Institut National des Postes et Télécommunications

Scalable Cloud Solutions

Report: Automated OpenStack Deployment with Kolla Ansible and external Ceph Integration

Submitted by: Zakarya Belaid Adil El Hihi

Academic Year: 2024/2025

Abstract

This project demonstrates the design and implementation of a scalable private cloud infrastructure, showcasing technical and functional expertise in the management and architecture of private clouds for both development and production environments. The private cloud was built using Kolla Ansible to automate the deployment of OpenStack across a multi-node architecture, comprising six nodes: three dedicated to OpenStack (one controller/deployer and two compute nodes) and three for an external Ceph cluster (one admin and two normal nodes deployed using Cephadm). The integration of Ceph delivers a resilient and scalable storage solution, while Vagrant was utilized to automate infrastructure setup, ensuring consistency and efficiency in deployment.

Throughout the project, several challenges were addressed, including complex networking implementations and optimizations, as well as Linux system administration tasks. These hurdles provided valuable hands-on experience in troubleshooting and optimizing infrastructure for performance and reliability. The result is a fully functional private cloud that is nearly ready for production use, requiring only minor configurations such as TLS/SSL certificate setup and ensuring seamless connectivity between the OpenStack servers and the external Ceph cluster.

To promote collaboration and knowledge sharing, the project's repository is openly available on GitHub at openstack.git. The repository contains all the necessary scripts and configurations to deploy OpenStack with an integrated Ceph cluster, enabling others to replicate or build upon this work.

This project not only highlights the successful deployment of a private cloud but also underscores the importance of automation, scalability, and resilience in modern infrastructure design. Key lessons learned include a deeper understanding of infrastructure design principles, enhanced proficiency in OpenStack administration, and advanced Linux system administration skills. This solution serves as a robust foundation for individuals seeking to deploy private cloud environments with a focus on automation, scalability, and operational efficiency.

Contents

Automated OpenStack Deployment with Kolla Ansible and external Ceph Integration

1 Introduction

In today's tech landscape, cloud computing has become the backbone of modern infrastructure. OpenStack offers a powerful solution for organizations to create and manage their cloud environments. OpenStack is a collection of open-source tools. Various OpenStack cloud deployment methods are available. Kolla Ansible is one of the recommended deployment methods.
Kolla Ansible is highly opinionated out of the box, but allows for complete customization. This permits operators with minimal experience to deploy OpenStack quickly and as experience grows modify the OpenStack configuration to suit the operator's exact requirements. Kolla Ansible provides production-ready containers and deployment tools for operating OpenStack clouds.
This project focuses on addressing these challenges by leveraging Kolla Ansible to automate the deployment of OpenStack across multiple nodes. Additionally, to ensure scalable and resilient storage, an external Ceph cluster was integrated into the architecture. The use of Vagrant further streamlined the infrastructure setup, ensuring consistency and efficiency throughout the deployment process.

2 Prerequisites and tools used:

tools used:

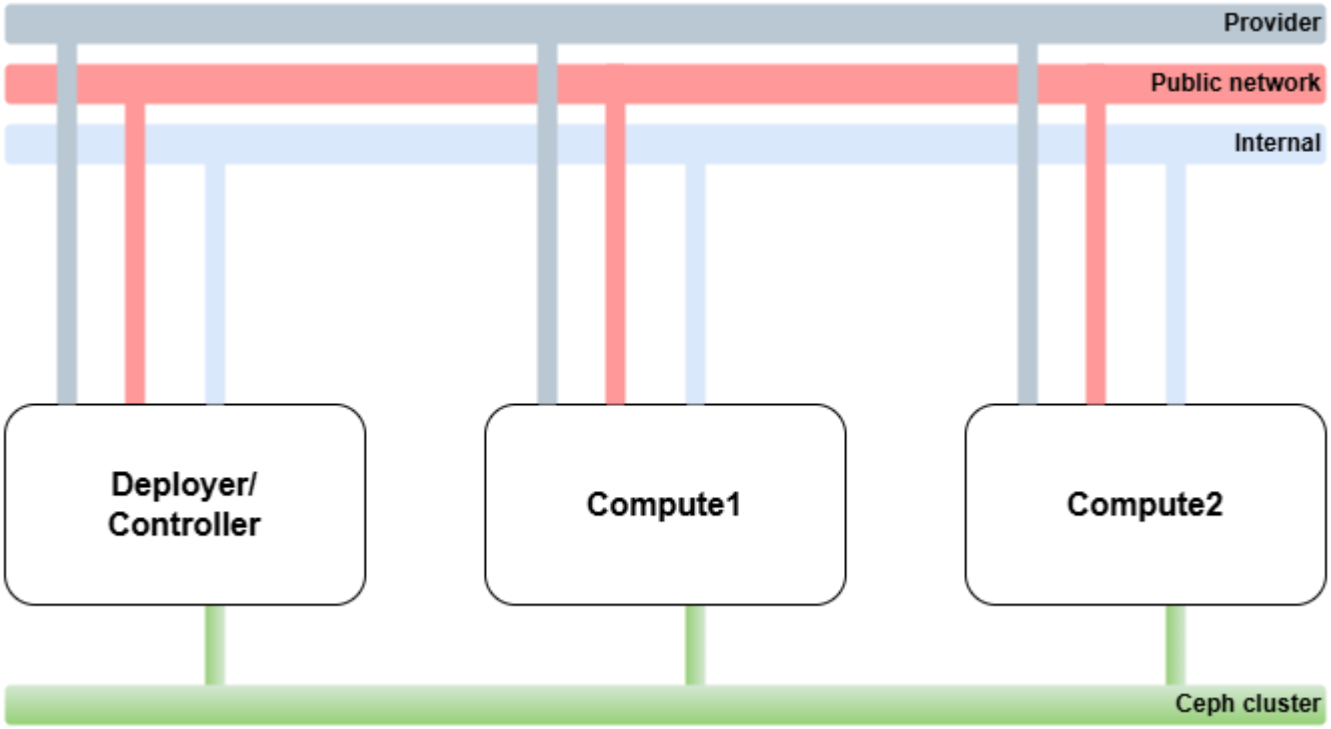
- Kolla-ansible.
- Cephadm: cephadm is a utility that is used to manage a Ceph cluster.
- Vagrant: building and maintaining portable virtual software development environments.
- Docker: managing containers.

Prerequisites:

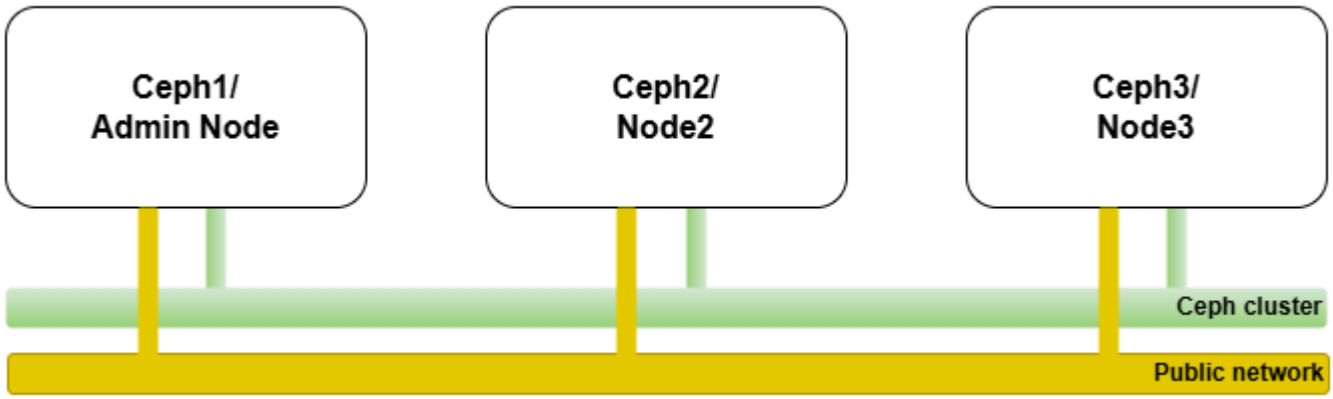
- Six virtual Servers (three controller+compute, three external Ceph)
- Ubuntu noble on each server.
- Ceph version squid
- Openstack version Epoxy (2025 release)

