**OpenStack**

**Manual Installation in Virtual Machines using Virtual Box**

Table Contents

[**Introduction** 4](#_Toc66303540)

[**Revision History** 4](#_Toc66303541)

[**Overview** 5](#_Toc66303542)

[**Architectural Design** 5](#_Toc66303543)

[Conceptual Architecture 6](#_Toc66303544)

[Logical Architecture 7](#_Toc66303545)

[OpenStack Services 7](#_Toc66303546)

[OpenStack Services – Core 8](#_Toc66303547)

[OpenStack Services - Advance 10](#_Toc66303548)

[**OpenStack Installation Steps** 12](#_Toc66303549)

[Network Environment 12](#_Toc66303550)

[Host Systems Operating System 12](#_Toc66303551)

[Passwords 12](#_Toc66303552)

[Host Addresses 13](#_Toc66303553)

[Host SSH Users 13](#_Toc66303554)

[Firewall & Default Ports 14](#_Toc66303555)

[**Controller Installation** 15](#_Toc66303556)

[Set-up - VM 15](#_Toc66303557)

[Configure Security, Networking, Install Linux Utilities 17](#_Toc66303558)

[Install & Configure Network Time Protocol 19](#_Toc66303559)

[Install Basic OpenStack Packages 19](#_Toc66303560)

[SQL Database – MariaDB 19](#_Toc66303561)

[Message Queue – RabbitMQ 20](#_Toc66303562)

[Memcached 20](#_Toc66303563)

[ETCD 21](#_Toc66303564)

[Install Keystone – Identity Management 22](#_Toc66303565)

[Install Glance – Image Service 24](#_Toc66303566)

[Install & Configure Nova(Compute Service) Controller 27](#_Toc66303567)

[Install & Configure Nova on Compute Nodes 32](#_Toc66303568)

[Install Neutron (Network Service) on Controller Node 32](#_Toc66303569)

[Install Cinder Block Storage Service on Controller Node 38](#_Toc66303570)

[Install Horizon Dashboard 40](#_Toc66303571)

[**Compute Installation** 42](#_Toc66303572)

[Set-up - VM 42](#_Toc66303573)

[Configure Security, Networking, Install Linux Utilities 43](#_Toc66303574)

[Install & Configure Network Time Protocol 45](#_Toc66303575)

[Install Basic OpenStack Packages 45](#_Toc66303576)

[Install Compute Service on Compute Node 45](#_Toc66303577)

[Install Neutron on Compute Node 48](#_Toc66303578)

[**Block Installation** 50](#_Toc66303579)

[Set-up VM 50](#_Toc66303580)

[Configure Security, Networking, Install Linux Utilities 52](#_Toc66303581)

[Install & Configure Network Time Protocols 54](#_Toc66303582)

[Install Basic OpenStack Packages 54](#_Toc66303583)

[Install Block Storage Service on Storage Node 54](#_Toc66303584)

**Introduction**:

The tutorial provides a step-by-step guide to manually install Open stack components in Virtual Machines using Virtual Box. This installation and configuration of Open stack is broadly categorized into 3 major parts.

1. **Controller Installation**
2. **Compute Installation**
3. **Block Installation**

We have configured the installation on 3 nodes, namely Controller, Compute, and Block. You can increase the nodes by adding more Computes & Block.

You can refer following link for complete OpenStack installation (Newton) on Ubuntu:

<https://docs.openstack.org/newton/install-guide-ubuntu/>

After becoming familiar with basic installation, configuration, operation, and troubleshooting of these OpenStack services, you should consider the following steps toward deployment using a production architecture:

* Determine and implement the necessary core and optional services to meet performance and redundancy requirements.
* Increase security using methods such as firewalls, encryption, and service policies.
* Implement a deployment tool such as Ansible, Chef, Puppet, or Salt to automate deployment and management of the production environment.

You will learn how easy it is to create Open stack in Virtual Machines using Virtual Box

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Changes Name** | **Updated By** | **Date** | **Version** |
| New Document | Sudhanshu Satyam | 2021/01/11 | 1.0.0 |
| Revision | Sudhanshu Satyam | 2021/02/16 | 1.0.1 |
|  |  |  |  |
|  |  |  |  |

# **Overview**

OpenStack is a software for building and managing cloud-computing platforms for public and private clouds. OpenStack is used to deploy virtual machines and other instances that handle different tasks for managing a cloud environment. OpenStack is considered as – Infrastructure as a Service (IaaS).

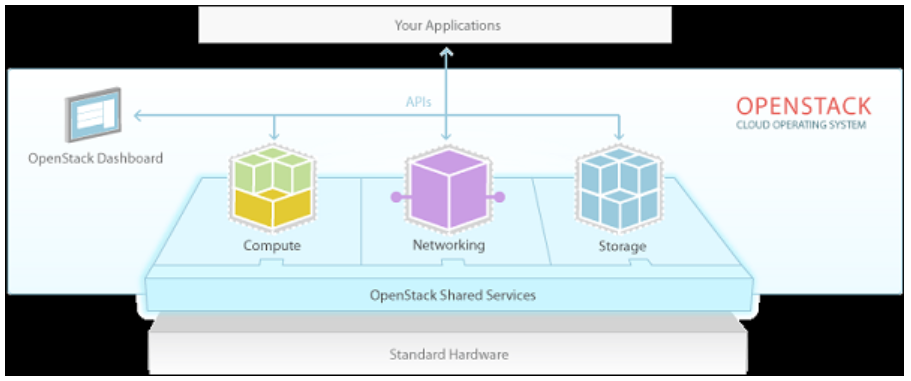


Image Source : <https://www.tutorialspoint.com/virtualization2.0/virtualization2.0_openstack.htm>

# **Architectural Design**

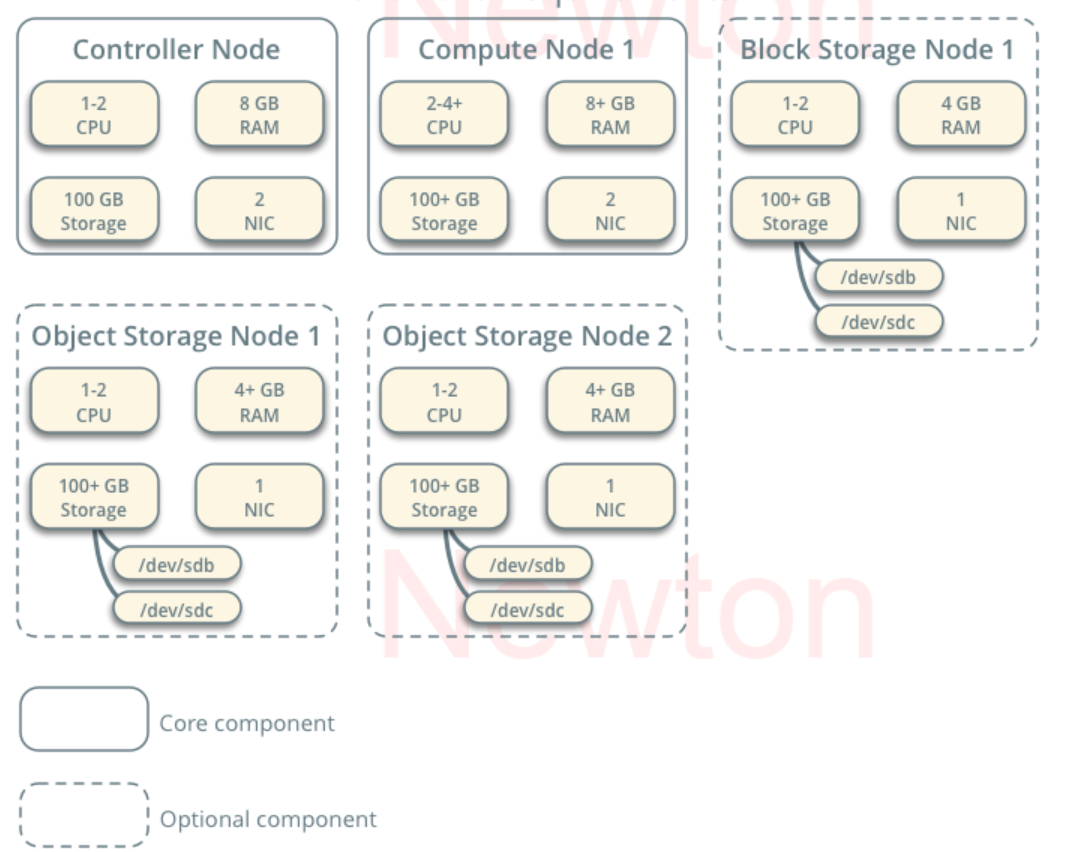


Image Source: <https://docs.openstack.org/install-guide/overview.html>

## Conceptual Architecture

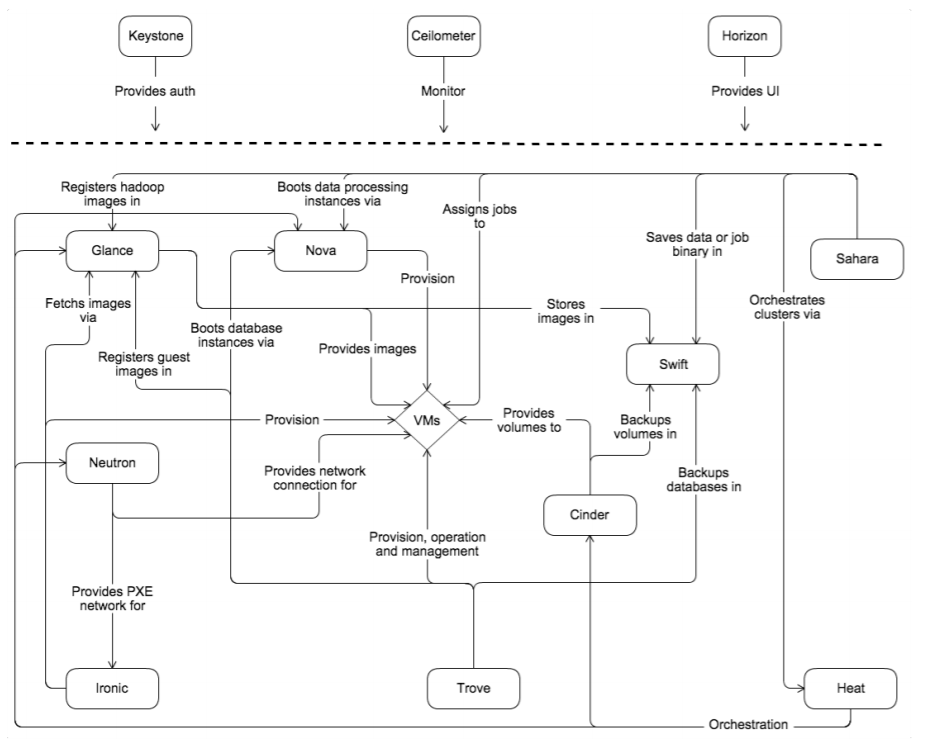


Image Source: <https://docs.openstack.org/install-guide/get-started-conceptual-architecture.html>

