OpenStack Installation on Ubuntu

What is OpenStack?

A free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS), whereby virtual servers and other resources are made available to customers.

Consists of several key services that are separately installed and work together depending on your cloud needs.

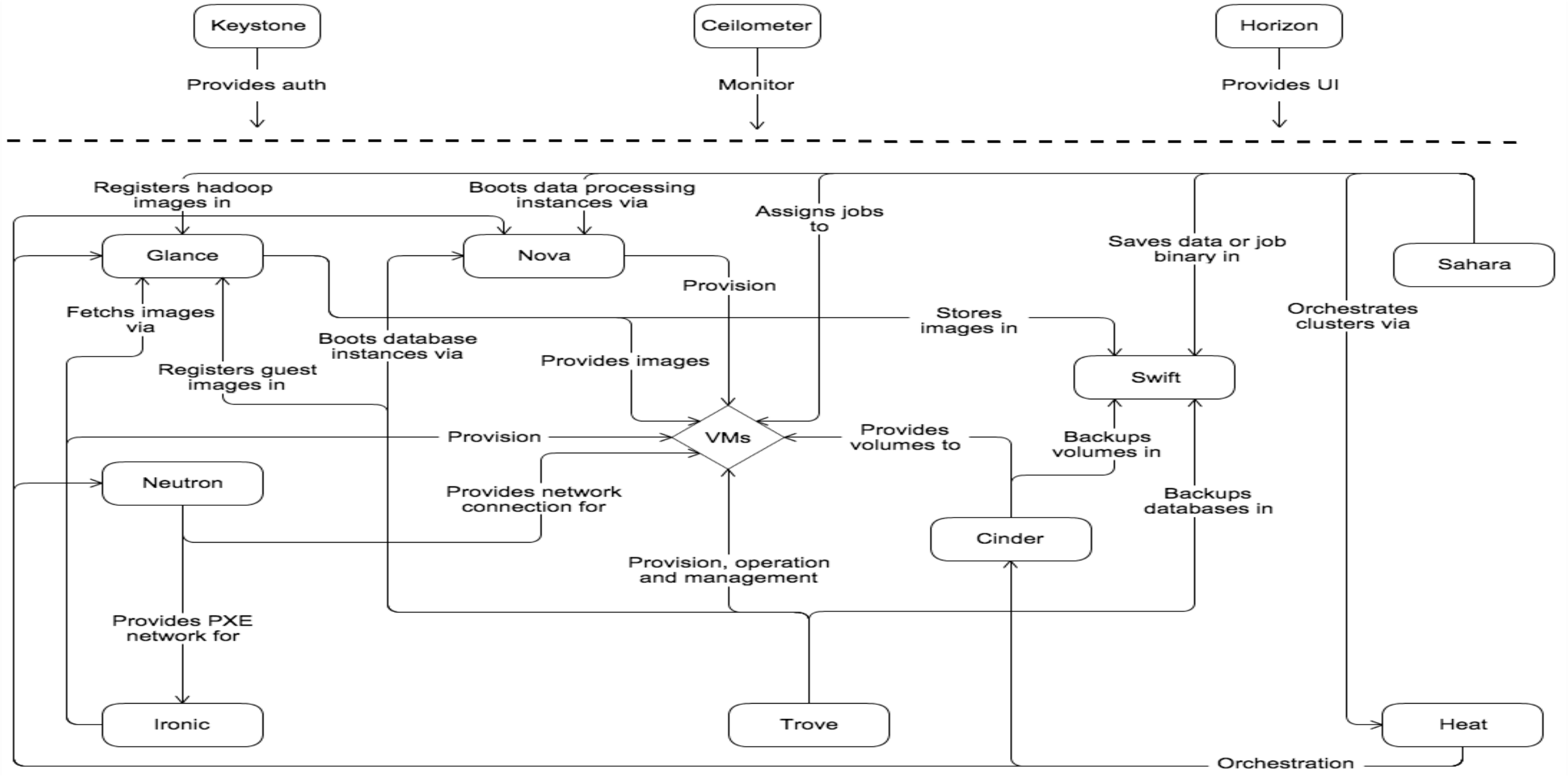
Services include – Compute, Identity, Networking, Image, Block Storage, Object Storage, Telemetry, Orchestration and Database

