CLOUD COMPUTING LAB

ASSIGNMENT 4:

OPENSTACK CLOUD INSTALLATION AND VIRTUAL MACHINE SETUP

NAME : SWAPNADEEP MISHRA

ROLL NO : 002211001115

GROUP MEMBER:
SAMUDRA ROY
(002211001114)

SECTION NO : A3

OpenStack Cloud Installation and Virtual Machine Setup Report

Objective:

The objective of this assignment is to familiarize students with the installation and configuration of OpenStack, a cloud computing platform, and to create and manage Virtual Machines (VMs) using OpenStack.

Task Overview:

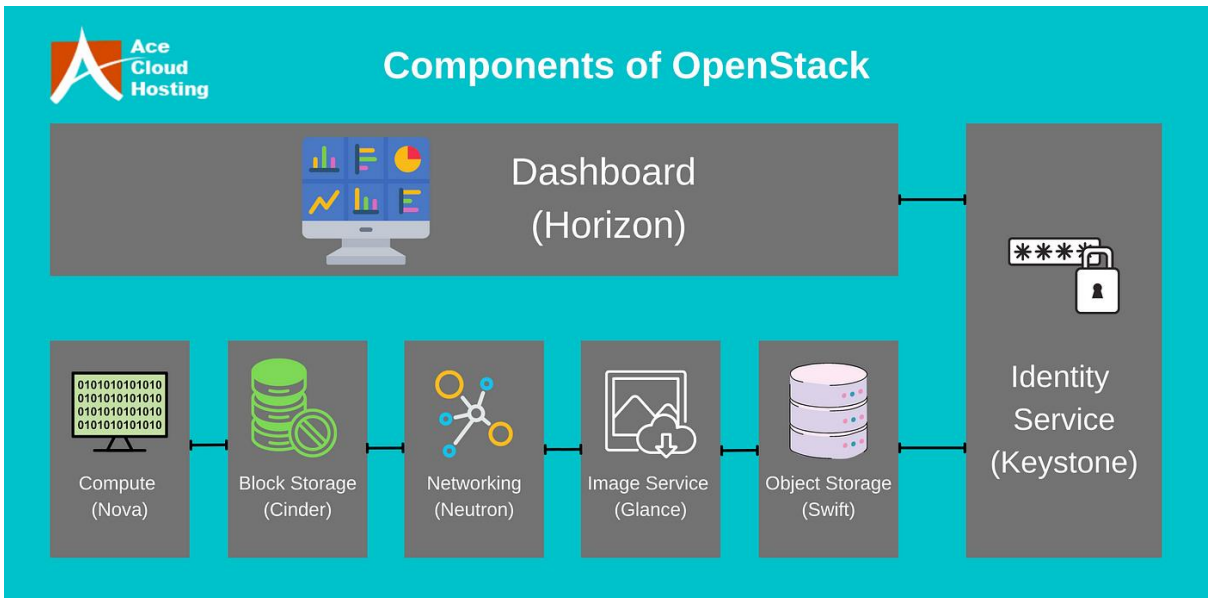
Install OpenStack using DevStack or PackStack on a local or virtual environment.
Set up a basic cloud infrastructure, including network configurations.
Create and manage at least two Virtual Machines (VMs) on the OpenStack cloud.
Demonstrate the functionality of the OpenStack cloud, including launching, pausing, and deleting VMs.
Submit a report documenting the process, challenges faced, and solutions implemented.

Introduction

OpenStack is a popular open-source cloud computing platform that allows users to create and manage cloud infrastructures. The platform consists of a number of components, including:

- **Keystone**: OpenStack's identity service for authentication.
- **Nova**: The compute service responsible for managing virtual machines (VMs).
- **Neutron**: The network service that handles networking for OpenStack.
- **Glance**: The image service that provides discovery, registration, and delivery services for disk and server images.
- **Horizon**: A web-based user interface for managing OpenStack services and components.

In this assignment, the goal is to install OpenStack, set up a basic cloud infrastructure, and create virtual machines (VMs).



Part 1: OpenStack Installation:

Installation Process

Step 1: Choose Installation Method

We chose **DevStack**, which is designed for a lightweight and development-friendly OpenStack installation. DevStack is often used for testing and development environments, making it the ideal choice for this assignment.

Step 2: Install Required Dependencies

We followed the DevStack installation guide to ensure all necessary dependencies were installed.

1. Install system packages and libraries needed for the installation.

```
deep@deep-VirtualBox:~$ sudo chmod +x /opt/stack
deep@deep-VirtualBox:~$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
```

```
deep@deep-VirtualBox:~$ sudo -u stack -i
stack@deep-VirtualBox:~$ git clone https://opendev.org/openstack/devstack
Command 'git' not found, but can be installed with:
sudo apt install git
stack@deep-VirtualBox:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 4,146 kB of archives.
After this operation, 21.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security/main amd64 git-man all 1:2.34.1-1ubuntu1.11 [955 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3,165 kB]
Fetched 4,146 kB in 3s (1,585 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 163563 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.34.1-1ubuntu1.11_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.11) ...
Selecting previously unselected package git.
```

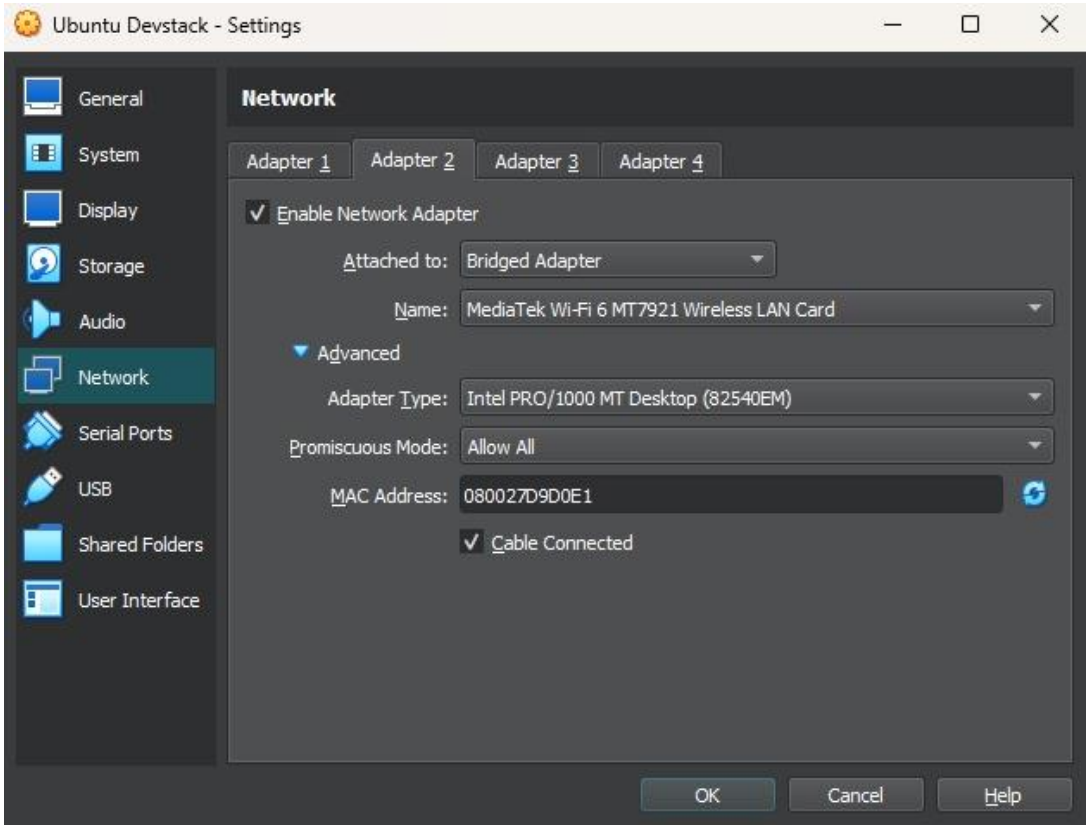
2. Clone the DevStack repository:

```
stack@deep-VirtualBox:~$ git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 51071, done.
remote: Counting objects: 100% (31062/31062), done.
remote: Compressing objects: 100% (10437/10437), done.
remote: Total 51071 (delta 30307), reused 20625 (delta 20625), pack-reused 20009
Receiving objects: 100% (51071/51071), 9.58 MiB | 3.15 MiB/s, done.
Resolving deltas: 100% (36266/36266), done.
stack@deep-VirtualBox:~$ cd devstack
stack@deep-VirtualBox:~/devstack$
```

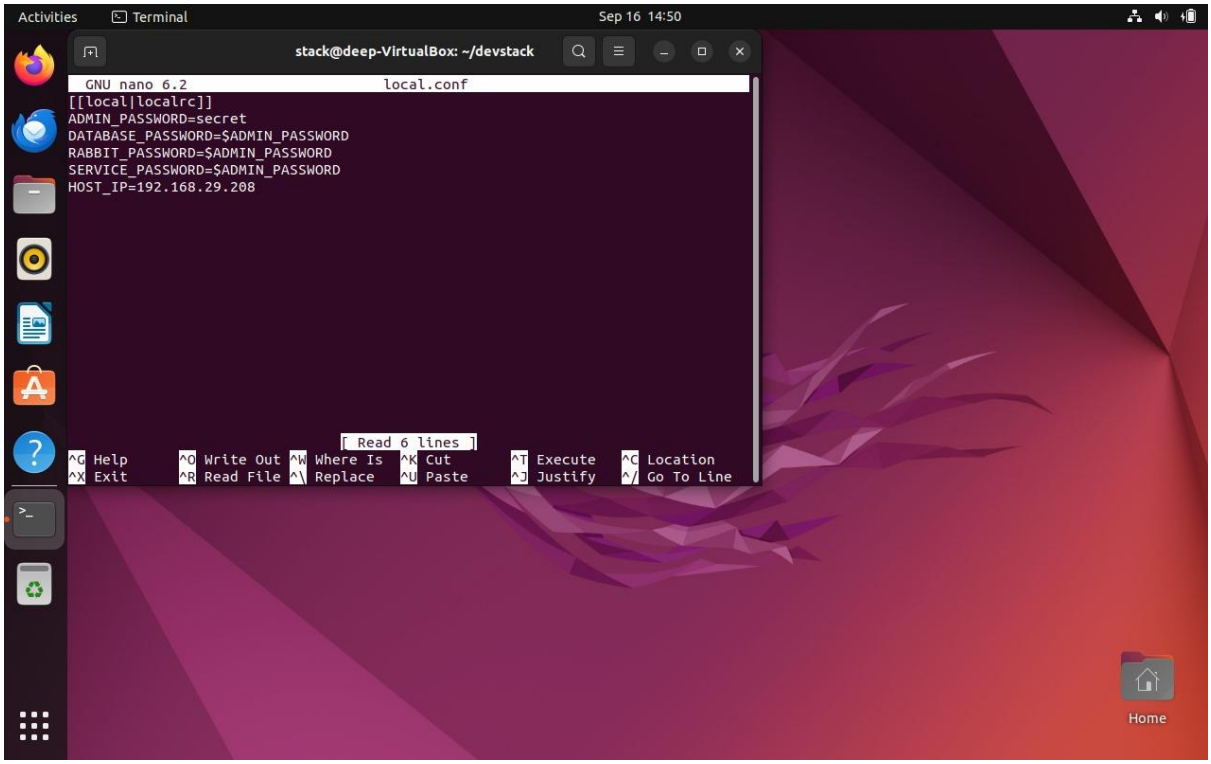
Step 3: Perform Installation on Ubuntu

Initially, I attempted to install DevStack on **Ubuntu 24 LTS**, but encountered errors during the process. After troubleshooting, I switched to **Ubuntu 22.04 LTS (Jellyfish)**, which resolved the installation issues.

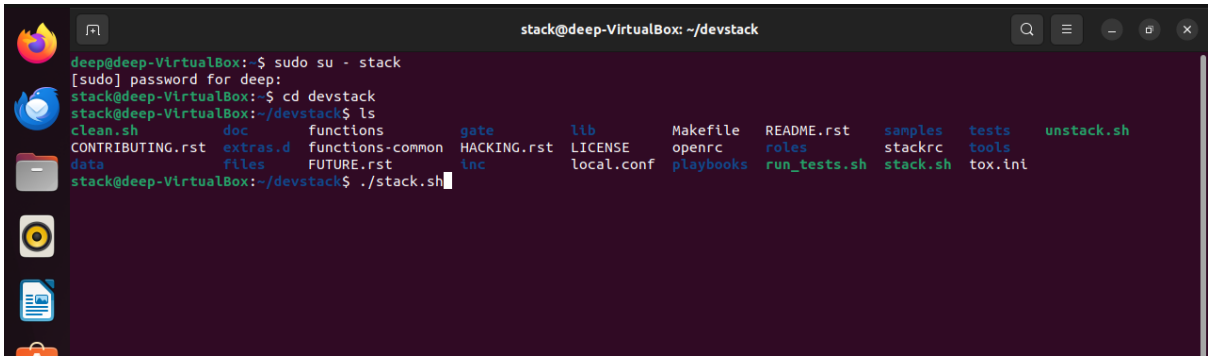
- Configuration of the ubuntu network access and IP setting:



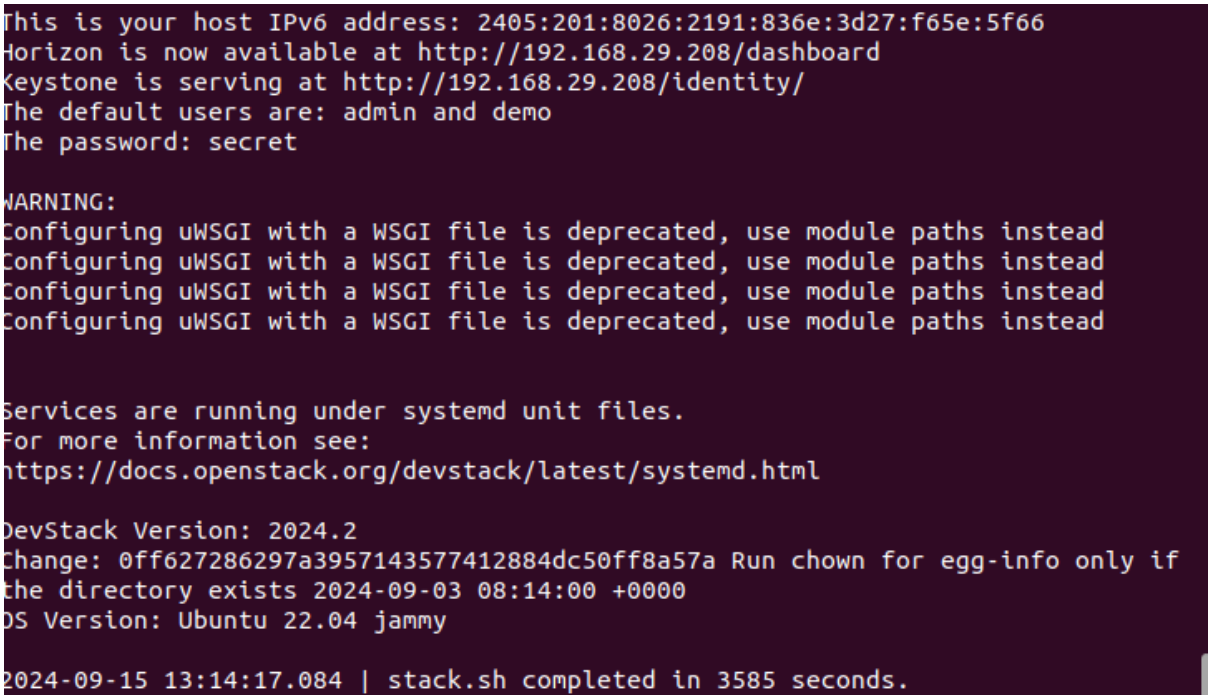
- Created a local.conf file with the following content:



- Ran the installation script:



This successfully installed DevStack and set up the necessary OpenStack components which took around 30-40 minutes of time.