## Logical Architecture

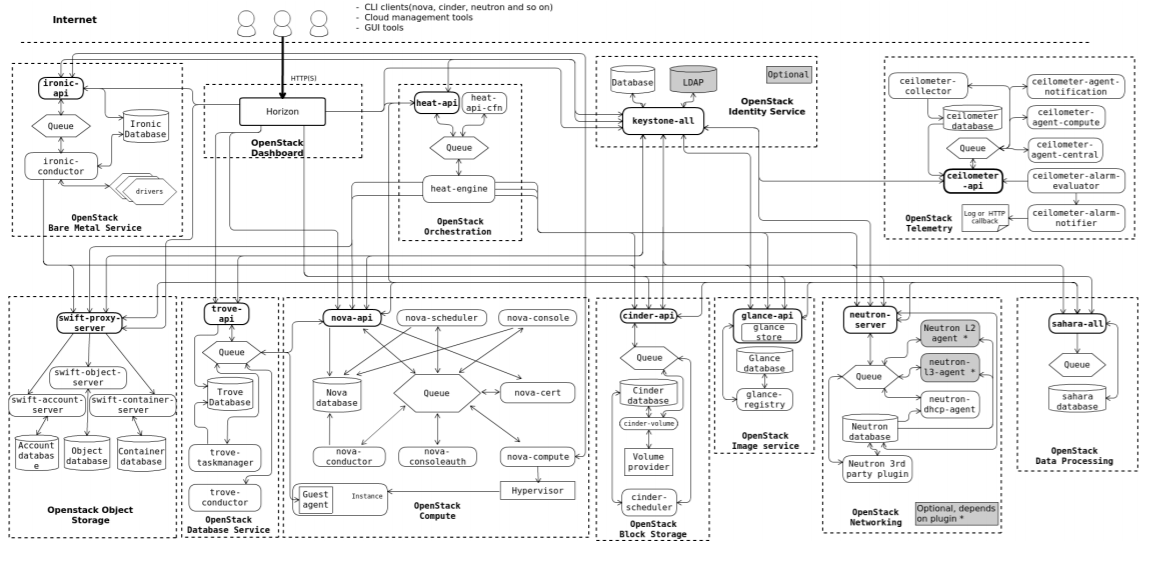
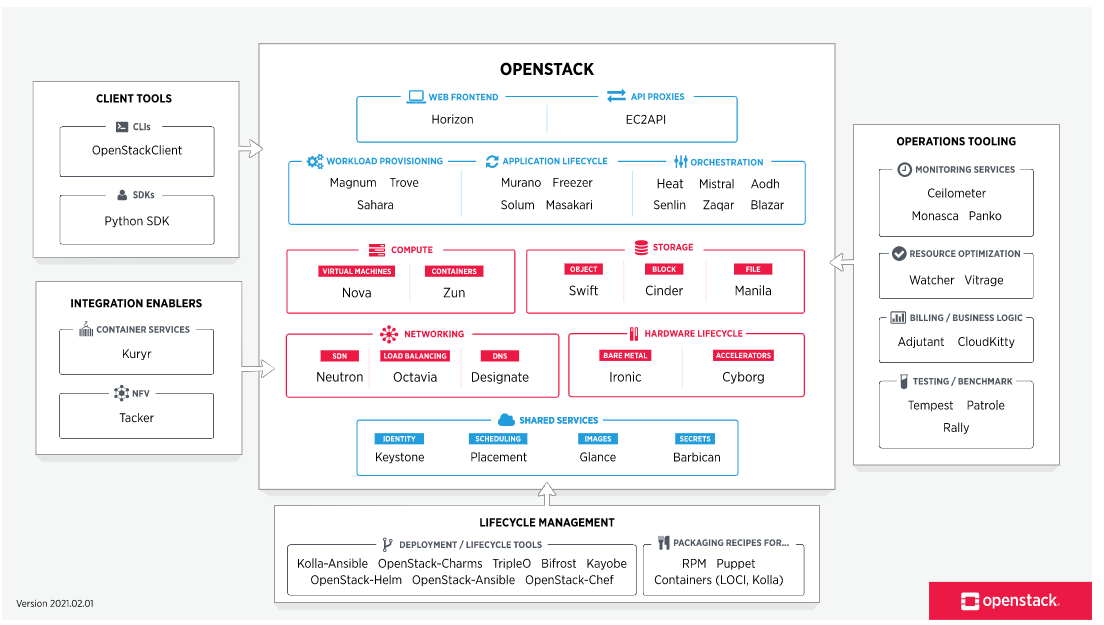


Image Source : <https://docs.openstack.org/install-guide/get-started-logical-architecture.html>

## OpenStack Services

OpenStack is broken up into services to allow you to plug and play components depending on your needs. The openstack map gives you an “at a glance” view of the OpenStack landscape to see where those services fit and how they can work together.



OpenStack Services are broadly classified into Core Services and Advanced Services. OpenStack Core services mainly includes 6 main services, that is frequently used during major deployments.

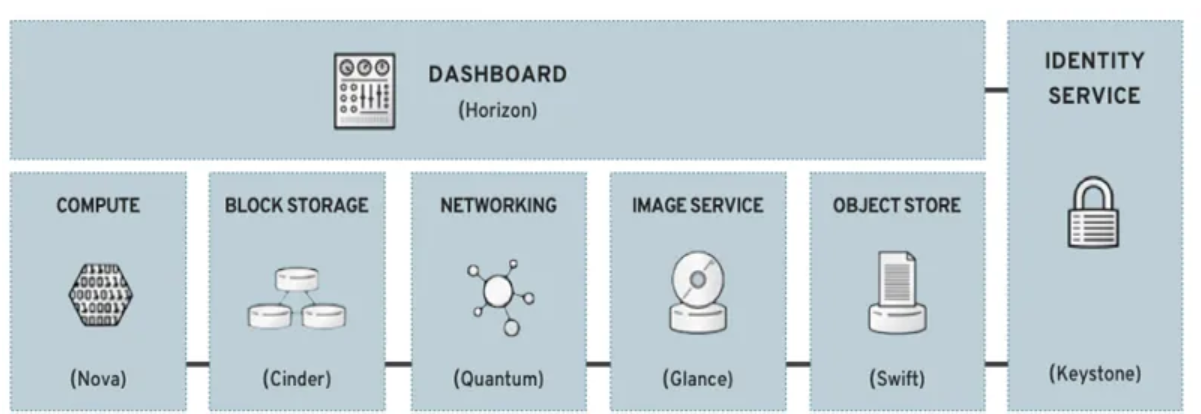


Image Source: <https://kartaca.com/en/components-of-openstack/>

## OpenStack Services – Core

* **Compute (Nova)** - OpenStack Compute is a cloud computing fabric controller, which manages pools of computer resources and work with virtualization technologies, bare metals, and high-performance computing configurations.
* **Image Service (Glance)** - OpenStack image service offers discovering, registering, and restoring virtual machine images. Glance has client-server architecture and delivers a user REST API, which allows querying of virtual machine image metadata and also retrieval of the actual image.
* **Object Storage (Swift)** - OpenStack Swift creates redundant, scalable data storage to store petabytes of accessible data. The stored data can be leveraged, retrieved and updated. It has a distributed architecture, providing greater redundancy, scalability, and performance, with no central point of control.
* **Dashboard (Horizon)** - Horizon is the authorized implementation of OpenStack’s Dashboard, which is the only graphical interface to automate cloud-based resources.
* **Identity Service (Keystone)** - Keystone provides a central list of users, mapped against all the OpenStack services, which they can access. It integrates with existing backend services such as LDAP while acting as a common authentication system across the cloud computing system.
* **Networking (Neutron)** - Neutron provides networking capability like managing networks and IP addresses for OpenStack. It ensures that the network is not a limiting factor in a cloud deployment and offers users with self-service ability over network configurations.

**Networking Option 1: Provider networks**

The provider networks option deploys the OpenStack Networking service in the simplest way possible with primarily layer-2 (bridging/switching) services and VLAN segmentation of networks. Essentially, it bridges virtual networks to physical networks and relies on physical network infrastructure for layer-3 (routing) services. Additionally, a DHCP service provides IP address information to instances.

