UNIVERSITY OF INFORMATION TECHNOLOGY &

VIETNAM - KOREA MEDIA

**Faculty of Computer & Electronics Engineering**



**CLOUD COMPUTING REPORT**

TOPIC

**Resource Provisioning in Cloud Computing: Hybrid Virtual Server with OpenStack.**

|  |  |  |
| --- | --- | --- |
| Student: | **Tran Vu Lap Thi**  **Nguyen An Phuc**  **Truong Van Khai**  **Nguyen Huu Khanh** | MSV: 21IT448  MSV: 21IT639  MSV: 21IT  MSV: 21IT |
| Instructor: | **Dr. Dang Quang Hien** | |
| Course: | Cloud Computing (4) | |

***Danang, May 2025***

UNIVERSITY OF INFORMATION TECHNOLOGY &

VIETNAM - KOREA MEDIA

**Faculty of Computer & Electronics Engineering**



**CLOUD COMPUTING REPORT**

TOPIC

**Resource Provisioning in Cloud Computing: Hybrid Virtual Server with OpenStack.**

|  |  |  |
| --- | --- | --- |
| Student: | **Tran Vu Lap Thi**  **Nguyen An Phuc**  **Truong Van Khai**  **Nguyen Huu Khanh** | MSV: 21IT448  MSV: 21IT639  MSV: 21IT  MSV: 21IT |
| Instructor: | **Dr. Dang Quang Hien** | |
| Course: | Cloud Computing (4) | |

***Da Nang, May 2025***

ACKNOWLEDGEMENTS

We would like to express our sincere dedicated gratitude to the school for their guidance, constructive feedback, and support, which has enabled us to complete the research project for the course "Cloud Computing" in the best possible way.

We also wish to thank Dr. Dang Quang Hien for his enthusiastic guidance throughout the process of working on this research report.

During the course of the research project, we have made every effort to do our best. However, there may still be some dangers. We look forward to receiving feedback from our professors and the committee so that our research project can be further improved.

We sincerely thank you!

TASK ASSIGNMENT TABLE

|  |  |  |
| --- | --- | --- |
| **STT** | **Member Names** | **% of Task** |
| 1 | Tran Vu Lap Thi | 25 % |
| 2 | Nguyen An Phuc | 25 % |
| 3 | Truong Van Khai | 25 % |
| 4 | Nguyen Huu Khanh | 25 % |

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENTS i](#_Toc197732088)

[TASK ASSIGNMENT TABLE ii](#_Toc197732089)

[LIST OF FIGURES v](#_Toc197732090)

[LIST OF ABBREVIATIONS vi](#_Toc197732091)

[INTRODUCTION ix](#_Toc197732092)

[1. Reason for Choosing the Topic ix](#_Toc197732093)

[2. Objectives of the Topic ix](#_Toc197732094)

[3. Direction of the Topic x](#_Toc197732095)

[4. Research object x](#_Toc197732096)

[5. Research scope x](#_Toc197732097)

[Chapter 1: Overview of Cloud Computing 1](#_Toc197732098)

[1.1. Overview of Cloud Computing 1](#_Toc197732099)

[1.1.1. What is Clouds? 1](#_Toc197732100)

[1.1.2. Six Characteristics of Clouds 2](#_Toc197732101)

[1.1.3. Deployment Models 3](#_Toc197732102)

[1.2. Service Models in Cloud Computing 5](#_Toc197732103)

[1.2.1. Infrastructure as a Service (IaaS) 5](#_Toc197732104)

[1.2.2. Platform as a Service (PaaS) 6](#_Toc197732105)

[1.2.3. Software as a Service (SaaS) 7](#_Toc197732106)

[1.2.4. Applications of Cloud Computing 8](#_Toc197732107)

[Chapter 2: Exploring Open Source Hybrid Cloud with OpenStack 10](#_Toc197732108)

[2.1. Introduction to Hybrid Cloud and OpenStack 10](#_Toc197732109)

[2.1.1. Defining Hybrid Cloud 10](#_Toc197732110)

[2.1.2. OpenStack - A Leading Open Source Solution 11](#_Toc197732111)

[2.1.3. The Role of OpenStack in Hybrid Cloud 12](#_Toc197732112)

[2.2. OpenStack Architecture 13](#_Toc197732113)

[2.2.1. Key Components of OpenStack 13](#_Toc197732114)

[2.2.2. OpenStack Deployment Models in Hybrid Cloud 14](#_Toc197732115)

[2.3. Benefits of OpenStack in Hybrid Cloud 15](#_Toc197732116)

[2.3.1. Flexibility and Scalability 15](#_Toc197732117)

[2.3.2. Cost Savings 15](#_Toc197732118)

[2.3.3. Security and Control 15](#_Toc197732119)

[2.3.4. Large Community Support 16](#_Toc197732120)

[Chapter 3: BUILDING LAB TO DEPLOY OPENSTACK HYBRID CLOUD 17](#_Toc197732121)

[3.1. INTRODUCTION DEPLOYMENT 17](#_Toc197732122)

[3.1.1. VMware Workstation infrastructure 17](#_Toc197732123)

[3.1.2. Network Configuration 18](#_Toc197732124)

[3.2. Deploying OpenStack with DevStack 18](#_Toc197732125)

[3.2.1. Install DevStack 18](#_Toc197732126)

[3.2.2. Configuration - Cinder with LVM 20](#_Toc197732127)

[3.2.3. Network Configuration – Neutron 21](#_Toc197732128)

[3.3. System configuration and demo 23](#_Toc197732129)

[3.3.1. Overview model 23](#_Toc197732130)

[3.3.2. Create Project and User 24](#_Toc197732131)

[3.3.3. Create Networks 24](#_Toc197732132)

[3.3.4. Create route 25](#_Toc197732133)

[3.3.5. Service Deployment 26](#_Toc197732134)

[3.3.6. Initialize Key Pairs 27](#_Toc197732135)

[3.3.7. Initialize Instance from Volume (Nova) 28](#_Toc197732136)

[3.3.8. Assign Floating IP (Neutron) 29](#_Toc197732137)

[3.3.9. System Demo 29](#_Toc197732138)

[3.4. End of chapter 3 32](#_Toc197732139)

[CONCLUDE 33](#_Toc197732140)

[REFERENCES 35](#_Toc197732141)

LIST OF FIGURES

[Figure 1.1. 1 : Cloud Computing Simulation 1](#_Toc197732322)

[Figure 1.1. 2 : Cloud Computing Deployment Models 3](#_Toc197732323)

[Figure 1.2. 1 : Infrastructure as a Service (IaaS) 5](#_Toc197732324)

[Figure 1.2. 2 : Platform as a Service (PaaS) 6](#_Toc197732325)

[Figure 1.2. 3 : Software as a Service (SaaS) 7](#_Toc197732326)

[Figure 1.2. 4 : Illustration of Cloud Computing 8](#_Toc197732327)

[Figure 1.2. 5 : Popular cloud computing technologies 9](#_Toc197732328)

[Figure 2.1. 1 : Hybrid Cloud 10](#_Toc197732329)

[Figure 2.1. 2 : OpenStack open source 11](#_Toc197732330)

[Figure 3.1. 1 : WMware configuration 17](#_Toc197732331)

[Figure 3.1. 2 : Network card configuration 18](#_Toc197732332)

[Figure 3.2. 1 :Create file local.conf 19](#_Toc197732333)

[Figure 3.2. 2 : Openstack Management Web 20](#_Toc197732334)

[Figure 3.2. 3 : Create lvmdriver-2 21](#_Toc197732335)

[Figure 3.2. 4 : Current network information 21](#_Toc197732336)

[Figure 3.2. 5 : NAT Rule 22](#_Toc197732337)

[Figure 3.3. 1 : System model 23](#_Toc197732338)

[Figure 3.3. 2 : Create project and user 24](#_Toc197732339)

[Figure 3.3. 3 : Networks 25](#_Toc197732340)

[Figure 3.3. 4 : Router configuration 25](#_Toc197732341)

[Figure 3.3. 5 : Routing setup 26](#_Toc197732342)

[Figure 3.3. 6 : Image 27](#_Toc197732343)

[Figure 3.37 : Create volume to store data 27](#_Toc197732344)

[Figure 3.3. 8 : List of ssh keys 28](#_Toc197732345)

[Figure 3.3. 9 : View created instances on web interface 28](#_Toc197732346)

[Figure 3.3. 10 : Floating IPs 29](#_Toc197732347)

[Figure 3.3. 11 : Graph model 29](#_Toc197732348)

[Figure 3.3. 12 : Perform ping from webserver 30](#_Toc197732349)

[Figure 3.3. 13 : Test ssh connection 30](#_Toc197732350)

[Figure 3.3. 14 : Web-server web interface 31](#_Toc197732351)

[Figure 3.3. 15 : Login page interface 31](#_Toc197732352)

[Figure 3.3. 16 : Login with user manager1 32](#_Toc197732353)

LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| Abbreviation | Full Form | Description |
| API | Application Programming Interface | A set of rules for software components to interact with each other. |
| CPU | Central Processing Unit | The main processing unit of a computer that executes instructions. |
| DB | Database | A structured collection of data. |
| DHCP | Dynamic Host Configuration Protocol | A protocol for dynamically assigning IP addresses to devices on a network. |
| DNS | Domain Name System | A system that translates domain names into IP addresses. |
| DMZ | Demilitarized Zone | A network area that acts as a buffer between internal systems and the Internet. |
| HA | High Availability | A system design approach to ensure operational continuity with minimal downtime. |
| HTTP/HTTPS | HyperText Transfer Protocol (Secure) | Protocols used for web communication (HTTPS is the secure version). |
| ICMP | Internet Control Message Protocol | A protocol used for error messages and operational queries (eg, ping). |
| IDS | Intrusion Detection System | A system that monitors and detects potential security resources. |
| IPS | Intrusion Prevention System | A system that monitors and blocks malicious activities. |
| IP | Internet Protocol | A protocol for sending data between devices across networks. |
| IaaS | Infrastructure as a Service | A cloud computing model provides virtualized computing resources. |
| LVM | Logical Volume Manager | A Linux subsystem for managing logical disk volumes. |
| NAT | Network Address Translation | A method of remapping one IP address space into another. |
| OVS | Open vSwitch | An open-source virtual switch for managing network traffic in virtualization. |
| OS | Operating System | System software that manages computer hardware and software resources. |
| PaaS | Platform as a Service | A cloud model provides a platform for developers to build and run applications. |
| SaaS | Software as a Service | A model for delivering software over the Internet. |
| SDN | Software Defined Networking | A networking approach that allows programmatic control of the network. |
| SSH | Secure Shell | A protocol for secure remote login and command execution. |
| UI | User Interface | The visual part of software with which users interact. |
| VM | Virtual Machine | A software emulation of a physical computer. |
| VLAN | Virtual Local Area Network | A logical subgroup within a LAN for better segmentation. |
| VPN | Virtual Private Network | A secure network connection over a public network such as the Internet. |
| VPC | Virtual Private Cloud | A private cloud hosted within a public cloud infrastructure. |
| VCPU | Virtual CPU | A virtualized CPU assigned to a VM. |
| OSPF | Open Shortest Path First | A routing protocol used in IP networks to find the best path. |
| SNMP | Simple Network Management Protocol | A protocol used to manage and monitor network devices. |
| GUI | Graphical User Interface | A visual way of interacting with a computer or software. |
| YAML | Yet Another Markup Language | A human-readable data-serialization language used for configuration files. |

INTRODUCTION

1. Reason for Choosing the Topic

In the context of the ongoing Industry 4.0 revolution, the demand for remote data processing, storage, and access has become increasingly essential. Cloud Computing has emerged as a core technology solution, enabling organizations and individuals to leverage computational resources in a flexible, cost-effective, and efficient manner. However, deploying resource provisioning models in cloud computing is not based solely on theory; it also requires a close integration of architecture models, software platforms, and open-source tools, such as OpenStack.