3 Overview

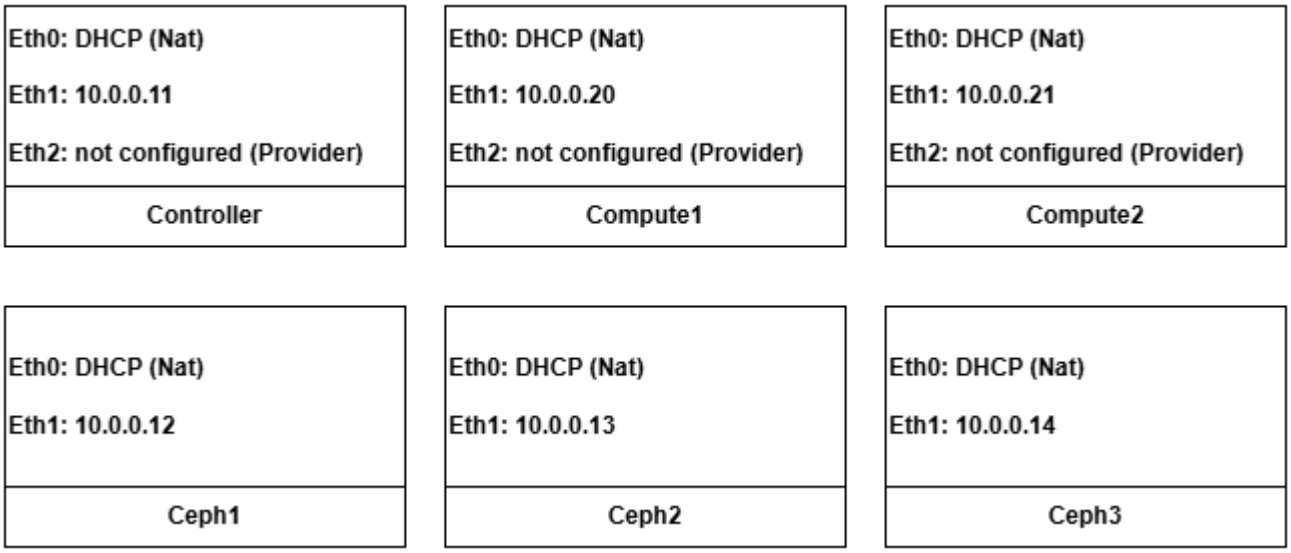
OpenStack Topology



Ceph Topology



IP Allocation



Three network adapters were configured to ensure seamless connectivity and functionality. The first adapter, a NAT adapter(Eth0), was used to provide internet connectivity to the virtual machines (VMs), enabling them to access external resources and updates. The second adapter, a host-only adapter(Eth1), facilitated communication between the six VMs, ensuring isolated and efficient internal networking. The third adapter, a NAT network adapter (Provider-Eth2), was reserved for OpenStack's internal IP allocation. Although not manually configured, this adapter allowed the VMs within OpenStack to automatically allocate IP addresses from a predefined pool.

4 Tasks performed

This project involves two main activities. First, a Ceph cluster is deployed using cephadm, a dedicated tool for managing Ceph clusters. Since Kolla Ansible does not support provisioning or configuring Ceph directly, cephadm is used to set up the cluster as external storage. Second, an OpenStack cluster is installed and integrated with the deployed Ceph storage. Once the Ceph cluster is operational, OpenStack's core services are installed.
All the scripts should have the proper permissions to run, and all commands should be run as root user.

4.1 Pre-config

Steps to configure the nodes:

1- create netplan config files for each node, grub file to erase new network interface names, a file to disable the cloud-init service on ubuntu and /etc/host file.
If we dont disable the cloud-init service, each time we reboot the new network config will be overwritten by the default config.