Some More Info on OpenStack

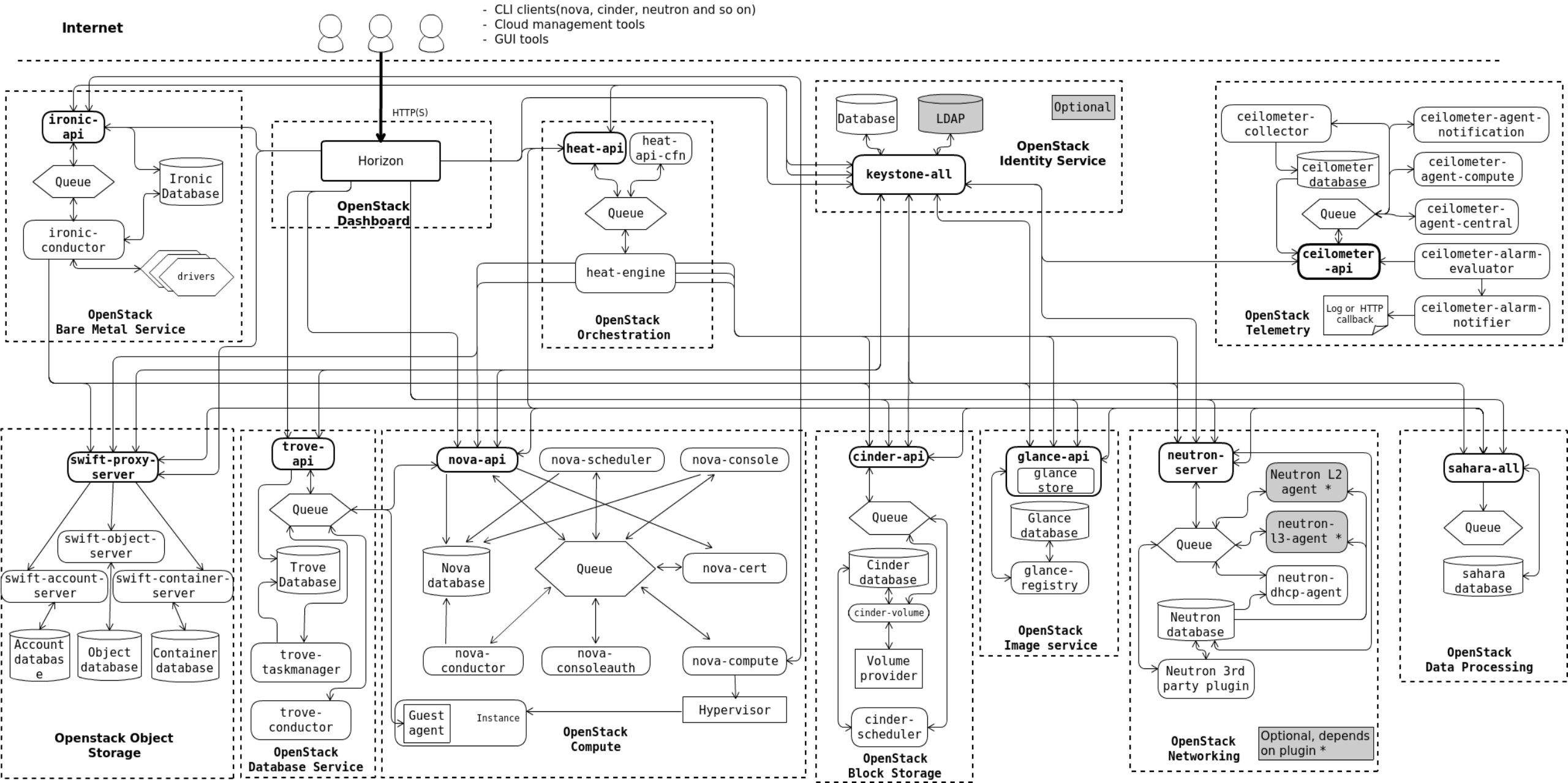
The OpenStack project is an open source cloud computing platform for all types of clouds, which aims to be simple to implement, massively scalable, and feature rich.

