

**University Of Petroleum and Energy Studies,**  
**Dehradun**



**Cloud Application Development**  
**(Lab)**

**Submitted by:**

Name- Vikash kumar

SAP: 500087705

Enrolment No. : R2142201906

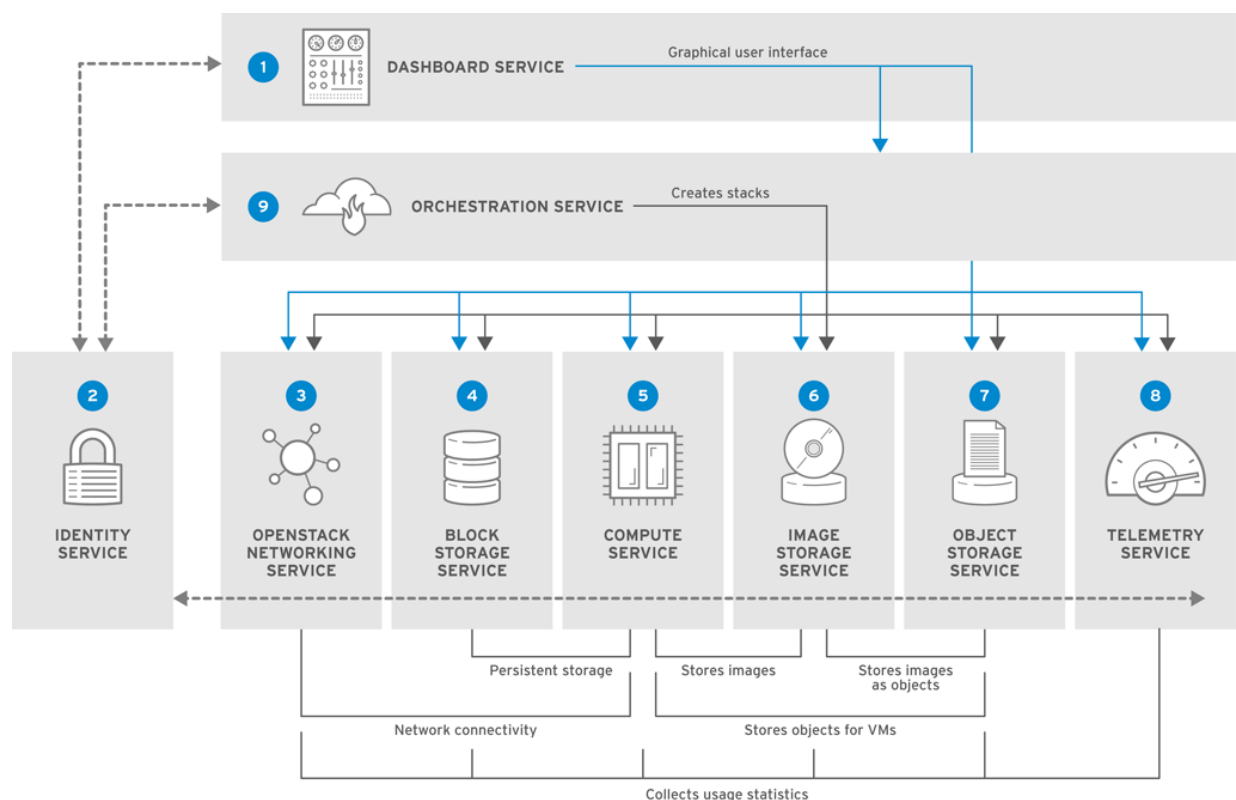
batch: B4.CCVT

**Submitted To:-** Saurabh Shanu sir

In order to run multiple applications or operating systems organizations often turn towards virtualization, which enables the use of multiple softwares or operating systems without any direct installation using the tools like hypervisor. This virtualization also allows abstraction of computing resources from physical Hardwares such as servers.

Now, openstack is a collection of software modules called projects that work together to create and manage private or public Cloud, using virtual resources pools. It supports massive scalability, easy to implement and each component or projects can be configured by the user using application programming interface(API).

Openstack supports open nature, so anyone can add additional components to openstack to meet their needs, but the openstack community has collaboratively identified six components as the core of openstack, officially maintained by the openstack community.