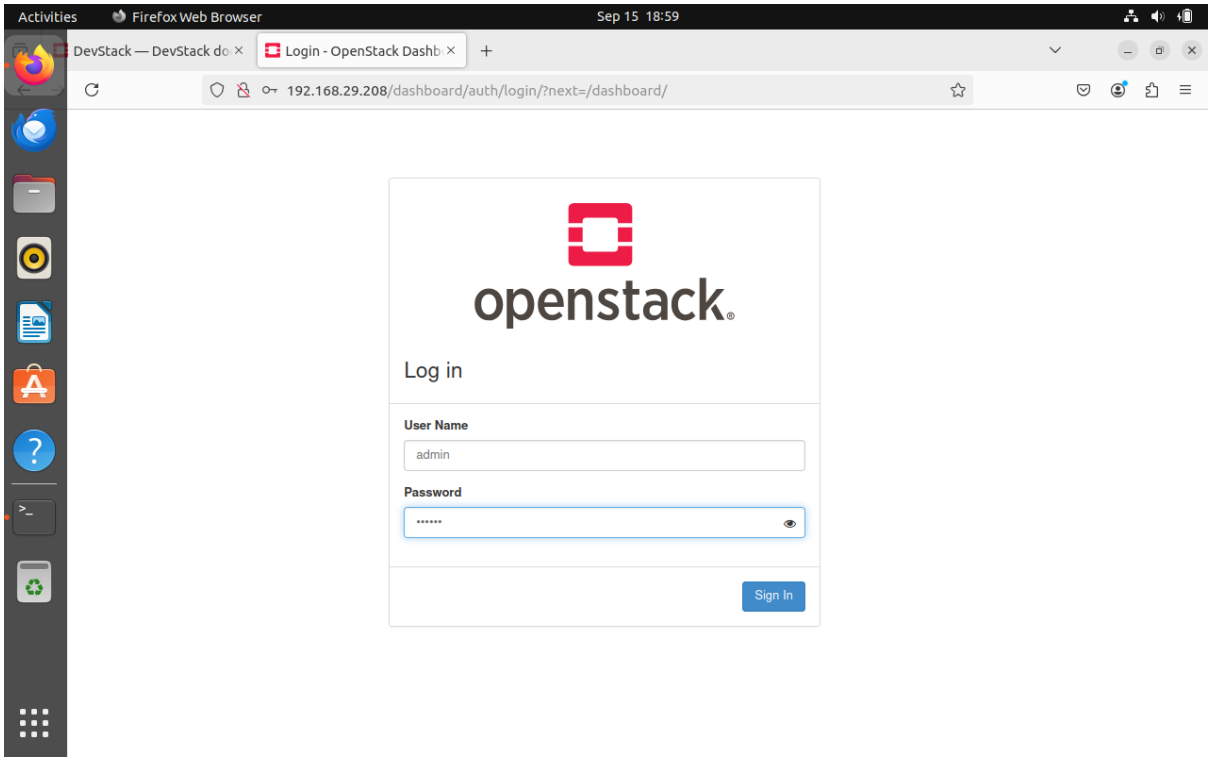


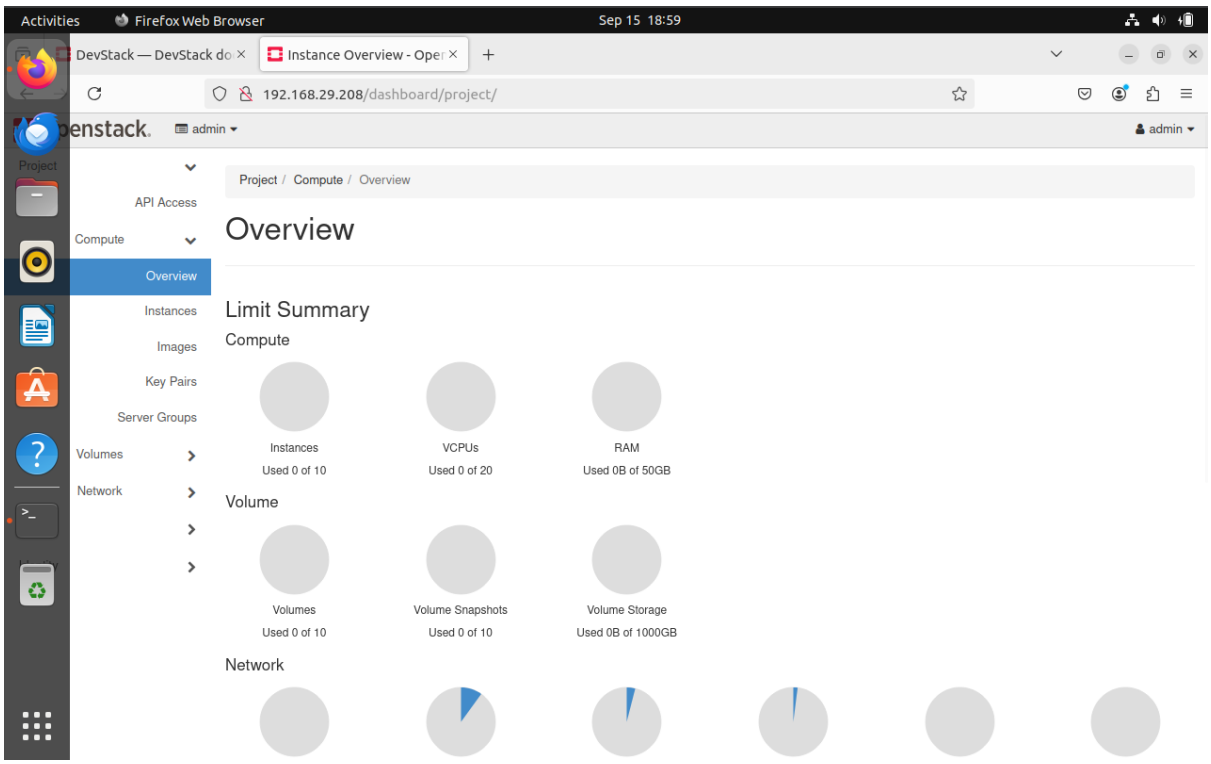
Step 4: Configure OpenStack Components

The key components configured during installation were:

- **Keystone**: Provides identity services for users and projects.
- **Nova**: Handles the management of compute instances.
- **Neutron**: Manages networking between the instances.
- **Glance**: Stores and retrieves VM images.

- Horizon Dashboard of Openstack :

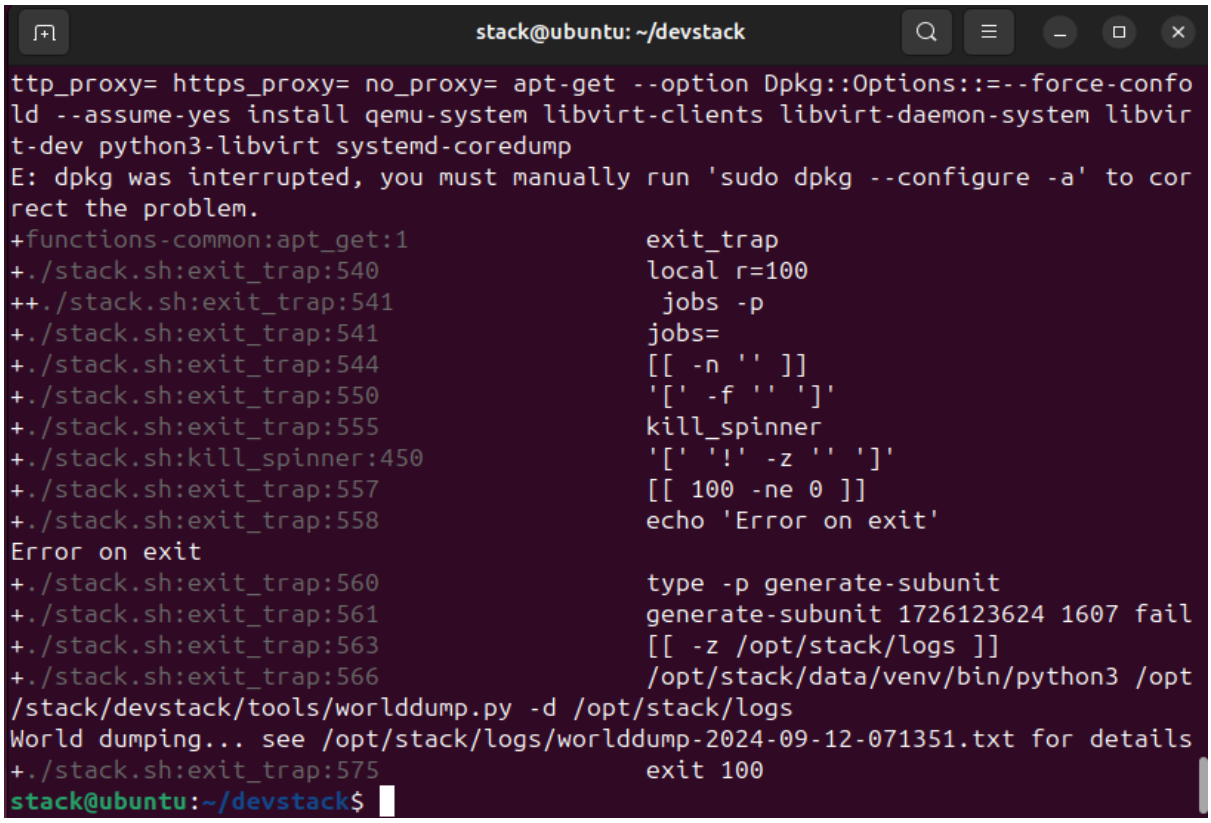


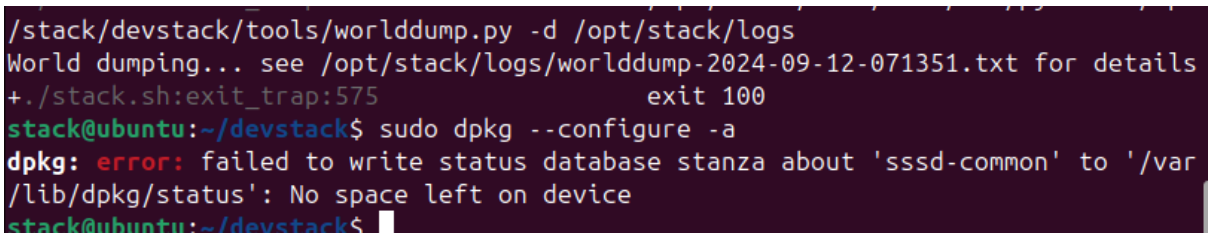


Challenges & Solutions

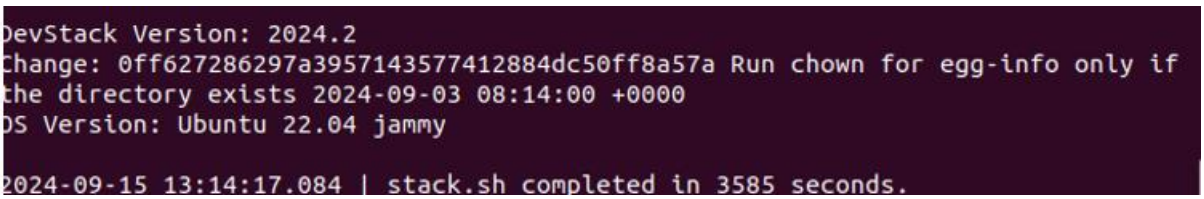
Challenge 1: Installation Errors on Ubuntu 24 LTS

During the installation process on **Ubuntu 24 LTS**, I encountered multiple dependency errors that prevented successful installation. After some research, I found that these issues were due to compatibility problems with DevStack on the latest version of Ubuntu.





Solution: I switched to **Ubuntu 22.04 LTS (Jellyfish)**, which is more stable and compatible with DevStack. This resolved the installation errors, and the process proceeded smoothly afterward.