Developers and cloud computing technologists from around the world create the OpenStack project.

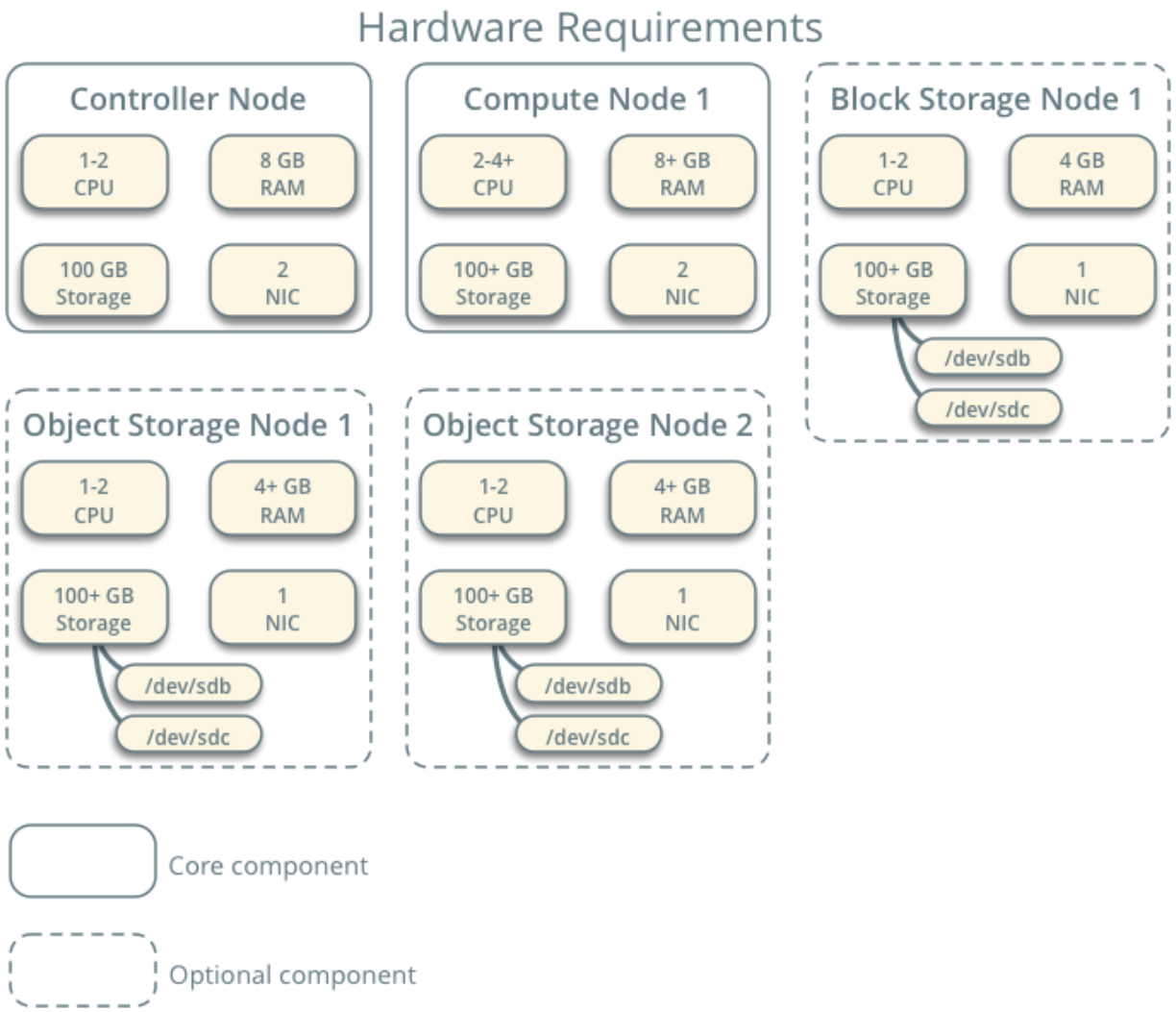
Conceptual Architecture



Logical Architecture



Example Architecture



Example Architecture Components

1. Controller: Runs the Identity, Image, Compute Management, Network Management, Networking Agent and Dashboard services. Also includes supporting services such as an SQL database, message queue and NTP.
2. Compute: Runs the hypervisor portion of Compute that operates instances.
3. Block Storage: Contains the disks that the Block Storage and Shared File System services provision for instances.
4. Object Storage: Contains the disks that the Object Storage service uses for storing accounts, containers and objects.