RHELOSP\_347192\_1015

These six components are:-

### **(1) Nova:-**

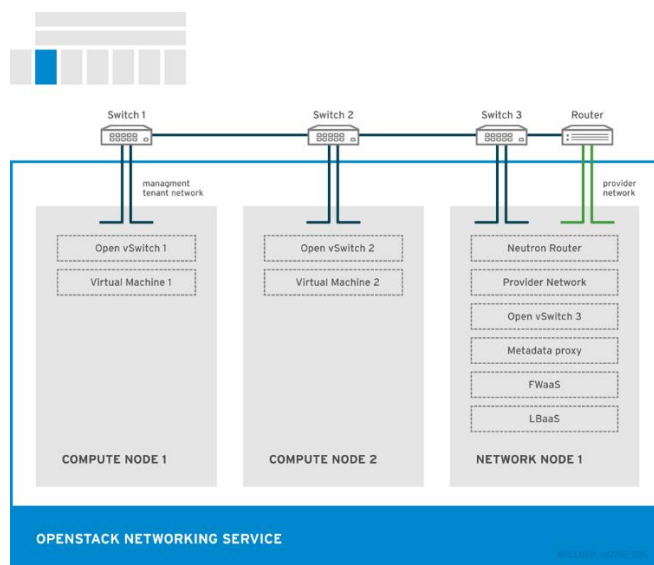
-- Nova which we can also termed as openstack compute provides virtual machines on demand and can be referred as core of openstack. It act just like a manager, what it does is it has got the software or api's bundle which goes and

talk to your hypervisor that's the whole purpose of nova and it manages the full life cycle as well starting, shutting ,downtime and then creating, deleting and then taking snapshots, everything are handled by nova..

- The major part of nova is that it takes care of VMs, nova got inbuilt messaging mechanism which is used to communicate between all its different modules, it also contains central database that is shared logically between all the components, nova also got a fault tolerant API and the messages are also designed in a very fault tolerant way. for example if you are requesting a server to create a new VM. when we are starting the request, suddenly the server went down so immediately we will get some type of message that server went down or instead of forming the error it immediately goes and checks for next servers as well. So, it has all that mechanism built in so that it goes ahead make sure that our VM is created and reports us back that the VM is up and then we can start managing our VM.
- Nova supports different types of hypervisors such as KVM, LXC, Hyper-V etc.
- Nova also helps to create VMs and manage the whole lifecycle and also used for hosting and managing our environments.

**(2) Neutron:-**

-- OpenStack Neutron, this is the networking service of OpenStack and it provides us network as a service. So that our instances can connect to a network which we can define using OpenStack Neutron.



but the question is that why did they ever introduce this networking service because for VMS there was nova already :-

- Actually, there was certain amount of problems there were layer to boundary limitations and there were VLAN limitations. Also because they were using the 802.1 queue standard which actually only allowed the VLANs to name from just 1 to 4094. so, we have limited amount of VLANs so that's why to avoid all of this to solve all of this problem open side Neutron was introduced. This was actually a standalone core project and it was created only because they wanted a completely separated software-defined networking stack for the OpenStack ecosystem.
- Neutron Network can be connected listen to different ports, we can define ports to which it will listen to and every request will come through the neutron router and security group.