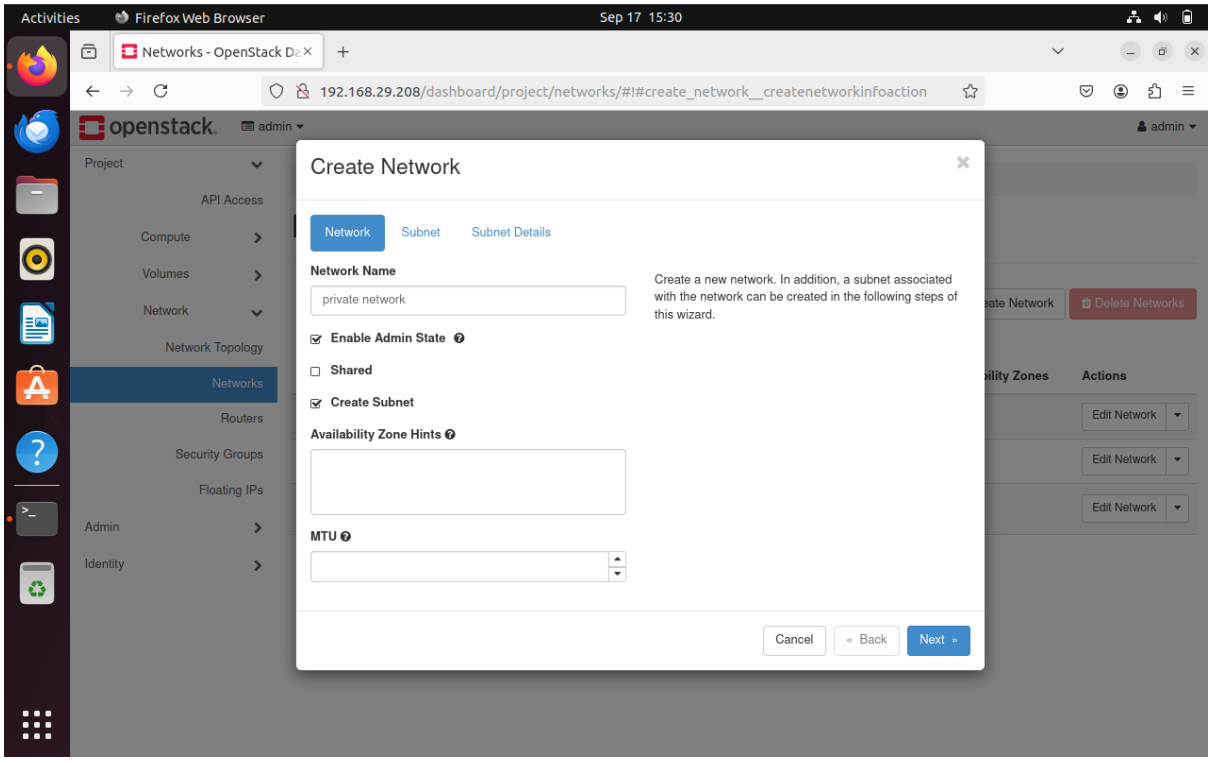
Part 2: Cloud Infrastructure Setup

VM Setup:

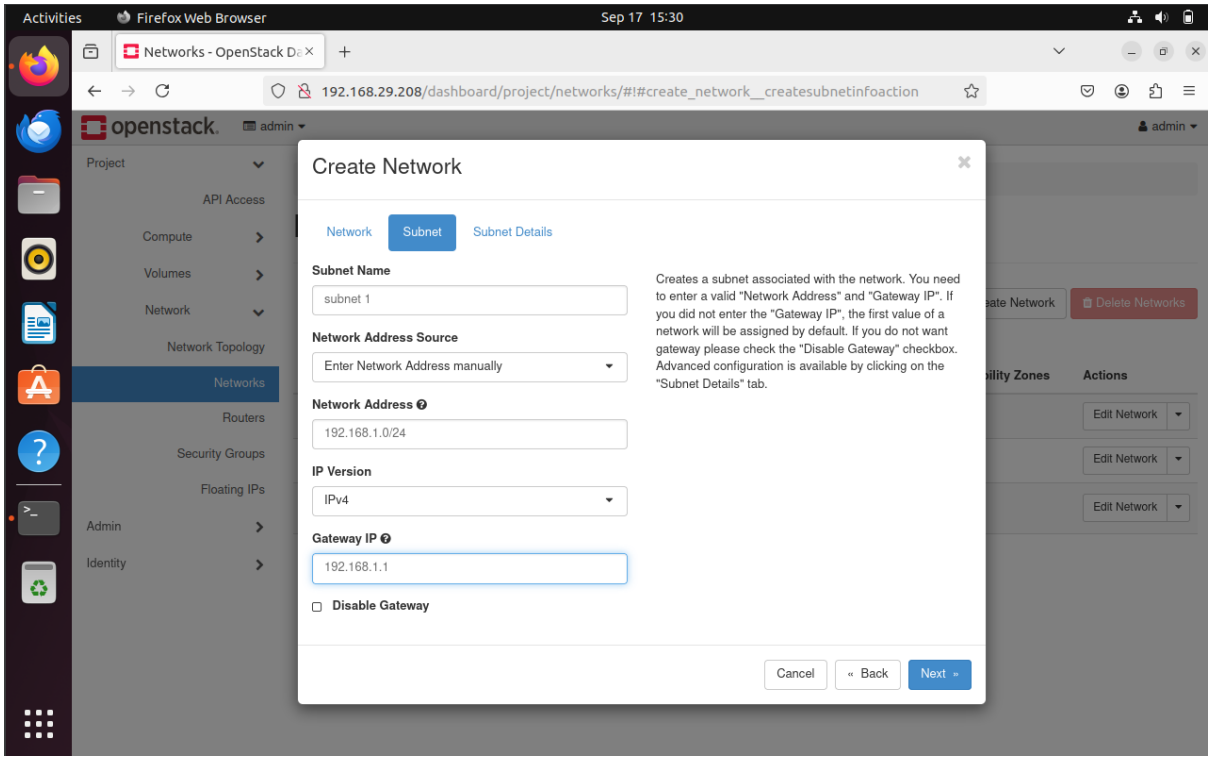
Step 1: Network Configurations

Private Network Setup

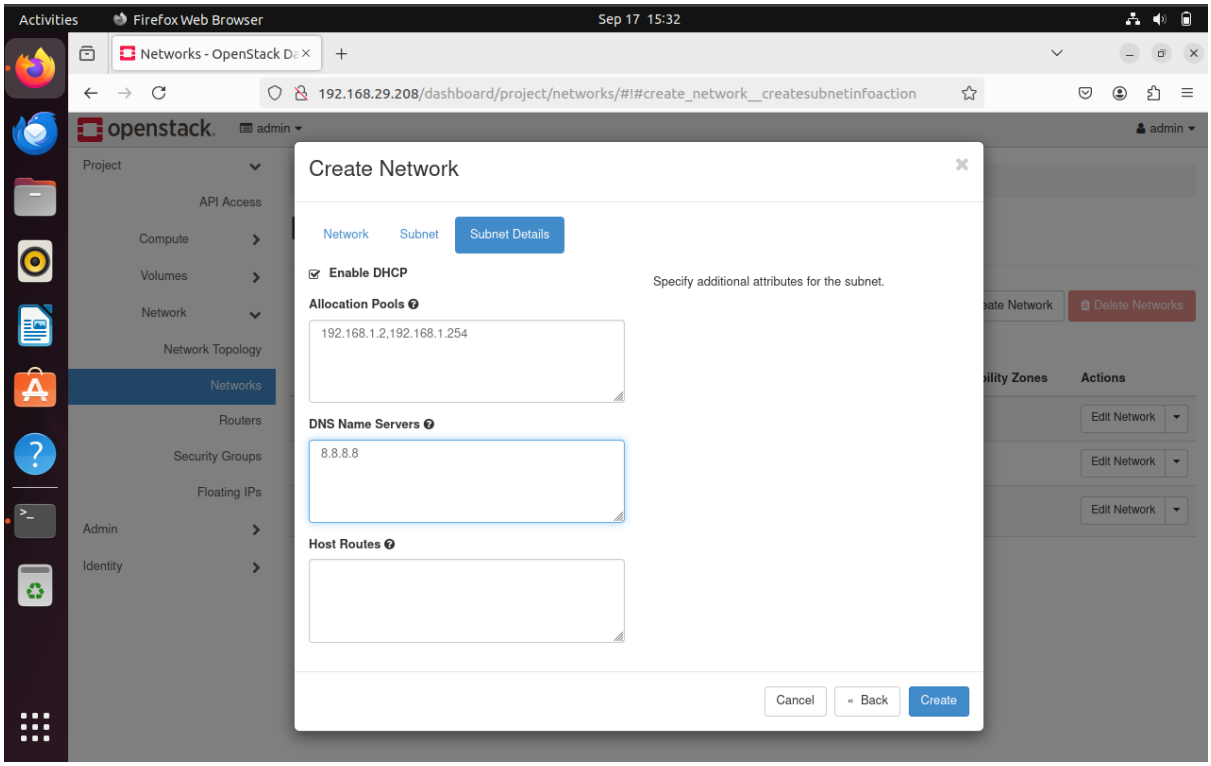
1. **Create a private network** for internal communication between virtual machines:
    - Navigate to the **Horizon Dashboard**.
    - Go to **Project > Network > Networks** and click **Create Network**.
    - Specify the name as `Private-Network`.
    - Set a subnet range (e.g., `192.168.1.0/24`) and configure the DHCP server.



2. **Subnet Creation**:
    - Create a subnet for `Private-Network` with the range `192.168.1.2,192.168.254`.

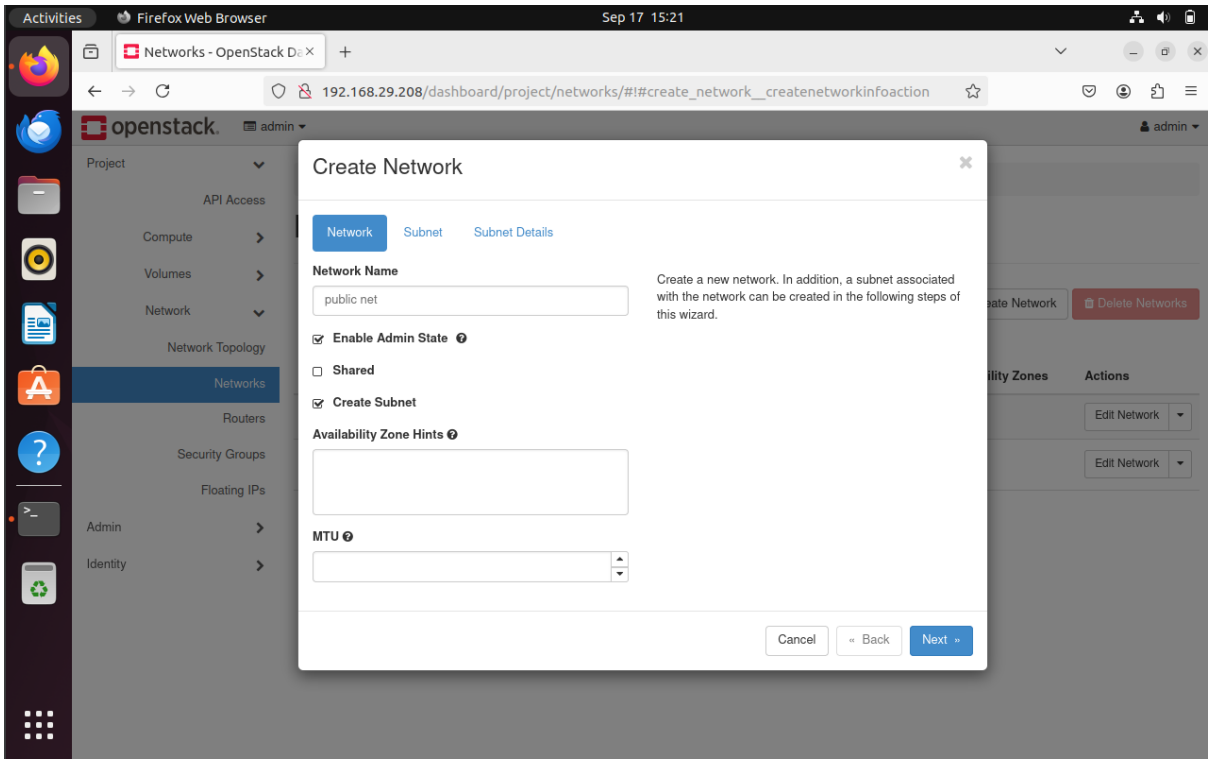


- Set up DHCP to assign IP addresses to VMs automatically.



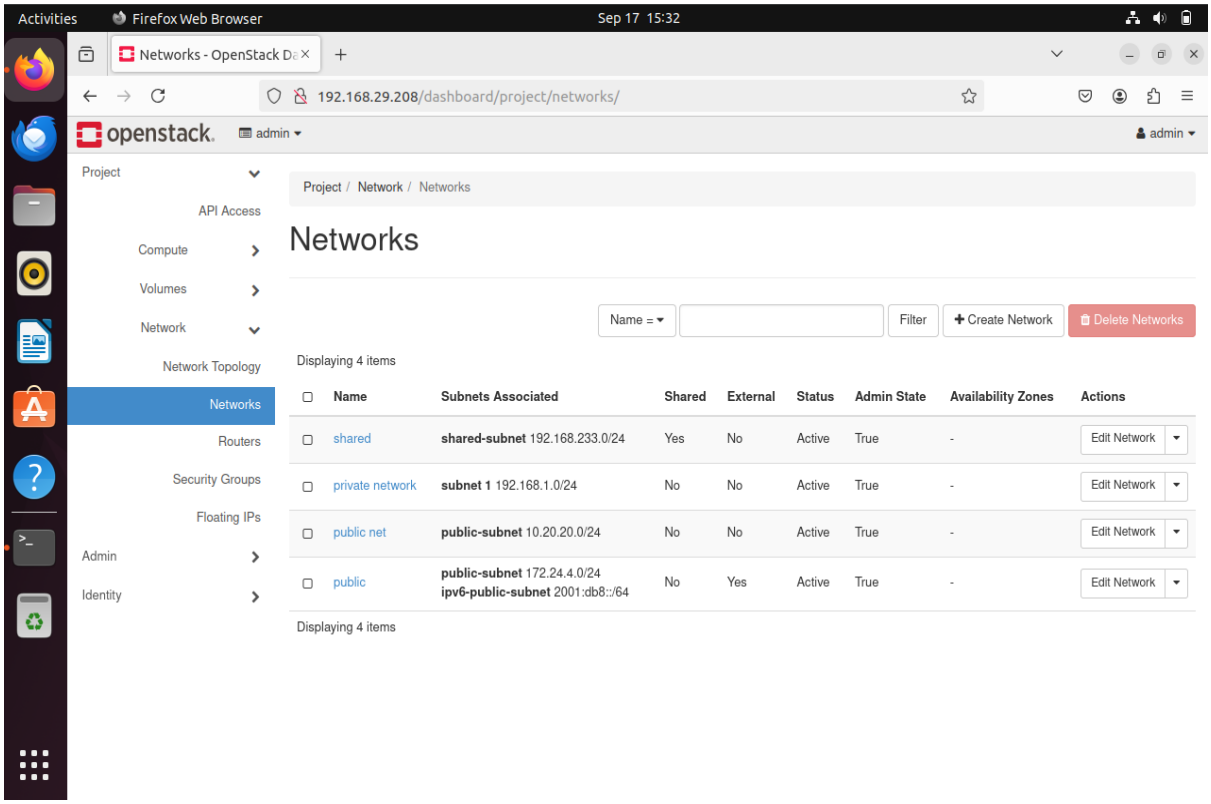
Public Network Setup

1. **Create a public network** to allow external access to virtual machines:
    - In the **Horizon Dashboard**, go to **Project > Network > Networks** and click **Create Network**.
    - Name it `Public-Net` and choose a valid IP range (e.g., `10.20.20.0/24`).



