qua hypervisor, gắn volume (Cinder) và cấu hình mạng (Neutron).
- Hệ thống hỗ trợ di chuyển, sao lưu, hoặc khôi phục máy ảo thông qua snapshot và live migration, đảm bảo tính liên tục và an toàn.

#### ***1.3.4 Neutron - Dịch vụ Mạng trong OpenStack***



*Hình 6 Logo Neutron*

Neutron là dịch vụ mạng thuộc hệ sinh thái OpenStack, chịu trách nhiệm cung cấp và quản lý tài nguyên mạng cho các máy ảo, container, hoặc ứng dụng trong môi trường đám mây. Neutron hỗ trợ tạo mạng ảo linh hoạt, phù hợp cho các kịch bản từ hệ thống đơn giản đến hạ tầng doanh nghiệp phức tạp.

Đặc điểm nổi bật:

- Quản lý mạng linh hoạt: Tạo, cấu hình và xóa mạng ảo, subnet, router, hoặc firewall theo nhu cầu.
- Khả năng mở rộng: Hỗ trợ nhiều công nghệ mạng, từ VLAN, VXLAN đến SDN, đáp ứng quy mô lớn.
- Tích hợp đa dịch vụ: Kết nối với Nova (tính toán), Cinder (lưu trữ) để cung cấp hạ tầng đám mây hoàn chỉnh.
- Tùy chỉnh cao: Hỗ trợ các plugin và driver để tích hợp với phần cứng hoặc phần mềm mạng bên thứ ba.

#### Thành phần chính:

- API Server: Tiếp nhận và xử lý yêu cầu API từ người dùng, chuyển tiếp đến các dịch vụ mạng liên quan.
- Core Plugin: Quản lý các tài nguyên mạng cơ bản như mạng, subnet, và port.
- Service Plugin: Hỗ trợ các dịch vụ nâng cao như Load Balancer, VPN, hoặc Firewall-as-a-Service.
- ML2 Plugin: Quản lý đa dạng công nghệ mạng (VLAN, VXLAN, GRE) thông qua các driver.
- Agents: Thực thi cấu hình mạng trên node tính toán hoặc node mạng, bao gồm L2 Agent (quản lý switch ảo), L3 Agent (quản lý router), và DHCP Agent (cung cấp địa chỉ IP).

#### Cơ chế bảo vệ dữ liệu:

- Virtual Networking: Tạo mạng ảo cách ly (isolated network) hoặc kết nối với mạng bên ngoài qua router ảo.
- Security Groups: Áp dụng quy tắc firewall để kiểm soát lưu lượng vào/ra máy ảo.
- Floating IP: Gán địa chỉ IP công cộng cho máy ảo để truy cập từ mạng bên ngoài.

Giao thức truy cập: Sử dụng RESTful API (GET, POST, PUT, DELETE) để quản lý tài nguyên mạng, tích hợp với các công cụ quản lý đám mây hoặc giao diện người dùng.

#### Quy trình hoạt động:

- Tài nguyên mạng được tổ chức dưới dạng: Network (mạng ảo), Subnet (dải địa chỉ IP), Port (điểm kết nối với máy ảo), Router (kết nối mạng nội bộ và bên ngoài).
- Người dùng gửi yêu cầu (tạo mạng, gán IP, cấu hình router) qua API đến API Server.
- API Server phối hợp với Core Plugin, Service Plugin, và ML2 Plugin để xử lý yêu cầu.
- Agents thực thi cấu hình trên node tính toán hoặc node mạng, đảm bảo máy ảo được kết nối đúng cách.



- Hệ thống hỗ trợ security groups, floating IP, và các dịch vụ nâng cao như load balancing để đảm bảo an toàn và hiệu quả mạng.

### ***1.3.5 Horizon - Giao diện Quản lý trong OpenStack***



*Hình 7 Logo Horizon*

Horizon là dịch vụ giao diện quản lý (Dashboard) thuộc hệ sinh thái OpenStack, cung cấp giao diện web trực quan để người dùng và quản trị viên quản lý các tài nguyên đám mây như máy ảo, mạng, lưu trữ, và hình ảnh. Horizon giúp đơn giản hóa việc vận hành môi trường OpenStack, phù hợp cho cả người dùng mới và quản trị viên hệ thống.

Đặc điểm nổi bật:

- Giao diện thân thiện: Cung cấp giao diện web dễ sử dụng để quản lý tài nguyên mà không cần dòng lệnh.
- Quản lý toàn diện: Hỗ trợ tương tác với các dịch vụ OpenStack như Nova, Cinder, Neutron, và Glance.

- Tùy chỉnh linh hoạt: Cho phép tùy chỉnh giao diện và tích hợp các tính năng bổ sung thông qua plugin.
- Phân quyền rõ ràng: Hỗ trợ vai trò người dùng và quản trị viên để kiểm soát truy cập tài nguyên.

Thành phần chính:

- Web Server: Xử lý yêu cầu HTTP từ người dùng, thường chạy trên Django framework với Apache hoặc Nginx.
- Dashboard Core: Cung cấp các panel chính để quản lý tài nguyên như máy ảo, volume, mạng, và hình ảnh.
- Plugins: Hỗ trợ mở rộng chức năng, tích hợp với các dịch vụ OpenStack hoặc bên thứ ba.
- API Client: Kết nối với API của các dịch vụ OpenStack (Nova, Cinder, Neutron, Glance) để thực thi yêu cầu.

Cơ chế quản lý:

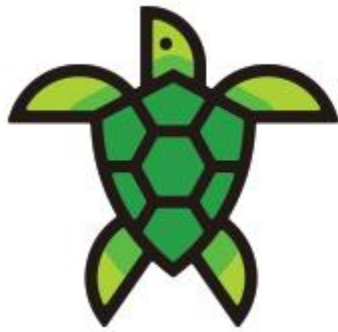
- Role-Based Access Control (RBAC): Phân quyền dựa trên vai trò (admin, user) để kiểm soát truy cập tài nguyên.
- Customization: Cho phép tùy chỉnh giao diện, thêm panel mới hoặc tích hợp dịch vụ bên ngoài qua plugin.
- Multi-Tenancy: Hỗ trợ quản lý nhiều dự án (tenant) trong một giao diện duy nhất.

Giao thức truy cập: Sử dụng giao diện web (HTTP/HTTPS) để tương tác, gửi yêu cầu đến API của các dịch vụ OpenStack thông qua API Client.

Quy trình hoạt động:

- Người dùng truy cập Horizon qua trình duyệt web, đăng nhập với thông tin xác thực từ Keystone (dịch vụ xác thực).
- Dashboard hiển thị các panel tương ứng với quyền truy cập của người dùng (máy ảo, mạng, lưu trữ).
- Người dùng thực hiện thao tác (tạo máy ảo, cấu hình mạng, gắn volume) qua giao diện web.
- Web Server chuyển yêu cầu đến API Client, gọi API của các dịch vụ liên quan (Nova, Neutron, Cinder).
- Kết quả được trả về và hiển thị trên giao diện, đảm bảo trải nghiệm quản lý liền mạch và trực quan.

### ***1.3.6 Keystone - Dịch vụ Xác thực trong OpenStack***



**KEYSTONE**  
*an OpenStack Community Project*

*Hình 8 Logo Keystone*

Keystone là dịch vụ xác thực thuộc hệ sinh thái OpenStack, chịu trách nhiệm quản lý danh tính, xác thực người dùng, và phân quyền truy cập vào các tài nguyên đám mây. Keystone cung cấp nền tảng bảo mật cốt lõi, đảm bảo các dịch vụ OpenStack như Nova, Cinder, và Neutron hoạt động an toàn và hiệu quả.

Đặc điểm nổi bật:

- Quản lý danh tính: Hỗ trợ xác thực người dùng, dịch vụ, và quản lý thông tin tài khoản.
- Phân quyền linh hoạt: Sử dụng vai trò và chính sách để kiểm soát truy cập tài nguyên.
- Hỗ trợ đa dự án: Quản lý nhiều dự án (tenant) trong môi trường đám mây.
- Tích hợp đa dạng: Hỗ trợ các giao thức xác thực như LDAP, OAuth, và OpenID Connect.

Thành phần chính:

- Identity Service: Quản lý thông tin người dùng, nhóm, và dự án, hỗ trợ tích hợp với hệ thống bên ngoài như LDAP.
- Authentication Service: Xác thực danh tính người dùng hoặc dịch vụ thông qua token hoặc thông tin đăng nhập.
- Authorization Service: Gán vai trò và chính sách để kiểm soát quyền truy cập vào tài nguyên OpenStack.
- Catalog Service: Cung cấp danh mục dịch vụ, chứa thông tin endpoint API của các dịch vụ OpenStack.

Cơ chế bảo mật:

- Token-based Authentication: Phát hành token cho người dùng hoặc dịch vụ sau khi xác thực, dùng để truy cập tài nguyên.
- Role-Based Access Control (RBAC): Gán vai trò (admin, user) để kiểm soát quyền truy cập dựa trên chính sách.

- Federation: Hỗ trợ xác thực liên kết với các hệ thống danh tính bên ngoài, cho phép đăng nhập một lần (SSO).

Giao thức truy cập: Sử dụng RESTful API (GET, POST, PUT, DELETE) để quản lý danh tính, vai trò, và dịch vụ, tích hợp với các giao diện như Horizon hoặc công cụ dòng lệnh.

Quy trình hoạt động:

- Thông tin được tổ chức dưới dạng: User (người dùng), Project (dự án), Role (vai trò), Service (dịch vụ), Endpoint (điểm truy cập API).
- Người dùng hoặc dịch vụ gửi yêu cầu xác thực (đăng nhập, token) qua API đến Keystone.
- Identity Service xác minh thông tin đăng nhập, Authentication Service phát hành token.
- Authorization Service kiểm tra vai trò và chính sách để cấp quyền truy cập tài nguyên.
- Catalog Service cung cấp danh sách endpoint để người dùng/dịch vụ tương tác với các dịch vụ OpenStack khác.

### ***1.3.7 Glance - Dịch vụ Hình ảnh trong OpenStack***



*Hình 9 Logo Glance*

Glance là dịch vụ hình ảnh thuộc hệ sinh thái OpenStack, chịu trách nhiệm lưu trữ, quản lý và cung cấp hình ảnh cho máy ảo hoặc container. Glance hỗ trợ triển khai nhanh

các máy ảo thông qua hình ảnh hệ điều hành hoặc ứng dụng, là thành phần cốt lõi cho dịch vụ tính toán như Nova.

Đặc điểm nổi bật:

- Quản lý hình ảnh linh hoạt: Lưu trữ, truy xuất, và cập nhật hình ảnh hệ điều hành hoặc ứng dụng.
- Tích hợp chặt chẽ: Hỗ trợ Nova, Cinder, và các dịch vụ khác để triển khai máy ảo hoặc lưu trữ dữ liệu.
- Hỗ trợ đa định dạng: Chấp nhận các định dạng hình ảnh như QCOW2, VMDK, ISO, và RAW.
- Khả năng mở rộng: Cho phép lưu trữ hình ảnh trên nhiều backend như Swift, Cinder, hoặc hệ thống tệp cục bộ.

Thành phần chính:

- API Server: Tiếp nhận và xử lý yêu cầu API từ người dùng, chuyển tiếp đến các dịch vụ lưu trữ hoặc quản lý hình ảnh.
- Registry Service: Quản lý metadata của hình ảnh, bao gồm thông tin như tên, định dạng, và thuộc tính.
- Storage Backend: Lưu trữ dữ liệu hình ảnh thực tế, hỗ trợ các backend như Swift, Cinder, hoặc hệ thống tệp (file system).
- Database: Lưu trữ thông tin metadata của hình ảnh để tra cứu và quản lý.