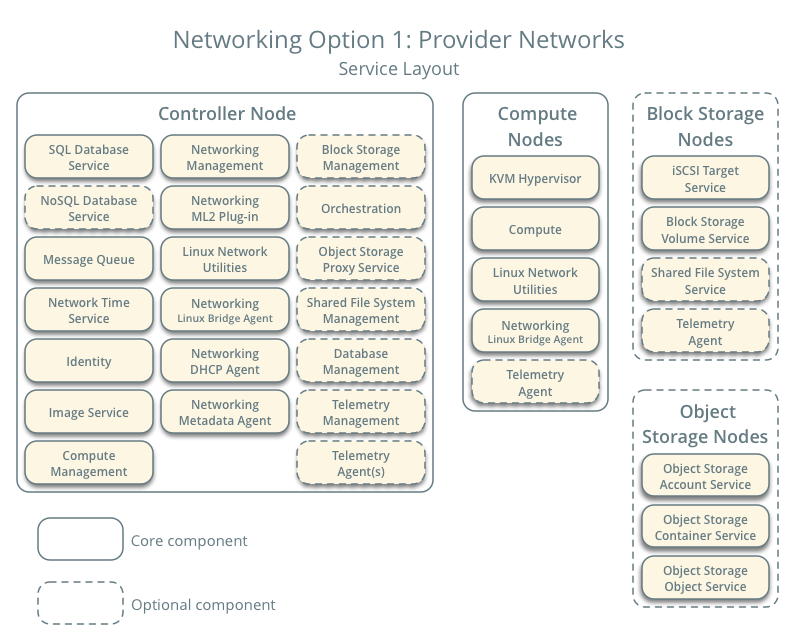


Image Source: <https://docs.openstack.org/install-guide/overview.html>

**Networking Option 2: Self-service networks¶**

The self-service networks option augments the provider networks option with layer-3 (routing) services that enable self-service networks using overlay segmentation methods such as VXLAN. Essentially, it routes virtual networks to physical networks using NAT. Additionally, this option provides the foundation for advanced services such as LBaaS and FWaaS.

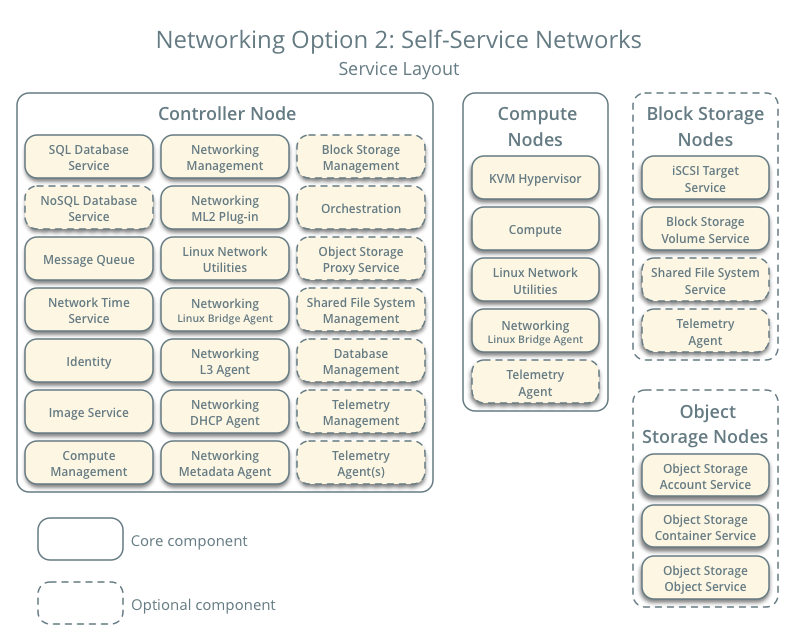


Image Source: <https://docs.openstack.org/install-guide/overview.html>

* **Block Storage (Cinder)** - OpenStack Cinder delivers determined block-level storage devices for application with OpenStack compute instances. A cloud user can manage their storage needs by integrating block storage volumes with Dashboard and Nova.

## OpenStack Services - Advance

* **Orchestration (Heat)** - Heat is a service to orchestrate multiple composite cloud applications through both the CloudFormation-compatible Query API and OpenStack-native REST API, using the AWS CloudFormation template format.
* **Workflow (Mistral)** - Mistral is a service that manages workflows. User typically writes a workflow using workflow language based on YAML and uploads the workflow definition to Mistral via its REST API. Then user can start this workflow manually via the same API or configure a trigger to start the workflow on some event.
* **Telemetry (Ceilometer)** - OpenStack Telemetry (Ceilometer) provides a Single Point Of Contact for billing systems, providing all the counters they need to establish customer billing, across all current and future OpenStack components. The delivery of counters is traceable and auditable, the counters must be easily extensible to support new projects, and agents doing data collections should be independent of the overall system.
* **Database (Trove)** - Trove is a database-as-a-service provisioning relational and a non-relational database engine.
* **Elastic map reduce (Sahara)** - Sahara is a component to easily and rapidly provision Hadoop clusters. Users will specify several parameters like the Hadoop version number, the cluster topology type, node flavor details (defining disk space, CPU and RAM settings), and others. After a user provides all of the parameters, Sahara deploys the cluster in a few minutes. Sahara also provides means to scale a preexisting Hadoop cluster by adding and removing worker nodes on demand.
* **Bare metal (Ironic)** - Ironic is an OpenStack project that provisions bare metal machines instead of virtual machines. It was initially forked from the Nova Baremetal driver and has evolved into a separate project. It is best thought of as a bare-metal hypervisor API and a set of plugins that interact with the bare-metal hypervisors. By default, it will use PXE and IPMI in concert to provision and turn on and off machines, but Ironic supports and can be extended with vendor-specific plugins to implement additional functionality.
* **Messaging (Zaqar/RabbitMQ)** - Zaqar is a multi-tenant cloud messaging service for Web developers. The service features a fully RESTful API, which developers can use to send messages between various components of their SaaS and mobile applications by using a variety of communication patterns. Underlying this API is an efficient messaging engine designed with scalability and security in mind. Other OpenStack components can integrate with Zaqar to surface events to end users and to communicate with guest agents that run in the "over-cloud" layer.
* **Shared file system (Manila)** - OpenStack Shared File System (Manila) provides an open API to manage shares in a vendor agnostic framework. Standard primitives include ability to create, delete, and give/deny access to a share and can be used standalone or in a variety of different network environments. Commercial storage appliances from EMC, NetApp, HP, IBM, Oracle, Quobyte, INFINIDAT and Hitachi Data Systems are supported as well as filesystem technologies such as Red Hat GlusterFS or Ceph.
* **DNS (Designate)** - Designate is a multi-tenant REST API for managing DNS. This component provides DNS as a Service and is compatible with many backend technologies, including PowerDNS and BIND. It doesn't provide a DNS service as such as its purpose is to interface with existing DNS servers to manage DNS zones on a per tenant basis.
* **Search (Searchlight)** - Searchlight provides advanced and consistent search capabilities across various OpenStack cloud services. It accomplishes this by offloading user search queries from other OpenStack API servers by indexing their data into ElasticSearch. Searchlight is being integrated into Horizon[124] and also provides a Command-line interface.
* **Key manager (Barbican)** - Barbican is a REST API designed for the secure storage, provisioning and management of secrets. It is aimed at being useful for all environments, including large ephemeral Clouds.
* **Container orchestration (Magnum)** - Magnum is an OpenStack API service developed by the OpenStack Containers Team making container orchestration engines such as Docker Swarm, Kubernetes, and Apache Mesos available as first class resources in OpenStack. Magnum uses Heat to orchestrate an OS image which contains Docker and Kubernetes and runs that image in either virtual machines or bare metal in a cluster configuration.
* **Root Cause Analysis (Vitrage)** - Vitrage is the OpenStack RCA (Root Cause Analysis) service for organizing, analyzing and expanding OpenStack alarms & events, yielding insights regarding the root cause of problems and deducing their existence before they are directly detected.
* **Rule-based alarm actions (Aodh)** - This alarming service enables the ability to trigger actions based on defined rules against metric or event data collected by Ceilometer or Gnocchi.

**Refer**: <https://www.openstack.org/software/project-navigator/openstack-components#openstack-services>

# **OpenStack Installation Steps**

## Network Environment

|  |  |  |  |
| --- | --- | --- | --- |
| **Network Type** | **CIDR** | **Gateway** | **Default Interface** |
| Management Network | 10.0.0.0/24 | 10.0.0.1 | eth0 |
| Provider Network | 203.0.113.0/24 | 203.0.113.1 | eth1 |

Reference: <https://docs.openstack.org/security-guide/networking/architecture.html>

## Host Systems Operating System

|  |  |
| --- | --- |
| **Name** | Ubuntu Server 16.04 LTS |
| **Download Link** | <https://www.ubuntu.com/download/server> |

## Passwords

|  |  |  |
| --- | --- | --- |
| **Description** | **Parameter** | **Value** |
| SQL Database 'root' Password | MySQL\_root | Openstack |
| '**admin**' User Password | ADMIN\_PASS | Openstack |
| Database password for Cinder | CINDER\_DBPASS | Openstack |
| '**cinder**' User Password | CINDER\_PASS | openstack |
| Database password for Horizon | DASH\_DBPASS | openstack |
| '**demo**' User Password | DEMO\_PASS | openstack |
| Database password for Glance | GLANCE\_DBPASS | openstack |
| '**glance**' User Password | GLANCE\_PASS | openstack |
| Database password for Keystone | KEYSTONE\_DBPASS | openstack |
| Secret for metadata server | METADATA\_SECRET | openstack |
| Database Password for Neutron | NEUTRON\_DBPASS | openstack |
| '**neutron**' User Password | NEUTRON\_PASS | openstack |
| Database Password for Nova | NOVA\_DBPASS | openstack |
| '**nova**' User Password | NOVA\_PASS | openstack |
| '**placement**' User Password | PLACEMENT\_PASS | openstack |
| RabbitMQ Password for '**openstack**' | RABBIT\_PASS | openstack |

**Note**: It is recommended to set username & password for every OpenStack Services as per standard rules defined in Security & Policy guidelines.