2- Configure passwordless access to the root user, hostname, and timezone from Ceph1 and Deployer(controller). First, we generate the ssh key and then copy the public key (id_rsa.pub) to other nodes located in .ssh/authorized_keys. After, we set the timezone for all nodes from either Ceph1 or Deployer.

3- Update and Upgrade packages on Each Node and install docker.

4.2 Install ceph cluster with cephadm as external storage

Steps to complete this task:

1- Ubuntu already includes up-to-date Ceph packages. In that case, we install cephadm directly.

2- Bootstrap a new cluster from Ceph1 using cephadm bootstrap command(precise the node static ip and cluster network).
Important: the bootstrap command will generate a new SSH key for the Ceph cluster and add it to the root user's /root/.ssh/authorized_keys file.
The output:

```
Ceph Dashboard is now available at:

            URL: https://ceph1:8443/
           User: admin
       Password: n79wr7896a

Enabling client.admin keyring and conf on hosts with "admin" label
Saving cluster configuration to /var/lib/ceph/681ee85c-06be-11f0-9a20-653fd97ed8b6/config directory
Enabling autotune for osd_memory_target
You can access the Ceph CLI as following in case of multi-cluster or non-default config:

       sudo /usr/sbin/cephadm shell --fsid 681ee85c-06be-11f0-9a20-653fd97ed8b6 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring

Or, if you are only running a single cluster on this host:

       sudo /usr/sbin/cephadm shell

Please consider enabling telemetry to help improve Ceph:

       ceph telemetry on

For more information see:

       https://docs.ceph.com/docs/master/mgr/telemetry/

Bootstrap complete.
root@ceph1:/home/vagrant#
```

3- Enable Ceph CLI on Ceph1 and verify Ceph command is accessible.

4- Adding hosts to the cluster by installing the cluster's public SSH key in the new hosts root users then tell Ceph that the new hosts are part of the cluster.

```
root@ceph1:/home/vagrant# for node in ceph{2..3}
do
  ceph orch host add $node
done
ceph orch host ls
Added host 'ceph2' with addr '10.0.0.13'
Added host 'ceph3' with addr '10.0.0.14'
HOST   ADDR       LABELS  STATUS
ceph1  10.0.0.12  _admin
ceph2  10.0.0.13
ceph3  10.0.0.14
3 hosts in cluster
root@ceph1:/home/vagrant# ceph orch apply osd --all-available-devices --method raw
Scheduled osd.all-available-devices update...
root@ceph1:/home/vagrant#
```

5- Adding storage by telling Ceph to consume any available and unused devices.