Cơ chế quản lý hình ảnh:

- Metadata Management: Gắn thẻ và thuộc tính cho hình ảnh để hỗ trợ tìm kiếm và lọc.
- Image Caching: Lưu trữ tạm hình ảnh trên node tính toán để tăng tốc độ triển khai máy ảo.
- Access Control: Kiểm soát quyền truy cập hình ảnh (public, private, hoặc shared) dựa trên dự án hoặc người dùng.

Giao thức truy cập: Sử dụng RESTful API (GET, POST, PUT, DELETE) để quản lý hình ảnh, tích hợp với giao diện như Horizon hoặc công cụ dòng lệnh.

Quy trình hoạt động:

- Hình ảnh được tổ chức dưới dạng: Image (tệp hình ảnh hệ điều hành/ứng dụng), Metadata (thuộc tính mô tả), Storage Backend (nơi lưu trữ dữ liệu).
- Người dùng gửi yêu cầu (tải lên, truy xuất, xóa hình ảnh) qua API đến API Server.
- Registry Service ghi hoặc truy xuất metadata từ Database, trong khi Storage Backend xử lý dữ liệu hình ảnh.

- Hình ảnh được cung cấp cho Nova để triển khai máy ảo, hoặc lưu trữ trên Swift/Cinder để sao lưu.

### ***1.3.8 Ironic - Dịch vụ Triển khai Phần cứng trong OpenStack***



*Hình 10 Logo Ironic*

Ironic là dịch vụ triển khai phần cứng (Bare Metal Provisioning) thuộc hệ sinh thái OpenStack, cho phép quản lý và triển khai máy chủ vật lý thay vì máy ảo, đáp ứng nhu cầu ứng dụng yêu cầu hiệu suất cao hoặc môi trường không ảo hóa. Ironic cung cấp khả năng quản lý vòng đời phần cứng, từ khởi tạo đến thu hồi, như máy ảo trong Nova.

Đặc điểm nổi bật:

- Quản lý phần cứng linh hoạt: Triển khai, cấu hình, và thu hồi máy chủ vật lý với quy trình tự động.
- Hiệu suất tối ưu: Cung cấp tài nguyên phần cứng trực tiếp, phù hợp cho ứng dụng tính toán hiệu năng cao.
- Tích hợp chặt chẽ: Làm việc với Nova, Neutron, và Glance để sử dụng hình ảnh và mạng như máy ảo.
- Hỗ trợ đa dạng phần cứng: Tương thích với nhiều loại máy chủ thông qua driver phần cứng.

Thành phần chính:

- API Server: Tiếp nhận và xử lý yêu cầu API từ người dùng, chuyển tiếp đến các dịch vụ liên quan.
- Conductor: Quản lý vòng đời phần cứng, điều phối thao tác triển khai, kiểm tra, và thu hồi.

- Drivers: Giao tiếp với phần cứng thông qua chuẩn như IPMI, iLO, hoặc Redfish để kiểm soát nguồn, khởi động, và cấu hình.
- Database: Lưu trữ thông tin trạng thái và cấu hình của máy chủ vật lý.

Cơ chế quản lý phần cứng:

- Hardware Inspection: Tự động phát hiện và thu thập thông tin phần cứng (CPU, RAM, ổ cứng) khi đăng ký máy chủ.
- Deployment: Triển khai hình ảnh từ Glance lên máy chủ vật lý, sử dụng mạng từ Neutron.
- Power Management: Kiểm soát trạng thái nguồn (bật, tắt, khởi động lại) thông qua driver phần cứng.

Giao thức truy cập: Sử dụng RESTful API (GET, POST, PUT, DELETE) để quản lý máy chủ vật lý, tích hợp với Horizon hoặc công cụ dòng lệnh.

Quy trình hoạt động:

- Máy chủ vật lý được tổ chức dưới dạng: Node (máy chủ), Port (giao diện mạng), Image (hình ảnh hệ điều hành).
- Người dùng gửi yêu cầu (triển khai, cấu hình, thu hồi) qua API đến API Server.
- Conductor sử dụng driver để kiểm tra phần cứng, tải hình ảnh từ Glance, và cấu hình mạng qua Neutron.
- Hình ảnh được triển khai lên máy chủ vật lý, kết nối với mạng và lưu trữ (nếu cần) từ Cinder.
- Hệ thống hỗ trợ kiểm tra trạng thái và quản lý vòng đời để đảm bảo vận hành liên tục và hiệu quả.

## 1.4 Các giải pháp NSM được triển khai trong thực tế

## CHƯƠNG 2. TRIỂN KHAI HỆ THỐNG OPENSTACK

### 2.1 Giới thiệu chương

Ở chương này sẽ trình bày các bước triển khai, cấu hình hệ thống OpenStack với 2 mục đích khác nhau:

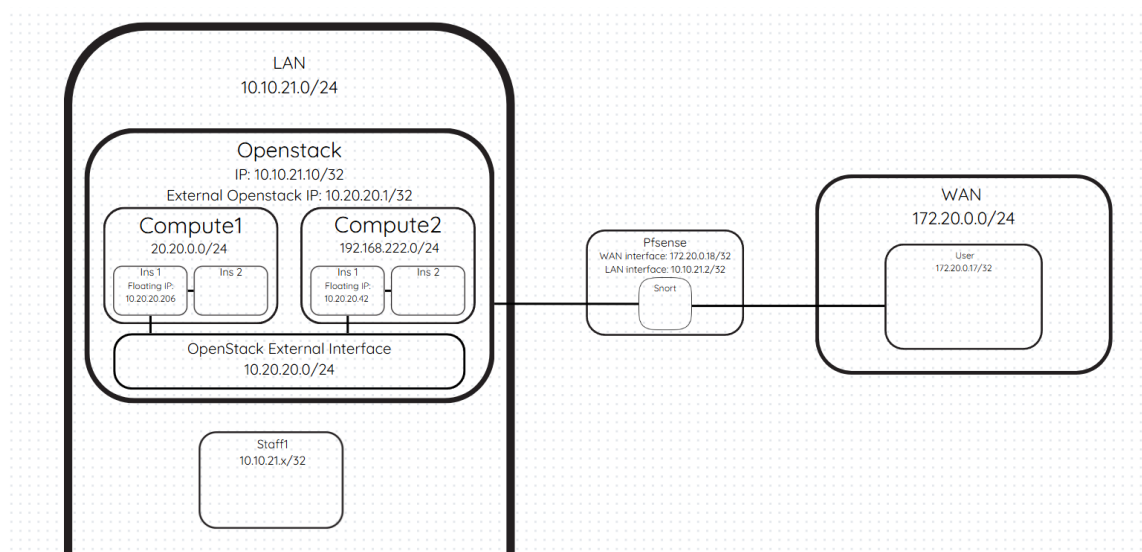
- Hệ thống OpenStack triển khai trên mạng vật lý thực
- Hệ thống OpenStack triển khai trên mạng ảo và triển khai nhiều cụm máy ảo khác nhau.

Chương này sẽ trình bày rõ ràng mô hình mạng được triển khai, các bước cấu hình hệ thống, các luật theo dõi trên giải pháp Snort cũng như demo những tác động lên hệ thống OpenStack để kiểm tra hoạt động của công cụ Snort.

### 2.2 Demo hệ thống triển khai mạng ảo trên OpenStack

#### 2.2.1 Mô hình hệ thống

Hệ thống xây dựng giả định sẽ cung cấp 2 cụm máy ảo cho 2 khách hàng riêng biệt. Máy chủ chạy OpenStack sẽ được đặt sau tường lửa chạy công cụ Snort để theo dõi các gói tin đi vào các instance.



Hình 11 Sơ đồ mạng hệ thống

Các máy user từ ngoài mạng WAN muốn truy cập vào các Instance trong OpenStack sẽ đi qua route như sau:

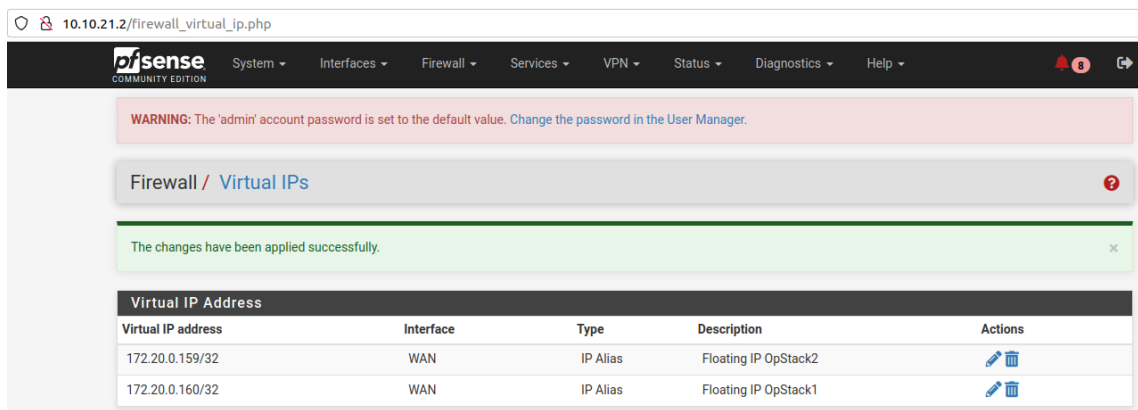
- Gói tin TCP gửi đến PfSense qua floating IP tạo bởi tường lửa
- Tường lửa chuyển tiếp gói tin TCP qua máy chủ OpenStack.
- Máy chủ OpenStack chuyển tiếp gói tin TCP đến Instance.

#### 2.2.2 Quá trình cài đặt

##### 2.2.2.1 Cài đặt tường lửa PfSense.

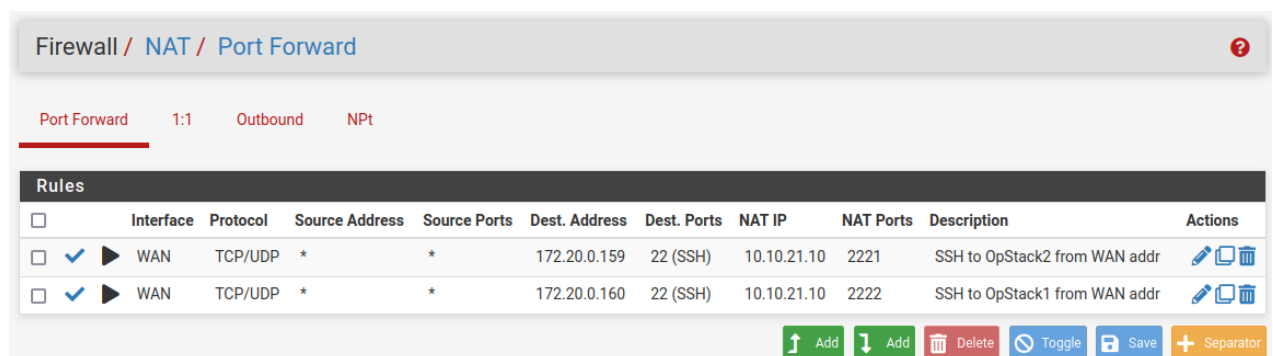
- Ta cần tạo các floating IP trên PfSense để kết nối instance với WAN interface