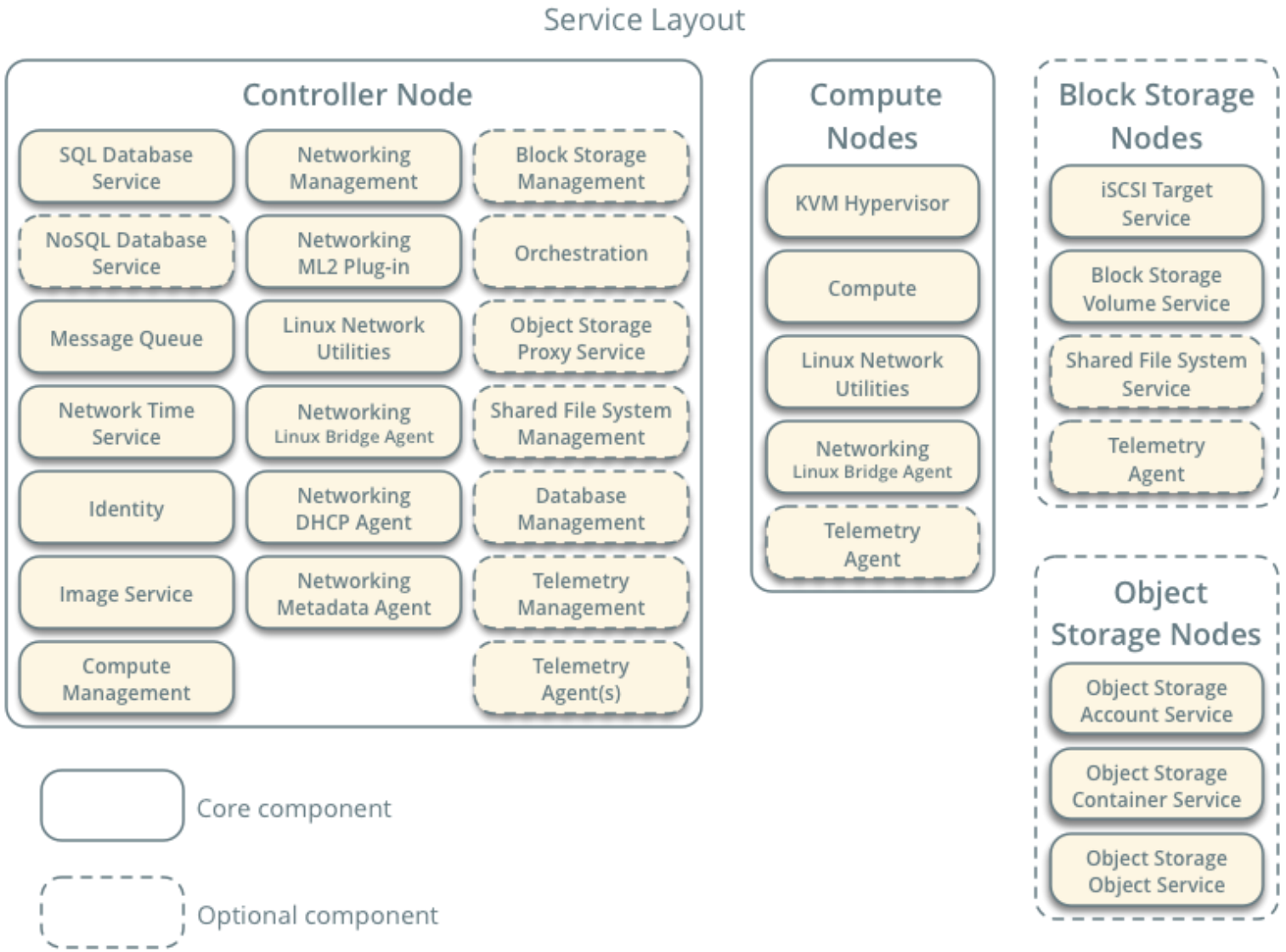
Networking Option 1: Provider Networks

Deploys the OpenStack Networking service in the simplest way possible with primarily layer-2 (bridging/switching) services and VLAN segmentation of networks.

Bridges virtual networks to physical networks and relies on physical network infrastructure for layer-3 (routing) services.