## Host Addresses

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **IPv4 Address** | **Netmask** | **DNS Nameserver** |
| controller | 10.0.0.11 | 255.255.255.0 | 8.8.8.8 |
| compute1 | 10.0.0.31 | 255.255.255.0 | 8.8.8.8 |
| compute2 | 10.0.0.32 | 255.255.255.0 | 8.8.8.8 |
| block1 | 10.0.0.41 | 255.255.255.0 | 8.8.8.8 |

We have created 3 VM’s namely, Controller, Compute, and Block for creating Virtual Box environment from scratch. You can further extend this environment by adding more Compute and Blocks, depending on your complex requirements.

## Host SSH Users

|  |  |  |
| --- | --- | --- |
| **Host** | **Username** | **Password** |
| controller | Openstack | openstack |
| compute1 | Openstack | openstack |
| compute2 | Openstack | openstack |
| block1 | Openstack | openstack |

**Note**: It is recommended to set username & password for every OpenStack Services as per standard rules defined in Security & Policy guidelines.

## Firewall & Default Ports

|  |  |  |
| --- | --- | --- |
| **Description** | **Service** | **Port** |
| Horizon Dashboard unencrypted | HTTP | 80 |
| SSL Enabled Services | HTTPS | 443 |
| Block Storage iSCSI target | iSCSI Target | 3260 |
| Common SQL Database | MariaDB | 3306 |
| Message Broker (AMPQ traffic) | RabbitMQ | 5672 |
| Block Storage Endpoints | Cinder | 8776 |
| Compute Endpoints | Nova | 8774 |
| Compute API | Nova | 8775 |
| Compute API | Nova | 8773 |
| Compute VM consoles | Nova | 5900-5999 |
| Compute VNC Proxy (browsers) | Nova | 6080 |
| Compute VNC Proxy (clients) | Nova | 6081 |
| Compute HTML5 console | Nova | 6082 |
| Identity Service admin endpoint | Keystone | 35357 |
| Identity Service public endpoint | Keystone | 5000 |
| Image Service API | Glance | 9292 |
| Image Service Registry | Glance | 9191 |
| Networking Service | Neutron | 9696 |

We have currently disabled firewall settings in the current environment.

|  |  |  |
| --- | --- | --- |
| sudo ufw status verbose | |  |
| sudo ufw disable | |  |
|  |  |  |

# **Controller Installation**

## Set-up - VM

**HW Config**

|  |  |  |
| --- | --- | --- |
| **Virtual** | Recommended | Actual |
| VCPU (cores) | 1-2+ | 2 |
| RAM | 4+ GB | 6 |
| Primary Disk | 10+ GB | 20 |

**VirtualBox Host-Only Network Ethernet Adapter #2**

Configure Adapter Manually IPv4 Addr 10.0.0.1

IPv4 Net Mask 255.255.255.0

DHCP Disabled

**NAT Network ProviderNetwork1**

CIDR 203.0.113.0/24 DHCP Disabled

**NAT Network NatNetwork1**

CIDR 10.10.10.0/24 DHCP Enabled

**Network Interfaces**



Promiscuous Mode: Allow All

**Operating System**

|  |  |  |
| --- | --- | --- |
| **Name** | Ubuntu Server 16.04 LTS |  |
| **Link** | https://www.ubuntu.com/download/server | |

**Operating System Installation Options**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Recommended |  | **Actual** |
| 1. Language |  |  | English |  | English |
| 2. Hit F4 to choose 'Modes' | |  | Install a Minimal Virtual Machine | | Install a Minimal Virtual Machine |
| 3. Press Enter to 'Install Ubuntu Server' | | |  |  |  |
| 4. Choose Language | |  | English-English |  | English-English |
| 5. Select your location | |  | United States |  | United States |
| 6. Detect keyboard layout? | |  | No |  | No |
| 7. Keyboard layout | |  | English (US) |  | English (US) |
| 8. Primary network interface | |  | enp0s3 |  | enp0s3 |
| 9. Network configration method | | | Configure network manualy | | Configure Network manualy |
| 10. IP address |  |  | 10.0.0.11 |  | 10.0.0.11 |
| 11. Netmask |  |  | 255.255.255.0 |  | 255.255.255.0 |
| 12. Gateway |  |  | <nothing> |  | <nothing> |
| 13. Name server address | |  | 8.8.8.8 |  | 8.8.8.8 |
| 14. Hostname |  |  | controller |  | controller |
| 15. Domain name | |  |  |  |  |
| 16. Full name of the new user | | |  |  | openstack |
| 17. Username for your account | | |  |  | openstack |
| 18. Choose password for the new user | | |  |  | openstack |
| 19. Encrypt your home directory? | | |  |  | no |
| 20. Select your time zone | |  |  |  | Eastern |
| 21. Partitioning method | |  | use entire disk and set up LVM | | use entire disk and set up LVM |
| 22. HTTP Proxy | |  |  |  | none |
| 23. How to manage upgrades? | | | No automatic updates | | No automatic updates |
| 24. Choose software to install | | | OpenSSH Server |  | OpenSSH Server |
| 25. Install GRUB? | |  | Yes |  | Yes |

## Configure Security, Networking, Install Linux Utilities

**Configure 'sudo' access for**

|  |  |  |
| --- | --- | --- |
| **sudo su** |  |  |
| **visudo** |  |  |

add following line at the bottom of the file:

|  |  |
| --- | --- |
| openstack **ALL=(ALL) NOPASSWD:ALL** |  |

save, exit and run sudo su again to test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Edit /etc/hosts** | | | | | |
| Remove 127.0.1.1 controller, if present | | | | | |
| Make sure following lines are present: | | | | | |
| **10.0.0.11 controller** | | | | | |
| **10.0.0.31 compute1** | | | | | |
| **10.0.0.41 block1** | | | | | |
|  | | | | | |
| Edit /etc/default/grub to include: | | | | | |
| **GRUB\_CMDLINE\_LINUX="net.ifnames=0 biosdevname=0"** | | | | | |
| Run command: | | | | | |
| **update-grub** | | | | | |
| **Reboot** | | | | | |
|  | | | | | |
| **Enable Network Interfaces** | | | | | |
| **sudo su** | | | | | |
| Edit **/etc/network/interfaces** | | | | | |
| Make sure following Interfaces definitions are present: | | | | | |
| **auto eth0** |  |  |  | |
| **iface eth0 inet static** | | |  | |
| **address 10.0.0.11** | |  |  | |
| **netmask 255.255.255.0** | | |  | |
| **dns-nameservers 8.8.8.8** | | |  | |
| **auto eth1** |  |  |  | |
| **iface eth1 inet manual** | | |  | |
| **up ip link set dev eth1 up** | | |  | |
| **down ip link set dev eth1 down** | | | | |
| **auto eth2** |  |  |  | |
| **iface eth2 inet dhcp** | |  |  | |
|  |  |  |  | |
| Reboot the system | | | |
| Run 'ifconfig' as superuser to verify settings. | | | |
| Verify connectivity to other hosts, once configured | | | |
| **ping -c 3 openstack.org** | | | |
| **ping -c 3 compute1** | | | |
| **ping -c 3 block1** | | | |
|  | | | |
| **Install basic Linux Utilities** | | | |
| Run following commands: | | | |
| **sudo su** | | | |
| **apt update** | | | |
| **apt install vim glances curl** | | | |
| **apt upgrade -y** | | | |

## Install & Configure Network Time Protocol

|  |
| --- |
| Install and Configure Components |
|  |
| **sudo su** |
| **apt install chrony** |
|  |
| Edit **/etc/chrony/chrony.conf**: |
| set **server** to your Orgaznization's NTP Server, if you have one |
| set **allow** to **10.0.0.0/24** |
| save and quit |
| Restart **chrony** service: |
| **service chrony restart** |
|  |
| Verify: |
| **chronyc sources** |

## Install Basic OpenStack Packages

|  |
| --- |
| **sudo su** |
| **apt install software-properties-common** |
| **add-apt-repository cloud-archive:pike** |
| **apt update && apt dist-upgrade** |
| **Reboot** |
| **apt install python-openstackclient** |

## SQL Database – MariaDB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Install and Configure Packages | | | | |
| **sudo su** | | | | |
| **apt install mariadb-server python-pymysql** | | | | |
|  | | | | |
| Create and edit MariaDB configuration file: /etc/mysql/mariadb.conf.d/99 openstack.cnf | | | | |
| Put following 7 lines in the file: | | | | |
| **[mysqld]** |  |  |  |
| **bind-address = 10.0.0.11** | | |  |
| **default-storage-engine = innodb** | | |  |
| **innodb\_file\_per\_table = on** | | |  |
| **max\_connections = 4096** | | |  |
| **collation-server = utf8\_general\_ci** | | | |
| **character-set-server = utf8** | | |  |

|  |
| --- |
| Restart MariaDB service: |
| **service mysql restart** |
|  |
| Secure the Database Service: |
| **mysql\_secure\_installation** |

## Message Queue – RabbitMQ

Install and Configure Packages:

|  |
| --- |
| **sudo su** |
| **apt install rabbitmq-server** |
|  |
| Add **openstack** user: |
| **rabbitmqctl add\_user openstack openstack** |
|  |
| Configure permissions for **openstack** user: |
| **rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"** |

## Memcached

|  |
| --- |
| Install and Configure Packages: |
| **sudo su** |
| **apt install memcached python-memcache** |
|  |
| Edit **/etc/memcached.conf** to define IP address: |
| **-l 10.0.0.11** |
|  |
| Restart Memcached Service: |
| **service memcached restart** |