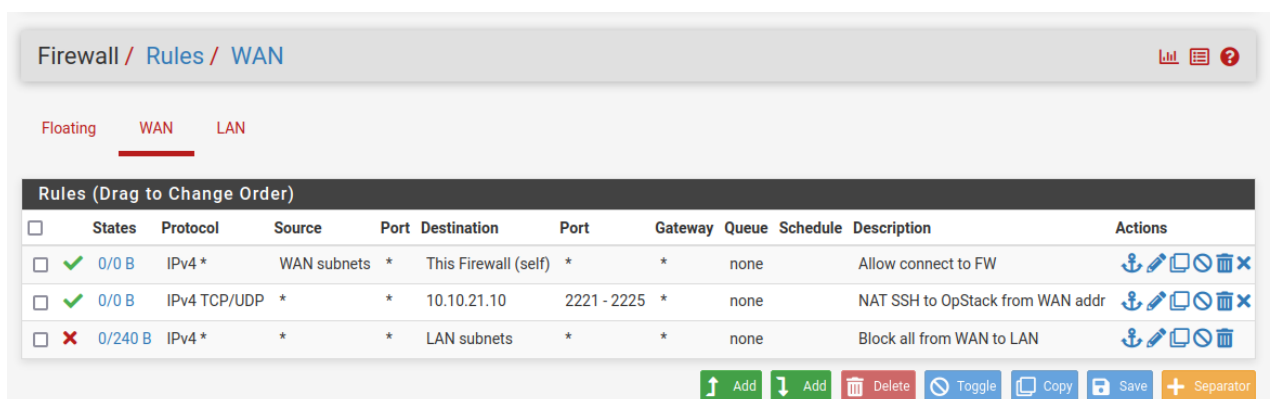
Hình 12 Cấu hình floating IP trên PfSense

- Tiếp theo ta cấu hình NAT forwarding để chuyển tiếp gói tin đến máy chủ OpenStack trên LAN interface, gửi gói tin cụ thể đến port cố định cấu hình riêng cho từng instance.



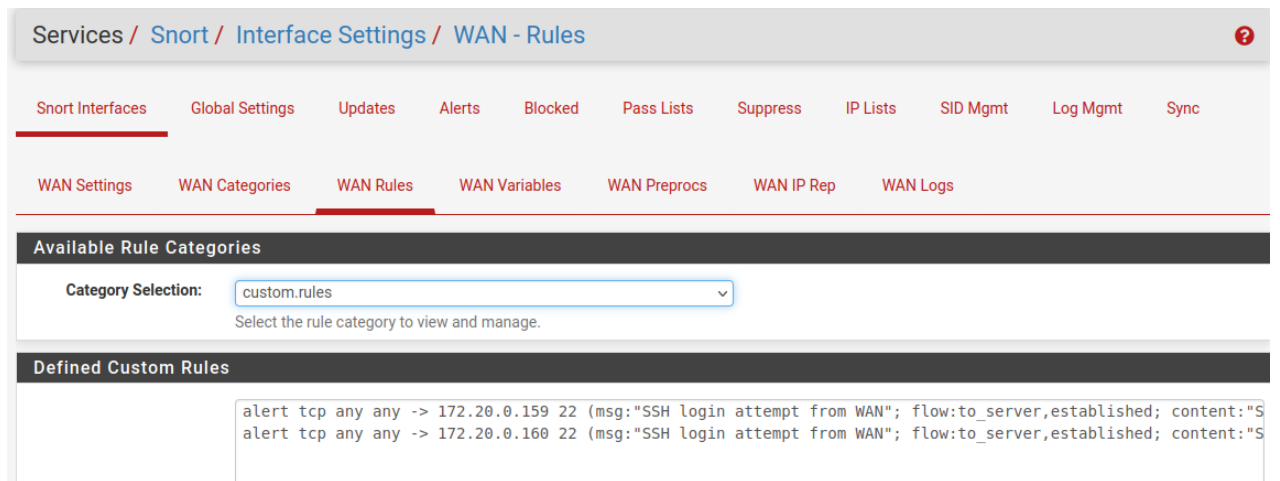
Hình 13 Cấu hình NAT forwarding trên PfSense

- Tiếp theo ta cần cấu hình rule để cho phép máy từ ngoài WAN có thể truy cập được và chặn các truy cập khác.

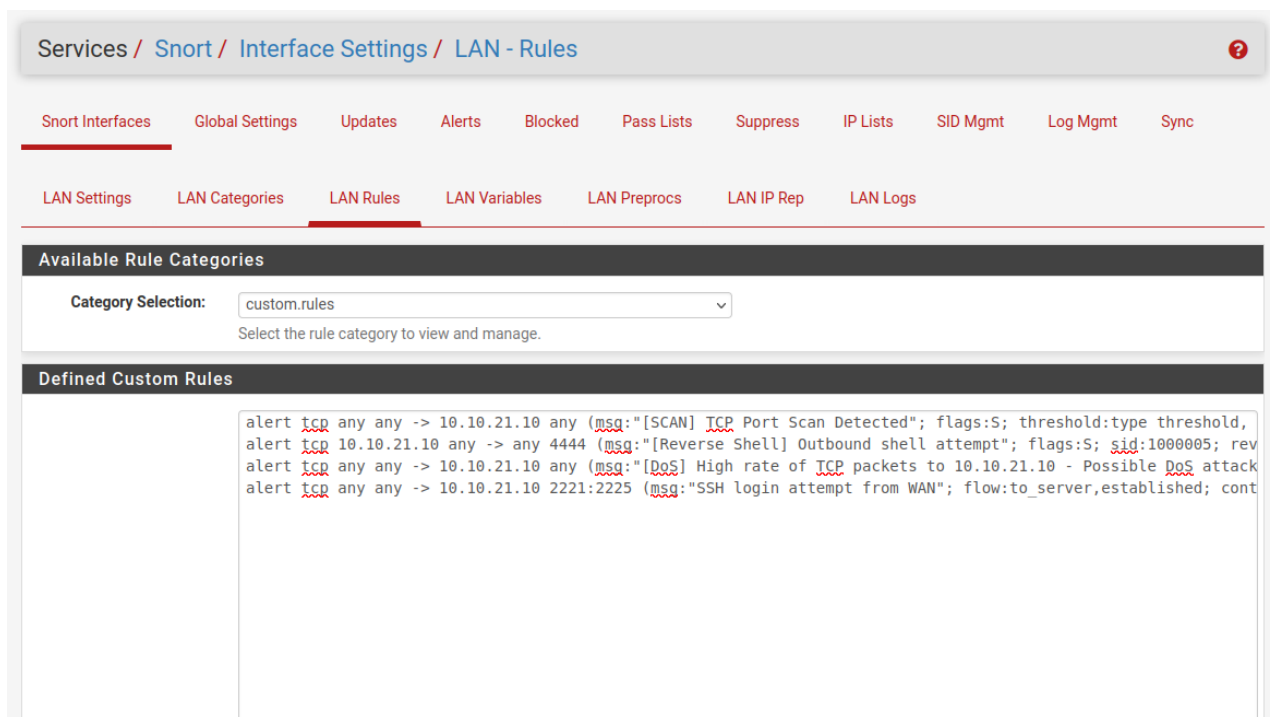


Hình 14 Cấu hình rule WAN trong PfSense

- Cấu hình các rule Snort trên tường lửa để bắt các gói tin truy cập đến các instance OpenStack.



Hình 15 Rule Snort trên WAN Interface



Hình 16 Rule Snort trên LAN interface

#### 2.2.2.2 Cài đặt máy chủ OpenStack

Hệ thống này sẽ sử dụng phiên bản OpenStack đơn giản và nhẹ hơn được cung cấp bởi Canonical (hãng phát triển Ubuntu) và được phân phối qua Snap.

- Để cài đặt hệ thống, ta chạy lệnh:

```
sudo snap install microstack --beta --devmode
```

- Tiếp theo, ta cài đặt openstack tự động theo mặc định bằng lệnh

```
sudo microstack init --auto --control
```

- Sau khi cài đặt xong hệ thống OpenStack, chạy lệnh sau để lấy mật khẩu người dùng OpenStack

```

trung@trung-virtual-machine:~$ sudo snap get microstack config.credentials.keystone-password
[sudo] password for trung:
j49YR03nXotu48b9596mSNDXaVBFrKzd
trung@trung-virtual-machine:~$

```

Hình 17 Lấy mật khẩu admin OpenStack

- Sau đó, trên trình duyệt truy cập vào ip của máy triển khai OpenStack, tài khoản mặc định là admin với mật khẩu là dãy kí tự lấy được ở trên.

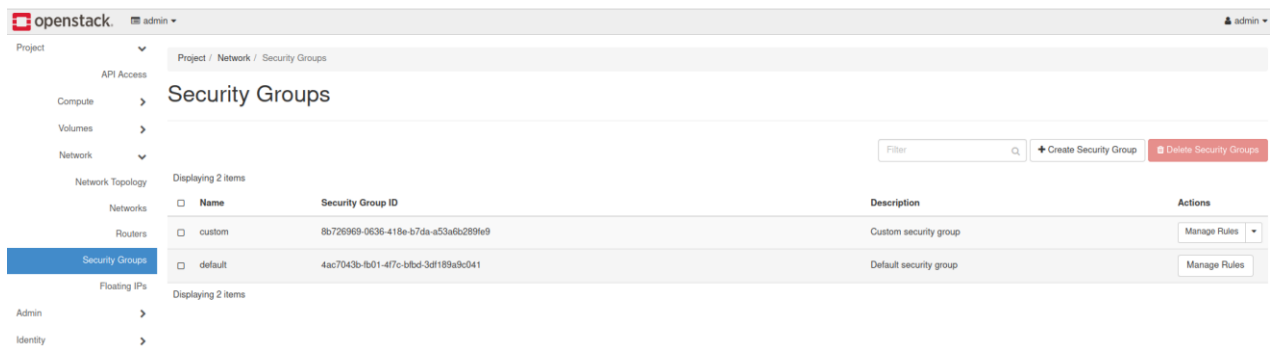
Hình 18 Giao diện đăng nhập OpenStack

- Sau khi đăng nhập vào giao diện quản lý OpenStack, ta tiến hành cấu hình mạng cho hệ thống OpenStack

Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
internal	internal-subnet 20.20.0.0/24	No	No	Active	UP	-	Edit Network
test	test-subnet 192.168.222.0/24	No	No	Active	UP	-	Edit Network
external	external-subnet 10.20.20.0/24	No	Yes	Active	UP	-	Edit Network

Hình 19 Các dải mạng được cấu hình

- Vì có 2 người dùng tách biệt nên ta cũng cần tạo 2 security group riêng biệt cho từng cụm instance để tách biệt chúng.