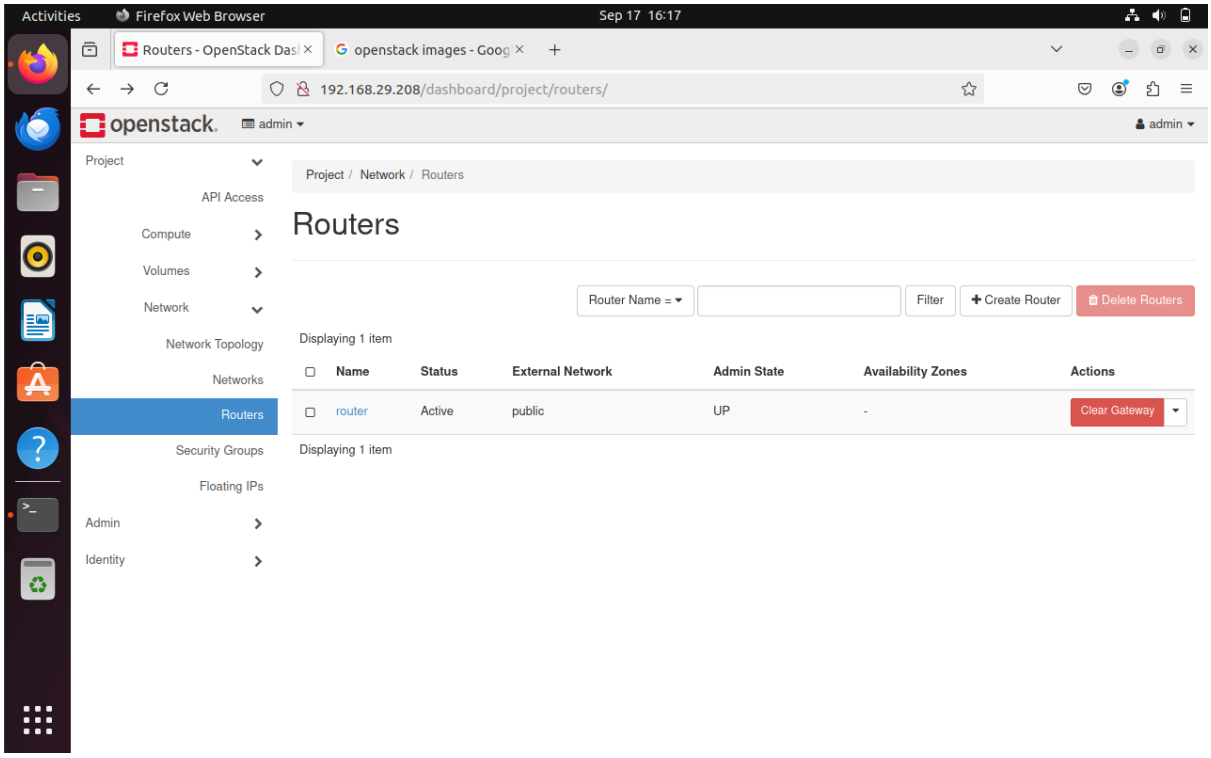
2. **Subnet for Public Network**:
    - Create a subnet for `Public-Network` with a valid external IP range.
    - Disable DHCP for this network since IPs will be manually assigned.

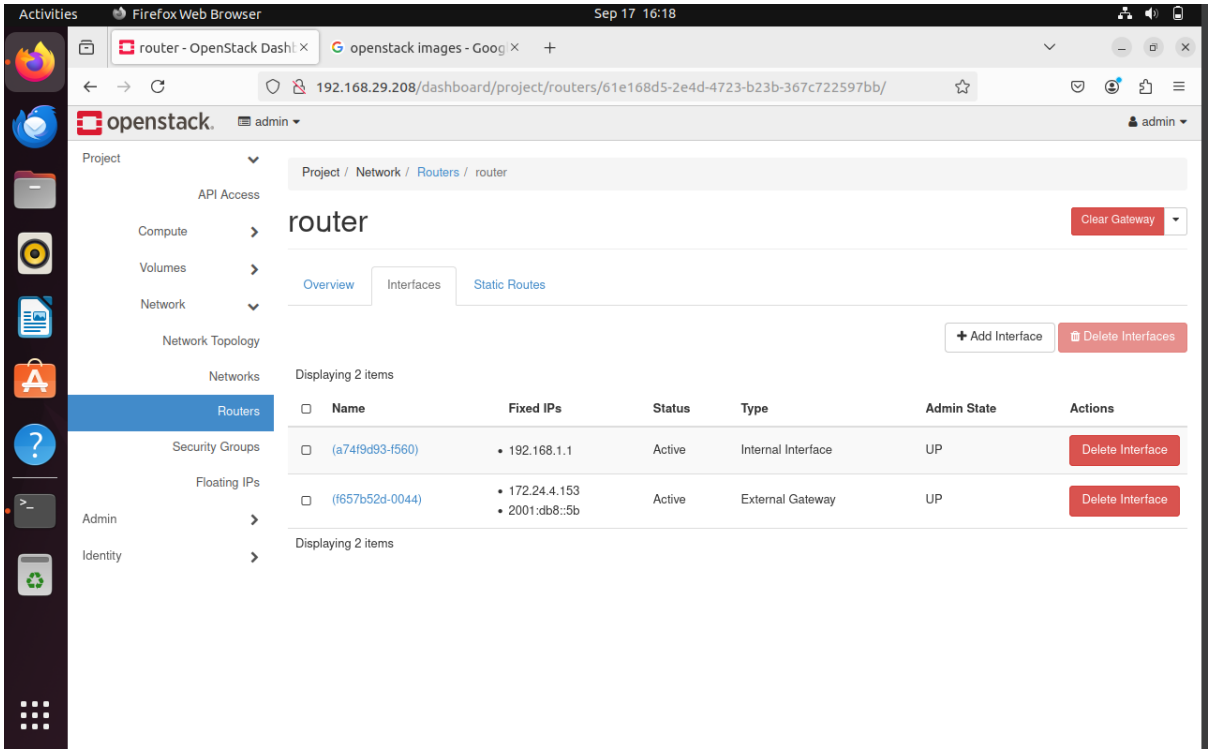
Created Networks:



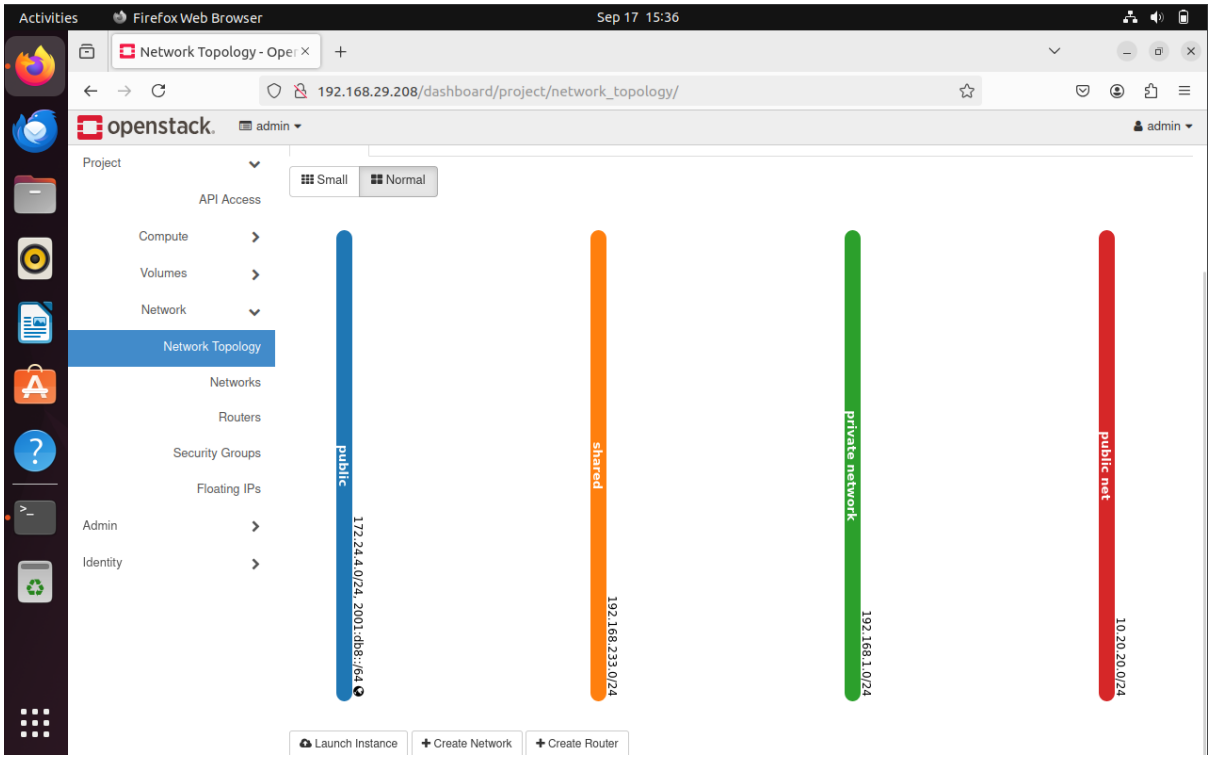
Configure Routers

1. Go to **Project > Network > Routers** and create a router.
2. Connect the router to both **Private-Network** and **Public-Network** to enable external access for VMs.



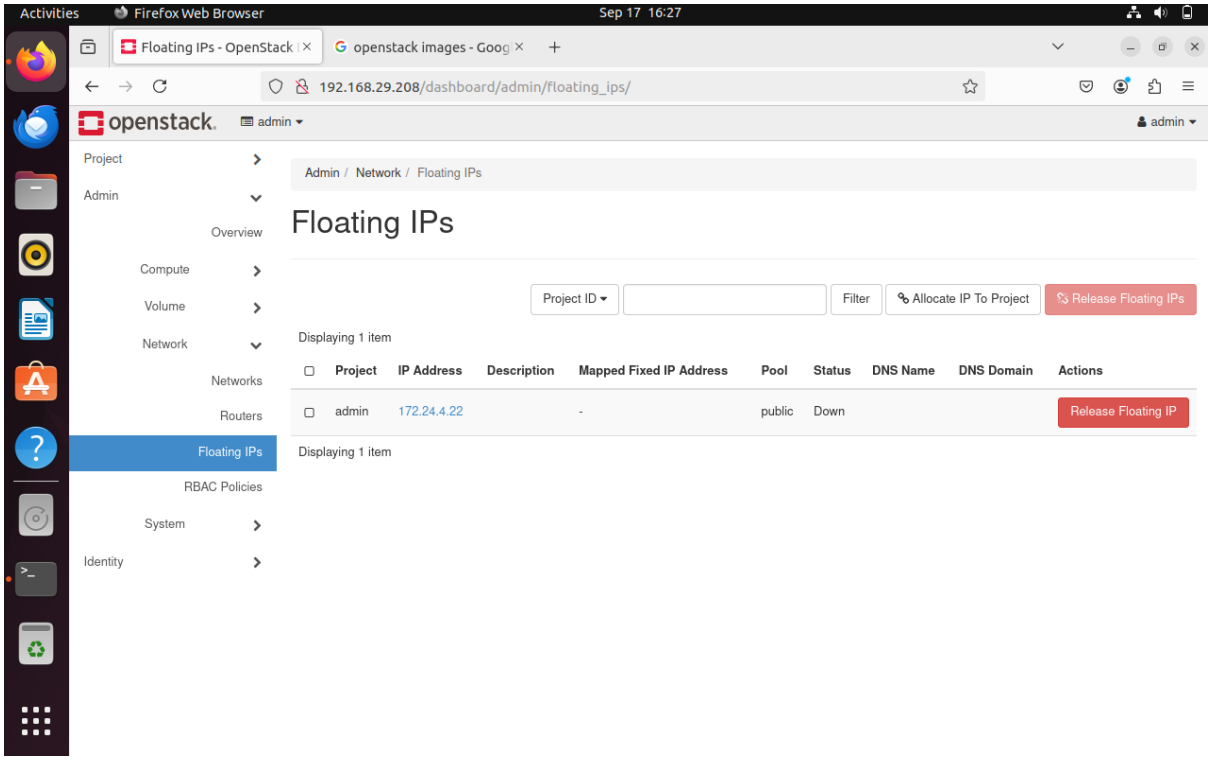


Network Topology:

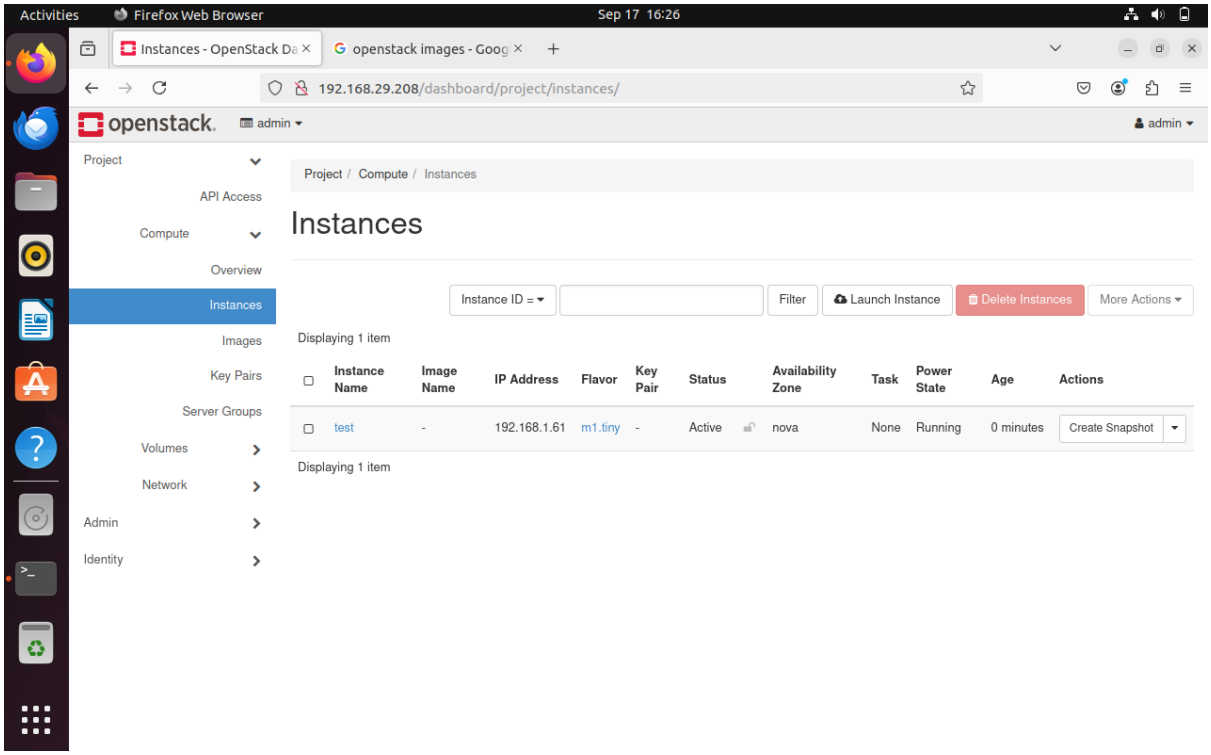


Floating IPs

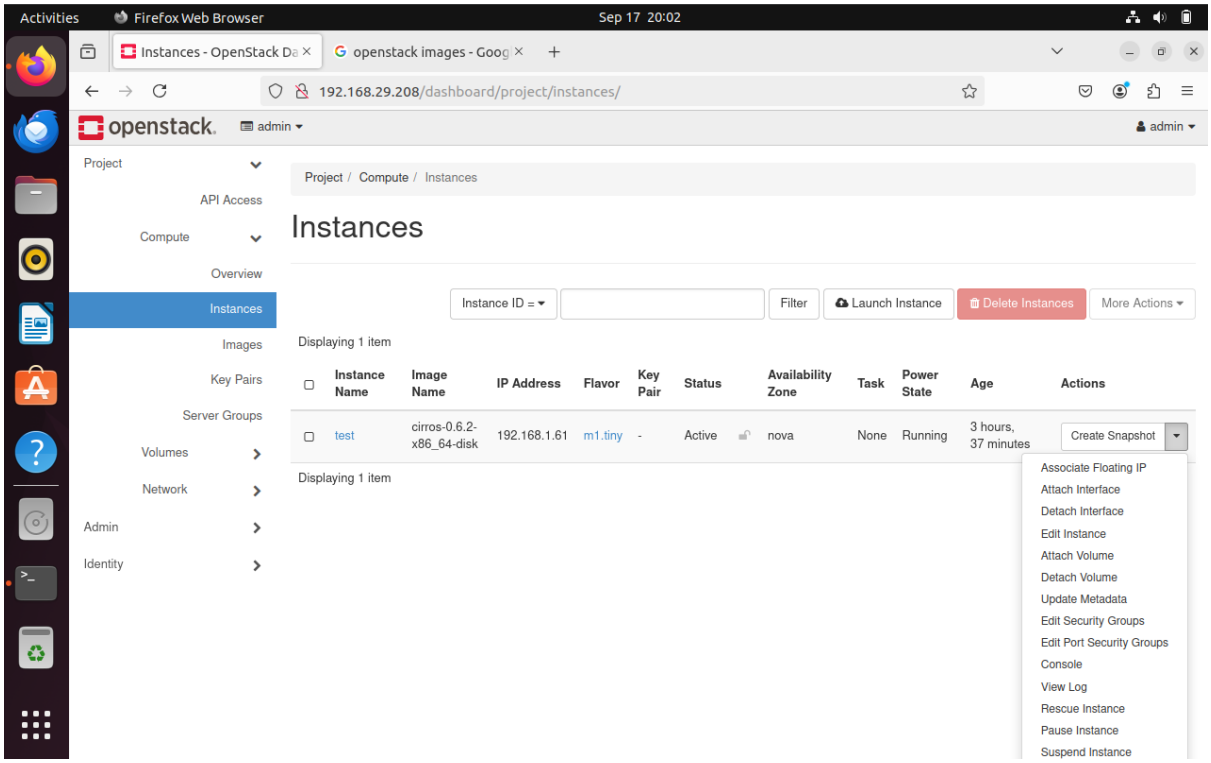
1. **Allocate floating IPs** to allow VMs on the private network to communicate externally.
    - Navigate to **Project > Network > Floating IPs** and click **Allocate IP to Project**.

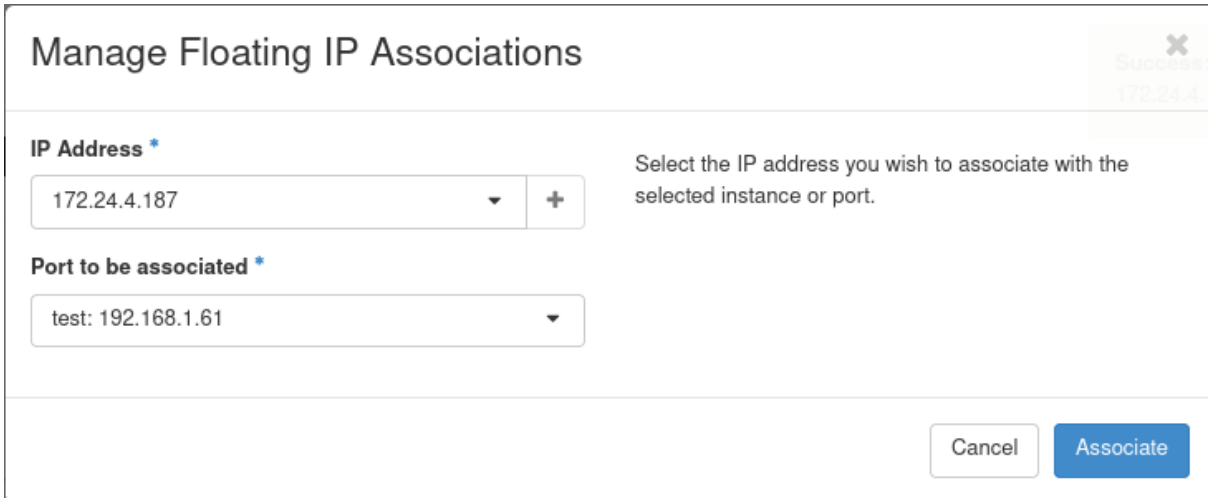


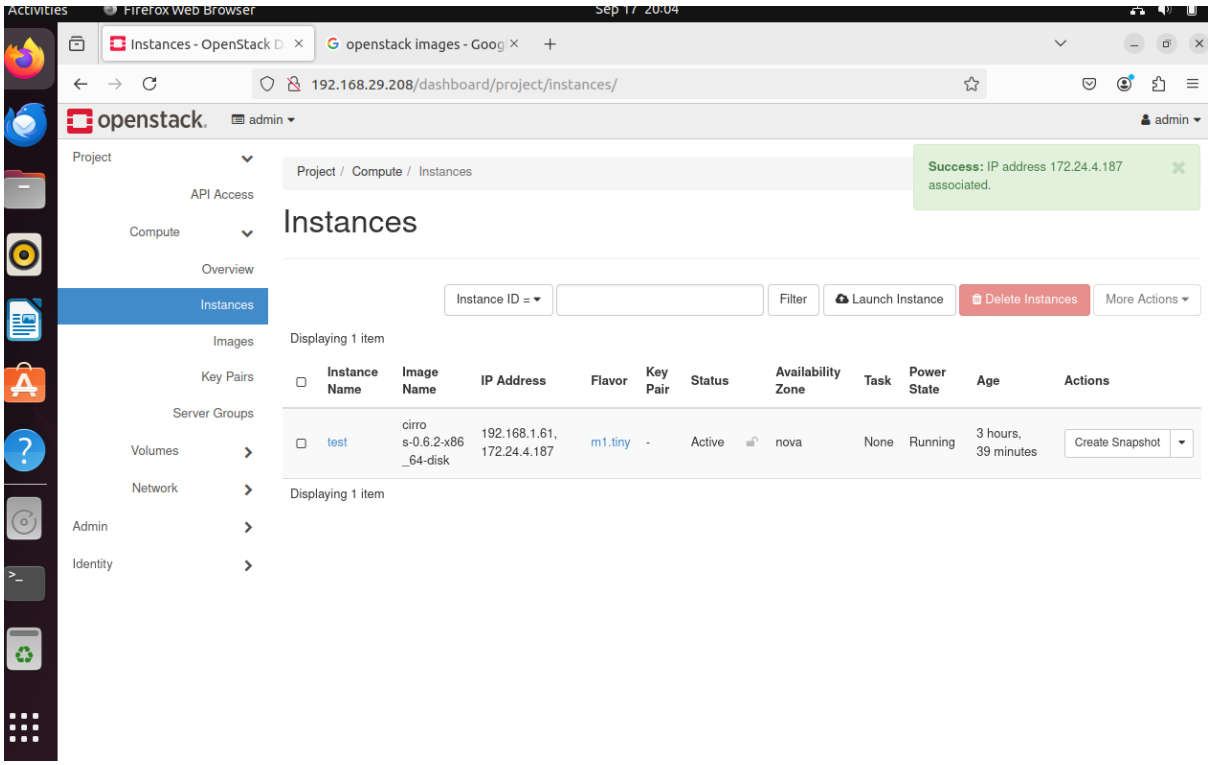
- Associate the floating IP with the external **Public-Network test** .