Currently, the hybrid cloud model is gaining significant attention due to its ability to integrate the benefits of both public and private clouds, balancing performance, flexibility, and security. Meanwhile, OpenStack has emerged as a powerful open-source platform, trusted by major organizations worldwide to build and manage cloud infrastructure.

Driven by these practical issues, the research team has chosen the topic "Research on Resource Provisioning Models in Cloud Computing – Case Study: Building a Hybrid Virtual Server with OpenStack" to delve deeper into the operational principles, capabilities, and practical application potential of cloud computing in the modern deployment virtualization environment.

1. Objectives of the Topic

Systematize the theoretical foundation of cloud computing, especially the popular resource provisioning and deployment models (IaaS, PaaS, SaaS, etc.) currently in use.

Analyze and present the theory related to resource provisioning models in cloud environments, clarifying the role and operational principles of each component within the system.

Develop a Hybrid Virtual Server model using the open-source OpenStack, presenting the installation, deployment, and configuration processes for basic services, as well as evaluating the system's performance.

Provide a reference platform for individuals and organizations seeking to explore or implement open-source cloud computing solutions in real-world environments, particularly in educational institutions, small and medium-sized enterprises.

1. Direction of the Topic

The direction of the topic " **Resource Provisioning in Cloud Computing: Hybrid Virtual Server with OpenStack.** " in the project report can be carried out through the following chapters:

* Chapter 1: Overview of Cloud Computing
* Chapter 2: Exploring Open-Source Hybrid Virtual Server with OpenStack
* Chapter 3: Implementation of the Program and Results

1. Research object

Cloud Computing Models and Architectures

* Service Delivery Models in Cloud Computing
* Core Components of OpenStack
* Methods of Deploying Cloud Systems Using Open Source
* Integration and Scalability Between Private and Public Cloud Models

1. Research scope

* Study the theory behind cloud computing provisioning and deployment models.
* Analyze the key components in OpenStack that facilitates resource provisioning.
* Develop and deploy a Hybrid Virtual Server model in a simulated or small-scale environment.
* Avoid delving into high-level security issues, performance optimization, or complex multi-platform integration.

# Overview of Cloud Computing

## Overview of Cloud Computing

### What is Clouds?

" Clouds" refers to servers that are accessed over the Internet, with software and databases running on those servers. Cloud servers are located in data centers distributed across the world. With cloud computing, users and companies do not need to directly manage their own servers or run software applications on their own machines.



Figure .1: Cloud Computing Simulation

Cloud computing allows users to access the same files and applications from almost any device, as the processing and storage occurs on servers located in data centers, rather than on the user's device. This is why users can log into their Facebook account on a new phone after their old phone is broken and still find their account with all their photos, videos, and chat history. It works similarly to email providers like Gmail or Microsoft Office 365, and cloud storage services like Dropbox or Google Drive.

### Six Characteristics of Clouds

**On-demand self-service** : Customers can be provided with resources such as servers or storage capacity automatically as per their request, without the need for intervention from the service provider.

**Broad network access** : With just an internet-connected application on any device, such as a desktop, laptop, or mobile device, users can access cloud resources .

**Location independent resource pooling** : Customers are unaware of and do not control the location of the provided resources; however, they can manage them through the service provider's advanced features. These resources may include storage, processing power, memory, and network bandwidth .

**Rapid elasticity** : Resources can be provisioned quickly and flexibly, with the ability to scale up or down depending on the customer's usage needs. For customers, cloud resources are always available and can be considered virtually unlimited, accessible at any time .

**Measured service** : Cloud computing systems have the ability to automatically manage and adjust resource usage by applying measurement methods at different levels for each type of service. Resource usage can be monitored and measured, and customers typically only pay for the amount of resources they consume .

**Cloud API:** It is an Application Programming Interface (API) used to build applications in the cloud computing market. Cloud APIs allow software to request data and computation from one or more services through a direct or indirect interface. Cloud APIs often expose their features through REST or SOAP .

### Deployment Models



Figure .2: Cloud Computing Deployment Models

#### Private cloud

**Definition** : A private cloud is a cloud computing service provided within an organization. These "clouds" exist behind the company's firewall and are managed directly by the enterprise. This is an inevitable trend for businesses looking to optimize their IT infrastructure . Target Users: Internal users within the organization who use and manage the private cloud.

**Advantages:**

* Active control over usage, upgrades, and management.
* Cost reduction due to resource optimization.
* Better security, as the infrastructure is controlled internally.

**Disadvantages:**

* Technological challenges during deployment and the cost of building and maintaining the system.
* Limited to internal use within the organization; external users cannot access the system.

#### Public cloud

**Definition:** Public cloud services are provided by a third-party vendor. They exist outside the company's firewall and are managed by the cloud service provider. Public clouds are built to serve the general public, and users sign up with the provider and pay for usage based on the provider's pricing policy. Public cloud is the most commonly used deployment model in cloud computing today .

**Target Users:** Includes external users from the internet. Management: Managed by the service provider.

**Advantages:**

* Can serve a larger number of users, not limited by space or time.
* Saves on server infrastructure, electricity, and labor costs for businesses.

**Disadvantages:**

* Businesses are dependent on the provider and do not have full control over management.
* Difficulties in storing internal documents and sensitive information.

#### Hybrid cloud

**Definition** : A hybrid cloud is a combination of private cloud and public cloud. It allows organizations to leverage the strengths of each model and provides an optimal usage method for users. These "clouds" are typically created by the business, with management responsibilities shared between the enterprise and the public cloud provider.

**Target Users** : Enterprises and providers manage the hybrid cloud based on mutual agreements. Users can access both the services provided by the cloud provider and the enterprise's own services .

**Advantages** : The enterprise can simultaneously utilize multiple services without limitations .

**Disadvantages** : Challenges in deployment and management. High costs involved .

## Service Models in Cloud Computing

### Infrastructure as a Service (IaaS)

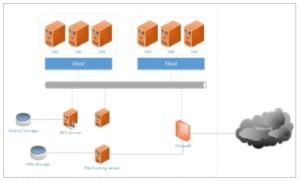


Figure .1: Infrastructure as a Service (IaaS)

**Definition:** IaaS (Infrastructure-as-a-Service) provides users with raw infrastructure, typically in the form of virtual machines, as a service. Users can deploy and run software on these virtual machines as if they were on physical servers, or they can upload personal data to the "cloud" for storage. Users do not have control over the physical infrastructure within the "cloud"; however, they have full management rights to the resources provided to them, as well as the ability to request an expansion of the resources they are permitted to use.

**Key Features of IaaS:**

* Resource Provisioning as a Service: Includes servers, networking equipment, memory, CPU, disk space, and data center equipment.
* Flexible Scalability: Allows resources to be scaled up or down based on demand.
* Cost Variability: Pricing is based on actual usage, meaning costs can change depending on resource consumption.
* Multi-tenancy: Multiple tenants can share the same resources, optimizing the use of infrastructure.
* Enterprise Level: Provides benefits for companies by offering a consolidated resource pool for computational needs.

Examples: Amazon EC2/S3, Elastra (Beta 2.0, February 2009), Nirvanix, AppNexus.

### Platform as a Service (PaaS)

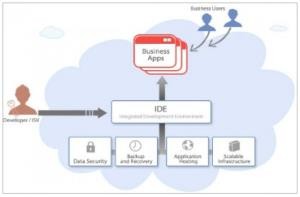


Figure .2: Platform as a Service (PaaS)

**Definition:** PaaS (Platform as a Service) is a model that provides a platform for application development in an abstracted environment. It supports the deployment of applications without worrying about the cost or complexity of equipping and managing the underlying hardware and software layers. PaaS offers all the necessary features to support the development and delivery of applications and web services ready to be deployed on the Internet, without requiring any software downloads or installations for developers, IT managers, or end-users. PaaS allows application developers to quickly create applications, as many of the complexities related to server setup and database management are handled by the PaaS provider.

**Key Features of PaaS:**

* Supports Development, Testing, Deployment, and OperationsWeb-based Interface Tools
* Unified Architecture
* Integration of Web Services and Databases
* Supports Team Collaboration
* Provides Additional Utility Tools

Examples: Google App Engine , OpenShift , Salesforce , Microsoft Azure

### Software as a Service (SaaS)



Figure .3: Software as a Service (SaaS)

**Definition** : SaaS (Software as a Service) is an application deployment model where the provider allows users to access the service on-demand. SaaS providers can either host the application on their servers or download the application to the client's device, deactivating it once the term ends. The on-demand functionalities can be controlled internally to share the rights of a third-party application provider.

**Key Features of SaaS:**

* **Ready-to-use Software** : The software is available and requires access and management via the network.
* **Centralized Management** : Activities are managed from a centralized location rather than at each customer's site, allowing customers to access services remotely via the web.
* **One-to-many Model** : SaaS offers a model that is more aligned with the one-to-many mapping rather than the traditional 1:1 model, including features related to architecture, pricing, and management.
* **Centralized Features for Upgrades** : Provides centralized features for upgrades, freeing users from having to download patches and updates.
* **Frequent Integration** : Regularly integrates software for communication across wide-area networks.
* **Examples** : Email services, Google Docs, Google Calendar, and other applications provided by Google.

### Applications of Cloud Computing



Figure .4: Illustration of Cloud Computing

**Website Hosting** : The issue of website hosting is always considered important and essential. When a company's system cannot meet its continuous growth, adopting cloud technology solves all related issues efficiently and effectively. Additionally, businesses only need to pay for actual usage when utilizing hosting services, ensuring that security systems are maintained fully and seamlessly.

**Business Management Applications** : By storing data on cloud storage systems, users can easily control and manage data most effectively. Nowadays, most platforms specialize in data analysis using cloud computing models. These platforms are capable of processing both structured and unstructured data comprehensively.

**Cloud Databases** : For businesses with large amounts of data but limited budgets and a lack of specialized operations teams, adopting cloud computing technology is a practical solution. Cloud databases run efficiently without the need for businesses to invest in physical servers for storage and operations. Service providers manage and monitor the systems to ensure the smooth functioning of your database operations.

**Data Storage and Sharing** : Businesses can back up and retrieve data from anywhere as long as there is an internet connection. Popular cloud storage applications such as **Google Drive** , **OneDrive** , and **Dropbox** are widely used for storing and sharing data.

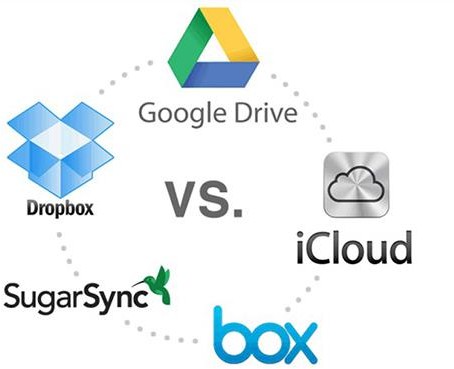


Figure .5: Popular cloud computing technologies

Google Drive, Dropbox, OneDrive, iCloud,... are typical examples of cloud services. Users only need to register an account and use the free and paid services according to their needs. They store documents on their "cloud" account and access them from any location as long as there is an internet connection.