Hình 20 Các security group

- Cấu hình các rule trong security group. Ở đây ta cho phép tất cả gói tin đi ra (Egress) và cho phép tất cả các gói tin đi vào (Ingress) với các instance cùng security group (Các instance cùng cụm). Ngoài ra ta cho phép gói tin TCP đến cổng 22 từ 10.20.20.1/32 (IP của máy chủ OpenStack trên External Network của hệ thống mạng OpenStack)

Displaying 6 items								<a href="#">+ Add Rule</a>	<a href="#">Delete Rules</a>
<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions	
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	<a href="#">Delete Rule</a>	
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	-	<a href="#">Delete Rule</a>	
<input type="checkbox"/>	Ingress	IPv4	Any	Any	-	custom	-	<a href="#">Delete Rule</a>	
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	-	<a href="#">Delete Rule</a>	
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	10.20.20.1/32	-	-	<a href="#">Delete Rule</a>	
<input type="checkbox"/>	Ingress	IPv6	Any	Any	-	custom	-	<a href="#">Delete Rule</a>	
Displaying 6 items									

Hình 21 Cấu hình rule trong security group

- Cấu hình router để kết nối mạng internal của cụm instance ra ngoài mạng external

openstack

admin

Project

API Access

Compute

Volumes

Network

Network Topology

Networks

Routers

Security Groups

Floating IPs

Admin

Identity

admin

Project / Network / Routers

Routers

Router Name  Filter [+ Create Router](#) [Delete Routers](#)

Displaying 2 items

<input type="checkbox"/>	Name	Status	External Network	Admin State	Availability Zones	Actions
<input type="checkbox"/>	test-router	Active	external	UP	-	<a href="#">Clear Gateway</a>
<input type="checkbox"/>	internal-router	Active	external	UP	-	<a href="#">Clear Gateway</a>

Displaying 2 items

Hình 22 Cấu hình router OpenStack

- Cấu hình interface tương ứng cho các router



Hình 23 Cấu hình interface cho các router để kết nối mạng internal với external

- Tiến hành tạo các instance trên OpenStack, lưu ý cần xác định rõ dải mạng và security group của các instance. Ở đây có 2 cụm instance, mỗi cụm có 2 instance và chỉ có một instance mỗi cụm được cấp floating IP để kết nối với bên ngoài.

openstack

admin

Project

API Access

Compute

Overview

Instances

Images

Key Pairs

Server Groups

Volumes

Network

Admin

Identity

Project / Compute / Instances

Instances

Displaying 4 items

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions	
<input type="checkbox"/>	test-2	cirros	192.168.222.87, 10.20.20.42	m1.tiny	-	Shutoff	us-east-1	nova	None	Shut Down	1 week, 3 days	Start Instance
<input type="checkbox"/>	test-1	cirros	192.168.222.220	m1.tiny	-	Shutoff	us-east-1	nova	None	Shut Down	1 week, 3 days	Start Instance
<input type="checkbox"/>	internal-2	cirros	20.20.0.155, 10.20.20.206	m1.tiny	-	Shutoff	us-east-1	nova	None	Shut Down	1 week, 3 days	Start Instance
<input type="checkbox"/>	internal-1	cirros	20.20.0.131	m1.tiny	-	Shutoff	us-east-1	nova	None	Shut Down	1 week, 3 days	Start Instance

Displaying 4 items

Instance ID

Filter

Launch Instance

Delete Instances

More Actions

Hình 24 Các instance trong hệ thống

- Tiến hành cấu hình Port Forwarding trên máy chủ OpenStack, cấu hình đường vào qua port cố định đến floating IP của instance cổng 22 để cho phép SSH và đường ra cấu hình đi từ floating IP của instance ra đến IP của máy chủ OpenStack trên External Network (10.20.20.1), ở đây ta sử dụng các lệnh:

```
sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport <port forward> -j DNAT --to-destination <Instance floating IP>:22
```

```
sudo iptables -t nat -A POSTROUTING -d <Instance floating IP> -p tcp --dport 22 -j SNAT --to-source 10.20.20.1
```

```

trung@trung-virtual-machine:~$ sudo iptables -t nat -L -n -v
[sudo] password for trung:
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain PREROUTING (policy ACCEPT 6 packets, 1172 bytes)
  pkts bytes target     prot opt in     out     source    destination
    4   240 DNAT      tcp  --  ens33  *      0.0.0.0/0 0.0.0.0/0      tcp dpt:2222 to:10.20.20.206:22
    2   120 DNAT      tcp  --  ens33  *      0.0.0.0/0 0.0.0.0/0      tcp dpt:2221 to:10.20.20.42:22

Chain INPUT (policy ACCEPT 6 packets, 1172 bytes)
  pkts bytes target     prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 21664 packets, 1491K bytes)
  pkts bytes target     prot opt in     out     source    destination

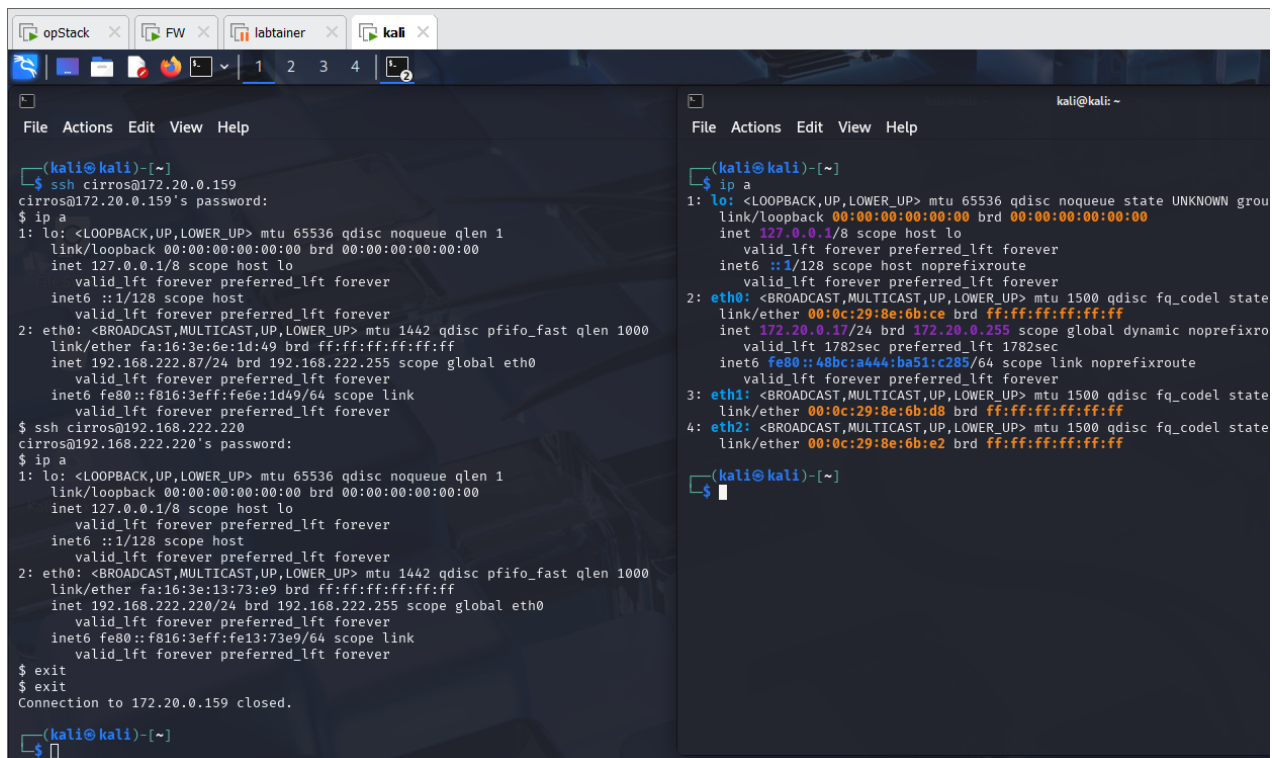
Chain POSTROUTING (policy ACCEPT 21653 packets, 1490K bytes)
  pkts bytes target     prot opt in     out     source    destination
    4   240 SNAT      tcp  --  *      *      0.0.0.0/0 10.20.20.206    tcp dpt:22 to:10.20.20.1
    2   120 SNAT      tcp  --  *      *      0.0.0.0/0 10.20.20.42     tcp dpt:22 to:10.20.20.1
trung@trung-virtual-machine:~$

```

Hình 25 Cấu hình Port Forwarding trên máy chủ OpenStack

### 2.2.3 Triển khai hệ thống

- Trước tiên, ta vào vị trí client truy cập từ ngoài WAN vào OpenStack, client có thể SSH vào instance qua các floating IP 172.20.0.159 và 172.20.0.160. Từ máy instance có floating IP SSH đến các instance còn lại trong cụm máy ảo.



```

(kali@kali)-[~]
└─$ ssh cirros@172.20.0.159
cirros@172.20.0.159's password:
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1442 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:6e:1d:49 brd ff:ff:ff:ff:ff:ff
    inet 192.168.222.87/24 brd 192.168.222.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe6e:1d49/64 scope link
        valid_lft forever preferred_lft forever
$ ssh cirros@192.168.222.220
cirros@192.168.222.220's password:
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1442 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:13:73:e9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.222.220/24 brd 192.168.222.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe13:73e9/64 scope link
        valid_lft forever preferred_lft forever
$ exit
$ exit
Connection to 172.20.0.159 closed.
(kali@kali)-[~]
└─$

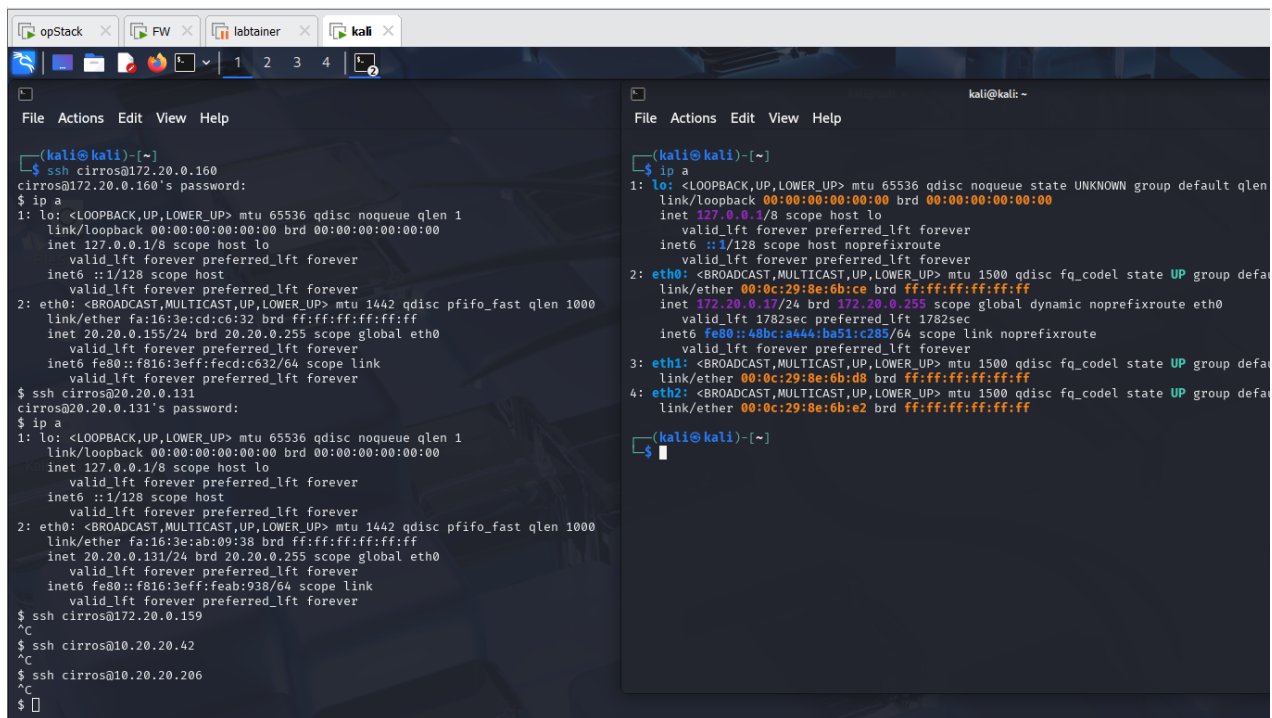
```

```

(kali@kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
    link/ether 00:0c:29:8e:6b:ce brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.17/24 brd 172.20.0.255 scope global dynamic noprefixro
        valid_lft 1782sec preferred_lft 1782sec
    inet6 fe80::48bc:a444:ba51:c285/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
    link/ether 00:0c:29:8e:6b:d8 brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
    link/ether 00:0c:29:8e:6b:e2 brd ff:ff:ff:ff:ff:ff
(kali@kali)-[~]
└─$