2. Once the IP is allocated, assign it to a specific VM through the VM's **Actions > Associate Floating IP**.

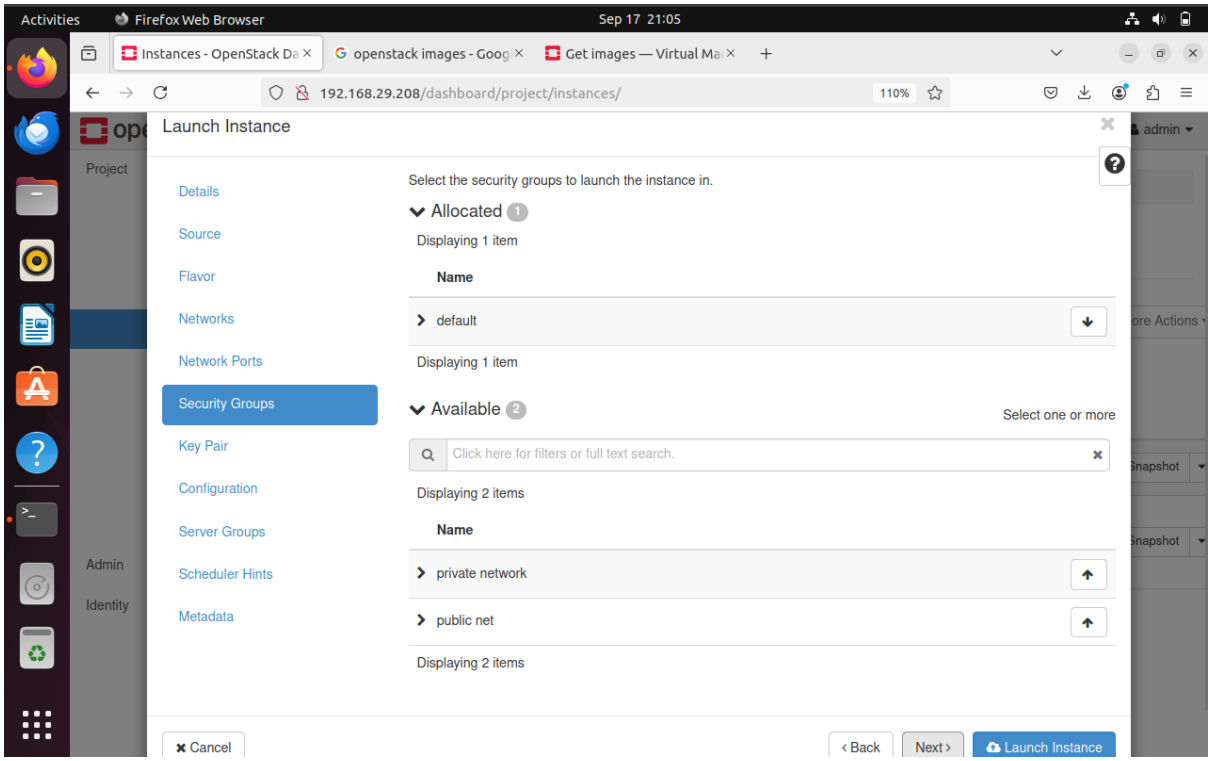




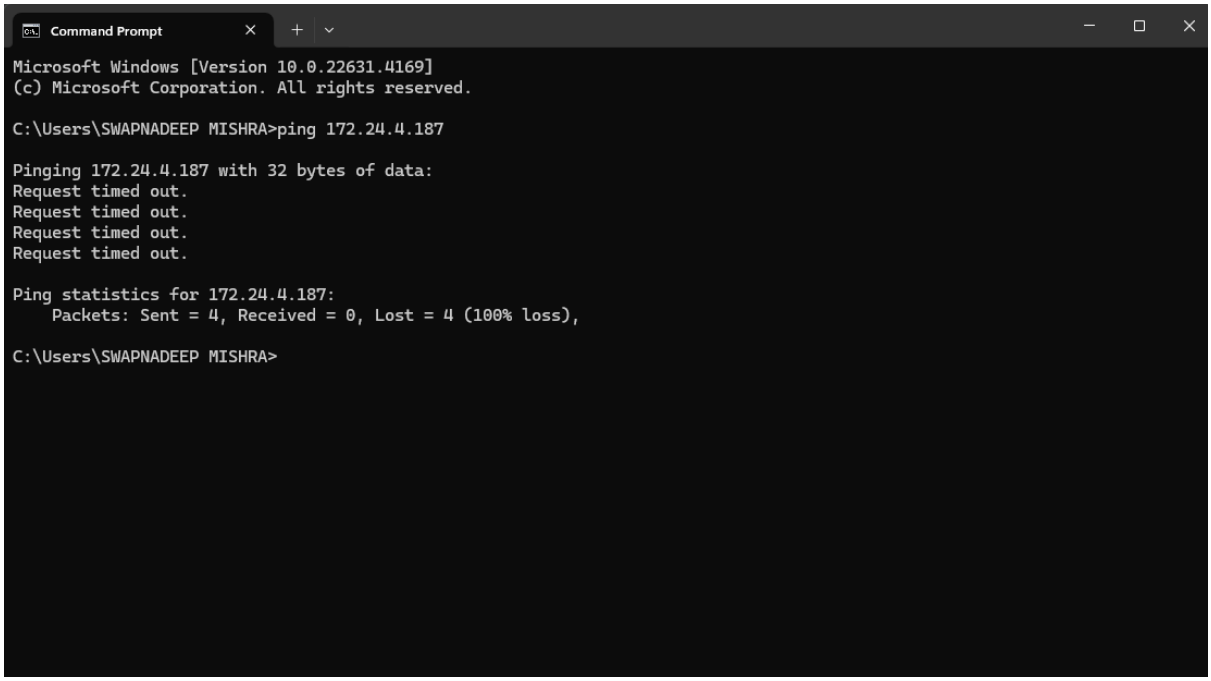


Step 2: Security Groups

1. **Default Security Group Configuration**:
   - By default, security groups block all traffic. To enable SSH and ICMP, modify the default security group.
   - Go to **Project > Network > Security Groups** and click on the default security group.

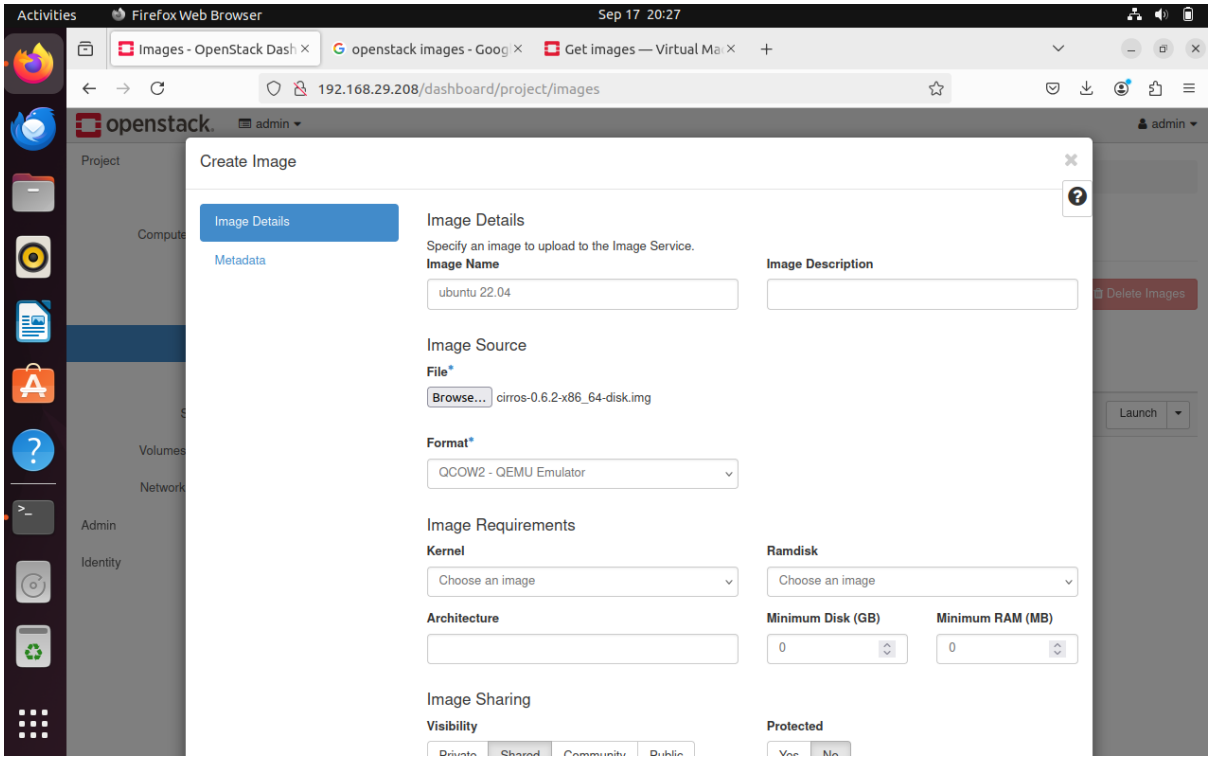


2. **Create Security Rules**:
   - **Add SSH Rule**: Enable SSH by adding a rule to allow incoming TCP traffic on IP address 172.24.4.187 from any source.
   - **Allow Ping (ICMP)**: Add a rule to allow all ICMP traffic to enable pinging VMs.

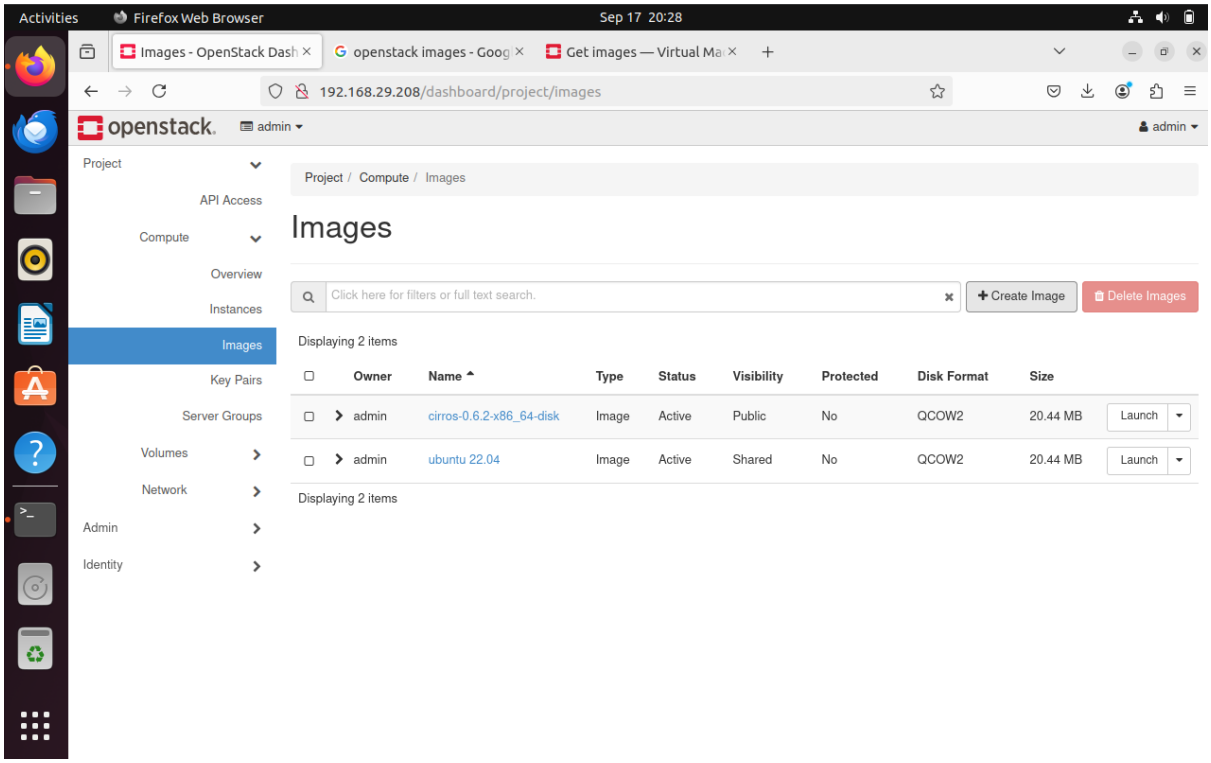


Step 3: Add an Image Using Glance

1. **Upload OS Image**:
    - Go to **Project > Compute > Images** and click **Create Image**.
    - Name the image `Ubuntu 22.04`.
    - Choose the image source file (download from Ubuntu official sources).
    - Set the disk format to `qcow2` (for efficient image storage) and the minimum disk size as 20.44MB.

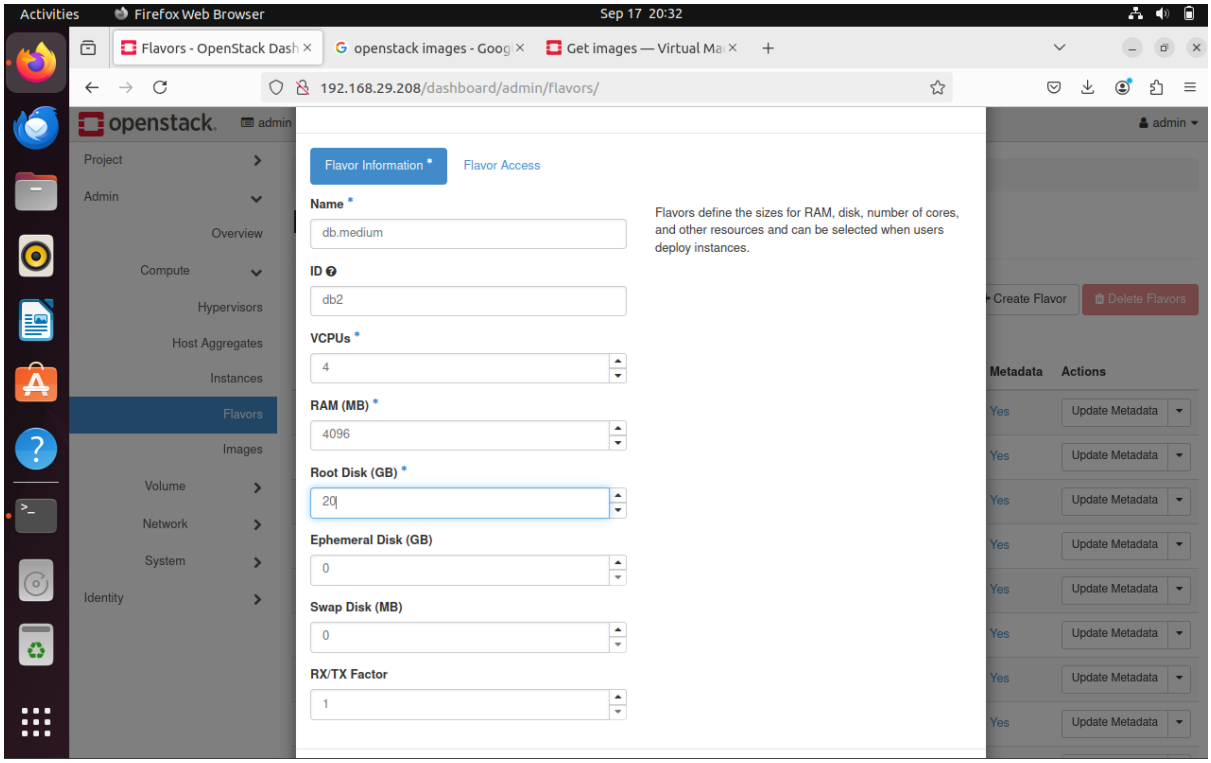


2. **Verify the Image**:
    - After uploading, ensure the image is available under **Images** and is in an active state.

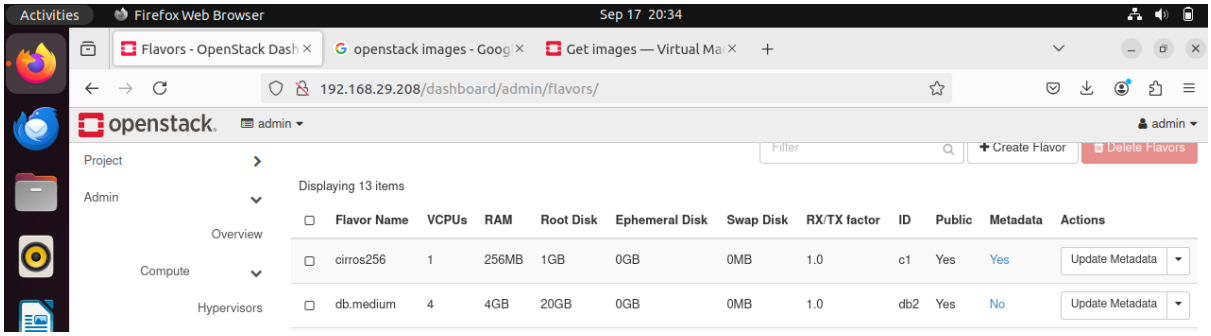


Step 4: Create a Flavor

1. **Define Compute, Storage, and Memory Resources**:
   - Go to **Admin > Compute > Flavors** and click **Create Flavor**.
   - Set the flavor name as `db.medium`.



2. **Configure the Resources**:
    - Set **vCPUs** to `4`, **RAM** to `4096MB` (4GB), and **disk space** to `20GB`.
    - Save the flavor and ensure it is available for selection when launching VMs.



Challenges & Solutions

Challenge 1: Image Upload

- Initially, I encountered an issue where the uploaded image failed to reach an active state due to a misconfiguration in the Glance service.
- **Solution**: I corrected the disk format setting to `qcow2` and ensured that the image size was compatible with the available storage. This resolved the issue, and the image was successfully activated.

Challenge 2: Floating IP Assignment

- The floating IPs were not being properly associated with VMs, leading to failed external connections.
- **Solution**: After verifying the router setup and network configurations, I identified that the router was not properly linked to the external network. Reconfiguring the router resolved the floating IP issue, and the VMs were able to connect to external networks.