# Exploring Open Source Hybrid Cloud with OpenStack

## Introduction to Hybrid Cloud and OpenStack

### Defining Hybrid Cloud

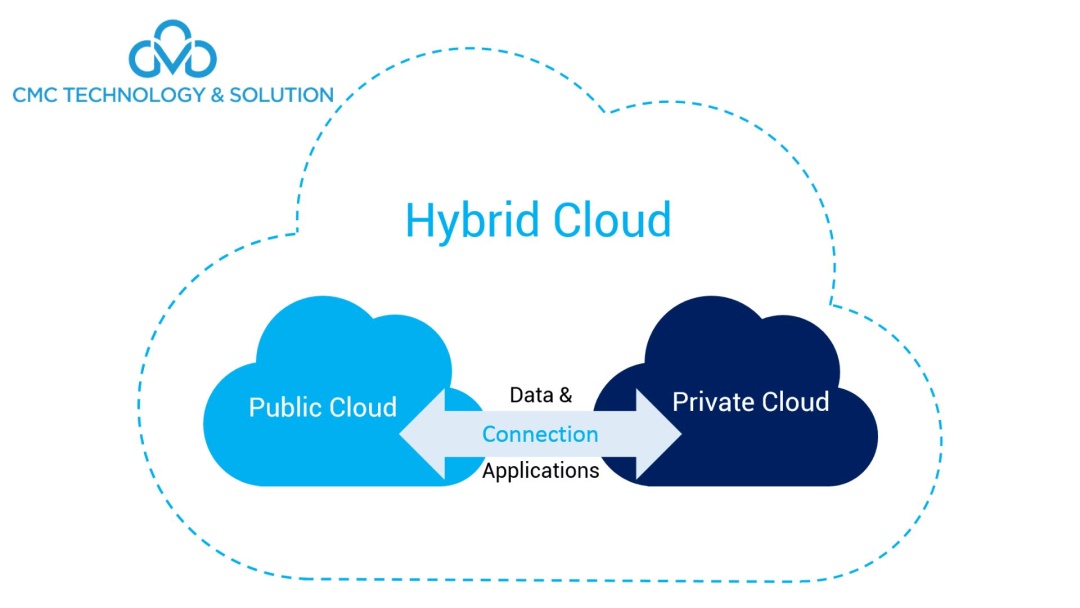


Figure .1: Hybrid Cloud

In the context of increasingly powerful digital transformation, the need for flexible and efficient infrastructure has become critical. Hybrid cloud has emerged as an optimal solution, combining the advantages of both on-premises environments and public cloud computing.

Hybrid Cloud is a cloud computing model that combines two or more distinct cloud infrastructures (eg, public cloud, private cloud, community cloud) or traditional IT infrastructure located on-premises, which are bound together by technology that enables data and application portability between them.

This model offers superior flexibility, allowing organizations to choose the most suitable environment for each specific workload. Factors driving the adoption of hybrid cloud include:

* **Flexible Scalability:** Easily scale resources as demand increases without significant upfront investment in physical infrastructure.
* **Cost Optimization:** Select the most cost-effective environment for each application and data set.
* **Deployment Flexibility:** Deploy new applications quickly and easily in the appropriate environment.
* **Maintaining Control:** Retain control over critical on-premises resources and data.
* **Regulatory Compliance:** Meet stringent security and legal compliance requirements by storing sensitive data on private infrastructure.

### OpenStack - A Leading Open Source Solution

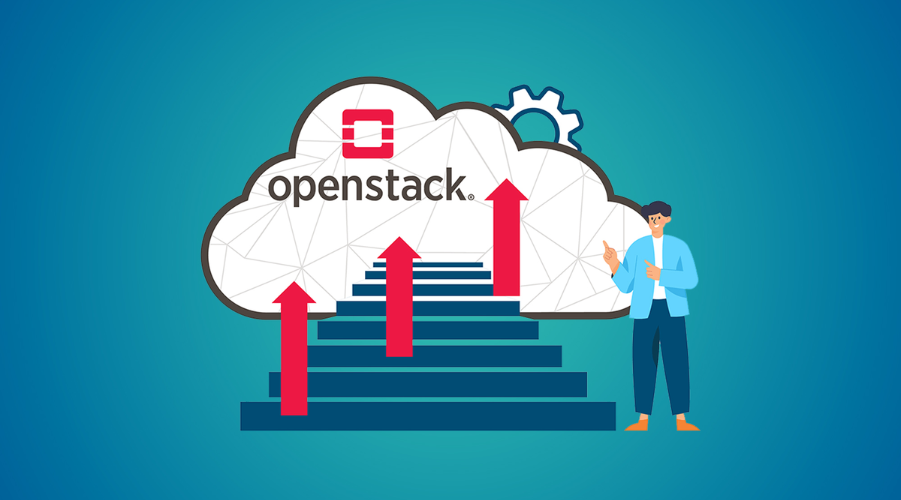


Figure .2: OpenStack open source

OpenStack is an open-source cloud computing platform that provides a robust set of tools for building and managing private cloud infrastructure or combining with other public clouds to create a hybrid environment. Initiated in 2010, OpenStack has rapidly become one of the largest and most important open-source projects in the field of cloud computing.

OpenStack's strengths lie in its flexibility, high customizability, and large development community. It enables organizations to build a cloud environment tailored to their specific needs, leverage existing hardware, and integrate with other systems easily.

The core components of OpenStack provide essential services such as:

* **Nova:** Manages virtual machines (compute).
* **Neutron:** Manages networking.
* **Cinder:** Manages block storage.
* **Swift:** Manages object storage.
* **Glance:** Manages virtual machine images (image service).
* **Keystone:** Provides authentication and authorization services (identity service).
* **Horizon:** Provides a web-based user interface (dashboard).

### The Role of OpenStack in Hybrid Cloud

OpenStack plays a pivotal role in building and managing hybrid cloud environments. It provides a consistent platform for managing compute, storage, and networking resources, regardless of their location – on-premises or in the public cloud.

specifically, OpenStack offers the following benefits in a Hybrid Cloud model:

* + **Consistency:** Provides an abstraction layer, allowing for unified management of resources across different environments.
  + **Interoperability:** Supports standard interfaces and APIs, facilitating seamless integration with other public cloud services.
  + **Control:** Enables organizations to maintain control over critical resources within their private environment while leveraging the scalability of the public cloud when needed.
  + **Flexibility:** Easily migrate applications and data between different environments based on business and technical requirements.
  + **Reduced Vendor Lock-in:** As an open-source platform, OpenStack minimizes dependence on a single vendor, offering greater flexibility and customization.

## OpenStack Architecture

### Key Components of OpenStack

The architecture of OpenStack includes numerous independent components that work together to deliver comprehensive cloud computing services. Here is a detailed description of the main components:

* + **Nova (Compute):** The core component responsible for managing the lifecycle of virtual machines (instances). Nova allows for the creation, launching, stopping, deleting, and management of compute resources. It interacts with other components such as Keystone (for authentication), Glance (to retrieve VM images), and Neutron (to configure networking).
  + **Neutron (Networking):** Provides virtual networking services, including the creation and management of networks, subnets, routers, firewalls, and load balancers. Neutron ensures flexible and secure network connectivity for virtual machines and other services within the OpenStack environment.
  + **Cinder (Block Storage):** Manages the provisioning and attachment of block storage devices (block volumes) to virtual machines. Cinder supports various storage backends and offers features such as snapshots and backups.
  + **Swift (Object Storage):** Provides highly scalable and durable object storage services. Swift is commonly used for storing unstructured data such as images, videos, and large files.
  + **Glance (Image Service):** Manages virtual machine images used to launch instances. Glance allows for the storage, discovery, and management of operating system and application templates.
  + **Keystone (Identity Service):** Provides centralized authentication and authorization services for all OpenStack components. Keystone manages users, tenants (projects), roles, and access permissions.
  + **Horizon (Dashboard):** Offers a web-based user interface (GUI) that allows administrators and users to interact with and manage OpenStack resources visually.
  + **Heat (Orchestration):** Provides an orchestration service to automate the deployment and management of complex applications and infrastructure within OpenStack through templates.
  + **Telemetry (Ceilometer & Aodh):** Collects and monitors performance metrics and resource utilization from OpenStack components. Aodh provides an alarming service based on these metrics.
  + **Trove (Database as a Service):** Provides on-demand database services, allowing users to easily deploy and manage various database instances.
  + **Sahara (Data Processing):** Provides a service for deploying and managing Hadoop and Spark clusters for big data processing tasks.

### OpenStack Deployment Models in Hybrid Cloud

The deployment of OpenStack in a Hybrid Cloud model can be implemented in various ways, depending on the specific requirements of the organization. Some common deployment models include:

* + **Virtual Private Network (VPN) or Dedicated Leased Lines:** Establishing secure and stable connections between the on-premises OpenStack environment and public cloud resources. This allows virtual machines and services in both environments to communicate as if they were on the same network.
  + **Utilizing Dedicated Cloud Connection Services:** Public cloud providers often offer dedicated connection services (eg, AWS Direct Connect, Azure ExpressRoute, Google Cloud Interconnect) to establish high-bandwidth, low-latency connections between an organization's data center and their cloud.
  + **Employing Hybrid Cloud Management Tools:** Tools such as Google's Anthos, Azure Arc, or other multi-cloud management platforms can be used to centrally manage and orchestrate OpenStack resources alongside other public cloud services.
  + **Deploying OpenStack on Public Cloud Infrastructure:** Some organizations may choose to deploy part or all of their OpenStack environment on a public cloud provider's infrastructure. This can be beneficial for scenarios requiring extreme scalability or leveraging specific services offered by the public cloud provider.

## Benefits of OpenStack in Hybrid Cloud

Using OpenStack as a platform for hybrid cloud offers several significant benefits for organizations:

### Flexibility and Scalability

OpenStack provides high flexibility in building and managing infrastructure. Organizations can customize OpenStack components to suit their specific needs. The scalability of OpenStack allows for easy addition of resources as required, accommodating the growth of applications and data without the rigid limitations of traditional infrastructure. In a hybrid environment, OpenStack enables seamless scaling of resources to the public cloud when on-premises demand spikes.

### Cost Savings

As an open-source platform, OpenStack helps reduce software licensing costs. Organizations can leverage existing hardware and choose OpenStack-supported service providers with competitive pricing. In a hybrid model, OpenStack helps optimize costs by allowing less critical workloads to run on the public cloud at a lower cost, while maintaining control over important applications and data on the on-premises OpenStack infrastructure.

### Security and Control

OpenStack offers robust security features and allows to maintain a high degree of control over their infrastructure and data. Features such as identity and access management (Keystone), firewalls (Neutron), and storage encryption (Cinder) help protect the hybrid environment from security threats. Deploying OpenStack on-premises allows organizations to comply with stringent legal and security regulations for sensitive data.

### Large Community Support

With a strong and active development community, OpenStack benefits from the contributions of thousands of developers and experts worldwide. This ensures that the platform is continuously improved, updated, and supported. The community provides extensive documentation, tutorials, and technical assistance, helping organizations deploy and manage their OpenStack environment effectively.