## ETCD

|  |
| --- |
| Create **etcd** User and directories: |
| **sudo su** |
| **groupadd --system etcd** |
| **useradd --home-dir "/var/lib/etcd" --system --shell /bin/false -g etcd etcd** |
| **mkdir -p /etc/etcd** |
| **chown etcd:etcd /etc/etcd** |
| **mkdir -p /var/lib/etcd** |
| **chown etcd:etcd /var/lib/etcd** |
|  |
| Download and install **etcd** tarball |
| **ETCD\_VER=v3.2.7** |
| **rm -rf /tmp/etcd && mkdir -p /tmp/etcd** |
| **curl -L https://github.com/coreos/etcd/releases/download/${ETCD\_VER}/etcd-${ETCD\_VER}-linux-amd64.tar.gz -o /tmp/etcd-${ETCD\_VER}-linux-amd64.tar.gz** |
| **tar xzvf /tmp/etcd-${ETCD\_VER}-linux-amd64.tar.gz -C /tmp/etcd --strip-components=1** |
| **cp /tmp/etcd/etcd /usr/bin/etcd** |
| **cp /tmp/etcd/etcdctl /usr/bin/etcdctl** |
|  |
| Create and edit the **/etc/etcd/etcd.conf.yml** file |
| **vim /etc/etcd/etcd.conf.yml** |
| and put following 9 lines in it: |
| **name: controller** |
| **data-dir: /var/lib/etcd** |
| **initial-cluster-state: 'new'** |
| **initial-cluster-token: 'etcd-cluster-01'** |
| **initial-cluster: controller=http://10.0.0.11:2380** |
| **initial-advertise-peer-urls: http://10.0.0.11:2380** |
| **advertise-client-urls: http://10.0.0.11:2379** |
| **listen-peer-urls: http://0.0.0.0:2380** |
| **listen-client-urls: http://10.0.0.11:2379** |
|  |
| Create and edit **/lib/systemd/system/etcd.service** file |
| **vim /lib/systemd/system/etcd.service** |
| and put following 13 lines in it: |
| **[Unit]** |
| **After=network.target** |
| **Description=etcd - highly-available key value store** |
|  |
| **[Service]** |
| **LimitNOFILE=65536** |
| **Restart=on-failure** |
| **Type=notify** |
| **ExecStart=/usr/bin/etcd --config-file /etc/etcd/etcd.conf.yml** |
| **User=etcd** |
|  |
| **[Install]** |
| **WantedBy=multi-user.target** |
|  |
| Enable and start etcd Service: |
| **systemctl enable etcd** |
| **systemctl start etcd** |

## Install Keystone – Identity Management

**Configure SQL Database for Keystone:**

|  |
| --- |
| Run these commands: |
| **sudo su** |
| **Mysql** |
| **CREATE DATABASE keystone;** |
| **GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'openstack';** |
| **EXIT;** |
|  |
| **Install and Configure Packages:** |
| Run these commands: |
| **sudo su** |
| # Install required packages + crudini to edit .conf files |
| **apt install keystone apache2 libapache2-mod-wsgi crudini -y** |
| # Configure Keystone database access, as set above |
| **crudini --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:openstack@controller/keystone** |
| # Set Fernet Token Provider |
| **crudini --set /etc/keystone/keystone.conf token provider fernet** |
| # Populate Identity Service Database |
| **su -s /bin/sh -c "keystone-manage db\_sync" keystone** |
| # Initialize Fernet Repositories |
| **keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone** |
| **keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone** |
| # Bootstrap Identity Service |
| **keystone-manage bootstrap --bootstrap-password openstack --bootstrap-admin-url http://controller:35357/v3/ --bootstrap-internal-url http://controller:5000/v3/ --bootstrap-public-url http://controller:5000/v3/ --bootstrap-region-id RegionOne** |
|  |
| **Configure Apache Server:** |
| Edit **/etc/apache2/apache2.conf** and add following line: |
| **ServerName controller** |
|  |
| Restart the apache2 service |
| **service apache2 restart** |
|  |
| **Configure OpenStack Client Environment Scripts** |
| Create **admin-openrc** Script (in Primary User's Home Directory, for example) |
| Insert following lines: |
| **export OS\_PROJECT\_DOMAIN\_NAME=Default** |
| **export OS\_USER\_DOMAIN\_NAME=Default** |
| **export OS\_PROJECT\_NAME=admin** |
| **export OS\_USERNAME=admin** |
| **export OS\_PASSWORD=openstack** |
| **export OS\_AUTH\_URL=http://controller:35357/v3** |
| **export OS\_IDENTITY\_API\_VERSION=3** |
| **export OS\_IMAGE\_API\_VERSION=2** |
|  |
| **Create demo-openrc Script** |
| Insert following lines: |
| **export OS\_PROJECT\_DOMAIN\_NAME=Default** |
| **export OS\_USER\_DOMAIN\_NAME=Default** |
| **export OS\_PROJECT\_NAME=demo** |
| **export OS\_USERNAME=demo** |
| **export OS\_PASSWORD=openstack** |
| **export OS\_AUTH\_URL=http://controller:5000/v3** |
| **export OS\_IDENTITY\_API\_VERSION=3** |
| **export OS\_IMAGE\_API\_VERSION=2** |
|  |
| **Verify Keystone operation** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack token issue** |
|  |
| **Create Projects, Users and Roles** |
| Run following commands: |
|  |
| **. admin-openrc** |
| # Create a service Project |
| **openstack project create --domain default --description "Service Project" service** |
| # Create a demo Project |
| **openstack project create --domain default --description "Demo Project" demo** |
| # Create a demo User |
| **openstack user create --domain default --password openstack demo** |
| # Create a user Role |
| **openstack role create user** |
| # Add the user role to User demo in Project demo |
| **openstack role add --project demo --user demo user** |
|  |
| **Verify User demo** |
| Run following commands: |
| **. demo-openrc** |
| **openstack token issue** |

## Install Glance – Image Service

|  |
| --- |
| **Configure SQL Database for Glance** |
| Run following commands: |
|  |
| **sudo su** |
| **Mysql** |
| **CREATE DATABASE glance;** |
| **GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'openstack';** |
| **EXIT;** |
|  |
| **Create glance User** |
| **. admin-openrc** |
| **openstack user create --domain default --password openstack glance** |
|  |
| **Add admin role to User glance in Project service** |
| **openstack role add --project service --user glance admin** |
|  |
| **Create glance Service** |
| **openstack service create --name glance --description "OpenStack Image" image** |
|  |
| **Create glance Service Endpoints** |
| **openstack endpoint create --region RegionOne image public http://controller:9292** |
| **openstack endpoint create --region RegionOne image internal http://controller:9292** |
| **openstack endpoint create --region RegionOne image admin http://controller:9292** |
|  |
| **Install and Configure Packages** |
| Run following commands: |
|  |
| **apt update -y** |
| **apt install glance -y** |
|  |
| **Configure /etc/glance/glance-api.conf Parameters** |
| Run following commands: |
| # Configure database access for glance |
| **crudini --set /etc/glance/glance-api.conf database connection mysql+pymysql://glance:openstack@controller/glance** |
| # Configure Identity Service access |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken username glance** |
| **crudini --set /etc/glance/glance-api.conf keystone\_authtoken password openstack** |
| **crudini --set /etc/glance/glance-api.conf paste\_deploy flavor keystone** |
| # Configure Glance to store Images on Local Filesystem |
| **crudini --set /etc/glance/glance-api.conf glance\_store stores "file,http"** |
| **crudini --set /etc/glance/glance-api.conf glance\_store default\_store file** |
| **crudini --set /etc/glance/glance-api.conf glance\_store filesystem\_store\_datadir /var/lib/glance/images/** |
|  |
| **Configure /etc/glance/glance-registry.conf Parameters** |
| Run following commands: |
| # Configure database access for glance |
| **crudini --set /etc/glance/glance-registry.conf database connection mysql+pymysql://glance:openstack@controller/glance** |
| # Configure Identity Service access |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken username glance** |
| **crudini --set /etc/glance/glance-registry.conf keystone\_authtoken password openstack** |
| **crudini --set /etc/glance/glance-registry.conf paste\_deploy flavor keystone** |
|  |
| **Populate the Image Service Database** |
| Run following commands: |
| **su -s /bin/sh -c "glance-manage db\_sync" glance** |
|  |
| **Restart glance Services** |
| **service glance-registry restart** |
| **service glance-api restart** |
|  |
| **Verify Glance Operation** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **wget http://download.cirros-cloud.net/0.3.5/cirros-0.3.5-x86\_64-disk.img** |
| **openstack image create cirros3.5 --file cirros-0.3.5-x86\_64-disk.img --disk-format qcow2 --container-format bare --public** |
| **openstack image list** |
|  |
| **Download Cloud Images:** |
| [**https://docs.openstack.org/image-guide/obtain-images.html**](https://docs.openstack.org/image-guide/obtain-images.html) |