```

Hình 26 SSH từ WAN vào instance trên dải 192.168.222.0/24



Hình 27 SSH từ WAN vào instance trên dải 20.20.0.0/24

Services / Snort / Alerts

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Alert Log View Settings

Interface to Inspect: WAN (em0) Auto-refresh view: ☒ 250 Save

Alert Log Actions: Download Clear

Alert Log View Filter

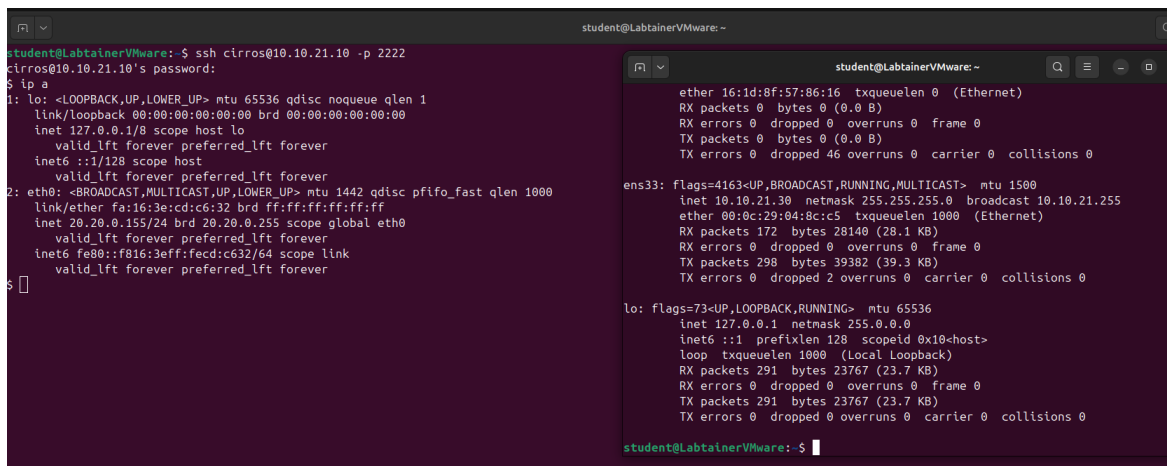
5 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-05-05 04:57:58	⚠	1	TCP	Attempted User Privilege Gain	172.20.0.17	42994	172.20.0.160	22	1:1000055	SSH login attempt from WAN
2025-05-05 04:57:49	⚠	1	TCP	Attempted User Privilege Gain	172.20.0.17	46784	172.20.0.160	22	1:1000055	SSH login attempt from WAN
2025-05-05 04:51:04	⚠	1	TCP	Attempted User Privilege Gain	172.20.0.17	47254	172.20.0.159	22	1:1000001	SSH login attempt from WAN
2025-05-05 04:50:08	⚠	1	TCP	Attempted User Privilege Gain	172.20.0.17	55268	172.20.0.159	22	1:1000001	SSH login attempt from WAN
2025-05-05 04:50:00	⚠	1	TCP	Attempted User Privilege Gain	172.20.0.17	55262	172.20.0.159	22	1:1000001	SSH login attempt from WAN

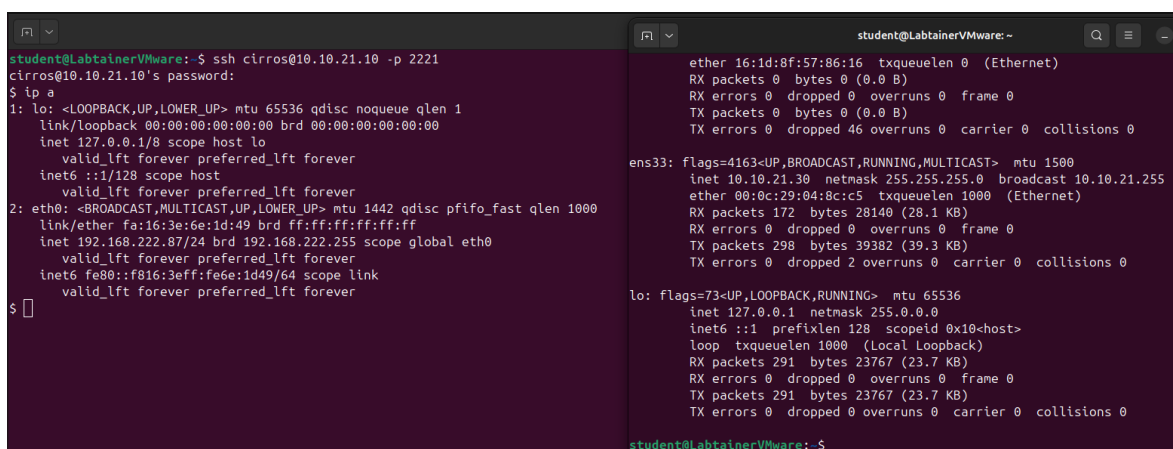
Hình 28 Alert trên Snort theo dõi các cố gắng SSH đến instance từ WAN

- Tiếp theo ta vào vai staff nằm trong LAN interface, người dùng này sẽ truy cập vào instance.





Hình 29 SSH từ LAN vào instance trên dải 20.20.0.0/24



Hình 30 SSH từ LAN vào instance trên dải 192.168.222.0/24

Services / Snort / Alerts

?

Snort Interfaces

Global Settings

Updates

Alerts

Blocked

Pass Lists

Suppress

IP Lists

SID Mgmt

Log Mgmt

Sync

Alert Log View Settings

Interface to Inspect

LAN (em1)

Choose interface..

☒ Auto-refresh view

250

Alert lines to display.

Save

Alert Log Actions

Download

Clear

Alert Log View Filter

+

2 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-05-05 06:01:29	⚠	1	TCP	Attempted User Privilege Gain	10.10.21.30	60954	10.10.21.10	2221	1:1000013	SSH login attempt from LAN
2025-05-05 06:01:23	⚠	1	TCP	Attempted User Privilege Gain	10.10.21.30	41378	10.10.21.10	2222	1:1000013	SSH login attempt from LAN

Hình 31 Alert trên Snort theo dõi các cố gắng SSH đến instance từ LAN

- Tiếp theo, ta tiến hành kiểm thử từ máy staff vào máy chủ OpenStack



```

student@LabtainerVMware: ~
student@LabtainerVMware:~$ sudo hping3 -S -p 80 --flood 10.10.21.10
[sudo] password for student:
HPING 10.10.21.10 (ens33 10.10.21.10): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.10.21.10 hping statistic ---
1578309 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
student@LabtainerVMware:~$ nmap -p- -T4 -Pn 10.10.21.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-05 07:27 PDT
Nmap scan report for 10.10.21.10
Host is up (0.00097s latency).
Not shown: 65508 closed tcp ports (conn-refused)
PORT      STATE SERVICE
443/tcp    open  https
2221/tcp   open  rockwell-csp1
2222/tcp   open  EthernetIP-1
3306/tcp   open  mysql
4369/tcp   open  epmd
5000/tcp   open  upnp
5672/tcp   open  amqp
5900/tcp   open  vnc
5901/tcp   open  vnc-1

```

Hình 32 Tiến hành tấn công DOS và quét cổng trên máy chủ OpenStack

Alert Log View Settings

Interface to Inspect

LAN (em1)

Choose interface..

☒ Auto-refresh view

250

Alert lines to display.

Save

Alert Log Actions

Download

Clear

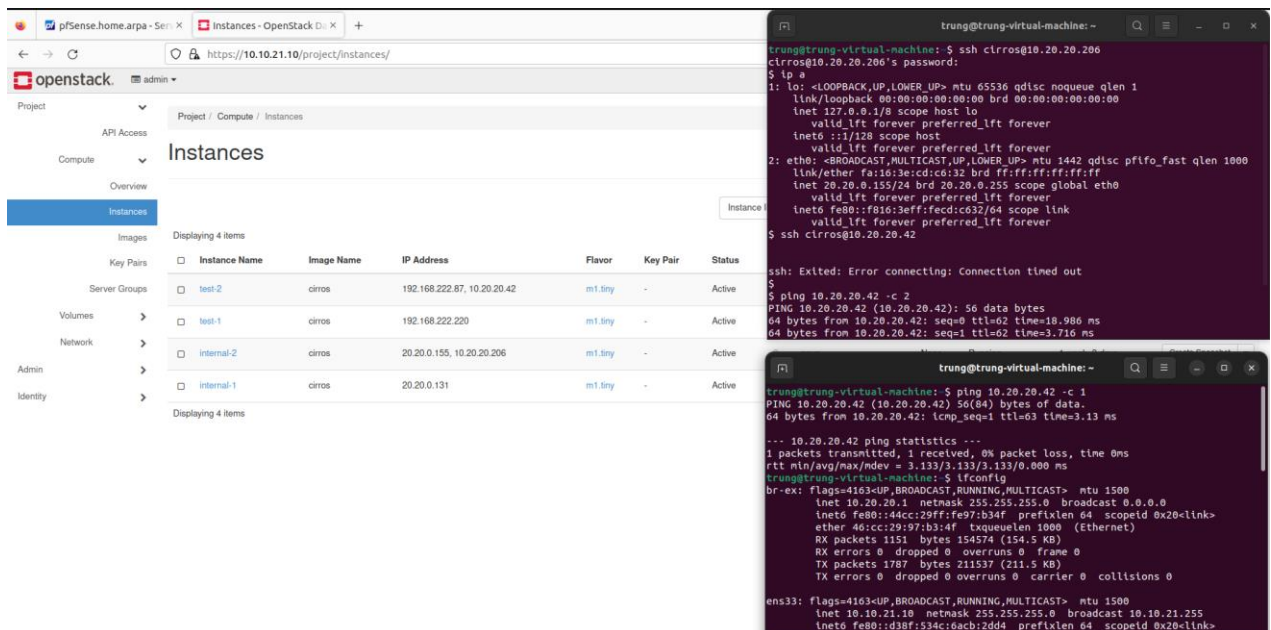
Alert Log View Filter

Most Recent 250 Entries from Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-05-05 06:05:10		0	TCP		10.10.21.10	35652	10.10.21.30	4444	1:1000005	[Reverse Shell] Outbound shell attempt
2025-05-05 06:04:45		0	TCP		10.10.21.10	49850	10.10.21.30	4444	1:1000005	[Reverse Shell] Outbound shell attempt
2025-05-05 06:04:10		0	TCP		10.10.21.30	51212	10.10.21.10	16018	1:1000002	[SCAN] TCP Port Scan Detected
2025-05-05 06:04:10		0	TCP		10.10.21.30	51212	10.10.21.10	16018	1:1000012	[DoS] High rate of TCP packets to 10.10.21.10 - Possible DoS attack
2025-05-05 06:04:10		0	TCP		10.10.21.30	35230	10.10.21.10	15445	1:1000012	[DoS] High rate of TCP packets to 10.10.21.10 - Possible DoS attack
2025-05-05 06:04:10		0	TCP		10.10.21.30	55262	10.10.21.10	19303	1:1000012	[DoS] High rate of TCP packets to 10.10.21.10 - Possible DoS attack
2025-05-05		0	TCP		10.10.21.30	38328	10.10.21.10	17227	1:1000012	[DoS] High rate of TCP packets to 10.10.21.10 - Possible