# 

# ****BUILDING LAB TO DEPLOY OPENSTACK**** HYBRID CLOUD

## INTRODUCTION DEPLOYMENT

This chapter aims to build an OpenStack Hybrid Cloud environment for testing and deploying real applications. The implementation is carried out in a virtualized environment using VMware Workstation, using DevStack to install OpenStack. The implementation includes configuring virtualized networks with Open vSwitch, storage with LVM, and deploying a real hybrid cloud model including public, DMZ, and internal networks.

### ****VMware Workstation**** infrastructure

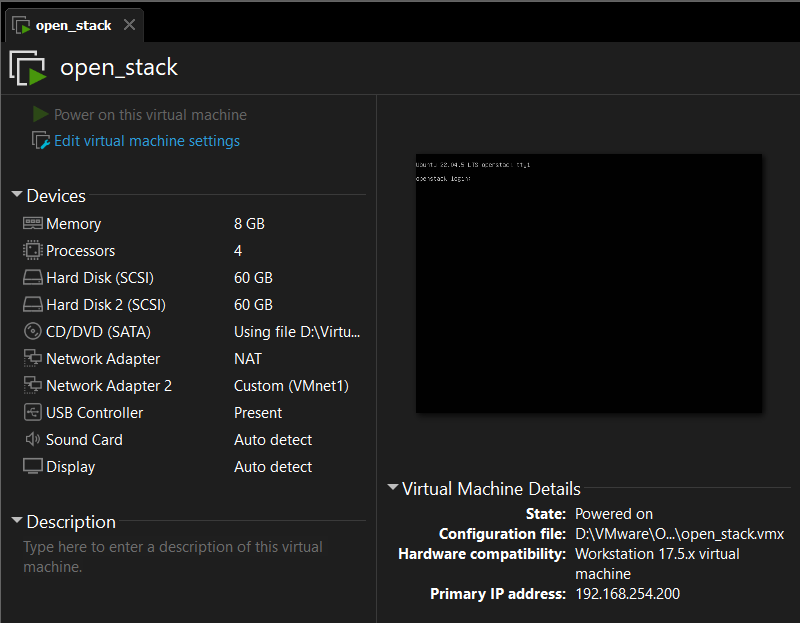


Figure .1: WMware configuration

Hardware:

CPU: 4 vCPU (Intel VT-x/AMD-V virtualization enabled).

RAM: 8GB.

Disk 1: 60GB (Operating system + LVM driver-1).

Disk 2: 60GB (LVM driver-2).

Network:

* Adapter 1: NAT (Internet Connection).
* Adapter 2: VMnet1 (Host-only for internal network).

OS Installation: Ubuntu Server 22.04 LTS.

### Network Configuration

Configure ens33 port ip as static ip and ens34 configuration independent of netplan to prevent network conflicts when installing openstack.

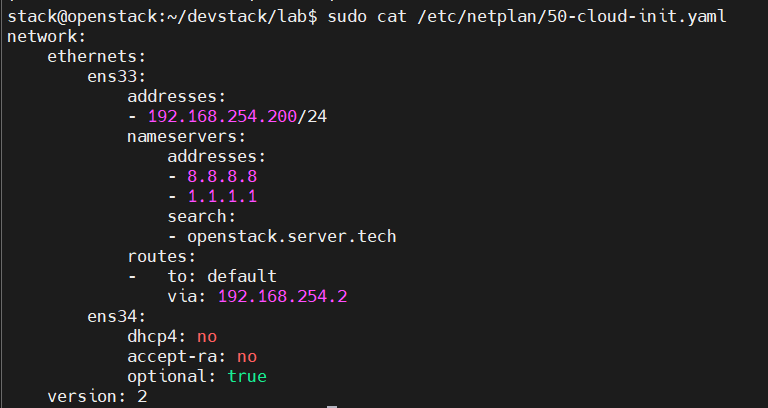


Figure .2: Network card configuration

## Deploying OpenStack with DevStack

### Install DevStack

**Step 1:** Add Stack User (optional)

* sudo useradd -s /bin/bash -d /opt/stack -m stack
* sudo chmod +x /opt/stack
* echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
* sudo -u stack -i

**Step 2:** Download DevStack:

Clone Destack deployment code from Github.

* git clone <https://opendev.org/openstack/devstack>

**Step 3:** Create a local.conf

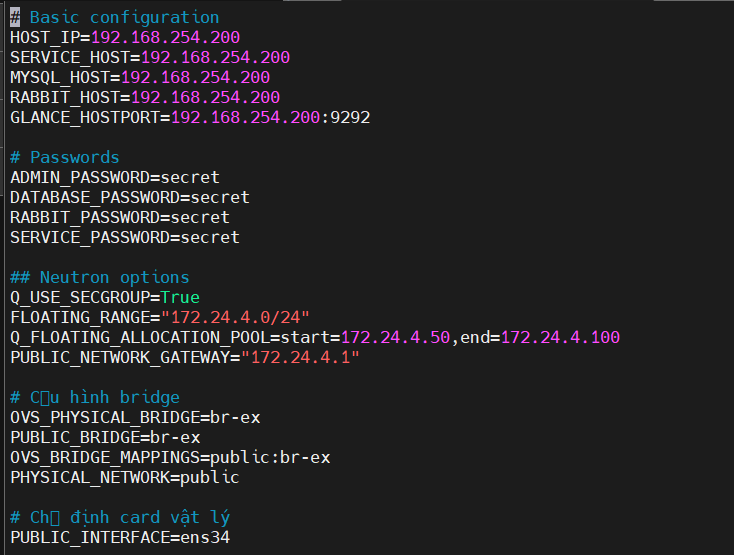


Figure .1:Create file local.conf

**Step 4:** Start Openstack Deployment on Ubuntu using DevStack

* devstack cd
* ./stack.sh

DevStack will install;

* **Keystone** – Identity Service
* **Glance** – Image Service
* **Nova** – Compute Service
* **Placement** – Placement API
* **Cinder** – Block Storage Service
* **Neutron** – Network Service
* **Horizon** – Openstack Dashboard

This will take 15 - 30 minutes, largely depending on the speed of your internet connection. Many git trees and packages will be installed during this process.

**Step 5:** Access OpenStack Dashboard

Copy the Horizon URL shown on the installation output and paste it into your web browser:

* <http://openstack.server.tech/dashboard>

Use the default users and configured password to login demo or admin

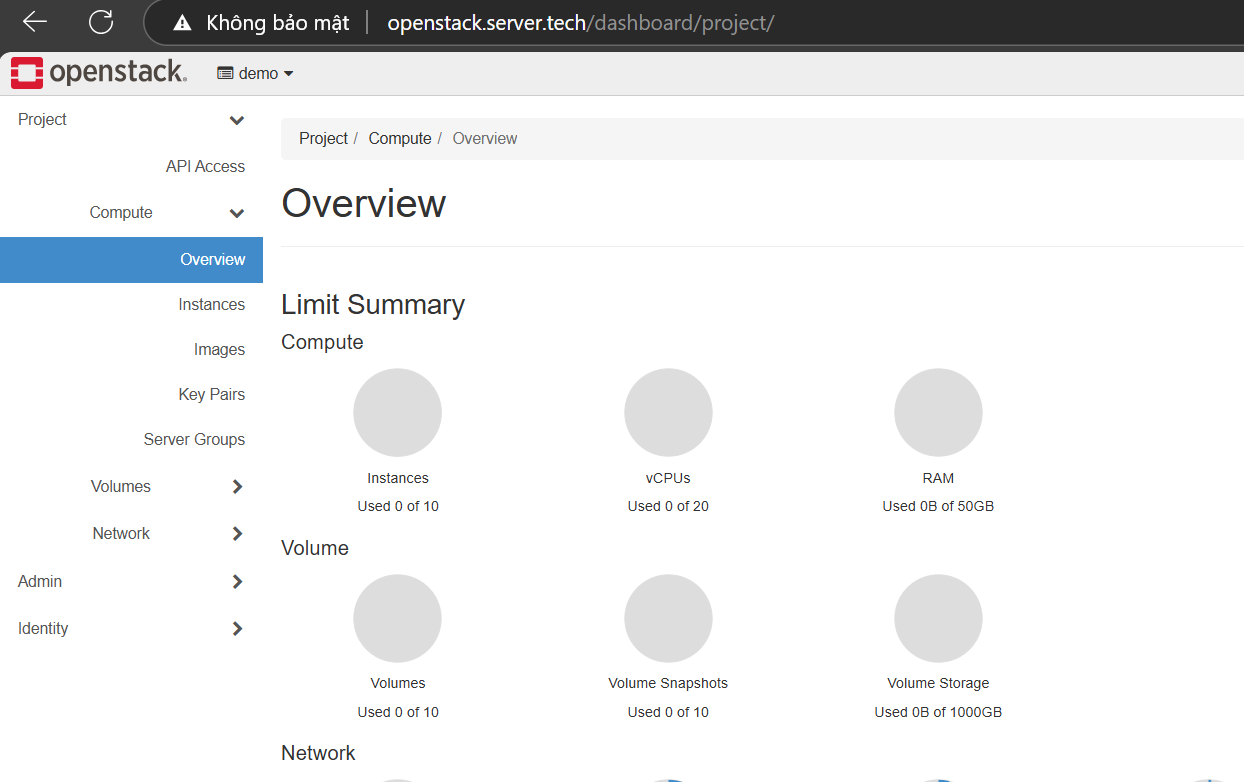


Figure .2: Openstack Management Web

You should see Openstack Management Web console after logging in.

If you want to use Openstack command-line tool to manage your devstack. You've arrived in your shell.source openrc

### ****Configuration -**** Cinder ****with LVM****

LVM is a flexible disk management system on Linux, allowing:

* Create virtual drives (Logical Volumes) from multiple physical drives (Physical Volumes).
* Dynamic resize (expand/reduce) without stopping the system.
* Snapshot, thin provisioning (provision capacity based on actual needs).

In OpenStack (Cinder), LVM is used to manage volumes (virtual disks) for virtual machines.

When configuring devstack installation, the system automatically generates lvmdriver-1, but due to low performance and limited capacity of only 30GB, using only lvmdriver-1 causes the Cinder service to fail, so we will create an additional lvmdriver-2 with high performance and a capacity larger than 60GB to use for creating virtual machine volumes.

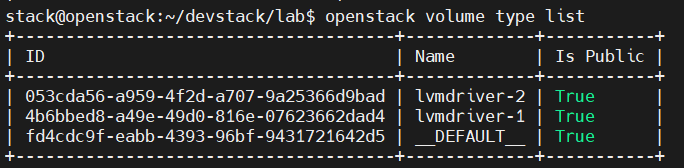


Figure .3: Create lvmdriver-2

### Network Configuration – Neutron

#### Check network on system

Execute the commands to check ip, check ovs: ip addsudo ovs-vsctl show



Figure .4: Current network information

Summary of network status information:

* ens33: Static configuration, used to access the Internet via gateway 192.168.254.2. Has IP address 192.168.254.200.
* ens34: Added to Open vSwitch br-ex bridge. Does not use DHCP.
* br-ex : Has IP 172.24.4.1/24 – often used as external network in OpenStack.
* br-int : Internal bridge, used for communication between components in OpenStack.

#### Check NAT Rules

Check the routing to see if br-ex is the gateway for VMs to access the Internet via ens33.