## Install & Configure Nova(Compute Service) Controller

|  |
| --- |
| **Configure SQL Databases for Nova** |
| Run following commands: |
|  |
| **sudo su** |
| **mysql** |
| **CREATE DATABASE nova\_api;** |
| **CREATE DATABASE nova;** |
| **CREATE DATABASE nova\_cell0;** |
| **GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'%' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'%' IDENTIFIED BY 'openstack';** |
|  |
| **Create Compute Service User and add admin role in service Project** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack user create --domain default --password openstack nova** |
| **openstack role add --project service --user nova admin** |
|  |
| **Create Compute Service & Endpoints** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack service create --name nova --description "OpenStack Compute" compute** |
| **openstack endpoint create --region RegionOne compute public http://controller:8774/v2.1** |
| **openstack endpoint create --region RegionOne compute internal http://controller:8774/v2.1** |
| **openstack endpoint create --region RegionOne compute admin http://controller:8774/v2.1** |
|  |
| **Create Placement Service User and add admin role in service Project** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack user create --domain default --password openstack placement** |
| **openstack role add --project service --user placement admin** |
|  |
| **Create Placement Service & Endpoints** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack service create --name placement --description "Placement API" placement** |
| **openstack endpoint create --region RegionOne placement public http://controller:8778** |
| **openstack endpoint create --region RegionOne placement internal http://controller:8778** |
| **openstack endpoint create --region RegionOne placement admin http://controller:8778** |
|  |
| **Install Nova Controller Packages** |
| Run following commands: |
|  |
| **sudo su** |
| **apt install -y nova-api nova-conductor nova-consoleauth nova-novncproxy nova-scheduler nova-placement-api** |
|  |
| **Configure MySQL & RabbitMQ parameters in /etc/nova/nova.conf** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf api\_database connection mysql+pymysql://nova:openstack@controller/nova\_api** |
| **crudini --set /etc/nova/nova.conf database connection mysql+pymysql://nova:openstack@controller/nova** |
| **crudini --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Configure Identity Service access** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf api auth\_strategy keystone** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken username nova** |
| **crudini --set /etc/nova/nova.conf keystone\_authtoken password openstack** |
|  |
| **Configure support for Networking Service** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf DEFAULT my\_ip 10.0.0.11** |
| **crudini --set /etc/nova/nova.conf DEFAULT use \_neutron True** |
| **crudini --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver** |
|  |
| **Configure vnc proxy on Controller Node** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf vnc enabled True** |
| **crudini --set /etc/nova/nova.conf vnc vncserver\_listen 10.0.0.11** |
| **crudini --set /etc/nova/nova.conf vnc vncserver\_proxyclient\_address 10.0.0.11** |
|  |
| **Configure Glance location** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf glance api\_servers http://controller:9292** |
|  |
| **Configure Lock Path for Oslo Concurrency** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp** |
|  |
| **Configure Placement API** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf placement os\_region\_name RegionOne** |
| **crudini --set /etc/nova/nova.conf placement project\_domain\_name Default** |
| **crudini --set /etc/nova/nova.conf placement project\_name service** |
| **crudini --set /etc/nova/nova.conf placement auth\_type password** |
| **crudini --set /etc/nova/nova.conf placement user\_domain\_name Default** |
| **crudini --set /etc/nova/nova.conf placement auth\_url http://controller:35357/v3** |
| **crudini --set /etc/nova/nova.conf placement username placement** |
| **crudini --set /etc/nova/nova.conf placement password openstack** |
|  |
| **Remove log\_dir parameter in DEFAULT section** |
| Run following command: |
|  |
| **crudini --del /etc/nova/nova.conf DEFAULT log\_dir** |
|  |
| **Populate nova\_api Database** |
| Run following commands: |
|  |
| **sudo su** |
| **su -s /bin/sh -c "nova-manage api\_db sync" nova** |
|  |
| **Register cell0 Database** |
| Run following command: |
|  |
| **su -s /bin/sh -c "nova-manage cell\_v2 map\_cell0" nova** |
|  |
| **Create cell1 Cell** |
| Run following command: |
|  |
| **su -s /bin/sh -c "nova-manage cell\_v2 create\_cell --name=cell1 --verbose" nova** |
|  |
| **Populate nova Database** |
| Run following command: |
|  |
| **su -s /bin/sh -c "nova-manage db sync" nova** |
|  |
| **Verify configuration of Cells** |
| Run following command: |
|  |
| **nova-manage cell\_v2 list\_cells** |
|  |
| **Restart Services** |
| Run following commands: |
|  |
| **service nova-api restart** |
| **service nova-consoleauth restart** |
| **service nova-scheduler restart** |
| **service nova-conductor restart** |
| **service nova-novncproxy restart** |

## Install & Configure Nova on Compute Nodes

|  |
| --- |
| **Discover Compute Nodes** |
| Run following command: |
|  |
| **su -s /bin/sh -c "nova-manage cell\_v2 discover\_hosts --verbose" nova** |
|  |
| **Verify Compute Service Installation** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack compute service list** |
| **openstack catalog list** |
| **openstack image list** |
| **nova-status upgrade check** |

## Install Neutron (Network Service) on Controller Node

|  |
| --- |
| **Create Neutron SQL Database** |
| Run following commands: |
|  |
| **sudo su** |
| **mysql** |
| **CREATE DATABASE neutron;** |
| **GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY 'openstack';** |
| **EXIT;** |
|  |
| **Create neutron User and add admin Role in service Project** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack user create --domain default --password openstack neutron** |
| **openstack role add --project service --user neutron admin** |
|  |
| **Create Neutron Service and Endpoints** |
| Run following commands: |
|  |
| **openstack service create --name neutron --description "OpenStack Networking" network** |
| **openstack endpoint create --region RegionOne network public http://controller:9696** |
| **openstack endpoint create --region RegionOne network internal http://controller:9696** |
| **openstack endpoint create --region RegionOne network admin http://controller:9696** |
|  |
| **Install Neutron Packages** |
| Run following commands: |
|  |
| **sudo su** |
| **apt install -y neutron-server neutron-plugin-ml2 neutron-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent neutron-metadata-agent** |
|  |
| **Configure SQL Database and RabbitMQ access for Neutron** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf database connection mysql+pymysql://neutron:openstack@controller/neutron** |
| **crudini --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Enable the Modular Layer 2 (ML2) plug-in, router service, and overlapping IP addresses** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf DEFAULT core\_plugin ml2** |
| **crudini --set /etc/neutron/neutron.conf DEFAULT service\_plugins router** |
| **crudini --set /etc/neutron/neutron.conf DEFAULT allow\_overlapping\_ips true** |
|  |
| **Configure Identity Service access** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf api auth\_strategy keystone** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken username neutron** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken password openstack** |
|  |
| **Configure Networking to notify Compute of network topology changes** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf DEFAULT notify\_nova\_on\_port\_status\_changes true** |
| **crudini --set /etc/neutron/neutron.conf DEFAULT notify\_nova\_on\_port\_data\_changes true** |
|  |
| **Configure Nova access** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf nova auth\_url http://controller:35357** |
| **crudini --set /etc/neutron/neutron.conf nova auth\_type password** |
| **crudini --set /etc/neutron/neutron.conf nova project\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf nova user\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf nova region\_name RegionOne** |
| **crudini --set /etc/neutron/neutron.conf nova project\_name service** |
| **crudini --set /etc/neutron/neutron.conf nova username nova** |
| **crudini --set /etc/neutron/neutron.conf nova password openstack** |
|  |
| **Configure ML2 Plugin** |
| Run following commands: |
|  |
| # Enable flat, VLAN and VXLAN Networks |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 type\_drivers flat,vlan,vxlan** |
| # Enable VXLAN Self-service Networks |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 tenant\_network\_types vxlan** |
| # Enable Linux Bridge and L2Population mechanisms |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers linuxbridge,l2population** |
| # Enable Port Security Extenstion Driver |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 extension\_drivers port\_security** |
| # Configure **provider** Virtual Network as flat Network |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_flat flat\_networks provider** |
| # Configure VXLAN Network Identifier Range for Self-service Networks |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_vxlan vni\_ranges 1:1000** |
| # Enable ipset to increase efficiency of Security Group Rules |
| **crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup enable\_ipset true** |
|  |
| **Configure the Linux Bridge Agent** |
| Run following commands: |
|  |
| # Configure **provider** Virtual Network mapping to Physical Interface |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings provider:eth1** |
| # Enable VXLAN for Self-service Networks, configure IP address of the Management Interface handling VXLAN traffic |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip 10.0.0.11** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true** |
| # Enable security groups and configure the Linux bridge iptables firewall driver |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver** |
|  |
| **Configure the Layer-3 Agent** |
| Run following command: |
|  |
| **crudini --set /etc/neutron/l3\_agent.ini DEFAULT interface\_driver linuxbridge** |
|  |
| **Configure the DHCP Agent** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT interface\_driver linuxbridge** |
| **crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT dhcp\_driver neutron.agent.linux.dhcp.Dnsmasq** |
| **crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT enable\_isolated\_metadata true** |
|  |
| **Configure Metadata Agent** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/metadata\_agent.ini DEFAULT nova\_metadata\_host controller** |
| **crudini --set /etc/neutron/metadata\_agent.ini DEFAULT metadata\_proxy\_shared\_secret openstack** |
|  |
| **Configure Compute Service to use Neutron** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf neutron url http://controller:9696** |
| **crudini --set /etc/nova/nova.conf neutron auth\_url http://controller:35357** |
| **crudini --set /etc/nova/nova.conf neutron auth\_type password** |
| **crudini --set /etc/nova/nova.conf neutron project\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf neutron user\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf neutron region\_name RegionOne** |
| **crudini --set /etc/nova/nova.conf neutron project\_name service** |
| **crudini --set /etc/nova/nova.conf neutron username neutron** |
| **crudini --set /etc/nova/nova.conf neutron password openstack** |
| **crudini --set /etc/nova/nova.conf neutron service\_metadata\_proxy true** |
| **crudini --set /etc/nova/nova.conf neutron metadata\_proxy\_shared\_secret openstack** |
|  |
| **Populate Neutron Database** |
| Run following Command: |
|  |
| **su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade head" neutron** |
|  |
| **Restart the Compute API Service** |
| Run following command: |
|  |
| **service nova-api restart** |
|  |
| **Restart Networking Services** |
| Run following Commands: |
|  |
| **service neutron-server restart** |
| **service neutron-linuxbridge-agent restart** |
| **service neutron-dhcp-agent restart** |
| **service neutron-metadata-agent restart** |
| **service neutron-l3-agent restart** |

**Install Neutron on Compute Nodes**

|  |
| --- |
| **Verify Installation** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack network agent list** |