Hình 33 Alert trên Snort theo dõi các cố gắng tấn công đến máy chủ OpenStack

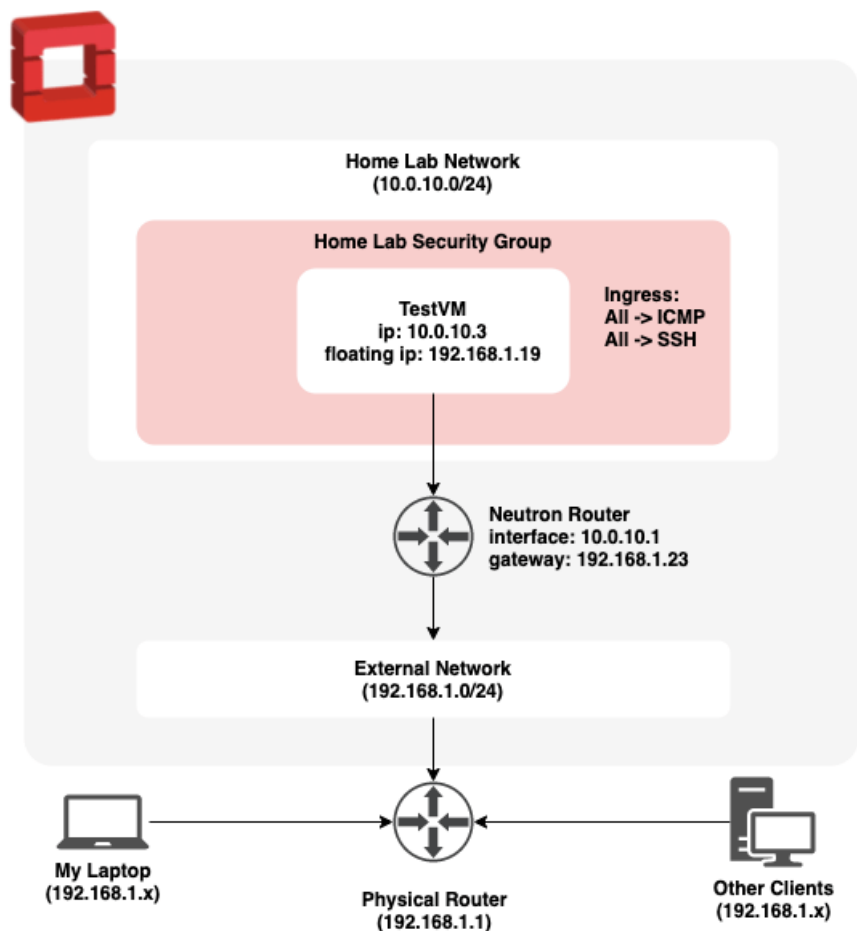
- Ta tiến hành thử SSH giữa 2 instance được cấp floating IP, từ đó ta nhận thấy các cụm instance đã được tách biệt. Ở đây ta thấy, theo rule của sec rule thì ta chỉ có thể ping giữa các instance để kiểm tra xem nó có thực sự kết nối được với nhau không. Ngoài ra các instance khác sec group không thể SSH được vào nhau mà chỉ cho phép SSH qua máy chủ OpenStack

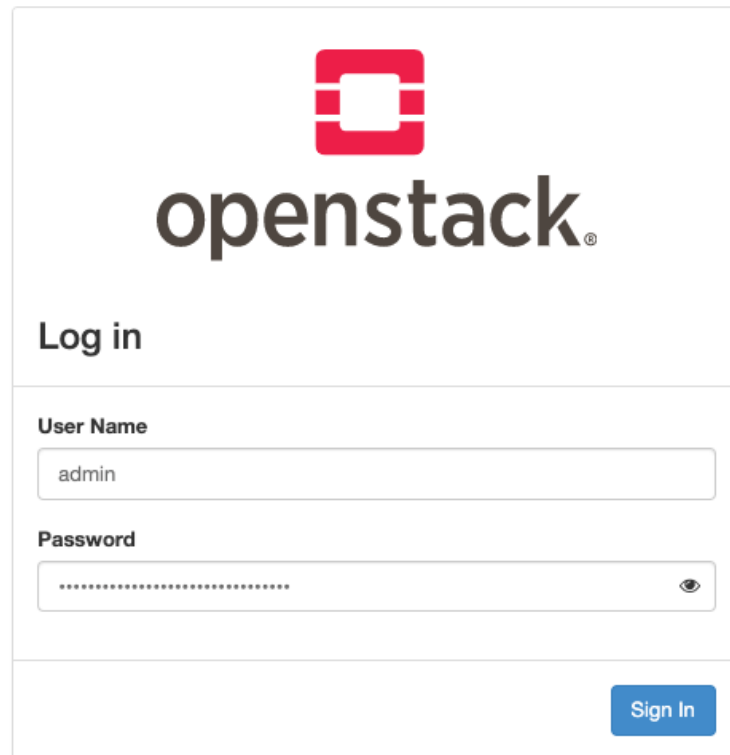



Hình 34 Kết nối giữa 2 instance khác security group

## 2.3 Demo hệ thống OpenStack triển khai trên mạng vật lý thật trên LAN

### 2.3.1 Cấu hình trước khi cài đặt






  
**openstack®**

**Log in**

---

**User Name**

**Password**



[Sign In](#)

Lấy mật khẩu bằng lệnh

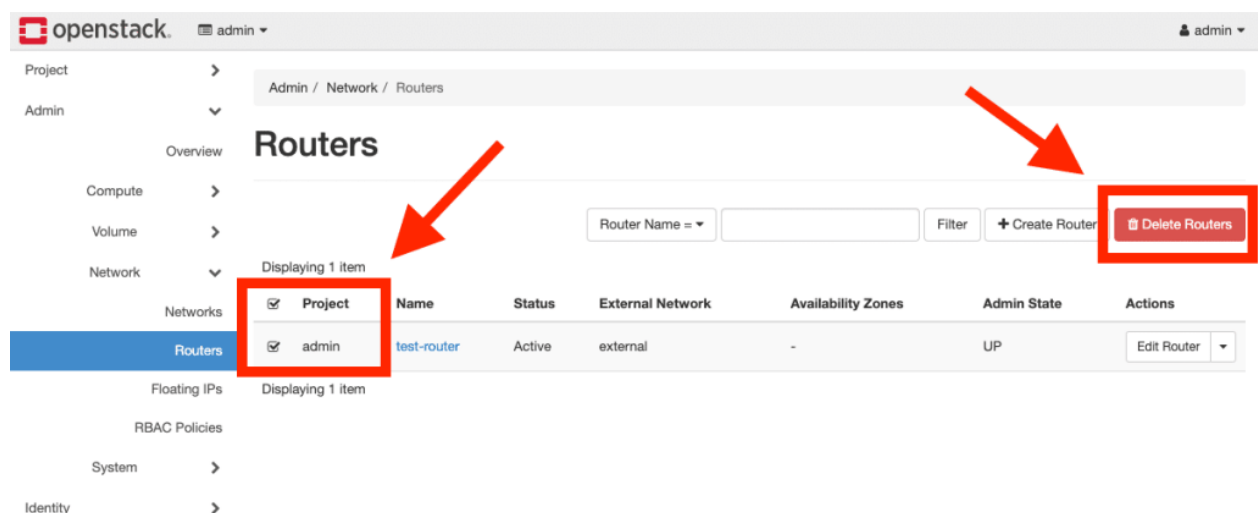
```
sudo snap get microstack config.credentials.keystone-password
```

## 2.3.2 Quá trình cài đặt

### 2.3.2.1 4. Xóa các mạng đang tồn tại

Để thay thế mạng ngoài mặc định, trước tiên ta cần phải xóa mạng đó. Ta cũng sẽ xóa mạng thử nghiệm vì nó không còn cần thiết nữa.

Cấu hình mặc định có một bộ định tuyến được kết nối với các mạng. Do đó, ta cần xóa nó trước khi xóa các mạng. Tìm “test-router” trong Admin > Network > Routers và xóa nó.



The screenshot shows the OpenStack Admin interface for the 'Routers' section. The breadcrumb trail is 'Admin / Network / Routers'. The left sidebar shows the navigation menu with 'Routers' selected under the 'Network' category. The main content area displays a table of routers. A red box highlights the 'Project' column header, and another red box highlights the 'Delete Routers' button. A red arrow points from the 'Delete Routers' button to the 'test-router' entry in the table.

Project	Name	Status	External Network	Availability Zones	Admin State	Actions
admin	test-router	Active	external	-	UP	Edit Router

Hình 35 aaaaa

Bây giờ hãy vào Admin > Network > Networks và xóa các mạng hiện có.

The screenshot shows the OpenStack Admin interface. The left sidebar has a menu with 'Networks' selected. The main content area is titled 'Networks' and shows a table of networks. Two red arrows point to specific elements: one points to the 'Delete Networks' button in the top right, and the other points to the first two rows of the network table. The table has columns for Project, Network Name, Subnets Associated, DHCP Agents, Shared, External, Status, Admin State, Availability Zones, and Actions.

Project	Network Name	Subnets Associated	DHCP Agents	Shared	External	Status	Admin State	Availability Zones	Actions
admin	external	external-subnet 10.20.20.0/24	0	No	Yes	Active	UP	-	Edit Network
admin	test	test-subnet 192.168.222.0/24	0	No	No	Active	UP	-	Edit Network

## 5. Cấu hình Mạng Vật lý

Mạng vật lý mới sẽ cho phép các mạng nội bộ kết nối với thế giới bên ngoài. Do đó, ta sẽ đặt tên nó là “public”. Vào Admin > Networks > Network và nhấp vào nút “Create Network”. Sử dụng thông tin tương tự như hình ảnh dưới đây.

## Create Network



Network \*

Subnet

Subnet Details

Name

public

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Project \*

admin

Provider Network Type \* ?

Flat

Physical Network \* ?

physnet1

☒ Enable Admin State ?

☐ Shared

☒ External Network

☒ Create Subnet

Availability Zone Hints ?

Cancel

« Back

Next »

Lưu ý: Tên mạng vật lý mặc định của neutron cho MicroStack là 'physnet1'. Sử dụng lệnh dưới đây nếu muốn kiểm tra lại cấu hình này.

Đối với cấu hình mạng con, vui lòng thay thế Địa chỉ Mạng và Địa chỉ Cổng tương ứng cho cấu hình mạng vật lý.

## Create Network



Network \*

Subnet

Subnet Details

### Subnet Name

public-subnet

### Network Address ?

192.168.1.0/24

### IP Version

IPv4

### Gateway IP ?

192.168.1.1

☐ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel

« Back

Next »

Tắt DHCP và thêm phạm vi IP từ mạng LAN sẽ được dành riêng cho OpenStack

## Create Network



Network \*

Subnet

Subnet Details

☐ Enable DHCP

Specify additional attributes for the subnet.

### Allocation Pools ?

192.168.1.11,192.168.1.60

### DNS Name Servers ?

### Host Routes ?

Cancel

« Back

Create

Kiểm tra xem máy chủ DHCP có phạm vi địa chỉ IP không xung đột với những địa chỉ được phân bổ cho OpenStack.

## 6. Cấu hình Home Lab Network

Truy cập Admin > Networks > Network, chọn “Create Network” và áp dụng các cài đặt từ hình ảnh dưới đây.