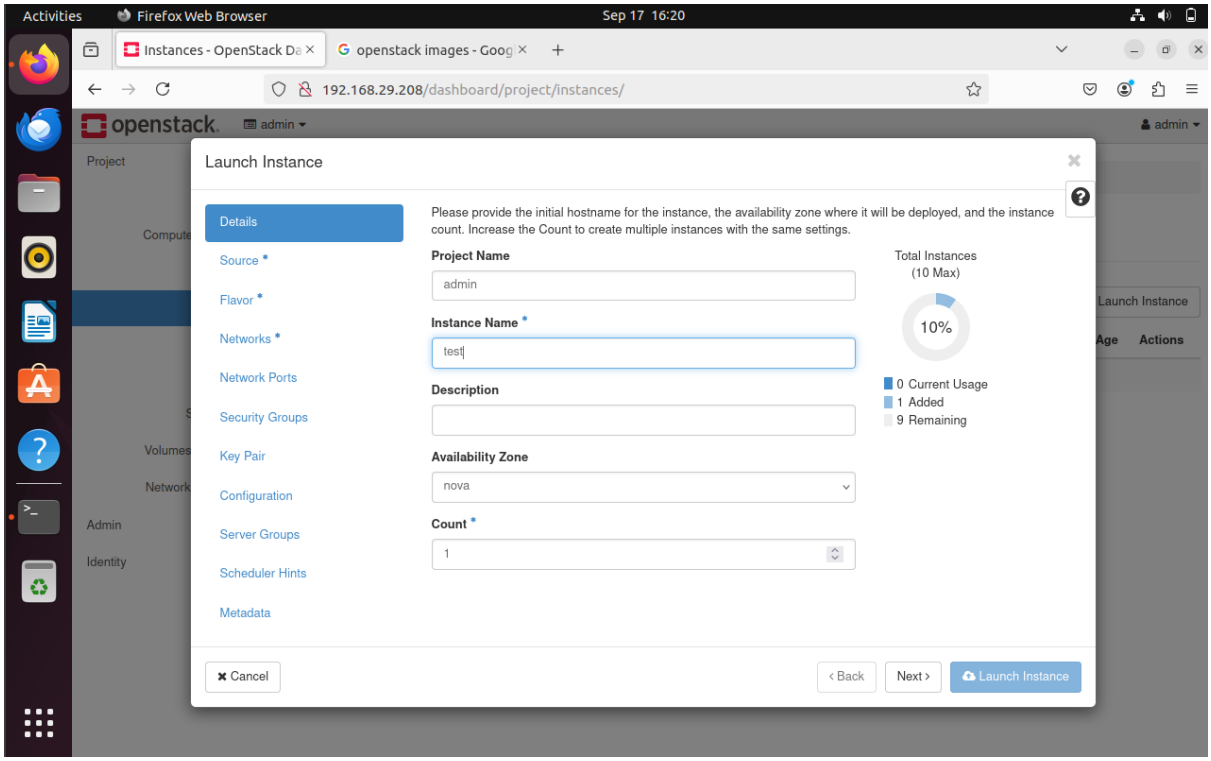
Part 3: Virtual Machine Creation and Management Report

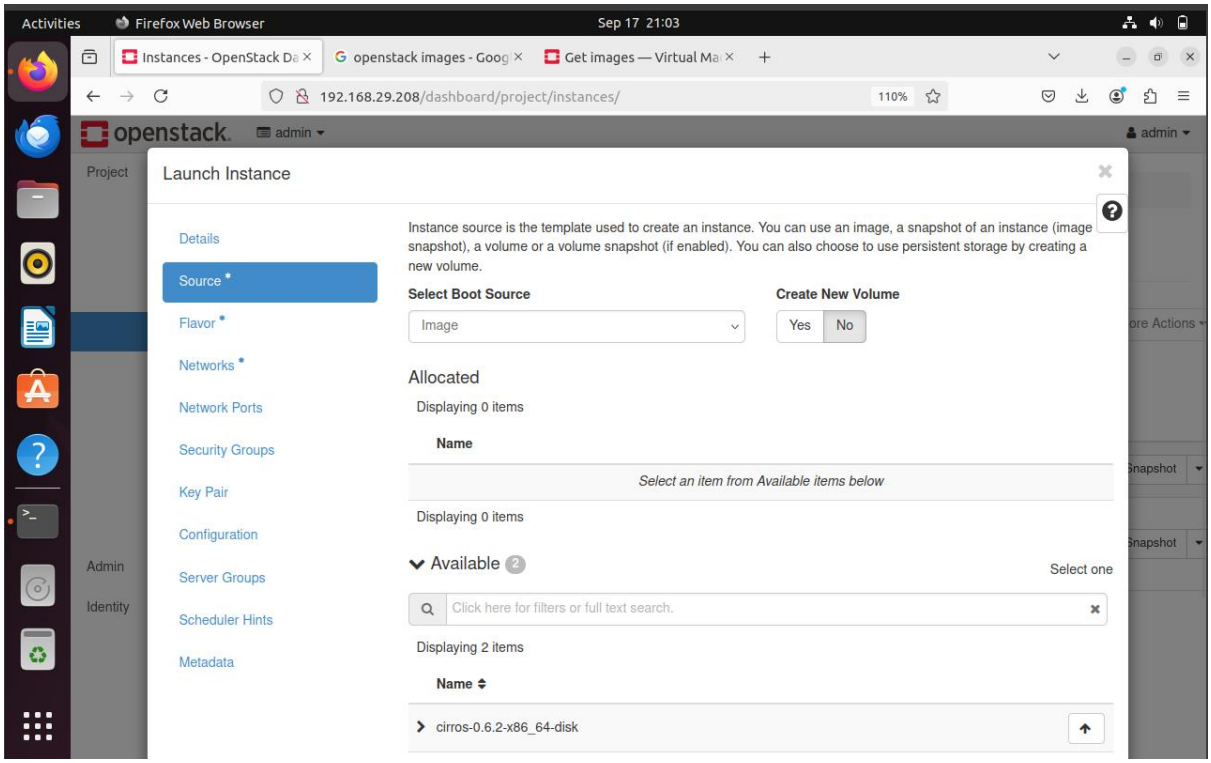
VM Setup

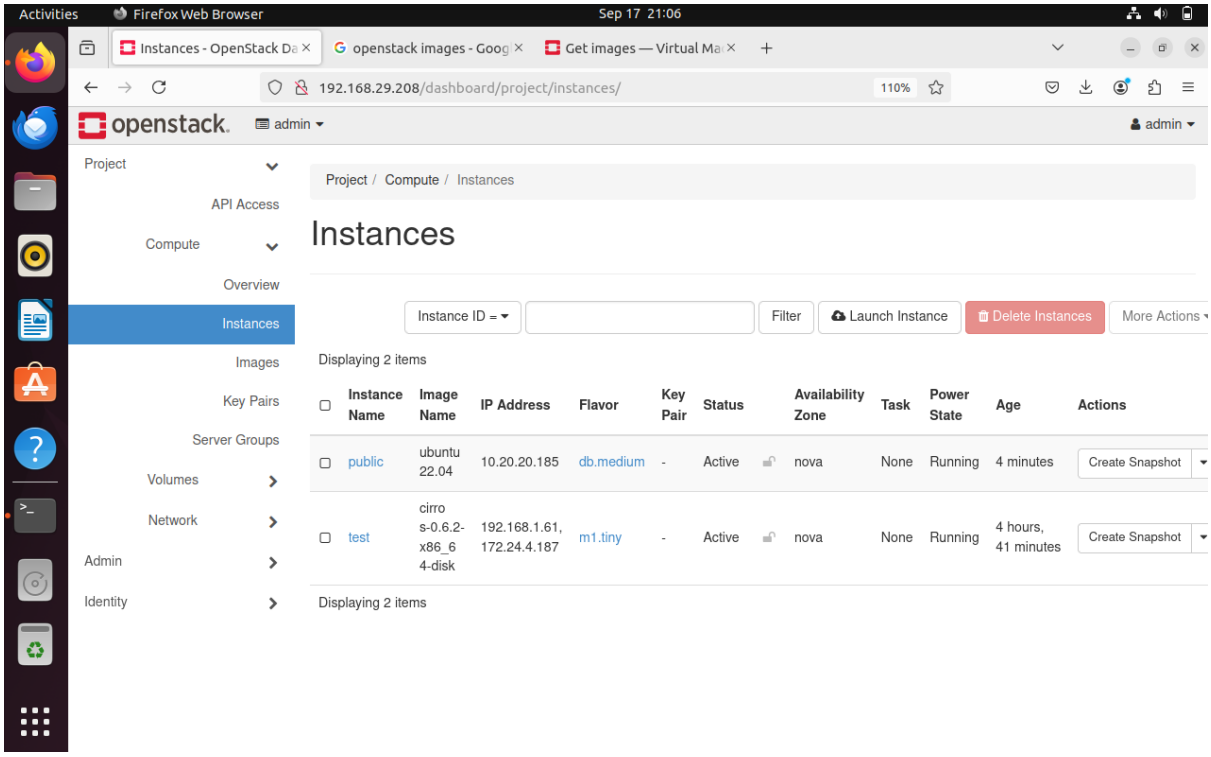
Step 1: Launching Virtual Machines (VMs)

Two virtual machines were successfully launched in the OpenStack environment. Both VMs were assigned to the appropriate networks as follows:

- **VM 1** was connected to the **Public Network**.
- **VM 2 named test** was connected to both the **Private-Network** and a **Floating IP** from the **Public-Network** for external access.





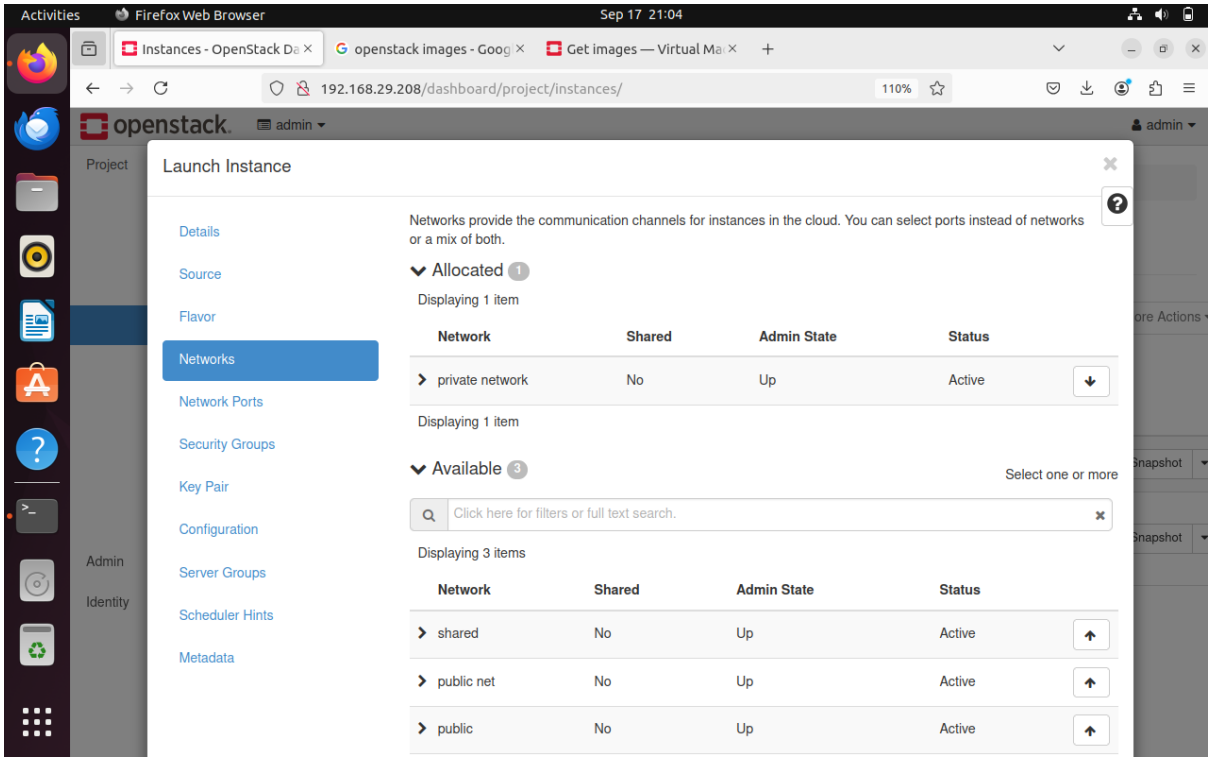


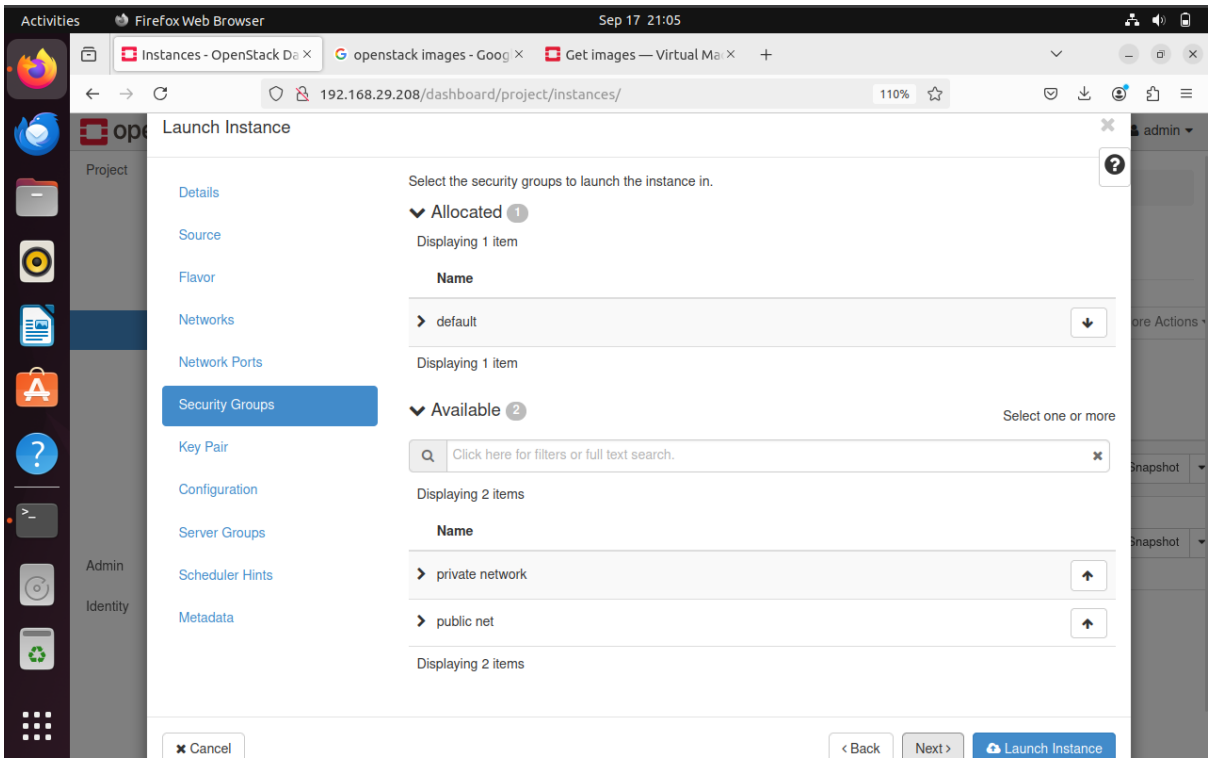
Step 2: Accessing the VMs

- **SSH Access**: Using the allocated floating IP, SSH was used to access both virtual machines. This confirmed successful external network communication.

