



ASSIGNMENT -1

20INMCA565- CLOUD COMPUTING

SUBMITTED TO: BINUMON SIR

SUBMITTED BY: ROHAN KURIAN MATHEW

ROLL NO: 47

OpenStack

OpenStack is an open-source cloud computing platform that provides a set of software tools for building and managing public and private clouds. It allows you to deploy and manage large networks of virtual machines, storage, and networking resources, similar to what commercial cloud providers like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure offer.

Key Components of OpenStack

- **Compute (Nova):** Nova is the component responsible for provisioning and managing large networks of virtual machines (VMs). It handles the creation, management, and scaling of virtual servers on demand.
- **Storage:**
 - **Block Storage (Cinder):** Provides persistent block storage to virtual machines. It's similar to attaching a virtual hard drive to a VM.
 - **Object Storage (Swift):** A distributed storage system for storing and retrieving unstructured data. It can store large amounts of data that can be accessed from anywhere, much like Amazon S3.
- **Networking (Neutron):** Manages the network connectivity and IP addressing within the OpenStack cloud. Neutron allows users to create and manage networks, subnets, routers, and firewalls.
- **Dashboard (Horizon):** A web-based interface for managing OpenStack services. Horizon allows users to interact with the various components of OpenStack without needing to use the command line.
- **Identity Service (Keystone):** Handles authentication and authorization for all OpenStack services. Keystone manages user accounts and API access, ensuring that only authorized users can access certain resources.
- **Image Service (Glance):** Provides discovery, registration, and delivery services for virtual machine images. Glance allows users to store and retrieve disk images that can be used to launch VMs.
- **Orchestration (Heat):** Provides a way to automate the deployment and management of cloud applications using templates. Heat allows users to define the infrastructure required for an application and automatically deploy it.
- **Telemetry (Ceilometer):** Collects and stores metering and usage data for billing, benchmarking, and scalability purposes.

Use Cases for OpenStack

Private Cloud: Many organizations use OpenStack to build private clouds, which provide the flexibility and scalability of a public cloud while maintaining control over data and resources.

Public Cloud: Some service providers use OpenStack to offer public cloud services, competing with other public cloud providers like AWS and Google Cloud.

Hybrid Cloud: OpenStack can be used to create a hybrid cloud environment, where resources are shared between private and public clouds, providing greater flexibility and cost efficiency.

Telecoms and NFV (Network Functions Virtualization): OpenStack is often used in telecom environments to virtualize network functions, replacing traditional hardware-based network appliances with software running on standard servers.

High-Performance Computing (HPC): Organizations with large-scale computing needs, such as research institutions, use OpenStack to manage and scale their HPC resources.

OpenStack Installation

OpenStack can be deployed in various configurations depending on your needs (single-node, multi-node, etc.). Below is a general guide for installing OpenStack using the OpenStack-Ansible method on a single-node environment, which is commonly used for testing and development purposes.

Step 1: Prepare the System

1. Update the System Packages:

```
sudo apt update && sudo apt upgrade -y
```

2. Install Essential Packages:

```
sudo apt install git python3-pip -y
```

3. Set Hostname:

```
sudo hostnamectl set-hostname <your_hostname>
```

4. Disable Swap (Required by OpenStack):

```
sudo swapoff -a  
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

5. Reboot the Server:

```
sudo reboot
```

Step 2: Install OpenStack-Ansible

1. Clone the OpenStack-Ansible Repository:

```
git clone https://opendev.org/openstack/openstack-ansible /opt/openstack-ansible
```

2. Navigate to the Directory and Check Out the Desired Branch:

```
cd /opt/openstack-ansible
git checkout stable/zed # Replace 'zed' with the
desired release
```

3. Run the Bootstrap Script:

```
sudo scripts/bootstrap-ansible.sh
```

4. Prepare the Configuration Files:

```
sudo cp -R /opt/openstack-
ansible/etc/openstack_deploy /etc/
cd /etc/openstack_deploy
sudo cp openstack_user_config.yml.example
openstack_user_config.yml
sudo cp user_variables.yml.example
user_variables.yml
```

5. Edit openstack_user_config.yml to Configure Your Network Interface:

```
openstack_user_config:
  compute_hosts:
    computel:
      ip: <your_controller_ip>
  control_plane:
    ip: <your_controller_ip>
```

Step 3: Run the OpenStack-Ansible Playbooks

1. Run the Playbooks to Set Up Hosts, Infrastructure, and OpenStack Services:

```
cd /opt/openstack-ansible/playbooks
sudo openstack-ansible setup-hosts.yml
```

```
sudo openstack-ansible setup-infrastructure.yml
sudo openstack-ansible setup-openstack.yml
```

Step 4: Accessing the OpenStack Dashboard

1. Once the Installation Completes:

- Access the Horizon Dashboard by navigating to the IP address of your controller in a web browser:

```
http://<your_controller_ip>/horizon
```

2. Login Using the Default Credentials:

- Username: admin
- Password: Retrieve from
/etc/openstack_deploy/admin_password (or set during installation)

Step 5: Post-Installation Tasks

1. Create and Manage Projects, Users, and Resources:

- You can manage these through the dashboard or OpenStack CLI.

2. Verify Services:

- Use the following commands to verify that services are up and running:

```
openstack service list
```

```
openstack compute service list
```

3. Manage OpenStack Using the CLI:

- Install the OpenStack client:

```
sudo apt install python3-openstackclient
```

- Authenticate:

```
source /etc/openstack_deploy/admin-openrc.sh
```

Troubleshooting

1. Logs:

- Check logs in `/var/log/` for any issues during installation.

2. Services:

- Ensure all OpenStack services are running. Use `'systemctl status'` to check individual services.