## Install Cinder Block Storage Service on Controller Node

|  |
| --- |
| **Create Cinder Database** |
| Run following commands: |
|  |
| **sudo su** |
| **Mysql** |
| **CREATE DATABASE cinder;** |
| **GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY 'openstack';** |
| **GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY 'openstack';** |
| **EXIT;** |
|  |
| **Create cinder User and Add admin Role in service Project** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack user create --domain default --password openstack cinder** |
| **openstack role add --project service --user cinder admin** |
|  |
| **Create cinderv2 and cinderv3 Services and their Endpoints** |
| Run following commands: |
|  |
| **openstack service create --name cinderv2 --description "OpenStack Block Storage" volumev2** |
| **openstack service create --name cinderv3 --description "OpenStack Block Storage" volumev3** |
| **openstack endpoint create --region RegionOne volumev2 public http://controller:8776/v2/%\(project\_id\)s** |
| **openstack endpoint create --region RegionOne volumev2 internal http://controller:8776/v2/%\(project\_id\)s** |
| **openstack endpoint create --region RegionOne volumev2 admin http://controller:8776/v2/%\(project\_id\)s** |
| **openstack endpoint create --region RegionOne volumev3 public http://controller:8776/v3/%\(project\_id\)s** |
| **openstack endpoint create --region RegionOne volumev3 internal http://controller:8776/v3/%\(project\_id\)s** |
| **openstack endpoint create --region RegionOne volumev3 admin http://controller:8776/v3/%\(project\_id\)s** |
|  |
| **Install Packages** |
| Run following command: |
|  |
| **apt install -y cinder-api cinder-scheduler** |
|  |
| **Configure Database and RabbitMQ Access** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder:openstack@controller/cinder** |
| **crudini --set /etc/cinder/cinder.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Configure Identity Service Access** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT auth\_strategy keystone** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken username cinder** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken password openstack** |
|  |
| **Configure my\_ip Parameter and Lock Path** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT my\_ip 10.0.0.11** |
| **crudini --set /etc/cinder/cinder.conf oslo\_concurrency lock\_path /var/lib/cinder/tmp** |
|  |
| **Populate Block Storage Database** |
| Run following command: |
|  |
| **su -s /bin/sh -c "cinder-manage db sync" cinder** |
|  |
| **Configure Compute Service to use Cinder** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf cinder os\_region\_name RegionOne** |
|  |
| **Restart Services** |
| Run following commands: |
|  |
| **service nova-api restart** |
| **service cinder-scheduler restart** |
| **service apache2 restart** |
|  |
| **Verify Cinder Operation** |
| Run following commands: |
|  |
| **. admin-openrc** |
| **openstack volume service list** |

## Install Horizon Dashboard

|  |
| --- |
| **Install Packages** |
| Run following commands: |
|  |
| **sudo su** |
| **apt install -y openstack-dashboard** |
|  |
| **Edit /etc/openstack-dashboard/local\_settings.py to include following settings:** |
|  |
| **OPENSTACK\_HOST = "controller"** |
|  |
| **SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'** |
|  |
| **CACHES = {** |
| **'default': {** |
| **'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',** |
| **'LOCATION': 'controller:11211',** |
| **}** |
| **}** |
|  |
| **OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST** |
|  |
| **OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True** |
|  |
| **OPENSTACK\_API\_VERSIONS = {** |
| **"identity": 3,** |
| **"image": 2,** |
| **"volume": 2,** |
| **}** |
|  |
| **OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "Default"** |
|  |
| **OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"** |
|  |
| **Edit /etc/apache2/conf-available/openstack-dashboard.conf to include following line:** |
|  |
| **WSGIApplicationGroup %{GLOBAL}** |
|  |
| **Reload Web Server Configuration** |
| Run following command: |
|  |
| **service apache2 reload** |
|  |
| **Verify Horizon Operation by pointing Web Browser to** |
|  |
| [**http://10.0.0.11/horizon**](http://10.0.0.11/horizon) |

# **Compute Installation**

**Compute Service Architecture**

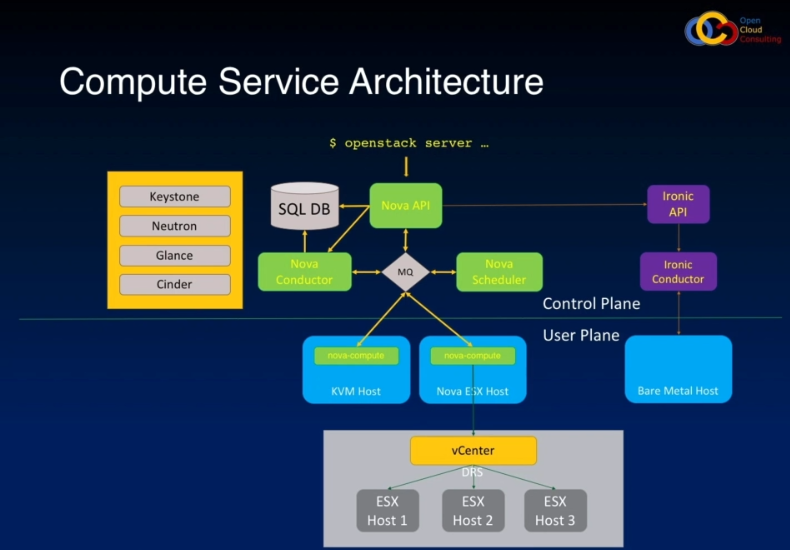


Image Source:

## Set-up - VM

**HW Config**

|  |  |  |
| --- | --- | --- |
| **Virtual** | Recommended | Actual |
| VCPU (cores) | 1-2+ | 1 |
| RAM | 4+ GB | 4 |
| Primary Disk | 10+ GB | 10 |

**VirtualBox Host-Only Network Ethernet Adapter #2**

Configure Adapter Manualy IPv4 Addr 10.0.0.1 IPv4 Net Mask 255.255.255.0 DHCP Disabled

**NAT Network ProviderNetwork1**

CIDR 203.0.113.0/24 DHCP Disabled

**NAT Network NatNetwork1**

CIDR 10.10.10.0/24 DHCP Enabled

**Network Interfaces**



Promiscuous Mode: Allow All

**Operating System**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Ubuntu Server 16.04 LTS |  |  |
| **Link** | https://www.ubuntu.com/download/server | |  |

**Operating System Installation Options**



## Configure Security, Networking, Install Linux Utilities

|  |
| --- |
| **Configure 'sudo' access for** |
| **sudo su** |
| **visudo** |
| add following line at the bottom of the file: |
| openstack **ALL=(ALL) NOPASSWD:ALL** |
| save, exit and run sudo su again to test |
|  |
| **Edit /etc/hosts** |
| Remove 127.0.1.1 compute1, if present |
| Make sure following lines are present: |
| **10.0.0.11 controller** |
| **10.0.0.31 compute1** |
| **10.0.0.41 block1** |
|  |
| Edit /etc/default/grub to include: |
| **GRUB\_CMDLINE\_LINUX="net.ifnames=0 biosdevname=0"** |
| Run command: |
| **update-grub** |
| **reboot** |
|  |
| **Enable Network Interfaces** |
| **sudo su** |
| Edit /etc/network/interfaces |
| Make sure following Interfaces definitions are present: |
|  |
| **auto eth0** |
| **iface eth0 inet static** |
| **address 10.0.0.31** |
| **netmask 255.255.255.0** |
| **dns-nameservers 8.8.8.8** |
| **auto eth1** |
| **iface eth1 inet manual** |
| **up ip link set dev eth1 up** |
| **down ip link set dev eth1 down** |
| **auto eth2** |
| **iface eth2 inet dhcp** |
|  |
| Reboot the system |
| Run 'ifconfig' as superuser to verify settings. |
| Verify connectivity to other hosts, once configured |
| **ping -c 3 openstack.org** |
| **ping -c 3 controller** |
| **ping -c 3 block1** |
|  |
| **Install basic Linux Utilities** |
| Run following commands: |
| **sudo su** |
| **apt update** |
| **apt install vim glances curl** |
| **apt upgrade -y** |

## Install & Configure Network Time Protocol

|  |
| --- |
| Install and Configure Components |
|  |
| **sudo su** |
| **apt install chrony** |
|  |
| Edit **/etc/chrony/chrony.conf**: |
| set **server** to **controller** |
| **server controller iburst** |
| comment out **pool 2.debian.pool.ntp.org offline iburst** line. |
| save and quit |
| Restart **chrony** service: |
| **service chrony restart** |
|  |
| Verify: |
| **chronyc sources** |

## Install Basic OpenStack Packages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sudo su** |  |  |  |  |
| **apt install software-properties-common** | | | |  |
| **add-apt-repository cloud-archive:pike** | | | |  |
| **apt update && apt dist-upgrade** | | |  |  |
| **reboot** |  |  |  |  |
| **apt install python-openstackclient** | | | |  |

## Install Compute Service on Compute Node

|  |
| --- |
| **Install Nova Compute Package** |
| Run following commands: |
|  |
| **sudo su** |
| **apt update** |
| **apt install -y nova-compute crudini** |
|  |
| **Configure RabbitMQ access** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Configure Identity Service access** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf api auth\_strategy keystone** |
| **crudini --set /etc/nova/nova.conf keystone\_auth auth\_uri http://controller:5000** |
| **crudini --set /etc/nova/nova.conf keystone\_auth auth\_url http://controller:35357** |
| **crudini --set /etc/nova/nova.conf keystone\_auth memcached\_servers controller:11211** |
| **crudini --set /etc/nova/nova.conf keystone\_auth auth\_type password** |
| **crudini --set /etc/nova/nova.conf keystone\_auth project\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf keystone\_auth user\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf keystone\_auth project\_name service** |
| **crudini --set /etc/nova/nova.conf keystone\_auth username nova** |
| **crudini --set /etc/nova/nova.conf keystone\_auth password openstack** |
|  |
| **Configure support for Networking Service** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf DEFAULT my\_ip 10.0.0.31** |
| **crudini --set /etc/nova/nova.conf DEFAULT use\_neutron True** |
| **crudini --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver** |
|  |
| **Configure vnc Remote Console access on Compute Node** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf vnc enabled True** |
| **crudini --set /etc/nova/nova.conf vnc vncserver\_listen 0.0.0.0** |
| **crudini --set /etc/nova/nova.conf vnc vncserver\_proxyclient\_address 10.0.0.31** |
| **crudini --set /etc/nova/nova.conf vnc novncproxy\_base\_url http://10.0.0.11:6080/vnc\_auto.html** |
|  |
| **Configure Glance location** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf glance api\_servers http://controller:9292** |
|  |
| **Configure Lock Path for Oslo Concurrency** |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp** |
|  |
| **Configure Placement API** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf placement os\_region\_name RegionOne** |
| **crudini --set /etc/nova/nova.conf placement project\_domain\_name Default** |
| **crudini --set /etc/nova/nova.conf placement project\_name service** |
| **crudini --set /etc/nova/nova.conf placement auth\_type password** |
| **crudini --set /etc/nova/nova.conf placement user\_domain\_name Default** |
| **crudini --set /etc/nova/nova.conf placement auth\_url** [**http://controller:35357/v3**](http://controller:35357/v3) |
| **crudini --set /etc/nova/nova.conf placement username placement** |
| **crudini --set /etc/nova/nova.conf placement password openstack** |
|  |
| **Remove log\_dir parameter in DEFAULT section** |
| Run following command: |
|  |
| **crudini --del /etc/nova/nova.conf DEFAULT log\_dir** |