Step 3: Testing VM Connectivity

- **VM-to-VM Connectivity**: Pinging between the two VMs within the **Private-Network** confirmed internal network functionality.
- **External Network Connectivity**: VMs connected to the public network were able to access external resources, ensuring proper configuration of floating IPs.



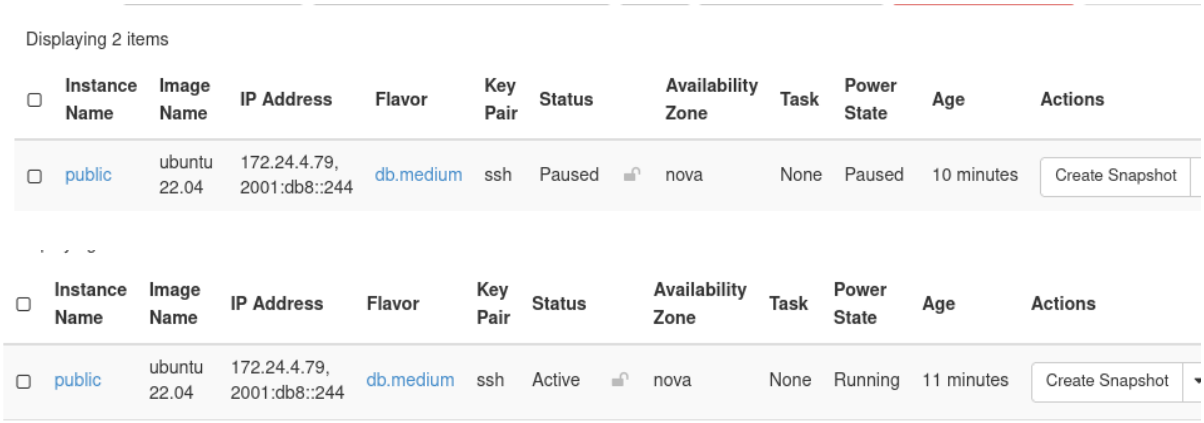


VM Management

Step 1: VM Operations

Several operations were performed to manage the VMs:

1. **Pause and Resume**:
    - VMs were paused to temporarily halt their processes and then successfully resumed.

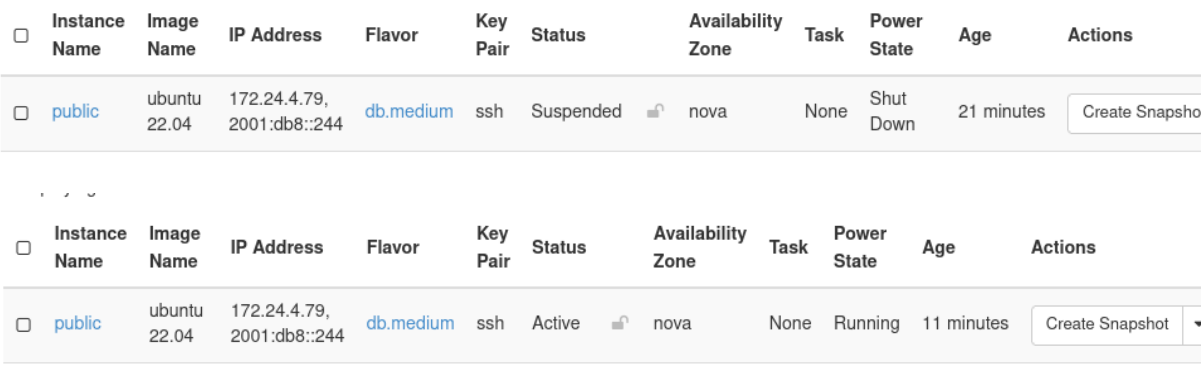


Displaying 2 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
public	ubuntu 22.04	172.24.4.79, 2001:db8::244	db.medium	ssh	Paused	nova	None	Paused	10 minutes	Create Snapshot

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
public	ubuntu 22.04	172.24.4.79, 2001:db8::244	db.medium	ssh	Active	nova	None	Running	11 minutes	Create Snapshot

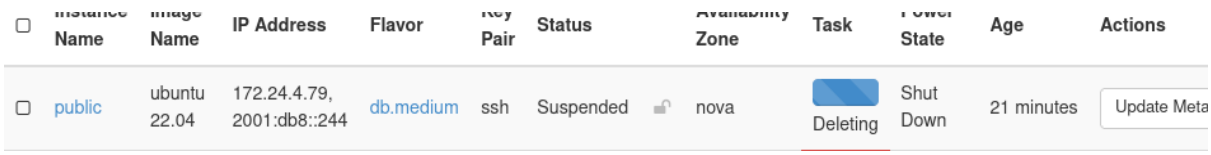
2. **Stop and Start**:
    - Both VMs were stopped and later restarted to ensure they could be managed without data loss or connectivity issues.



Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
public	ubuntu 22.04	172.24.4.79, 2001:db8::244	db.medium	ssh	Suspended	nova	None	Shut Down	21 minutes	Create Snapsho

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
public	ubuntu 22.04	172.24.4.79, 2001:db8::244	db.medium	ssh	Active	nova	None	Running	11 minutes	Create Snapshot

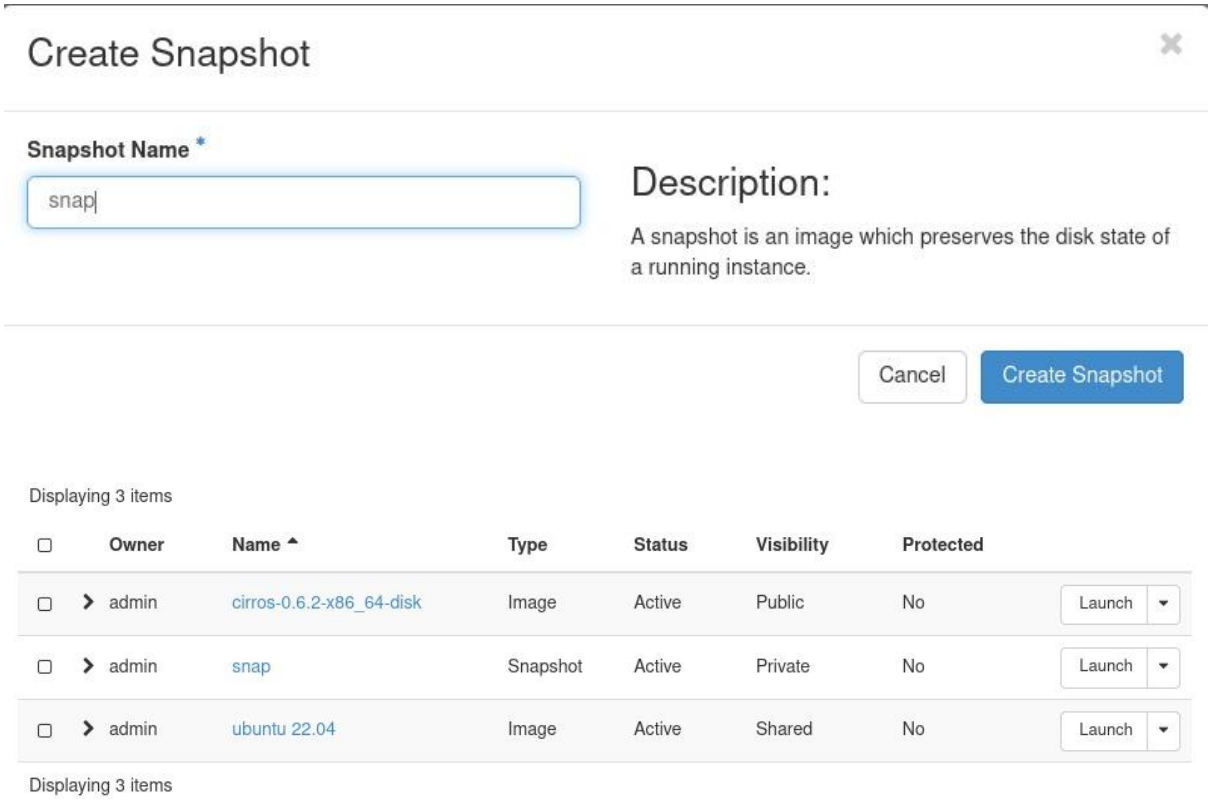
3. **Delete**:
    - After testing, one of the VMs was deleted successfully from the system, demonstrating full lifecycle management.



Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
public	ubuntu 22.04	172.24.4.79, 2001:db8::244	db.medium	ssh	Suspended	nova	Deleting	Shut Down	21 minutes	Update Meta

Step 2: VM Snapshot and Auto-Scaling

- **Snapshot Creation**:
  - A snapshot of one of the VMs was taken to save its current state. The snapshot feature worked as expected, allowing the VM state to be restored later.



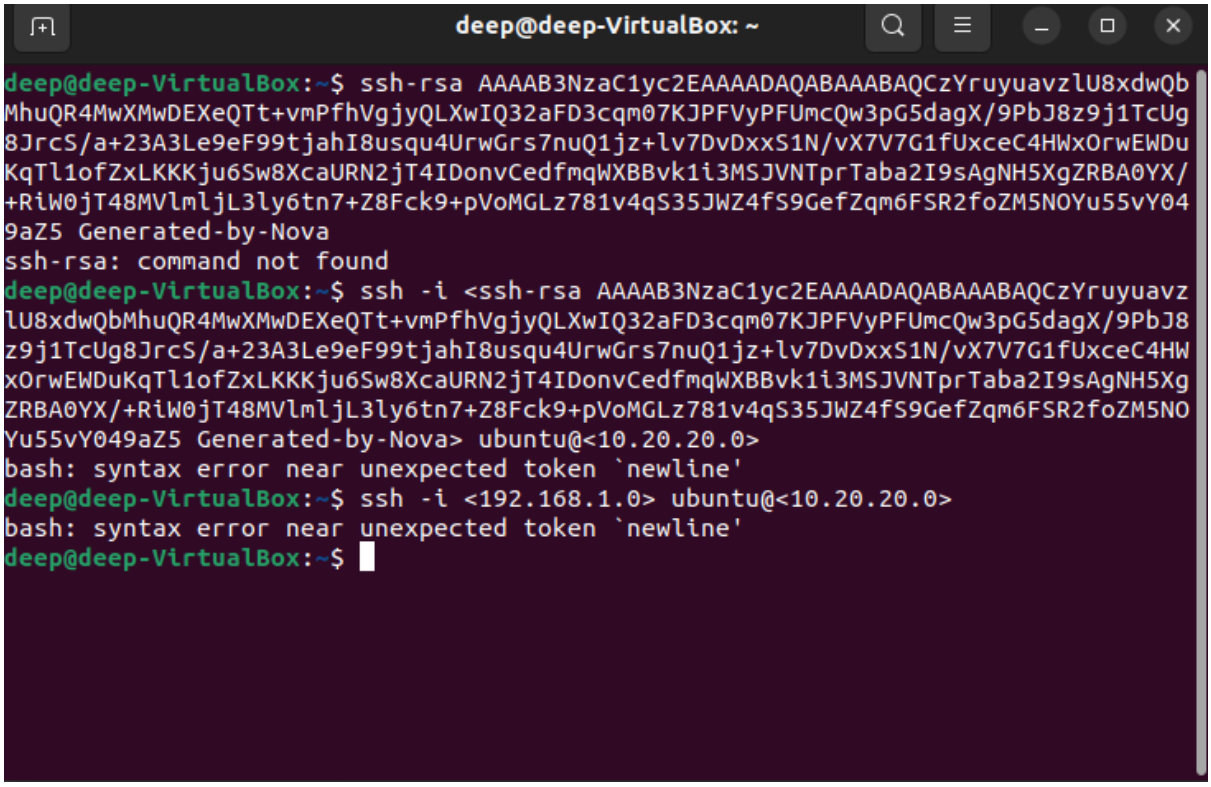
- **Auto-Scaling** (if possible):
  - Although auto-scaling was not demonstrated due to the setup, the infrastructure supports scaling resources based on demand.

Challenges & Solutions

Challenge 1: SSH Syntax Error Due to Incorrect Key Formatting

- **Issue**: A "syntax error near unexpected token newline" occurred when trying to use SSH with an improperly formatted key.
- **Solution**: Use the correct syntax by referencing the private key file:

  ssh -i /path/to/private_key.pem ubuntu@<ip_address>



Challenge 2: VM Connectivity

- **Issue**: There was difficulty pinging between the two VMs initially.
- **Solution**: After reviewing the security groups, the necessary ICMP rules were added to allow pinging, resolving the connectivity issue.

Conclusion:

This assignment provided valuable hands-on experience with OpenStack installation, configuration, and virtual machine management. Setting up networks, security groups, and floating IPs helped me understand the fundamentals of cloud infrastructure. Although there were challenges, especially with installation and networking, the solutions applied improved my problem-solving skills. Overall, this experience gave me practical knowledge of cloud environments, which will be useful in future projects.