OpenStack user needs to have more information about the underlying network infrastructure to create a virtual network to exactly math the infrastructure.

Networking Option 1: Provider Networks



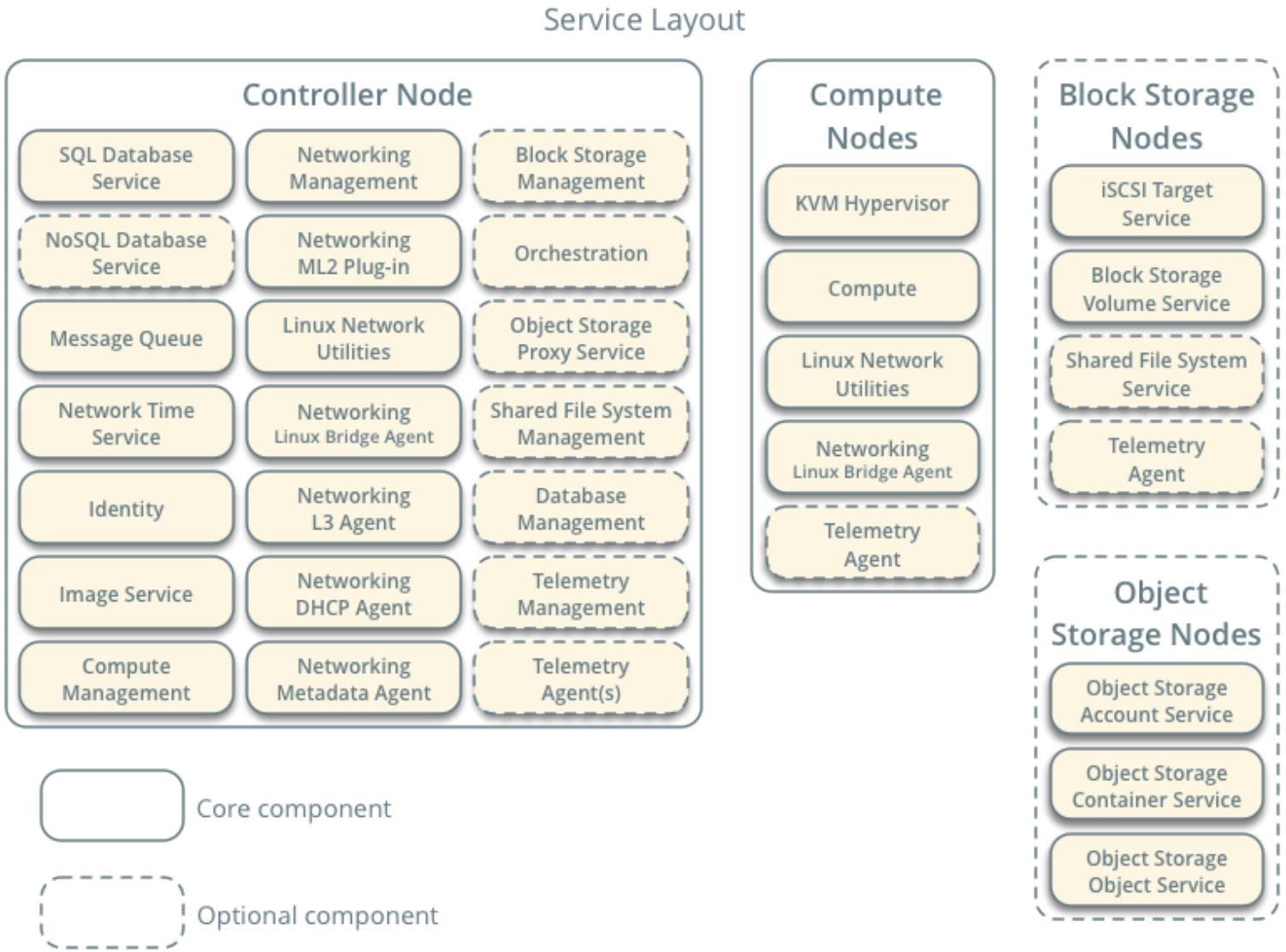
Networking Option 2: Self-Service Networks