**Use the command: sudo iptables –t nat –L –n -v**

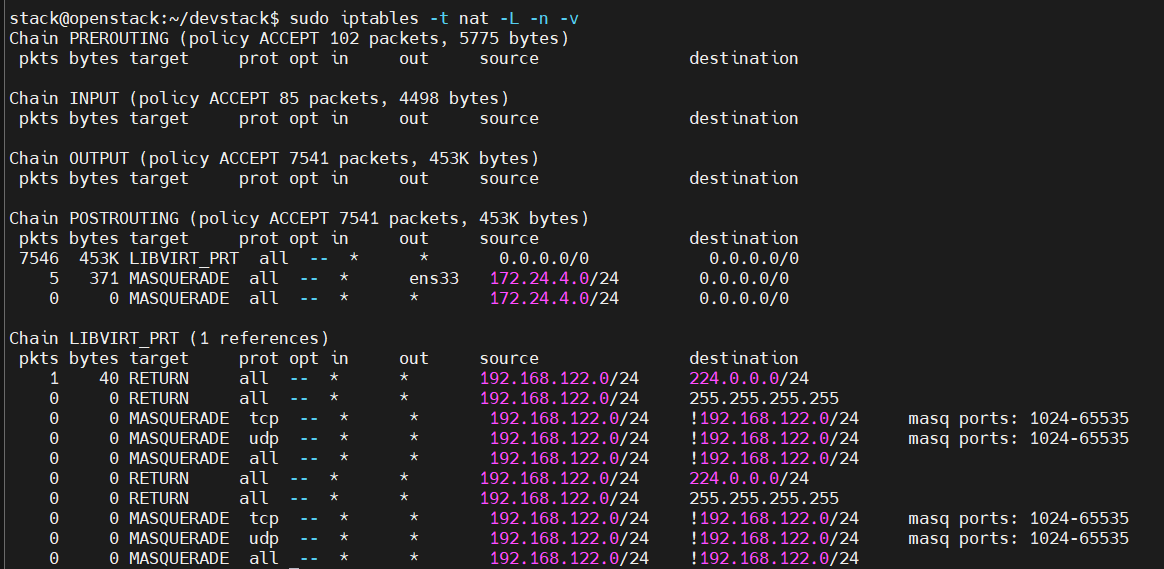


Figure .5: NAT Rule

Shows nat array 172.24.4.0/24 via port ens33

## System configuration and demo

### Overview model

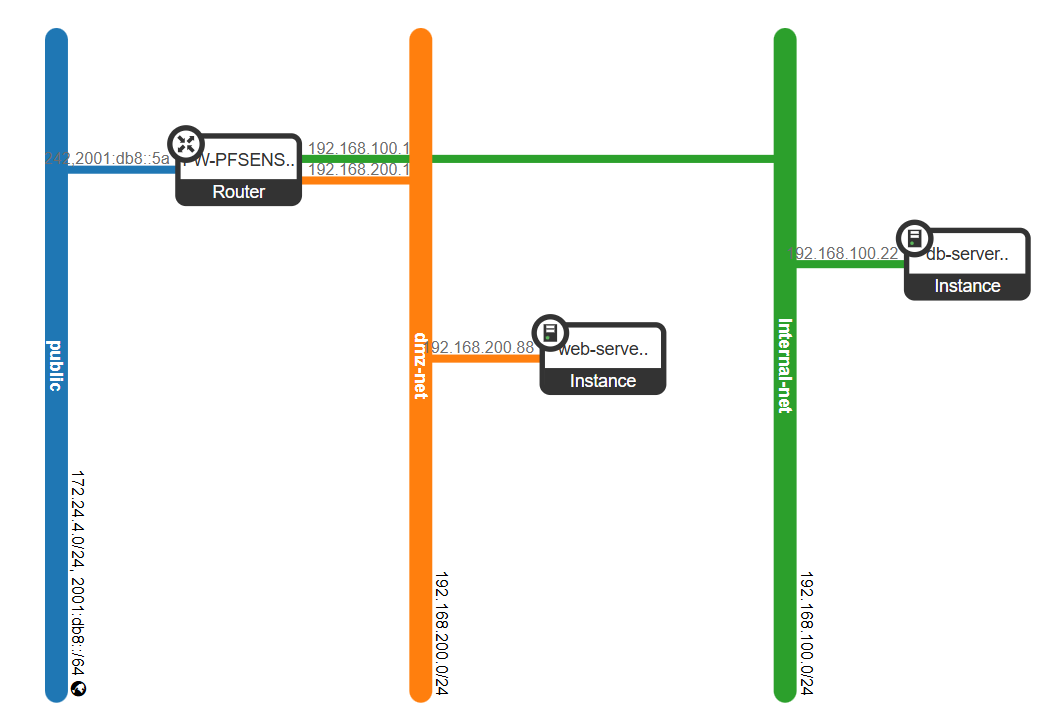


Figure .1: System model

#### Detailed network partitioning:

|  |  |  |
| --- | --- | --- |
| Network name | IP range | Purpose |
| Public Network | 172.24.4.0/24 | Provide Floating IP for instances.  Connect to the Internet via NAT Gateway. |
| DMZ Network | 192.168.200.0/24 | Host public services (Web Server).  Apply Security Groups to open only port 80/443/22. |
| Internal Network | 192.168.100.0/24 | Contains internal systems (Database, Backup).  Only allow access from DMZ via port 3306. |

#### List of devices and roles

|  |  |  |
| --- | --- | --- |
| Equipment/Components | Role | IP/Specifications |
| OpenStack Controller | Manage Nova, Neutron, Cinder, Glance services | 192.168.254.200 |
| Storage Backend | Volume storage (LVM driver-1/driver-2) | 2x60GB disk |
| Router | Inter-network routing + NAT/Firewall | WAN: 172.24.4.0/24  LAN: 192.168.200.1  LAN2: 192.168.100.1 |
| Web Server | Web application hosting (Nginx + Spring Boot) | 192.168.200.88 |
| Database Server | Internal data storage (MariaDB) | 192.168.100.22 |

### Create Project and User

We create a project named Hybrid\_Cloud then create a user named anphuc then add it to the project and set the highest permission as admin for this project.

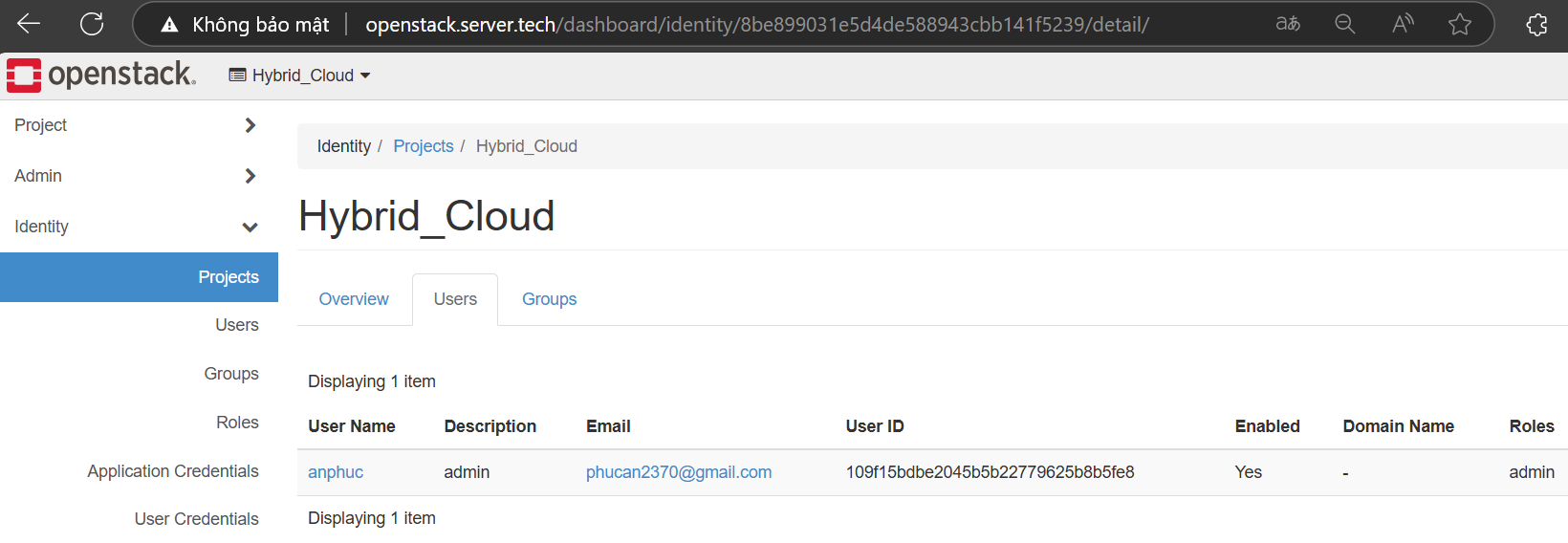


Figure .2: Create project and user

Then log in to anphuc and create configuration in this project.

### Create Networks

Use command or create directly on web interface for easy viewing

* openstack network create Internal-net
* openstack subnet create Internal-subnet --network Internal-net --subnet-range 192.168.100.0/24
* openstack network create dmz-net
* openstack subnet create dmz-subnet --network dmz-net --subnet-range 192.168.200.0/24

Assign the external (public) network as the primary gateway for the FW-PFSENSE router so that virtual machines can access the Internet.

