

Building Software Systems

Lecture 3.4

Introduction to Private Clouds and OpenStack

SAURABH SRIVASTAVA

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IIT (ISM) DHANBAD



What is a Private Cloud

We have seen an example of a Public Cloud Platform in the previous two Lectures

- We saw how AWS can be useful for us in building cloud-native applications
- There are solutions at the levels of IaaS, PaaS and SaaS

There is one drawback though, of using AWS

- You only have partial control over where your data is stored
- This may be an important factor for some organisations

While not an easy assignment, such organisations can build their “own” cloud platform

- Such a platform will require them to manage a Data Centre
- But if they can afford (and are willing) to do so, there is a common open-source solution can help
- The solution is called *OpenStack*

What is OpenStack

OpenStack is a complex and flexible cloud computing platform

- It can be configured to execute on your own hardware
- It is free to use and is open-sourced

DevStack is a version that you may try out on a VM

- Installing OpenStack is a challenging task
- This is because it has a number of components, and many possible deployment architectures
- If you just wish to see, if OpenStack is right for your organization's use, you may try out DevStack
- It is supposed to provide a testing environment for developers as well as a means to try out OpenStack
- You may clone the following repo to start:
`https://opendev.org/openstack/devstack`

You can define some basic settings in a local file called `local.conf`

- The next step is to invoke the script `stack.sh` and the installation will start in an autopilot mode

OpenStack Components – Nova

Overview:

- Nova is the primary computing engine behind OpenStack
- It's responsible for managing the lifecycle of compute instances in cloud environments

Key Features:

- **Instance Management:** Provision and manage virtual machines with various configurations
- **Compute Resources:** Dynamically allocate resources (CPU, memory, and storage) to meet demand
- **Scalability:** Designed to scale horizontally on standard hardware, supporting a vast number of VMs
- **APIs:** Rich APIs for integration and automation with other OpenStack services and third-party tools

Use Cases:

- Web hosting, application development, testing environments, ...
- ... and any scenario requiring scalable compute resources

OpenStack Components – Neutron

Overview:

- Neutron provides "networking-as-a-service" between interface devices (e.g., vNICs) managed by other OpenStack services like Nova

Key Features:

- **Pluggable Architecture:** Supports different networking technologies and vendors
- **Network Services:** Offers services like LBaaS (Load Balancer as a Service), FWaaS (Firewall as a Service), and VPNaaS (VPN as a Service)
- **Tenant Networks:** Enable users to create their own networks, routers, IP addresses, and manage their connectivity
- **Scalability and Performance:** Designed to scale and meet the networking needs of a large number of virtual machines

Use Cases:

- Custom network topologies, network segmentation for security, and scalable multi-tenant cloud environments

OpenStack Components – Cinder

Overview:

- Cinder provides persistent block storage to running instances
- Its architecture is designed to be scalable and to integrate with various storage backends

Key Features:

- **Volume Management:** Create, attach, detach, and delete block storage volumes to instances
- **Snapshot Management:** Take snapshots for volumes, allowing for data backup and recovery
- **Volume Types:** Support for different storage backends and performance characteristics
- **Multi-Backend Configuration:** Configure Cinder to work with different storage solutions simultaneously

Use Cases:

- Database storage, file system storage, and storage for applications requiring high performance and persistence

OpenStack Components – Swift

Overview:

- Swift is designed to store and retrieve unstructured data at a large scale
- It offers a durable, scalable, and fault-tolerant storage system for data such as files, videos, and images

Key Features:

- **High Availability:** Data is replicated across different physical locations to ensure durability and availability
- **Scalability:** Capable of scaling to multiple petabytes, handling large amounts of data effortlessly
- **RESTful API:** Offers a simple API for storing and retrieving files, making it accessible from anywhere
- **Security:** Supports container-level access control to manage read and write permissions

Use Cases:

- Web and mobile applications, backup and archive, content distribution, and big data analytics

OpenStack Components – Keystone

Overview:

- Keystone provides identity services for OpenStack, including user authentication and authorization, service discovery, and distributed multi-tenant authorization

Key Features:

- **Unified Authentication:** Supports multiple forms of authentication including username/password, token-based, and third-party systems
- **Service Catalogue:** Keeps a registry of all services available in the OpenStack cloud, facilitating service discovery
- **Tenant Management:** Organizes users into projects and groups for managing access and quotas
- **Role-Based Access Control (RBAC):** Fine-grained access control mechanisms to manage who can do what

Use Cases:

- Secure API access, centralized user management, service discovery, and multi-tenancy support

OpenStack Components – Glance

Overview:

- Glance provides discovery, registration, and delivery services for disk and server images
- It supports a variety of image formats and storage backends

Key Features:

- **Image Storage:** Store and manage images used to boot instances
- **Image Discovery:** Easily find images with search and filtering capabilities
- **Snapshot Management:** Create snapshots of running instances for backup or replication
- **Multiple Storage Backends:** Compatible with object storage (like Swift), block storage, and filesystems

Use Cases:

- Rapid deployment of instances, image catalogue management, and backup/restore operations

OpenStack Components – Horizon

Overview:

- Horizon is the official web-based dashboard for managing and using OpenStack services
- It provides a user-friendly interface for administrators and users to interact with the OpenStack cloud

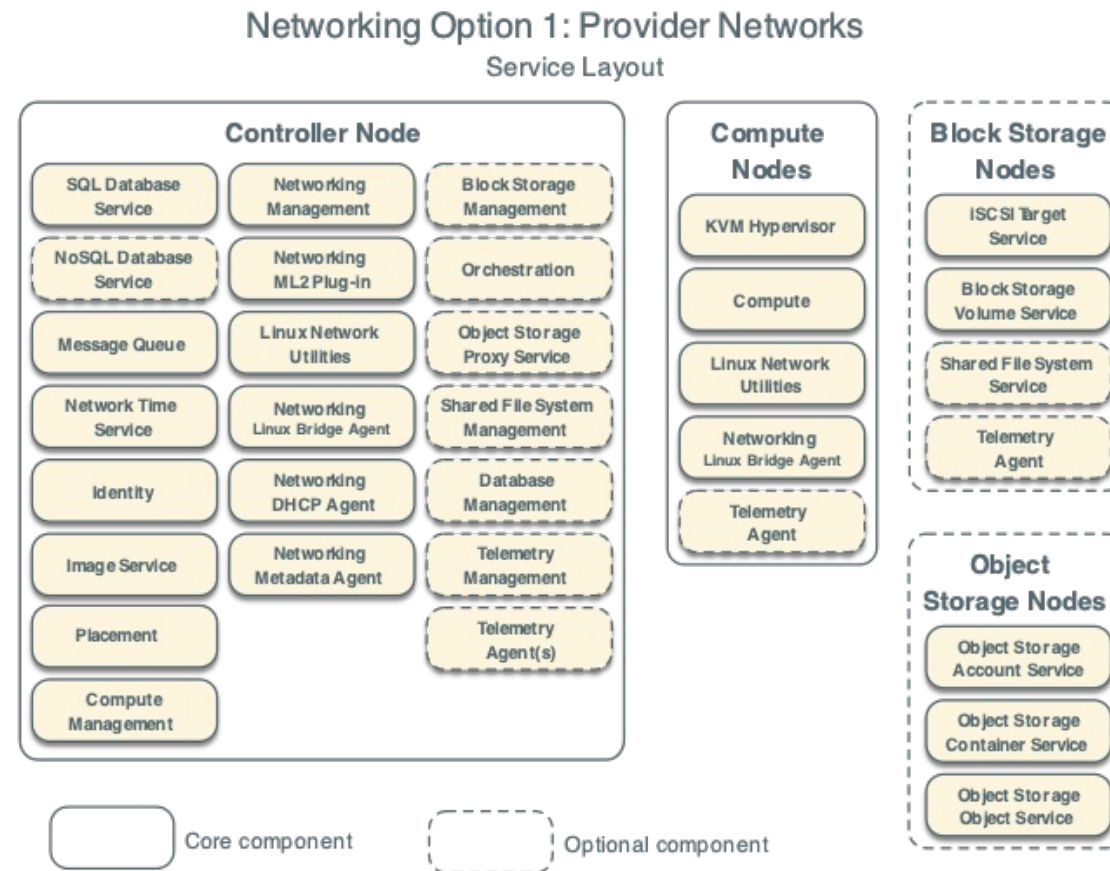
Key Features:

- **Ease of Use:** Intuitive graphical interface that simplifies cloud management tasks
- **Customizable:** Modular design allowing for customization and extension to meet specific needs
- **Integrated:** Access and manage all core OpenStack services through a single dashboard
- **Real-Time Updates:** Monitor resources and view logs in real-time for efficient cloud operations

Use Cases:

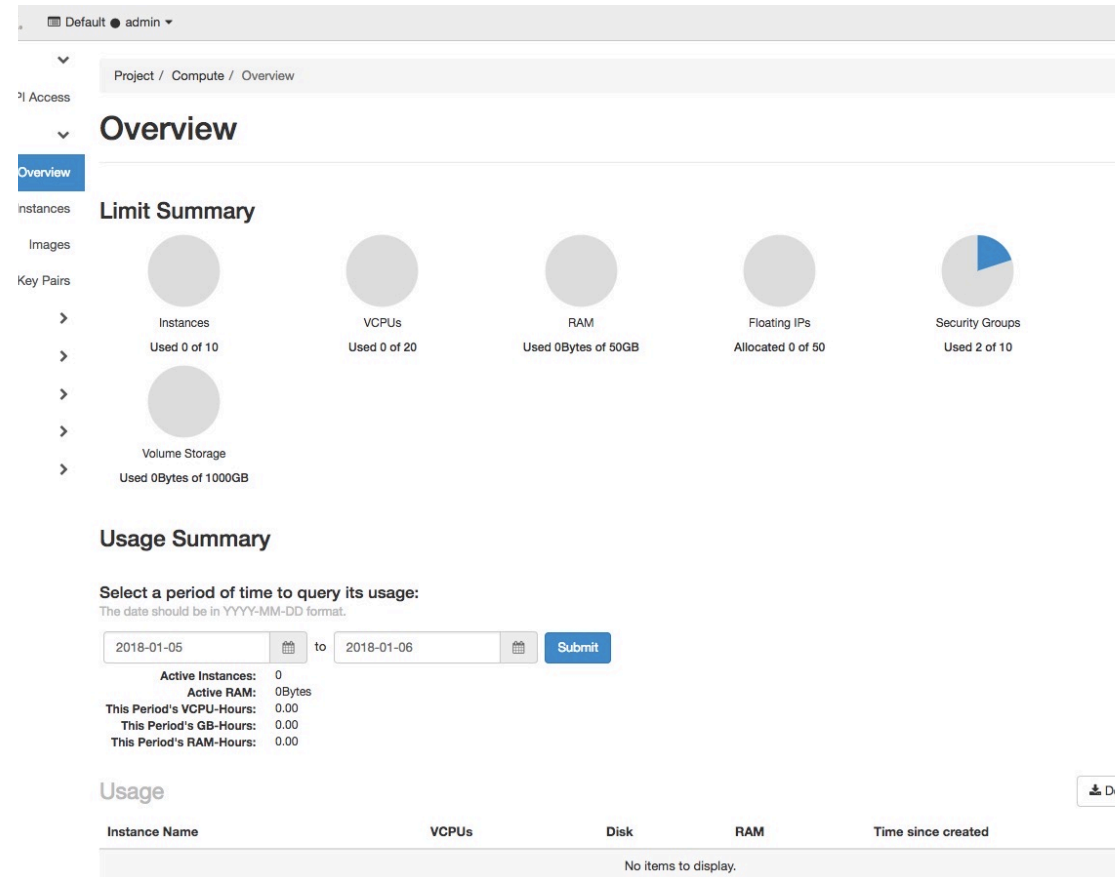
- Cloud resource management, monitoring and diagnostics, user and project management, and accessing OpenStack services

Sample Architecture for OpenStack



Source: [OpenStack Installation Overview](#)

The Horizon Dashboard



Source: [Horizon Dashboard Login](#)

Homework

If possible, create a VM on your laptop with 4 GB of Primary Memory and around 50 GB of storage

- Install Ubuntu 22.04 LTS on the same
- Install OpenStack through DevStack over the VM
- This link may be helpful:

<https://medium.com/@kcoupal/how-to-install-openstack-on-ubuntu-22-04-with-devstack-3336c01ddcfa>