Augments option 1 with layer-3 (routing) services that enable self-service networks using overlay segmentation methods such as VXLAN.

Routes virtual networks to physical networks using NAT

OpenStack user can create virtual networks without the knowledge of underlying infrastructure on the data network.

Networking Option 1: Provider Networks



Minimum Requirements for Proof-Of Concept

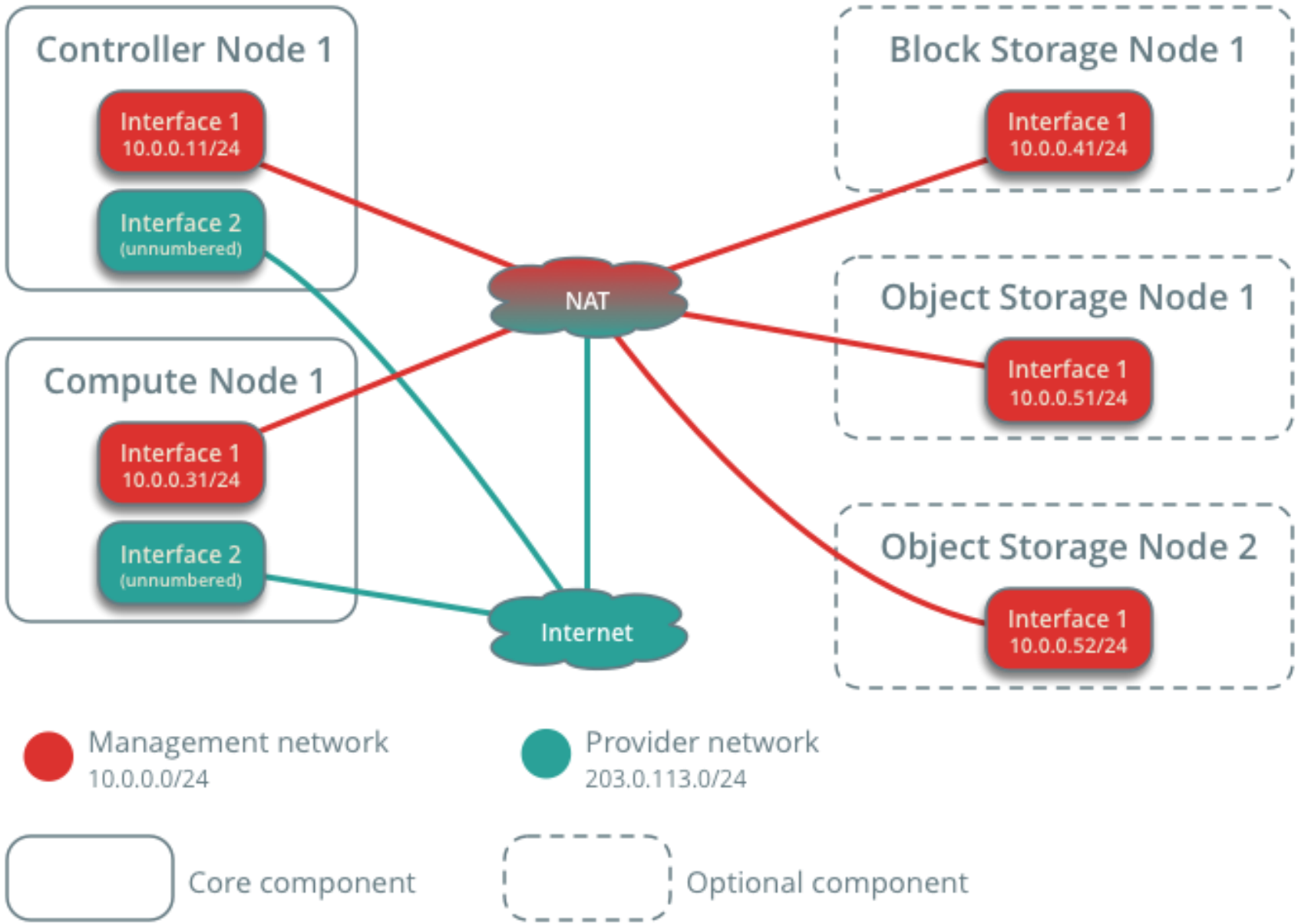
Controller Node

- ✓1 Processor
- ✓4 GB Memory (RAM)
- ✓5 GB Storage

Compute Node

- ✓1 Processor
- ✓2 GB Memory (RAM)
- ✓10 GB Storage

Network Layout



Configuring Controller Node

Configure Network Interfaces

1. Configure the first interface as the management interface:

   IP Address: 10.0.0.11

   Network Mask: 255.255.255.0 (/24)

   Default Gateway: 10.0.0.1

2. Configure the second interface as the provider interface by editing the /etc/network/interfaces file:

   ```
   # The provider network interface
   auto INTERFACE_NAME
   iface INTERFACE_NAME inet manual
   up ip link set dev $IFACE up
   down ip link set dev $IFACE down
   ```

3. Reboot the system to activate the changes.

Configure Name Resolution

1. Set the hostname of the node to controller.

2. Edit the /etc/hosts file to contain the following:

   ```
   # controller
   10.0.0.11    controller
   #compute1
   10.0.0.31    compute1
   #block1
   10.0.0.41    block1
   #object1
   10.0.0.51    object1