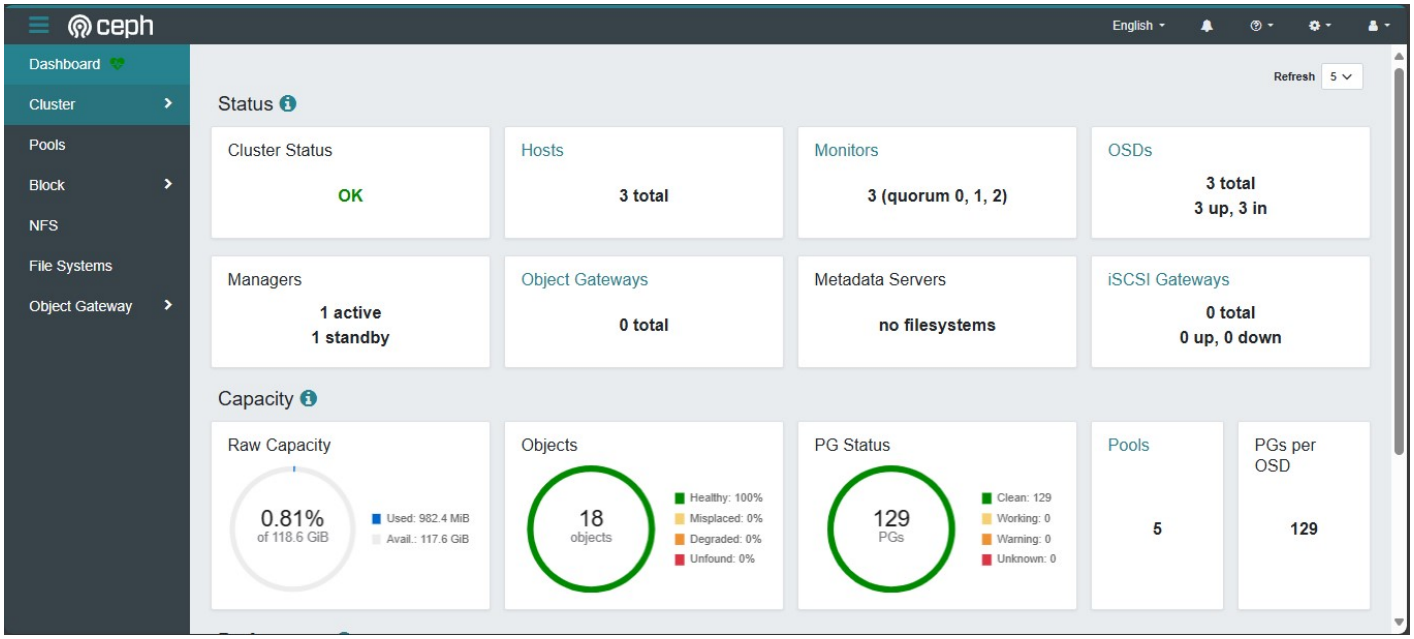
```
root@ceph1:/home/vagrant# ceph orch device ls --refresh
HOST   PATH      TYPE  DEVICE ID   SIZE  AVAILABLE  REFRESHED  REJECT REASONS
ceph1  /dev/sdb  hdd              10.0M  No         2m ago     Has a FileSystem, Insufficient space (<5GB)
ceph1  /dev/sdc  hdd              39.5G  No         2m ago     Has BlueStore device label, Has a FileSystem
ceph2  /dev/sdb  hdd              10.0M  No         2m ago     Has a FileSystem, Insufficient space (<5GB)
ceph2  /dev/sdc  hdd              39.5G  No         2m ago     Has BlueStore device label, Has a FileSystem
ceph3  /dev/sdb  hdd              10.0M  No         6s ago     Has a FileSystem, Insufficient space (<5GB)
ceph3  /dev/sdc  hdd              39.5G  Yes        6s ago
root@ceph1:/home/vagrant#
```

6- Set label *mon* and *osd* on all nodes.

7- Create pool for OpenStack (volumes images backups vms pool names)

8- Create Ceph keyring.

We can access now to Ceph dashboard on port 8843:



Ceph dashboard

Host Details

Hostname	Service Instances	Labels	Model	CPUs	Cores	Total Memory	Raw Capacity	HDDs	Flash	NICs
ceph1 *(10.0.0.12)*	crash: 1, grafana: 1, prometheus: 1, alertmanager: 1, mon: 1, mgr: 1, osd: 1, node-exporter: 1	_admin, mon, osd	Virtual Machine (VirtualBox)	1	2	2.9 GiB	79.7 GiB	11	0	3
ceph2 *(10.0.0.13)*	osd: 1, node-exporter: 1, mgr: 1, mon: 1, crash: 1	mon, osd	Virtual Machine (VirtualBox)	1	1	1.9 GiB	79.7 GiB	11	0	3
ceph3 *(10.0.0.14)*	osd: 1, mon: 1, crash: 1, node-exporter: 1	mon, osd	Virtual Machine (VirtualBox)	1	1	1.9 GiB	79.7 GiB	11	0	3

0 selected / 3 found / 3 total

Host details

4.3 Install OpenStack with Kolla ansible

Steps required to install OpenStack using deployment tools kolla-ansible:

1- Update System, Install Python build and virtual environment dependencies on Controller.

2- Activating the virtual environment and installing ansible, kolla-ansible and its dependencies using pip

3- Configuring ansible:

- Copying *globals.yml* and *passwords.yml* to /etc/kolla directory.
- Install Ansible Galaxy dependencies.
- Edit file multinode to match our setup.
- Generate Password for every openstack services.
- Create kolla config directory for nova, glance, and cinder.
- Copy file ceph.conf and ceph keyring to kolla config directory.