**Set-up #1 - Virtual Machines: use QEMU Emulator**

|  |
| --- |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova-compute.conf libvirt virt\_type qemu** |

**Set-up #2 - Bare Metal Hosts: use KVM**

|  |
| --- |
| **Verify Compute Host Capabilities** |
| Run following commands: |
|  |
| **sudo su** |
| **kvm-ok** |
| **uname -m** |
|  |
| **Install KVM & Utilities** |
| Run following command: |
|  |
| **apt-get install -y qemu-kvm libvirt-bin bridge-utils** |
|  |
| **Verify KVM Installation** |
| Run following command: |
|  |
| **virsh list –all** |
|  |
| Run following command: |
|  |
| **crudini --set /etc/nova/nova-compute.conf libvirt virt\_type kvm** |

|  |
| --- |
| **Restart Nova Compute service** |
| Run following command: |
|  |
| **service nova-compute restart** |

## Install Neutron on Compute Node

|  |
| --- |
| **Install Packages** |
| Run following commands: |
|  |
| **sudo su** |
| **apt update** |
| **apt install -y neutron-linuxbridge-agent** |
|  |
| **Configure RabbitMQ access** |
| Run following command: |
|  |
| **crudini --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Configure Indetity Service Accesss** |
| Run following commands: |
|  |
| **crudini --set /etc/neutron/neutron.conf DEFAULT auth\_strategy keystone** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_uri http://controller:5000** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken username neutron** |
| **crudini --set /etc/neutron/neutron.conf keystone\_authtoken password openstack** |
|  |
| **Configure the Linux Bridge Agent** |
| Run following commands: |
|  |
| # Configure **provider** Virtual Network mapping to Physical Interface |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings provider:eth1** |
| # Enable VXLAN for Self-service Networks, configure IP address of the Management Interface handling VXLAN traffic |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip 10.0.0.31** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true** |
| # Enable security groups and configure the Linux bridge iptables firewall driver |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true** |
| **crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver** |
|  |
| **Configure Compute Service to use Neutron** |
| Run following commands: |
|  |
| **crudini --set /etc/nova/nova.conf neutron url http://controller:9696** |
| **crudini --set /etc/nova/nova.conf neutron auth\_url http://controller:35357** |
| **crudini --set /etc/nova/nova.conf neutron auth\_type password** |
| **crudini --set /etc/nova/nova.conf neutron project\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf neutron user\_domain\_name default** |
| **crudini --set /etc/nova/nova.conf neutron region\_name RegionOne** |
| **crudini --set /etc/nova/nova.conf neutron project\_name service** |
| **crudini --set /etc/nova/nova.conf neutron username neutron** |
| **crudini --set /etc/nova/nova.conf neutron password openstack** |
|  |
| **Restart Services** |
| Run following commands: |
|  |
| **service nova-compute restart** |
| **service neutron-linuxbridge-agent restart** |
|  |

# **Block Installation**

## Set-up VM

**HW Config**

|  |  |  |
| --- | --- | --- |
| **Virtual** | Recommended | Actual |
| VCPU (cores) | 1-2+ | 1 |
| RAM | 4+ GB | 4 |
| Primary Disk | 10+ GB | 20 |

**VirtualBox Host-Only Network Ethernet Adapter #2**

Configure Adapter Manualy IPv4 Addr 10.0.0.1

IPv4 Net Mask 255.255.255.0

DHCP Disabled

**NAT Network ProviderNetwork1**

CIDR 203.0.113.0/24 DHCP Disabled

**NAT Network NatNetwork1**

CIDR 10.10.10.0/24 DHCP Enabled

**Network Interfaces**



Promiscuous Mode: Allow All

**Operating System**

|  |  |  |
| --- | --- | --- |
| **Name** | Ubuntu Server 16.04 LTS |  |
| **Link** | https://www.ubuntu.com/download/server | |

**Operating System Installation Options**



## Configure Security, Networking, Install Linux Utilities

|  |
| --- |
| **Configure 'sudo' access for** |
| **sudo su** |
| **visudo** |
| add following line at the bottom of the file: |
| openstack **ALL=(ALL) NOPASSWD:ALL** |
| save, exit and run sudo su again to test |
|  |
| **Edit /etc/hosts** |
| Remove 127.0.1.1 block1, if present |
| Make sure following lines are present: |
| **10.0.0.11 controller** |
| **10.0.0.31 compute1** |
| **10.0.0.41 block1** |
|  |
| Edit /etc/default/grub to include: |
| **GRUB\_CMDLINE\_LINUX="net.ifnames=0 biosdevname=0"** |
| Run command: |
| **update-grub** |
| **reboot** |
|  |
| **Enable Network Interfaces** |
| **sudo su** |
| Edit /etc/network/interfaces |
| Make sure following Interfaces definitions are present: |
|  |
| **auto eth0** |
| **iface eth0 inet static** |
| **address 10.0.0.41** |
| **netmask 255.255.255.0** |
| **dns-nameservers 8.8.8.8** |
| **auto eth2** |
| **iface eth2 inet dhcp** |
|  |
| Reboot the system |
| Run 'ifconfig' as superuser to verify settings. |
| Verify connectivity to other hosts, once configured |
| **ping -c 3 openstack.org** |
| **ping -c 3 controller** |
| **ping -c 3 compute1** |

|  |
| --- |
| **Install basic Linux Utilities** |
| Run following commands: |
| **sudo su** |
| **apt update** |
| **apt install vim glances curl** |
| **apt upgrade -y** |
| **reboot** |

## Install & Configure Network Time Protocols

|  |
| --- |
| Install and Configure Components |
|  |
| **sudo su** |
| **apt install chrony** |
|  |
| Edit **/etc/chrony/chrony.conf**: |
| set **server** to **controller** |
| **server controller iburst** |
| comment out **pool 2.debian.pool.ntp.org offline iburst** line. |
| save and quit |
| Restart **chrony** service: |
| **service chrony restart** |
|  |
| Verify: |
| **chronyc sources** |

## Install Basic OpenStack Packages

|  |
| --- |
| **sudo su** |
| **apt install software-properties-common** |
| **add-apt-repository cloud-archive:pike** |
| **apt update && apt dist-upgrade** |
| **Reboot** |
| **apt install python-openstackclient** |

## Install Block Storage Service on Storage Node

|  |
| --- |
| **Install Supporting Packages** |
| Run following commands: |
|  |
| **sudo su** |
| **apt update** |
| **apt install -y lvm2 thin-provisioning-tools crudini** |
|  |
| **Verify sdb Disk** |
| Run following command: |
|  |
| **fdisk -l** |
|  |
| **Create LVM Physical Volume /dev/sdb** |
| Run following command: |
|  |
| **pvcreate /dev/sdb** |
|  |
| **Create LVM Volume Group "cinder-volumes"** |
| Run following command: |
|  |
| **vgcreate cinder-volumes /dev/sdb** |
|  |
| **Edit LVM Configuration File /etc/lvm/lvm.conf to include following line in devices section** |
|  |
| **filter = [ "a/sda/", "a/sdb/", "r/.\*/"]** |
|  |
| **Install Cinder Packages** |
| Run following command: |
|  |
| **apt install -y cinder-volume** |
|  |
| **Configure Database and RabbitMQ Access** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder:openstack@controller/cinder** |
| **crudini --set /etc/cinder/cinder.conf DEFAULT transport\_url rabbit://openstack:openstack@controller** |
|  |
| **Configure Identity Service Access** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT auth\_strategy keystone** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_uri = http://controller:5000** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_url http://controller:35357** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken memcached\_servers controller:11211** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_type password** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_domain\_name default** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken user\_domain\_name default** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_name service** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken username cinder** |
| **crudini --set /etc/cinder/cinder.conf keystone\_authtoken password openstack** |
|  |
| **Configure my\_ip Parameter** |
| Run following command: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT my\_ip 10.0.0.41** |
|  |
| **Configure LVM Backend** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf lvm volume\_driver cinder.volume.drivers.lvm.LVMVolumeDriver** |
| **crudini --set /etc/cinder/cinder.conf lvm volume\_group cinder-volumes** |
| **crudini --set /etc/cinder/cinder.conf lvm iscsi\_protocol iscsi** |
| **crudini --set /etc/cinder/cinder.conf lvm iscsi\_helper tgtadm** |
|  |
| **Enable LVM Backend** |
| Run following command: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT enabled\_backends lvm** |
|  |
| **Configure Location of Image Service and Lock Path** |
| Run following commands: |
|  |
| **crudini --set /etc/cinder/cinder.conf DEFAULT glance\_api\_servers http://controller:9292** |
| **crudini --set /etc/cinder/cinder.conf oslo\_concurrency lock\_path /var/lib/cinder/tmp** |
|  |
| Restart Services |
| Run following commands: |
|  |
| **service tgt restart** |
| **service cinder-volume restart** |