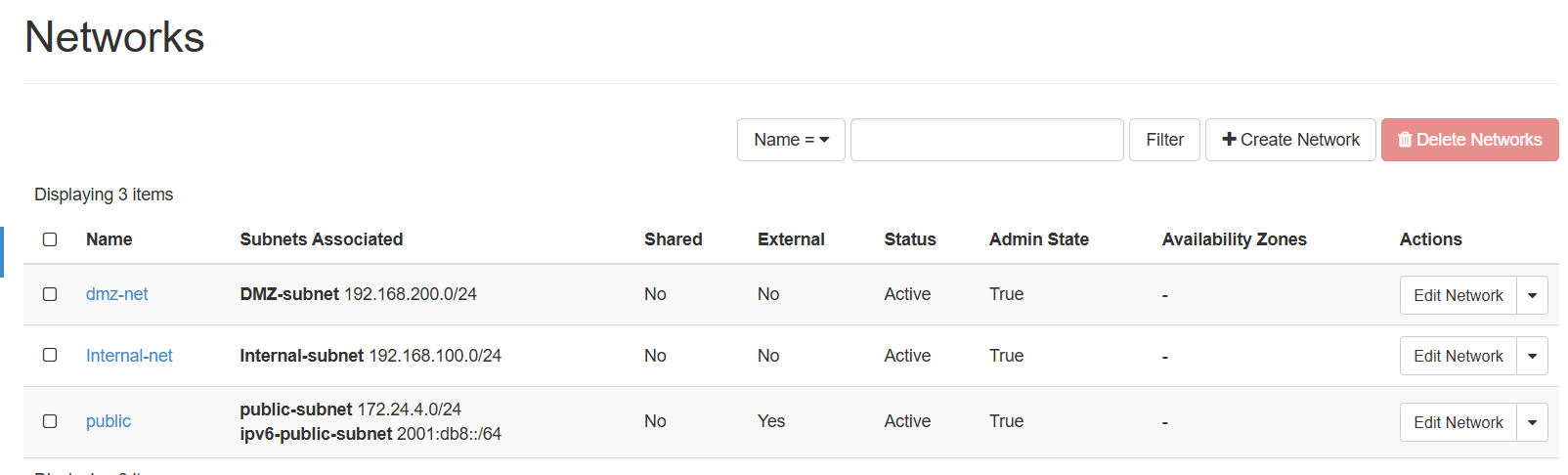


Figure .3: Networks

### Create route

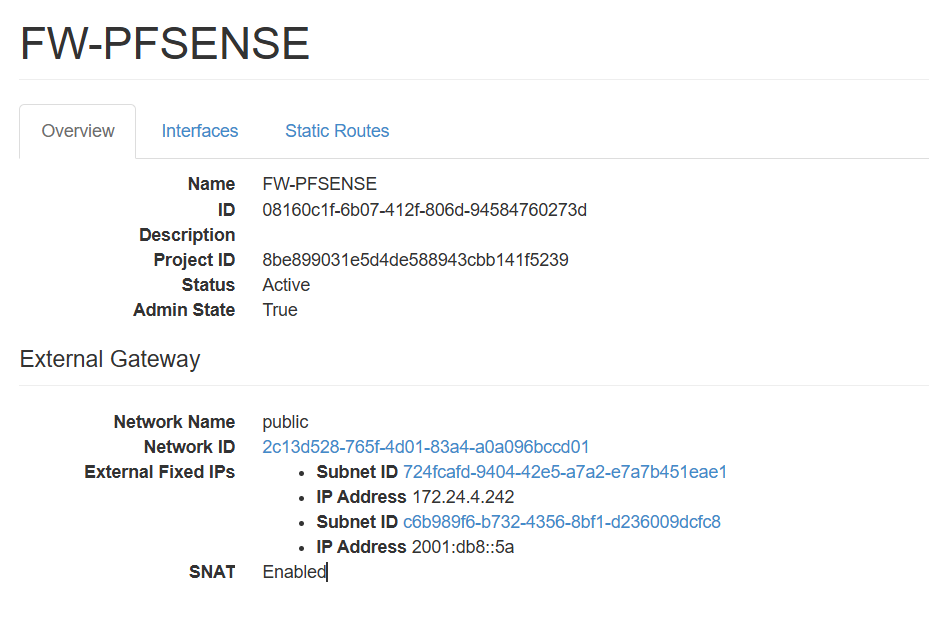


Figure .4: Router configuration

#### Command used to create and attach routers to other networks

* openstack router create FW-PFSENSE
* openstack router set FW-PFSENSE --external-gateway public
* openstack router add subnet FW-PFSENSE Internal-subnet
* openstack router add subnet FW-PFSENSE dmz-subnet

#### Set up routing

Set routing to 0.0.0.0/0 (Internet) via net hop 172.24.4.1/24

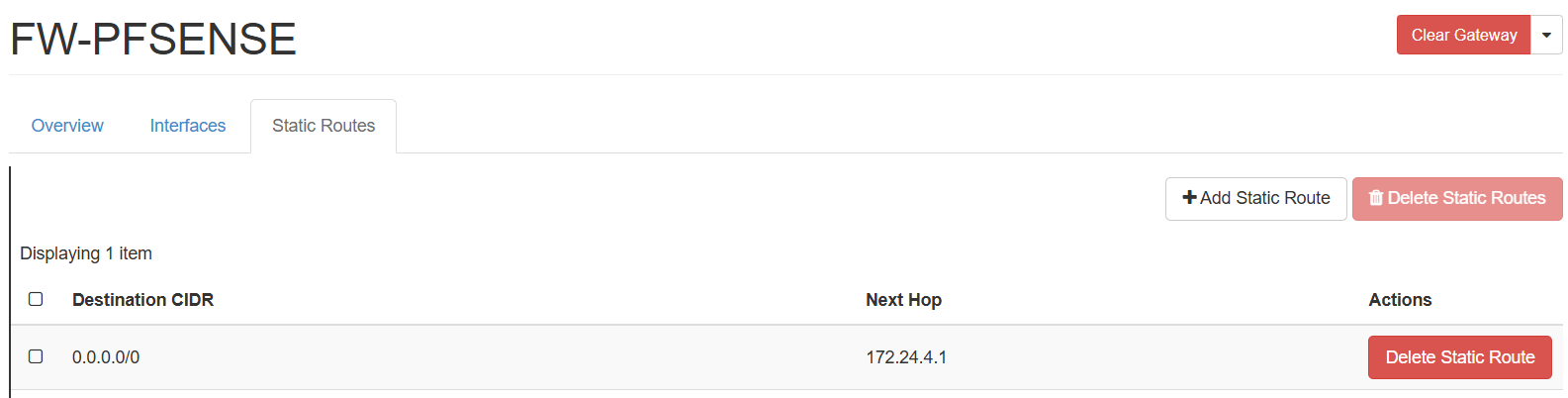


Figure .5: Routing setup

### Service Deployment

#### Create Image (Glance)

Image is a template containing the basic operating system/configuration to initialize instances.

Helps to quickly deploy multiple virtual machines in the same environment.

Download the debian image and upload the image to OpenStack

* openstack image create \
* --file ./debian-12-genericcloud-amd64.qcow2 \
* --disk-format qcow2 \
* --container-format bare \
* --public \
* --property name='Debian-12' \
* --property description='Debian 12 for demo' \
* debian-12

Check image:

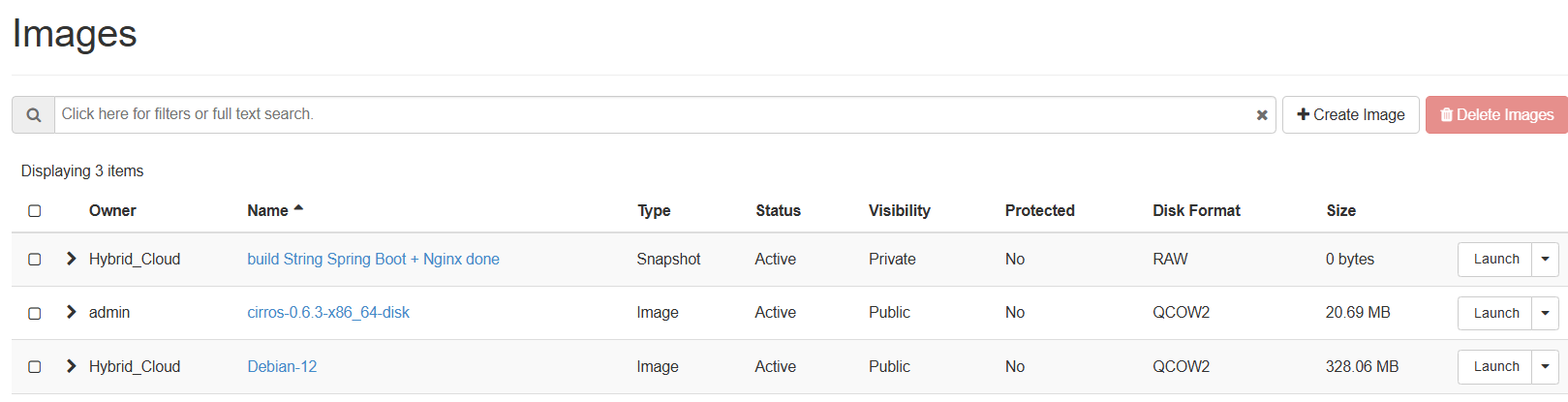


Figure .6: Image

#### Create Volume from Image (Cinder)

A volume is a separate disk that stores data independently of the instance lifecycle. Allows instantiation of instances from volumes (persistent storage).

* openstack volume create \
* --size 20 \
* --image Debian-12\
* --type lvmdriver-2 \
* vol-dbserver-data

--image: Specifies the template to initialize the volume.

--type lvmdriver-2: Use extended LVM backend.

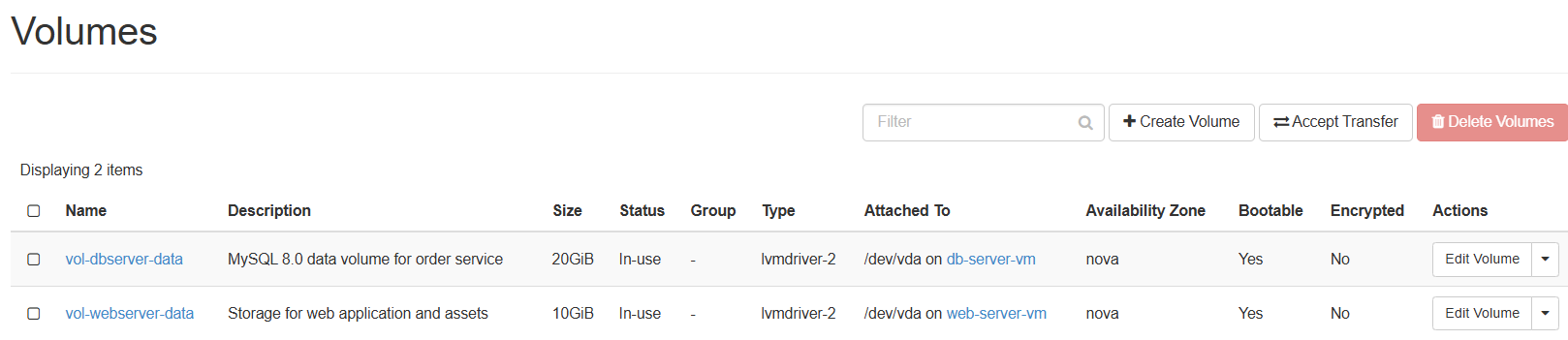


Figure .7: Create volume to store data

### Initialize Key Pairs

Command: openstack keypair create ssh-key > ssh-key.pem

Then edit the access restrictions:

chmod 400 ssh-key.pem

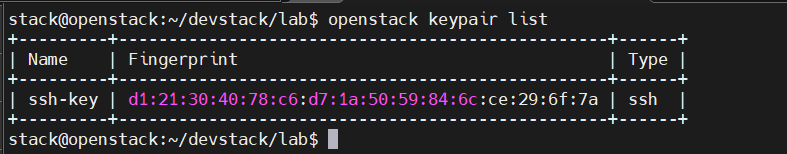


Figure 3.3.8: ssh keys

### Initialize Instance from Volume (Nova)

Virtual machine uses volume as root disk → data persists even when instance is deleted.

* openstack server create \
* --volume vol-dbserver-data \
* --flavor m1.small \
* --network Internal-net \
* --security-group db-access \
* --key-name ssh-key \
* db-server-vm

Parameter explanation:

--volume: Mount volume as primary drive.

--flavor: CPU/RAM configuration ( 1 vCPU, 2GB RAM)

--key-name: SSH key to access the instance.