4- Globals.yml should match this:

```
pc 1@PC MINGW64 ~/Desktop/Docs/Openstack/Openstack/kolla (master)
$ grep -v "#" globals.yml | tr -s '[:space:]'
---
workaround_ansible_issue_8743: yes
config_strategy: "COPY_ALWAYS"
kolla_base_distro: "ubuntu"
kolla_internal_vip_address: "10.0.0.10"
network_interface: "eth1"
neutron_external_interface: "eth2"
neutron_plugin_agent: "openvswitch"
keepalived_virtual_router_id: "51"
enable_haproxy: "yes"
enable_cinder: "yes"
enable_fluentd: "no"
enable_heat: "yes"
enable_horizon: "yes"
enable_openvswitch: "{{ enable_neutron | bool and neutron_plugin_agent != 'linux
bridge' }}"
ceph_glance_user: "glance"
ceph_glance_pool_name: "images"
ceph_cinder_user: "cinder"
ceph_cinder_pool_name: "volumes"
ceph_cinder_backup_user: "cinder-backup"
ceph_cinder_backup_pool_name: "backups"
ceph_nova_user: "{{ ceph_cinder_user }}"
ceph_nova_pool_name: "vms"
fernet_token_expiry: 86400
glance_backend_ceph: "yes"
cinder_backend_ceph: "yes"
cinder_volume_group: "cinder-volumes"
nova_backend_ceph: "yes"
nova_compute_virt_type: "qemu"
nova_console: "novnc"
```

globals.yml

5- Deploy Openstack with the bootstrap commands.

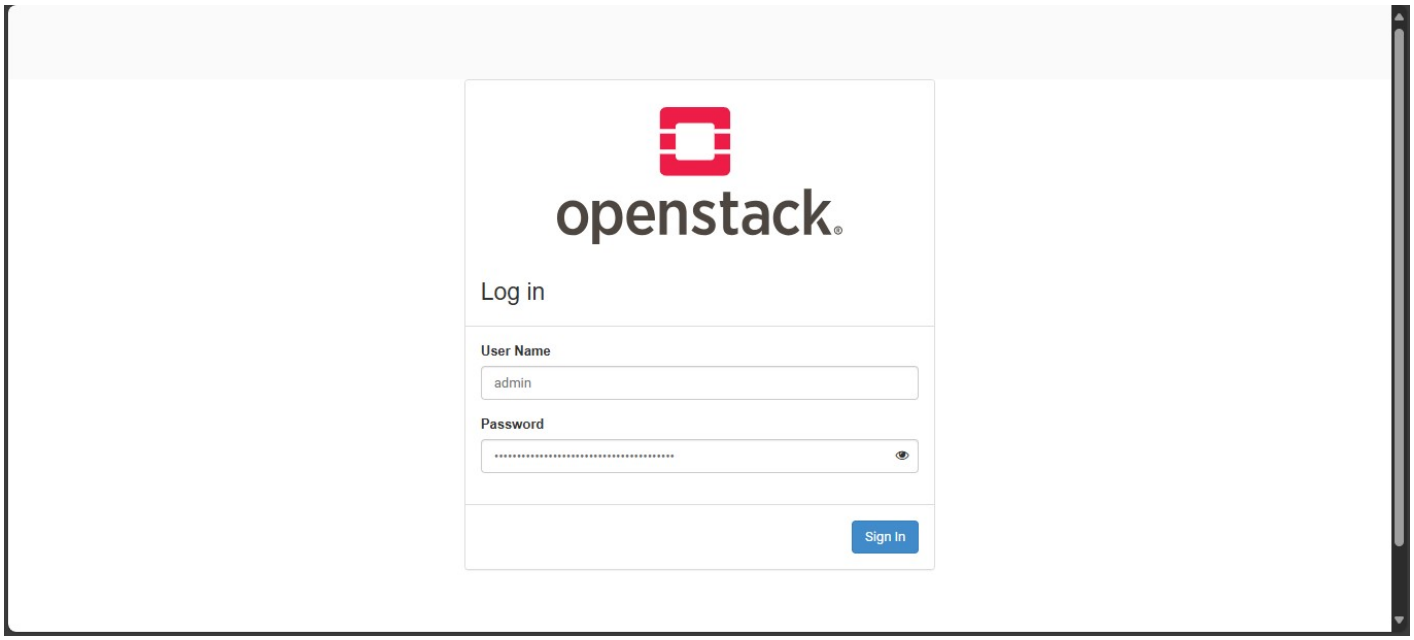
6- Install OpenStack Client.