   #object2
   10.0.0.52    object2
   ```

Configuring Compute Node

Configure Network Interfaces

1. Configure the first interface as the management interface:

   IP Address: 10.0.0.31

   Network Mask: 255.255.255.0 (/24)

   Default Gateway: 10.0.0.1

2. Configure the second interface as the provider interface by editing the /etc/network/interfaces file:

   ```
   # The provider network interface
   auto INTERFACE_NAME
   iface INTERFACE_NAME inet manual
   up ip link set dev $IFACE up
   down ip link set dev $IFACE down
   ```

3. Reboot the system to activate the changes.

Configure Name Resolution

1. Set the hostname of the node to compute1.

2. Edit the /etc/hosts file to contain the following:

   ```
   # controller
   10.0.0.11    controller
   #compute1
   10.0.0.31    compute1
   #block1
   10.0.0.41    block1
   #object1
   10.0.0.51    object1
   #object2
   10.0.0.52    object2
   ```

Configuring Block Storage Node

1. **Configure a storage node.**

2. Configure Network Interfaces: Configure the management interface:

IP Address: 10.0.0.41

Network Mask: 255.255.255.0 (/24)

Default Gateway: 10.0.0.1

3. Configure Name Resolution:

1. Set the hostname of the node to block1.
2. Edit the /etc/hosts file to contain the following:

```
# controller
10.0.0.11    controller
#compute1
10.0.0.31    compute1
#block1
10.0.0.41    block1
#object1
10.0.0.51    object1
#object2
10.0.0.52    object2
```