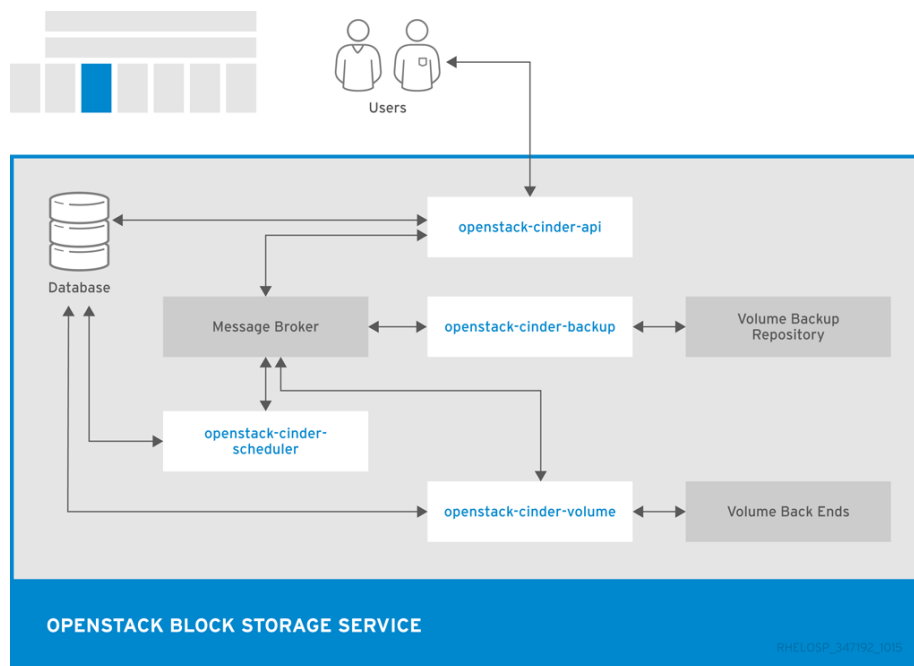
so what are security groups:-

- security groups are just a named collection of network access rules that are used to limit the types of traffic that has access to the instances. so we can define different security group, we can define different ports in the security groups to the one which will listen to. generally we actually use 422 port to get all those API requests and the Neutron API does that it gets all kind of API requests and it routes them to different components like plug-in agents.

### **(3) Cinder:-**

-- Cinder is a component of the OpenStack cloud computing platform that provides a simple API that allows users to create and manage virtual storage volumes, as well as attach and detach them from virtual machines. One of the main features of Cinder is its support for Block Storage-as-a-Service (BSaaS), which allows users to create and manage virtual storage volumes on demand, without the need for physical storage infrastructure.

Cinder enables users to scale storage capacity on demand, providing a flexible and cost-effective way to manage storage resources



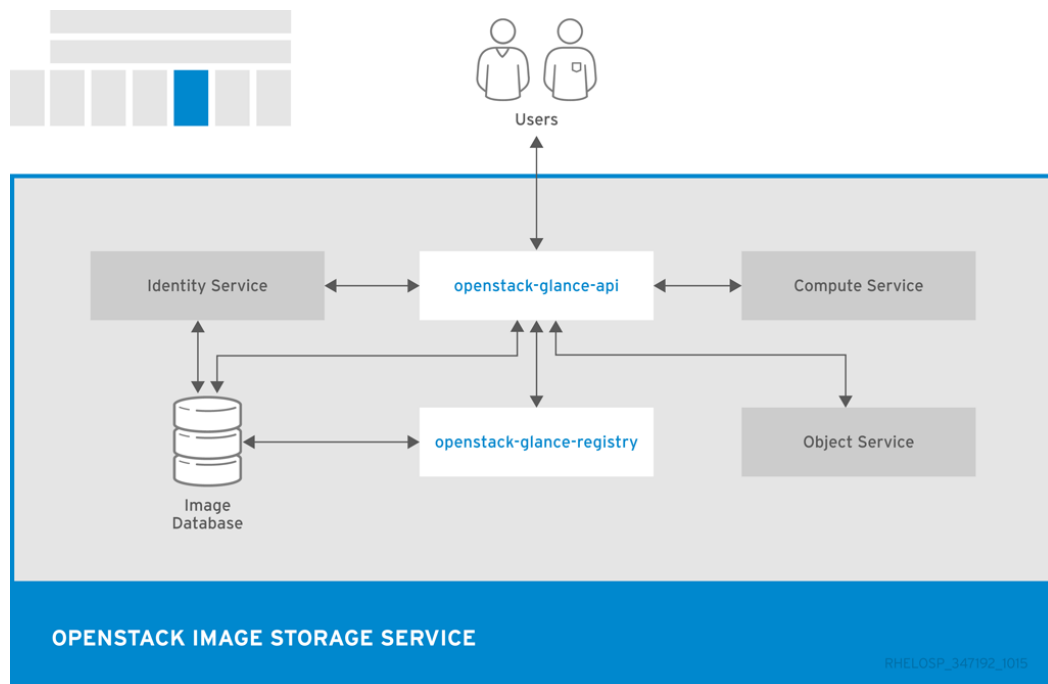
- It also supports a variety of storage technologies and allows users to create and manage multiple storage pools, including local and shared storage.
- Cinder also provides advanced features such as snapshots, backups, and replication, it can also be integrated with other OpenStack components, such as Nova for compute services , to provide a complete cloud computing platform.

#### **(4) Glance:-**

-- Glance is a component of the OpenStack cloud computing platform that provides a simple API that allows users to create and manage virtual machine images, as well as discover and retrieve images.

- It also supports a variety of image formats, and allows users to create and manage multiple image stores, including local and remote stores. One of the main features of Glance is its support for Image Management-as-a-Service (IMaaS), which allows users to create and manage virtual machine images on demand, without the need for physical storage infrastructure. Glance also provides advanced features such as image metadata, image tagging, and image sharing.

- OpenStack Glance has a client-server architecture that provides a REST API to the user through which requests to the server can be performed.