7- Configure the *init-runonce* file to match our network setup to provide the openstack vms internet connectivity.

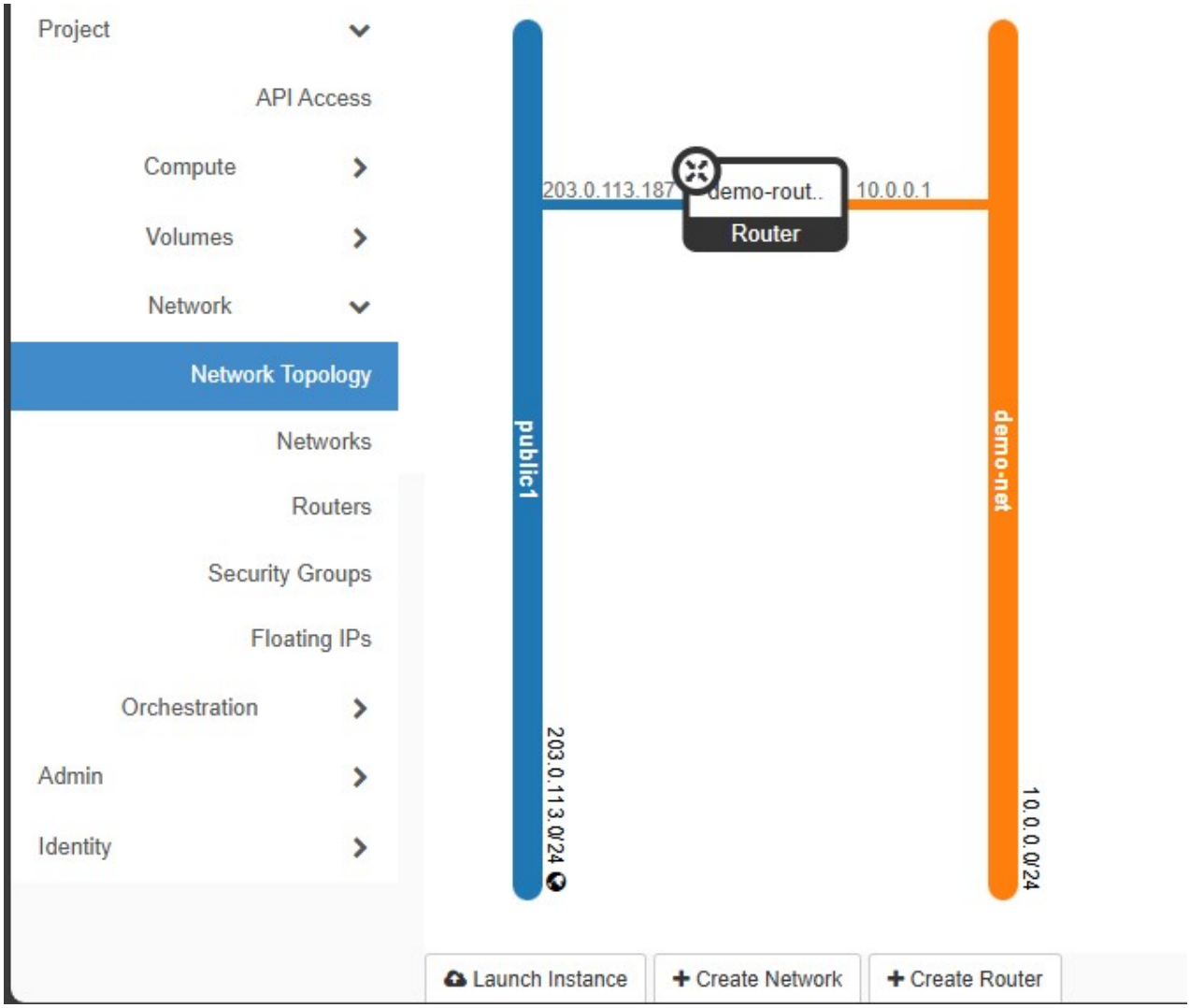
```
IP_VERSION=${IP_VERSION:-4}

DEMO_NET_CIDR=${DEMO_NET_CIDR:-'10.0.0.0/24'}
DEMO_NET_GATEWAY=${DEMO_NET_GATEWAY:-'10.0.0.1'}
DEMO_NET_DNS=${DEMO_NET_DNS:-'8.8.8.8'}

# This EXT_NET_CIDR is your public network,that you want to connect to the internet via.
ENABLE_EXT_NET=${ENABLE_EXT_NET:-1}
EXT_NET_CIDR=${EXT_NET_CIDR:-'203.0.113.0/24'}
EXT_NET_RANGE=${EXT_NET_RANGE:-'start=203.0.113.150,end=203.0.113.199'}
EXT_NET_GATEWAY=${EXT_NET_GATEWAY:-'203.0.113.1'}
```