### Create Network ✕

---

Network \*

Subnet

Subnet Details

**Name**

home-lab-network

**Project** \*

admin

**Provider Network Type** \* ?

Local

☒ **Enable Admin State** ?

☐ **Shared**

☐ **External Network**

☒ **Create Subnet**

**Availability Zone Hints** ?

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Cancel

« Back

Next »

Đặt tên cho mạng con và xác định khối CIDR tương ứng.

## Create Network



Network \*

Subnet

Subnet Details

### Subnet Name

home-lab-subnet

### Network Address ?

10.0.10.0/24

### IP Version

IPv4

### Gateway IP ?

☐ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel

« Back

Next »

Giữ tùy chọn DHCP được chọn và nhấp vào nút "Create".



## Create Network



Network \*

Subnet

Subnet Details

☒ Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ?

DNS Name Servers ?

Host Routes ?

Cancel

« Back

Create

Kết nối mạng Home Lab Network với mạng công cộng. Truy cập Admin > Network > Routers, nhấp vào “Create Router” và sử dụng thông tin từ hình ảnh.

## Create Router



Router Name

public-router

### Description:

Creates a router with specified parameters.

Project \*

admin

Enable SNAT will only have an effect if an external network is set.

☒ Enable Admin State ?

External Network

public

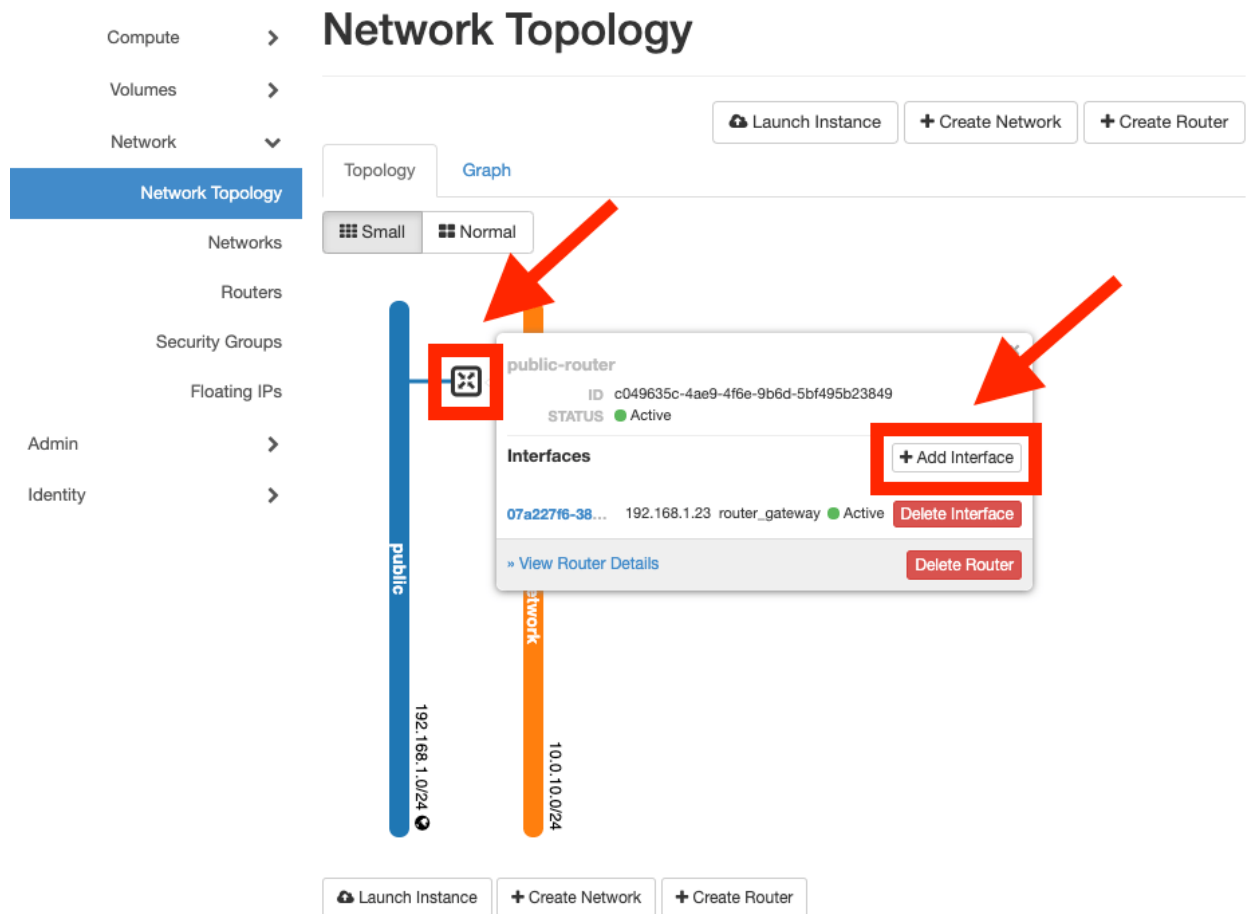
☒ Enable SNAT

Availability Zone Hints ?

Cancel

Create Router

Truy cập Project > Network > Network Topology, bộ định tuyến sẽ hiển thị kết nối với mạng công cộng. Kết nối bộ định tuyến với Home Lab Network bằng cách nhấp vào bộ định tuyến, sau đó chọn nút “Add Interface”.



Chọn “home-lab-network” từ danh sách thả xuống và nhấp vào “Submit”.

## Add Interface

**Subnet \***

home-lab-network: 10.0.10.0/24 (home-lab-s... ▾)

**IP Address (optional) ⓘ**

**Description:**

You can connect a specified subnet to the router.  
If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router.  
If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Cancel

Submit

Cấu trúc mạng kết quả sẽ trông giống như hình dưới đây.

Compute >

Volumes >

Network >

Network Topology

Networks

Routers

Security Groups

Floating IPs

Admin >

Identity >

Network Topology

Graph

Small

Normal

Launch Instance

Create Network

Create Router

public-router

ID c049635c-4ae9-4f6e-9b6d-5bf495b23849

STATUS Active

Interfaces

+ Add Interface

07a227f6-38... 192.168.1.23 router\_gateway Active Delete Interface

722fb0c0-06... 10.0.10.1 router\_interface Active Delete Interface

> View Router Details

Delete Router

Launch Instance

Create Network

Create Router

## 7. Kiểm tra Máy Ảo và Nhóm Bảo Mật

Kiểm tra xem cấu hình có hoạt động hay không. Theo sơ đồ từ phần trước, tiến hành tạo máy ảo (VM) và nhóm bảo mật tương ứng.

### Floating IP

Trước tiên, phân bổ một IP thả nổi (công cộng) để máy ảo truy cập vào mạng vật lý. Truy cập Project > Network > Floating IPs, nhấp vào “Allocate IP to Project”, giữ nguyên cấu hình mặc định và chọn “Allocate IP”.

43

## Allocate Floating IP



Pool \*

public

Description

DNS Domain

DNS Name

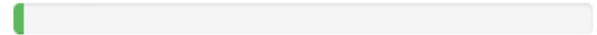
### Description:

Allocate a floating IP from a given floating IP pool.

### Project Quotas

Floating IP

0 of 50 Used



Cancel

Allocate IP

Danh sách sẽ hiển thị Floating IP đã được phân bổ.

Compute >

Volumes >

Network >

Network Topology

Networks

Routers

Security Groups

**Floating IPs**

Admin >

Identity >

## Floating IPs

Floating IP Address =  Filter [Allocate IP To Project](#)

Release Floating IPs

Displaying 1 item

<input type="checkbox"/>	IP Address	Description	DNS Name	DNS Domain	Mapped Fixed IP Address	Pool	Status	Actions
<input type="checkbox"/>	192.168.1.19				-	public	Down	<a href="#">Associate</a> <input type="button" value="v"/>

Displaying 1 item

### Nhóm Bảo Mật

Tạo nhóm bảo mật trước khi tạo máy ảo. Truy cập Project > Network > Security Groups, nhấp vào “Create Security Group”, đặt tên là “test-sg” và chọn “Create Security Group”.

Sau khi tạo, hệ thống sẽ chuyển hướng đến trang quản lý quy tắc. Nếu không, truy cập Project > Network > Security Groups, nhấp vào nút “Manage Rules” của nhóm “test-sg”.

Thêm một số quy tắc cho bài kiểm tra. Một quy tắc để cho phép ping máy ảo và một quy tắc để kết nối qua SSH. Nhấp vào nút “Add Rule”, chọn “All ICMP” từ danh sách thả xuống của quy tắc như hình ảnh dưới đây.

## Add Rule

Rule \*

Custom TCP Rule

Custom TCP Rule

Custom UDP Rule

Custom ICMP Rule

Other Protocol

All ICMP

All TCP

All UDP

DNS

HTTP

HTTPS

IMAP

IMAPS

LDAP

MS SQL

MYSQL

POP3

POP3S

RDP

SMTP

SMTPS

SSH

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

**Rule:** You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

**Open Port/Port Range:** For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

**Remote:** You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add

Nhấp vào nút “Add” và lặp lại quy trình tương tự cho quy tắc “SSH”.

## 8. Máy Ảo

Tạo máy ảo (VM). Truy cập Project > Compute > Instances, nhấp vào “Launch Instance” và sử dụng thông tin sau để tạo phiên bản:

- Tên Phiên bản: test-instance
- Chọn Nguồn Khởi động: Image
- Tạo Volume Mới: Không
- Hình ảnh Được Phân bổ: Chọn hình ảnh cirros trong phần “Available” bằng cách nhấp vào mũi tên lên.
- Flavor: Chọn “m1.tiny” từ danh sách “Available” bằng cách nhấp vào mũi tên lên tương ứng.
- Mạng Được Phân bổ: Chọn “home-lab-network” trong danh sách “Available” bằng cách nhấp vào mũi tên lên tương ứng.
- Nhóm Bảo Mật: Xóa nhóm “default” bằng cách nhấp vào mũi tên xuống tương ứng, sau đó chọn “home-lab-network” trong danh sách “Available” bằng cách nhấp vào mũi tên lên tương ứng.

Liên kết một Floating IP (Public) với máy ảo để kết nối với mạng vật lý. Từ danh sách thả xuống “Actions”, chọn “Associate Floating IP”.

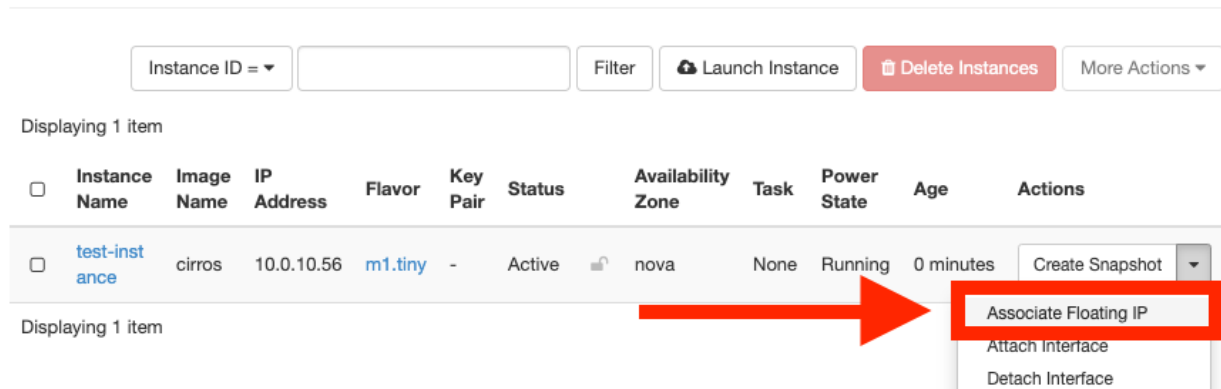
## Instances

Instance ID =  Filter Launch Instance Delete Instances More Actions ▾

Displaying 1 item

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	test-instance	cirros	10.0.10.56	m1.tiny	-	Active	nova	None	Running	0 minutes	<div>Create Snapshot Associate Floating IP Attach Interface Detach Interface</div>

Displaying 1 item



Sau đó cài Floating IP

## Manage Floating IP Associations

**IP Address \***  

Select an IP address ▾ +

Select an IP address  
192.168.1.19 ▾

Select the IP address you wish to associate with the selected instance or port.