Check instance:

Use the command “openstack server list” to check the existing instances

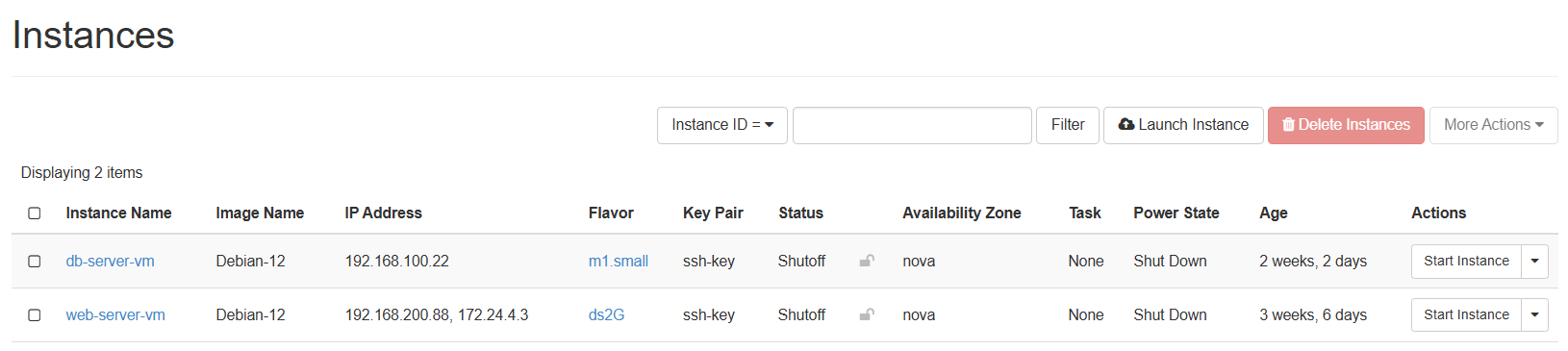


Figure .9: View created instances on web interface

### Assign Floating IP (Neutron)

Purpose:

* Allows instance access from the Internet via public IP.
* Open public IP to allow access to the website from outside

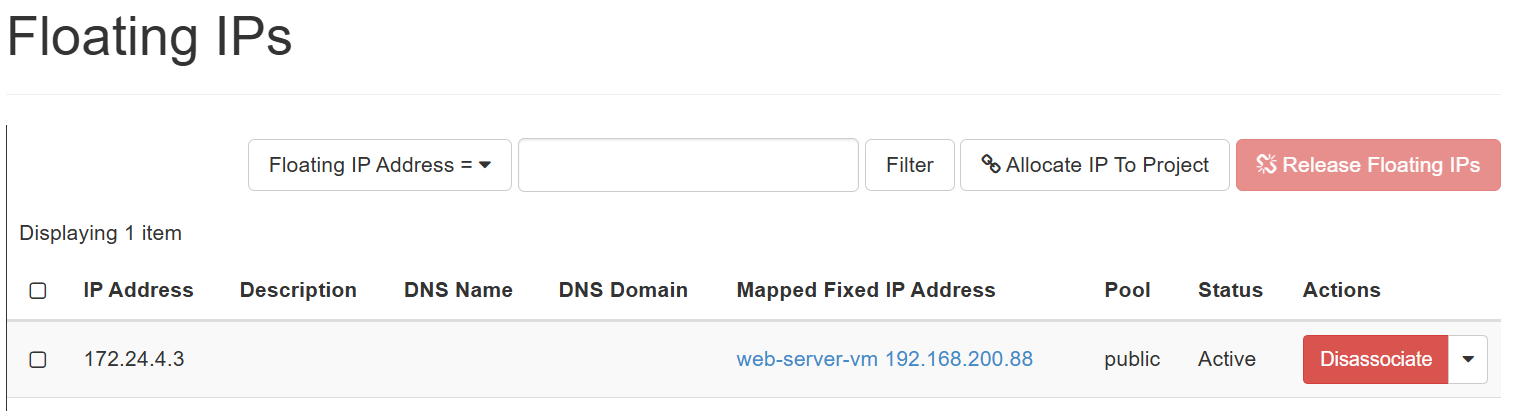


Figure .10: Floating IPs

### System Demo

#### Graph model of the system

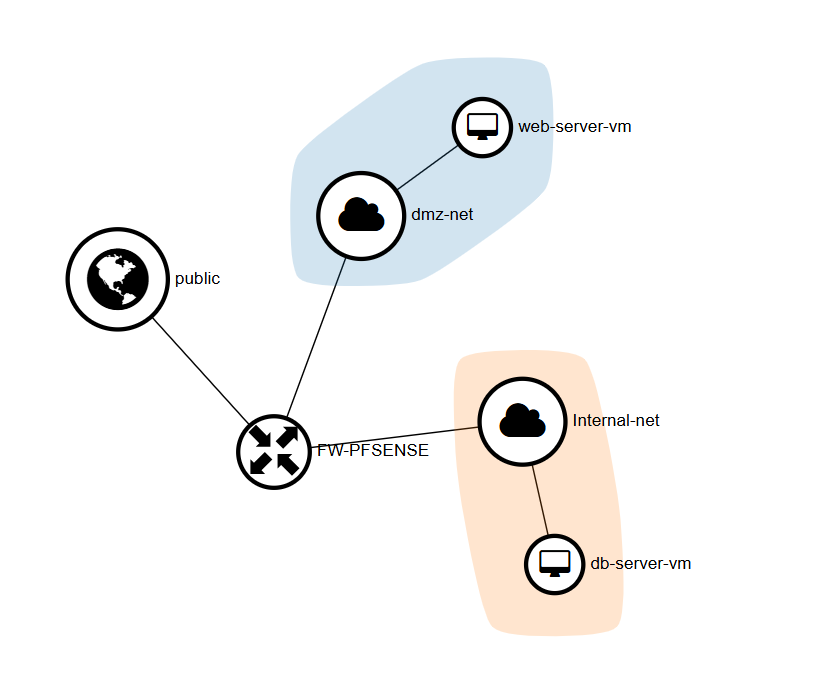


Figure .11: Graph model

#### Check connection

Start 2 web-server and db-server instances:

Test the connection between two devices using ICMP protocol.

Perform ping from webserver to db-server

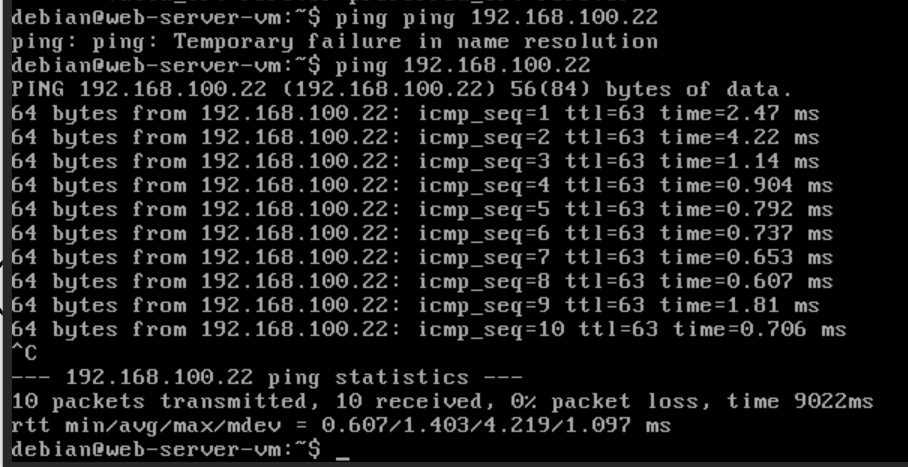


Figure .12: Perform ping from webserver

Result: Ping from web-server to db-server is successful, icmp rule is active and router is routing correctly through 2 networks.

ssh test from openstack host to web-server to check rule, ssh is working properly or not.

**Use the command:** ssh –i ssh-key.pem debian@172.24.4.3

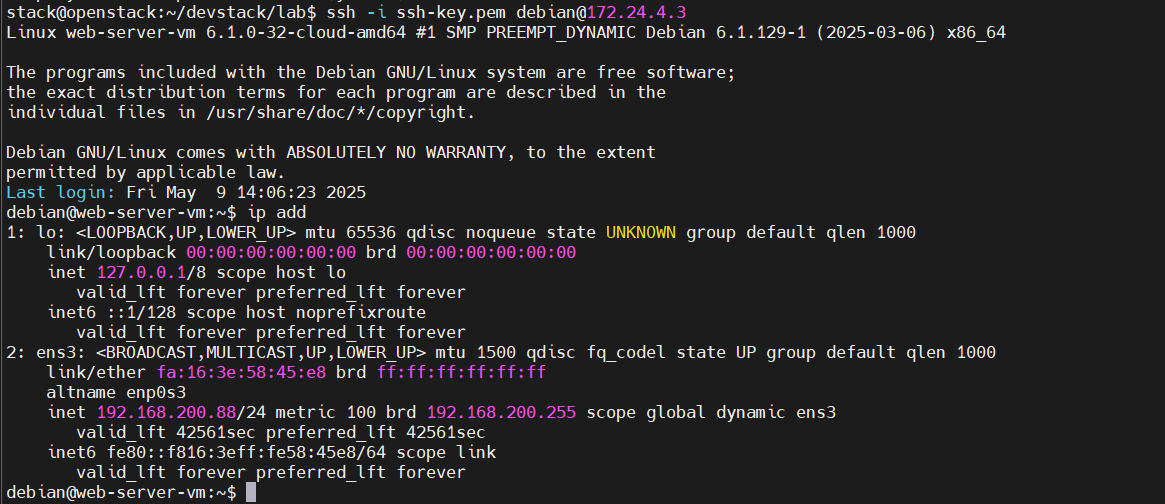
****

Figure .13: ssh connection

Result: access successful, security group is working properly and ssh is enabled

#### Access to webserver

From a machine outside the internet, access the webserver to check if the connection to the public address has been granted.