A Glance Domain Controller manages the internal server operations that is divided into layers. Specific tasks are implemented by each layer.

- All the file (Image data) operations are performed using glancestore library, which is responsible for interaction with external storage back ends and (or) local filesystem(s).
- The glancestore library provides a uniform interface to access the backend stores.
- Glance uses a central database (Glance DB) that is shared amongst all the components in the system and is sql-based by default.

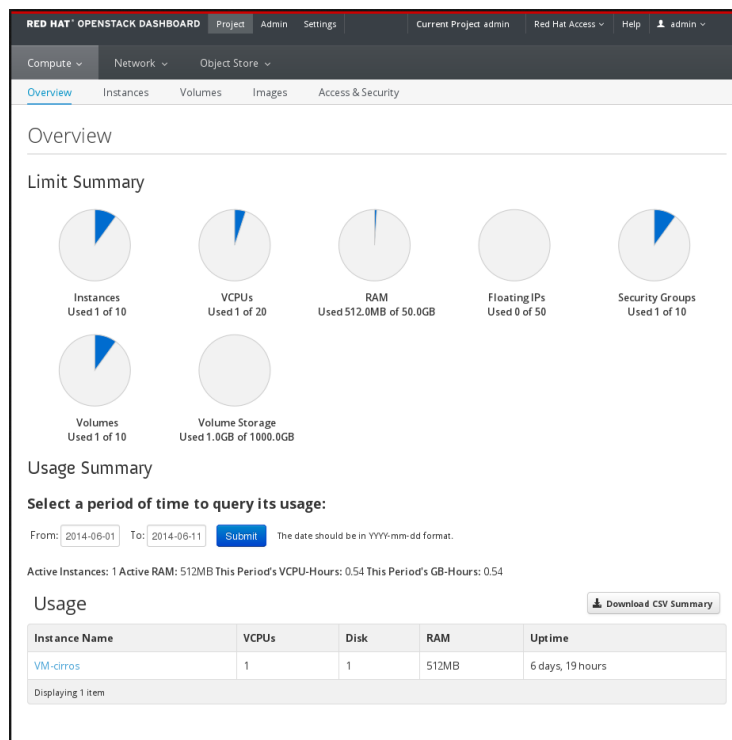
#### Components of Glance

- A client - any application that makes use of a Glance server.
- REST API - Glance functionalities are exposed via REST.
- Database Abstraction Layer (DAL) - an application programming interface (API) that unifies the communication between Glance and databases.
- Glance Domain Controller - middleware that implements the main Glance functionalities such as authorization, notifications, policies, database connections.

- Glance Store - used to organize interactions between Glance and various data stores.
- Registry Layer - optional layer that is used to organise secure communication between the domain and the DAL by using a separate service.

## **(5)Horizon:-**

-- Horizon is the official dashboard for OpenStack, it is a web-based user interface for OpenStack services that allows users to manage their cloud resources through a web browser. It provides a simple and user-friendly interface for creating and managing virtual machines, networks, storage, and other resources.



- Horizon provides a centralized point of access for managing all aspects of an OpenStack cloud. It allows users to create and manage virtual machines, networks, storage, and other resources using a web-based user interface. It also includes built-in monitoring and alerting features to help users keep track of their cloud resources.
- One of the main features of Horizon is its support for multi-tenancy, which allows users to create and manage multiple projects, each with their own resources, users, and access controls. This allows different teams or departments within an organization to have their own separate cloud environments.

- Horizon also provides advanced features such as role-based access control, integration with other OpenStack services such as Keystone for authentication and Glance for image management, and support for third-party plugins.

## **(6)Swift:-**

--We use OpenStack Object Storage to store our files. To do this, we can use a tool provided, named swift. This allows us to operate our OpenStack Object Storage environment by allowing us to create containers, upload files, retrieve them, and set required.

- Swift is a highly available, distributed, eventually consistent object/blob store. Organizations can use Swift to store lots of data efficiently, safely, and cheaply.
- OpenStack Object Storage, also known as Swift, is the service that allows massively scalable and highly redundant storage on commodity hardware. This service is analogous to Amazon's S3 storage service and is managed in a similar way under OpenStack.
- With OpenStack Storage, we can store many objects of virtually unlimited size—restricted by the available hardware—and grow our environment as needed, to accommodate our storage. The highly redundant nature of OpenStack Object Storage is ideal for archiving data (such as logs) as well as providing a storage system that OpenStack Compute can use for virtual machine instance templates.