Associate

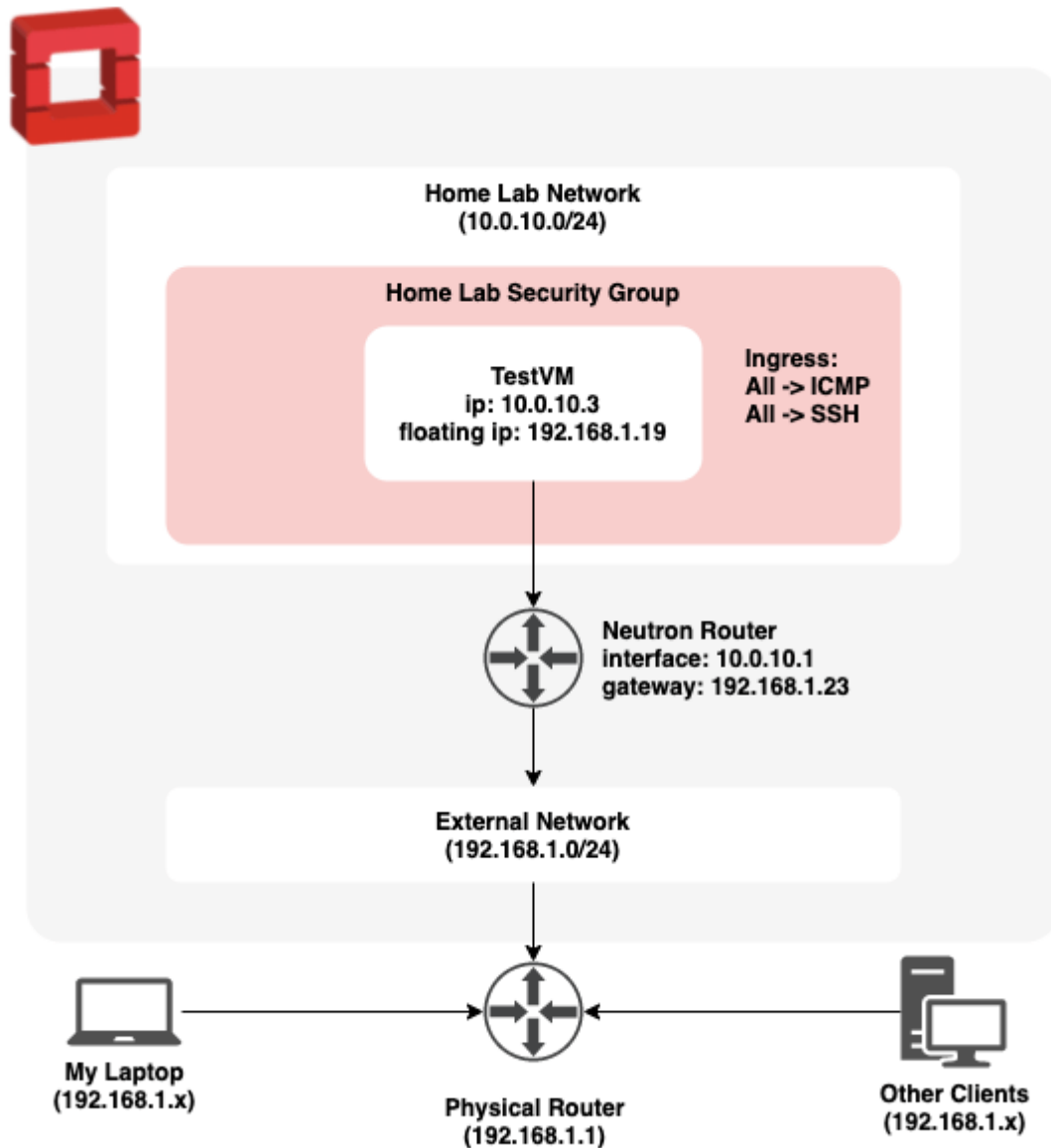
Kiểm tra OpenStack, kiểm tra kết nối mạng.

Tiếp theo, kết nối với máy ảo qua SSH. Mật khẩu mặc định cho người dùng cirros là “gocubsgo”.

Sử dụng lệnh sau để kiểm tra kết nối internet từ máy ảo:

Kết nối internet của máy ảo được coi là thành công nếu nhận được phản hồi HTTP/1.1 200 OK

## 9. Cấu hình pfSense



Từ sơ đồ ta cần thay pfSense hoạt động như một router thay cho Physical Router, như vậy snort sẽ có thể bắt được cả gói tin từ cả trong lẫn ngoài trước khi gói tin có thể truy cập được vào mạng Neuron.

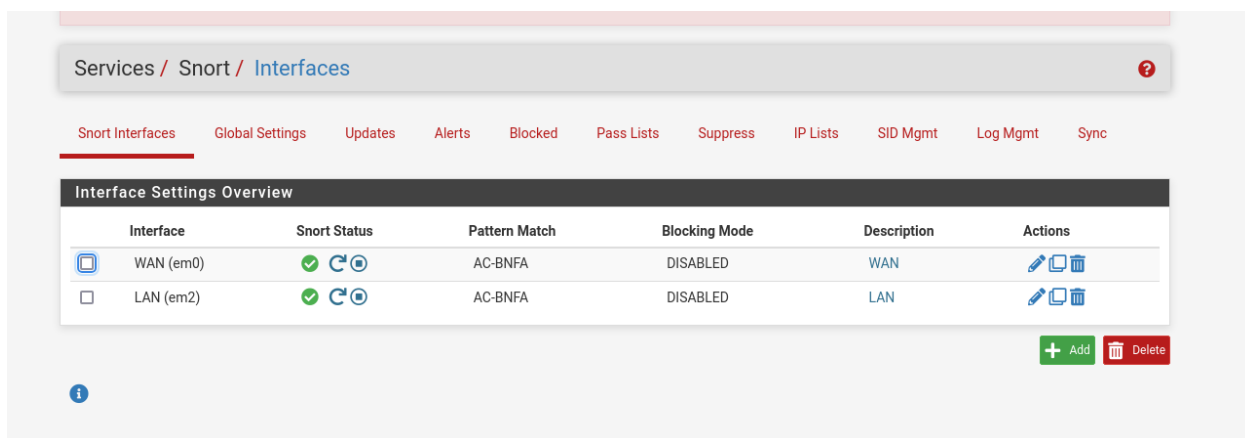
Thực hiện port forwarding từ địa chỉ của máy ảo tới địa chỉ của pfSense

Port Forward 1:1 Outbound NPt											
Rules											
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	WAN address	2222	192.168.1.41	22 (SSH)			

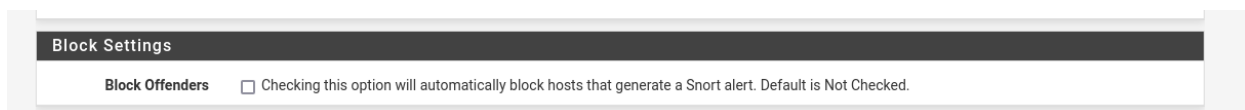
Như vậy khi người dùng ngoài truy cập họ có thể truy cập vào instance bằng cách truy cập qua cổng 2222 với địa chỉ WAN của pfSense

## 10. Cấu hình Snort

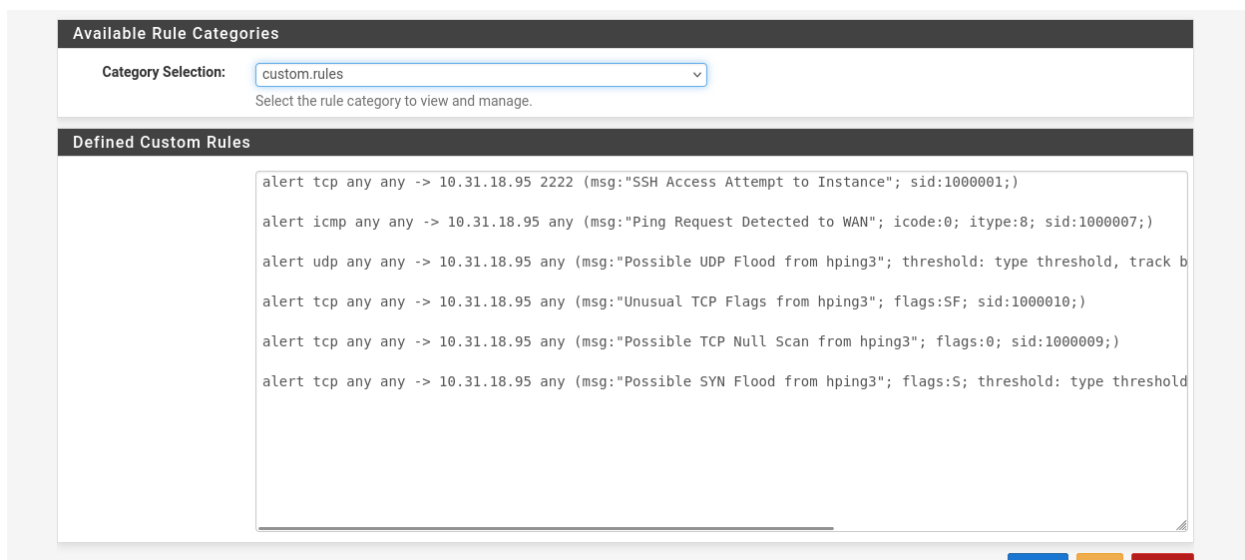
Thêm một WAN interface cho snort



Tùy nhu cầu sử dụng có thể cài đặt cho Snort hoạt động như một IDS hoặc một IPS bằng cách tích vào phần sau



Cài đặt rules để lọc gói tin qua pfSense












## 11. Thử nghiệm tấn công









Sử dụng các lệnh và kiểm tra kết quả

### Kết quả









Phát hiện ping từ bên ngoài

2025-04-21 20:07:03		0	ICMP	10.31.18.199  	10.31.18.95  	1:1000007   	Ping Request Detected to WAN
------------------------	---	---	------	---	--	--	------------------------------

Phát hiện truy cập từ bên ngoài

2025-04-21 20:07:11		0	TCP	10.31.18.199  	2446	10.31.18.95  	2222	1:1000001   	SSH Access Attempt to Instance
------------------------	---	---	-----	---	------	--	------	--	--------------------------------

Phát hiện bị tấn công DOS

2025-04-21 18:58:57		0	TCP	10.31.18.199  	1432	10.31.18.95  	2222	1:1000010   	Unusual TCP Flags from hping3
------------------------	---	---	-----	---	------	--	------	--	-------------------------------

## TÀI LIỆU THAM KHẢO

- [1] TS.Nguyễn Ngọc Điệp, Bài giảng Kỹ thuật theo dõi, giám sát an toàn mạng, 2021
- [2] OpenStack 2024.2 Administrator Guides, <https://docs.openstack.org/2024.2/admin/>