Ip pool

We can access now to openstack dashboard "Horizon":



Dashboard



Network topology

Hypervisor | Compute Host | Resource Provider

Filter

Displaying 2 items

Host	Availability zone	Status	State	Time since update	Actions
compute1	nova	Enabled	Up	0 minutes	Disable Service
compute2	nova	Enabled	Up	0 minutes	Disable Service

Displaying 2 items

Compute nodes status

5 Summary and Future Work

This project focused on designing and implementing a scalable private cloud infrastructure using OpenStack, Ceph, and Kolla Ansible. The goal was to automate the deployment of OpenStack across a multi-node architecture while integrating Ceph for resilient and scalable storage. Vagrant was utilized to streamline infrastructure setup, ensuring consistency and efficiency throughout the deployment process.

The deployment process involved configuring three network adapters: a NAT adapter for internet connectivity, a host-only adapter for internal communication between VMs, and a NAT network adapter for OpenStack's internal IP allocation. This multi-adapter setup ensured robust connectivity and isolation for both internal and external communication. The OpenStack cluster was deployed across three nodes (1 controller/deployer and 2 compute nodes), while the Ceph cluster was set up on three additional nodes (1 admin and 2 normal nodes) using Cephadm.

The project also provided valuable hands-on experience in overcoming complex challenges, such as networking configurations, system optimizations, and troubleshooting. These experiences not only enhanced technical skills but also deepened the understanding of infrastructure design principles and best practices.

In conclusion, this project successfully delivered a fully functional private cloud infrastructure that is nearly production-ready. It highlights the importance of automation, scalability, and resilience in modern cloud environments and serves as a practical foundation for organizations seeking to deploy private clouds. Future work could include implementing SSL certificates, optimizing performance, and scaling the infrastructure to support larger workloads.

6 Appendix and References

References and Resources

The following resources were consulted during the design and implementation of this project:

- **GitHub Repository:**
  The complete scripts and configurations for deploying OpenStack with Ceph are available in the project repository:
  https://github.com/zakaryadev03/Openstack.git
- **OpenStack Documentation:**
  Official documentation for Kolla Ansible:
  https://docs.openstack.org/kolla-ansible/latest/
- **Ceph Installation Guide:**
  Step-by-step guide for installing and setting up a Ceph storage cluster on Ubuntu:
  https://kifarunix.com/install-and-setup-ceph-storage-cluster-on-ubuntu-2204/
- **Cephadm Documentation:**
  Official documentation for deploying and managing Ceph clusters using Cephadm:
  https://docs.ceph.com/en/latest/cephadm/install/
- **External Ceph Integration Guide:**
  Official guide for integrating an external Ceph cluster with OpenStack using Kolla Ansible:
  https://docs.openstack.org/kolla-ansible/latest/reference/storage/external-ceph-guide.html