4. Reboot the system to activate the changes.

Verifying Connectivity

1. From controller node, test access to the Internet:

   ping –c 4 openstack.org

2. From controller node, test access to the management interface on the compute node:

   ping –c 4 compute1

3. From the compute node, test access to the Internet:

   ping –c 4 openstack.org

4. From the compute node, test access to the management interface on the controller:

   ping –c 4 controller

Network Time Protocol (NTP)

You should install Chrony, an implementation of NTP, to properly synchronize services among nodes. We recommend that you configure the controller node to reference more accurate (lower stratum) servers and other nodes to reference the controller node. Steps for configuring NTP using Chrony are:

1. Configure Chrony for Controller Node
2. Configure Chrony for other nodes.
3. Verify operation.

Configuring NTP for Controller Node

1. Install the packages: *sudo apt install chrony*

2. Edit the /etc/chrony/chrony.conf file and add, change or remove these keys as necessary: *server NTP_SERVER iburst*

3. To allow other nodes to connect to the chrony daemon on the controller node, add this key to the /etc/chrony/chrony.conf file: *allow 10.0.0.0/24*

4. Restart the NTP service: *sudo service chrony restart*

Configuring NTP for other nodes

1. Install the packages: *sudo apt install chrony*

2. Edit the /etc/chrony/chrony.conf file and add, change or remove these keys as necessary: *server controller iburst*

3. Comment out the *pool 2.debian.pool.ntp.org offline iburst* line.

4. Restart the NTP service: *sudo service chrony restart*

Verifying NTP Synchronization

1. **Run *chronyc sources* on the controller node:** Contents in the Name/IP address column should indicate the hostname or IP Address of one or more NTP servers. Contents in the MS column should indicate * for the server to which the NTP service is currently synchronized.

2. **Run the same command on all the other nodes:** Contents in the Name/IP Address column should indicate the hostname of the controller node.

Installing OpenStack Packages

1. **Enable the OpenStack Repository by running the following the following commands:**

   sudo apt install software-properties-common

   sudo add-apt-repository cloud-archive:ocata

2. **Finalize the installation by running the following commands:**

   sudo apt update && apt dist-upgrade

   sudo apt install python-openstackclient

Configuring SQL database

Install and configure the database components on the controller node:

sudo apt-install mariadb-server python-pymysql

Create and edit the */etc/mysql/mariadb.conf.d/99-openstack.cnf* file:

```
[mysqld]
bind-address = 10.0.0.11 #ControllerNode
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Configuring SQL database

Finalize Installation:

sudo service mysql restart

Secure the database service by running the *mysql_secure_installation* script. In particular, choose a suitable password for the database *root* account.

Configuring the Message Queue

OpenStack uses a message queue to coordinate operations and status information among services. The message queue typically runs on the controller node. OpenStack supports several message queue services including RabbitMQ, Qpid and ZeroMQ.

1. *sudo apt install rabbitmq-server*
2. *rabbitmqctl add_user openstack RABBIT_PASS*
3. Permit configuration, write and read access fot the *openstack* user.

   rabbitmqctl set_permissions openstack "." ".*" ".*"*

Configuring Memcached

The Identity service authentication mechanism for services uses Memcached to cache tokens. The Memcached service typicall runs on the controller node. Run the following commands to configure Memcached:

1. *sudo apt install memcached python-memcache*
2. Edit the /etc/memcached.conf file by changing *–l 127.0.0.1* to *–l 10.0.0.11*.
3. *sudo service memcached restart*

Installing OpenStack Services

At minimal, you need to install – in this order – the following components:

1. Identity Service (Keystone)
2. Image Service (Glance)
3. Compute Service (Nova)
4. Networking Service (Neutron)
5. Dashboard (Horizon)
6. Block Storage Service (Cinder)

Installation of each service is covered in the consecutive presentations.