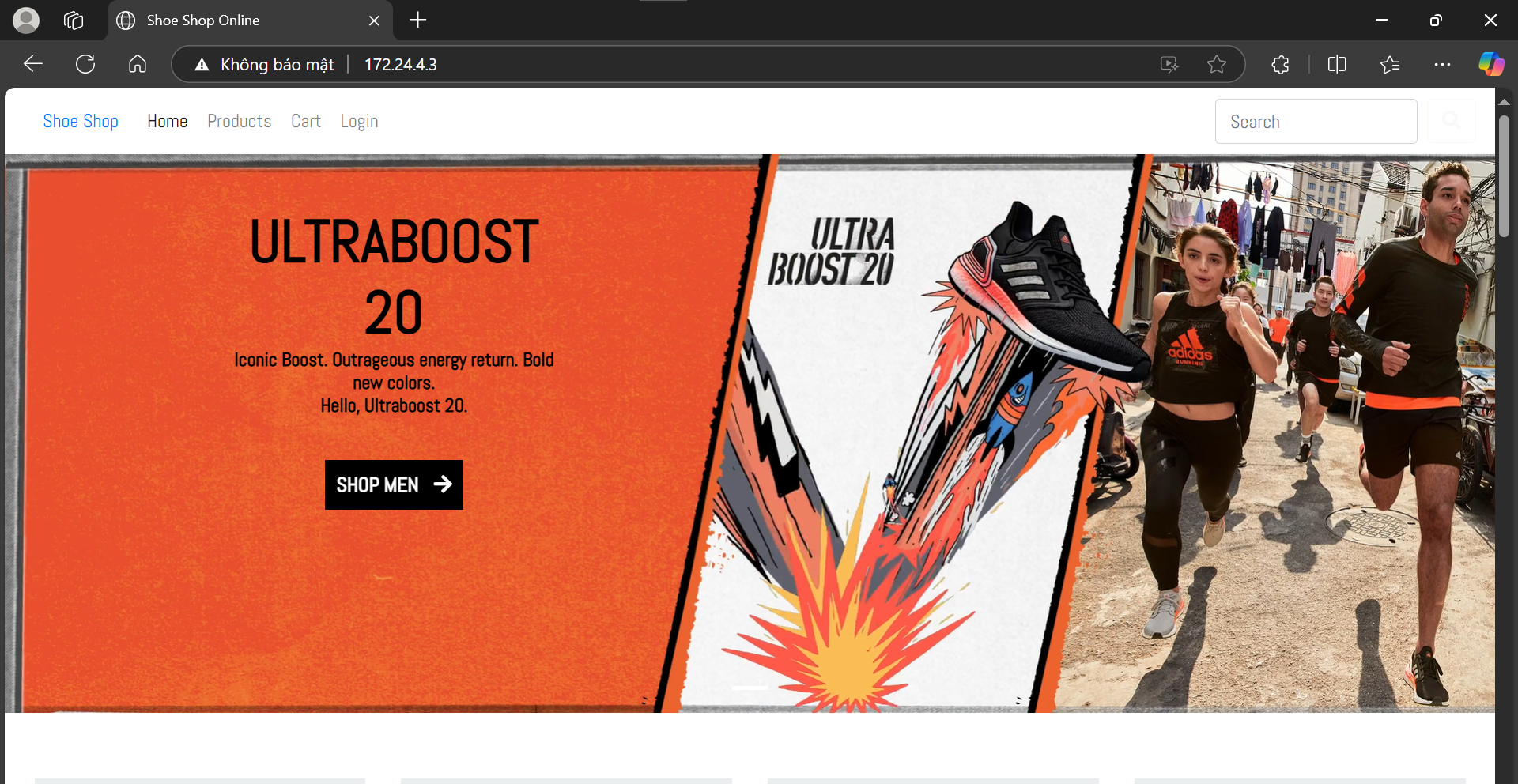


Figure .14: Web-server web interface

Result: The website is now accessible from outside the internet. Floating IPs are working properly.

Check the login to see if you can connect to the db-server.

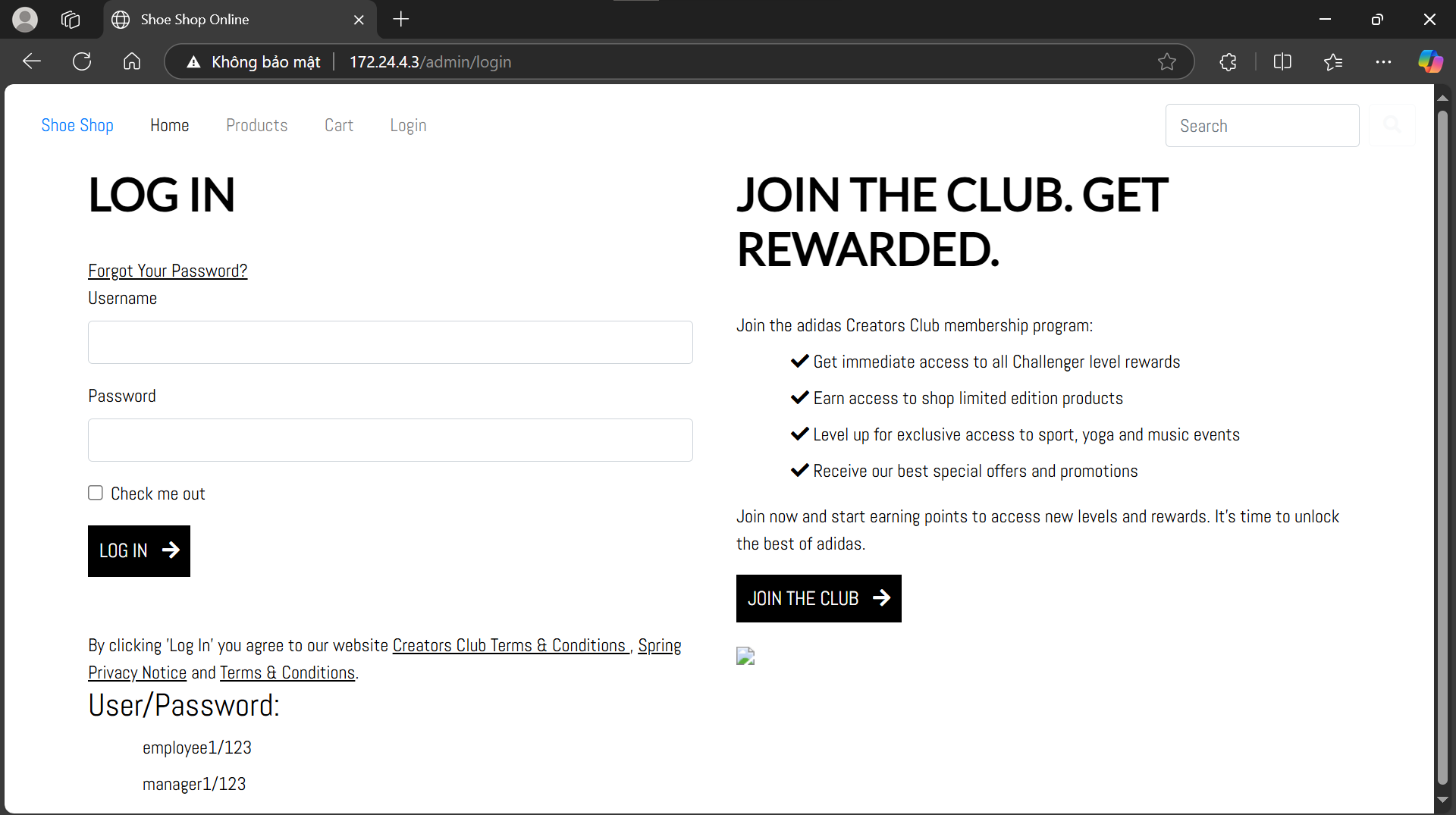


Figure .15: Login page interface

Log in with user manager1 and password 123

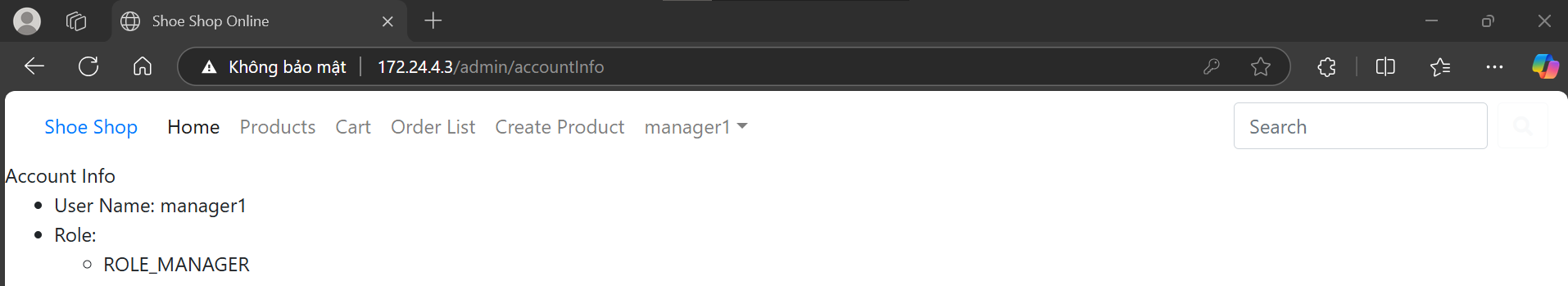


Figure .: Login with user manager1

Result: Login successful, proving that the db-server has provided data to the web-server.

## End of chapter 3

In this chapter, the OpenStack Hybrid Cloud system has been successfully deployed on VMware Workstation virtualization environment using DevStack tool. The deployment process includes infrastructure configuration, installation of OpenStack core services such as Keystone, Glance, Nova, Cinder, Neutron and Dashboard Horizon. The system uses Open vSwitch to build a flexible network architecture with three clear partitions: Internal, DMZ and Public , ensuring a three-layer network model that is popular in practice.

In addition, the system also integrates an LVM storage solution with two separate backends, ensuring higher performance for volumes created from images. The demo process demonstrated the system's stable operation by initializing, connecting and accessing services such as Web Server and Database Server from the Internet via Floating IPs.

However, the DevStack usage model is mainly suitable for research and testing purposes, due to its limitations in long-term maintainability and stability in production environments. In the following chapters , this model can be extended or upgraded to suit more realistic deployment scenarios such as using OpenStack in production with distributed services.

CONCLUDE

1. RESULTS ACHIEVED

Through the process of implementing the topic "Building an OpenStack Hybrid Cloud system", I would like to draw some conclusions as follows:

**FAVORABLE**

Enhanced professional knowledge: During the course, I have a better understanding of concepts related to virtualization, cloud computing, OpenStack architecture, SDN networking and hybrid cloud models.

Strong documentation and community: OpenStack has detailed official documentation and a large development community, making it easy to research and troubleshoot.

Flexible practice environment: Deployment on VMware Workstation with DevStack fully simulates the necessary components of a cloud system, making it easy to configure and test.

**HARD**

Limited system resources: Running multiple OpenStack components on virtual machines requires high hardware resources (CPU, RAM), making it difficult to deploy on personal machines.

Complex configuration: Some services like Neutron, Cinder require a good understanding of virtual network systems (bridge, OVS) and storage partitions, leading to long configuration times and error-prone.

Time constraints: Large workload with multiple test steps requires a lot of time to implement and document the results fully.

1. ACHIEVED AND UNACHIEVED ASPECTS

**The achieved side**

Successfully deployed cloud system: OpenStack system fully deployed components such as Keystone, Nova, Neutron, Glance, Horizon, Cinder, allowing creation and management of virtual machines.

Hybrid cloud network setup: A layered network model including public, DMZ and internal was successfully built and tested.

Configure storage and boot from volume: Volumes are created successfully and virtual machines can boot from volumes as in a production environment.

Practical application: The deployment model can be expanded and applied as a lab system for training or small business environment.

**Unsatisfactory face**

No HA (High Availability) configuration: The current model does not integrate HA clustering for important services such as Keystone or Controller Node.

Lack of advanced monitoring integration: Monitoring systems like Prometheus, Zabbix, or performance alerting systems have not been implemented.

No failure scenarios simulated: Due to time constraints, failure scenarios such as service disconnection, downtime, or disaster recovery have not been tested.

1. DIRECTIONS FOR EXTENDED RESEARCH

To improve and enhance the quality of the system in the future, I propose the following development directions:

Monitoring System Integration: Connect OpenStack with tools like Zabbix, Prometheus, or Grafana to monitor resource and service performance in real-time.

Build HA cluster: Deploy High Availability for Controller and core services using Pacemaker/Corosync or Keepalived to ensure system availability.

Deployment Automation: Use tools like Ansible, Terraform, or Kolla to automate the process of installing and updating OpenStack systems.

Enhanced security: Add security layers like firewalls, IDS/IPS (e.g. Suricata) or more granular authorization in Keystone.

REFERENCES

[1] OpenStack Documentation – https://docs.openstack.org

[2] DevStack – A Tool for Installing OpenStack – https://docs.openstack.org/devstack/latest/

[3] Open vSwitch (OVS) – https://docs.openvswitch.org/en/latest/

[4] OpenStack Networking Guide – https://docs.openstack.org/neutron/latest/

[5] Volume Service (Cinder) – https://docs.openstack.org/cinder/latest/

[7] Prometheus Documentation –https://prometheus.io/docs/introduction/overview/

[8] OpenStack HA Guide – https://docs.openstack.org/ha-guide/latest/