OPENSTACK CLOUD INSTALLATION GUIDE

# Introduction:

In the past decade cloud computing has been gaining popularity at tremendous rate. Cloud computing enables convenient and on demand access to a shared pool of configurable computing resources which can be rapidly provident and released with minimal effort.

Open Stack is a free and and open source software for building and managing own cloud interface. The software platform contains interrelated components that control hardware pool of processing of storage, network resource throughout the data centre. Users are managed through web based dashboard, through command line tools or through a bunch of APIs.

Services need to install in Open Stack

|  |  |  |
| --- | --- | --- |
| SERVICE | CODE NAME | DESCRIPTION |
| Identity Service | Keystone | User Management |
| Compute Service | Nova | Virtual Machine Management |
| Image Service | Glance | Manages Virtual image like kernel image or disk image |
| Network Service | Neutron | Virtual Networking Management |
| Dashboard | Horizon | Provides GUI console for web browser |
| Object Storage (optional) | Swift | Provides Cloud Storage |

# HARDWARE REQUIREMENTS

Architecture requires at least two nodes (hosts) to launch a basic virtual machine or instance.

### Controller

The controller node runs the Identity service, Image service, management portions of Compute, management portion of Networking, various Networking agents, and the dashboard.

For this node required specification

1. CPU >=i3 processor 2. RAM >=8GB

3. Storage >=5GB 4. NIC >=2

### Compute

The compute node runs the hypervisor portion of Compute that operates instances. By default, Compute uses the KVM hypervisor. The compute node also runs a Networking service agent that connects instances to virtual networks and provides firewalling services to instances via security groups.

1. CPU >=i3processor 2. RAM >=8GB

3. Storage >=10GB 4. NIC >=2

# Environment Setup for Installation

NOTE: Your distribution does not enable a restrictive [firewall](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-firewall) by default.

## Configure network interfaces

Edit the /etc/network/interfaces file

Replace INTERFACE\_NAME with the actual interface name. For example, eth1 or ens224

*# 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

Reboot the system to activate the changes

## Configure name resolution

Set the hostname of the node to controller (any name) and compute node. Additional compute node can be added based on requirement on other ip's. Object and block node are optional.

Edit the /etc/hosts file to contain the following

*# controller*

10.0.0.11 controller

*# compute1*

10.0.0.31 compute1

Warning: **Do not remove the 127.0.0.1 entry.**

 Verify network connectivity to the Internet and among the nodes before proceeding further.

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

# ping -c 4 openstack.org

PING openstack.org (174.143.194.225) 56(84) bytes of data.

64 bytes from 174.143.194.225: icmp\_seq=1 ttl=54 time=18.3 ms

64 bytes from 174.143.194.225: icmp\_seq=2 ttl=54 time=17.5 ms

64 bytes from 174.143.194.225: icmp\_seq=3 ttl=54 time=17.5 ms

64 bytes from 174.143.194.225: icmp\_seq=4 ttl=54 time=17.4 ms

--- openstack.org ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3022ms

rtt min/avg/max/mdev = 17.489/17.715/18.346/0.364 ms

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

# ping -c 4 compute1

PING compute1 (10.0.0.31) 56(84) bytes of data.

64 bytes from compute1 (10.0.0.31): icmp\_seq=1 ttl=64 time=0.263 ms

64 bytes from compute1 (10.0.0.31): icmp\_seq=2 ttl=64 time=0.202 ms

64 bytes from compute1 (10.0.0.31): icmp\_seq=3 ttl=64 time=0.203 ms

64 bytes from compute1 (10.0.0.31): icmp\_seq=4 ttl=64 time=0.202 ms

--- compute1 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3000ms

rtt min/avg/max/mdev = 0.202/0.217/0.263/0.030 ms

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

ping -c 4 controller

PING controller (10.0.0.11) 56(84) bytes of data.

64 bytes from controller (10.0.0.11): icmp\_seq=1 ttl=64 time=0.263 ms

64 bytes from controller (10.0.0.11): icmp\_seq=2 ttl=64 time=0.202 ms

64 bytes from controller (10.0.0.11): icmp\_seq=3 ttl=64 time=0.203 ms

64 bytes from controller (10.0.0.11): icmp\_seq=4 ttl=64 time=0.202 ms

--- controller ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3000ms

rtt min/avg/max/mdev = 0.202/0.217/0.263/0.030 ms

## Install and configure components

1. Install the packages:

# apt install chrony

Chrony can synchronize the system clock with NTP servers, reference clocks (e.g. GPS receiver), and manual input using wristwatch and keyboard.

2. Edit the /etc/chrony/chrony.conf file and add, change, or remove these keys as necessary for your environment:

Replace NTP\_SERVER with the hostname or IP address of a suitable more accurate (lower stratum) NTP server. The configuration supports multiple server keys.

server NTP\_SERVER iburst

3. To enable 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:

# service chrony restart

Other nodes reference the controller node for clock synchronization. Perform these steps on all other nodes.

1.Install the packages:

1. # apt install chrony
2. Edit the /etc/chrony/chrony.conf file and comment out or remove all but one server key. Change it to reference the controller node:

server controller iburst

1. Comment out the pool 2.debian.pool.ntp.org offline iburst line.
2. Restart the NTP service:

# service chrony restart

To verify the operation run the following command

chronyc sources

210 Number of sources = 2

MS Name/IP address Stratum Poll Reach LastRx Last sample

===============================================================================

^- 192.0.2.11 2 7 12 137 -2814us[-3000us] +/- 43ms

^\* 192.0.2.12 2 6 177 46 +17us[ -23us] +/- 68ms

Contents in the Name/IP address column should indicate the hostname or IP address of one or more NTP servers. Contents in the S column should indicate \* for the server to which the NTP service is currently synchronized.

Run the same command on *all other* nodes:

# chronyc sources

210 Number of sources = 1

MS Name/IP address Stratum Poll Reach LastRx Last sample

===============================================================================

^\* controller 3 9 377 421 +15us[ -87us] +/- 15ms

# Enable the OpenStack repository

# apt install software-properties-common

# add-apt-repository cloud-archive:newton

## Finalize the installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/environment-packages.html#finalize-the-installation)

Upgrade the packages on your host:

# apt update && apt dist-upgrade

Note: Reboot the host to activate.

Install the OpenStack client:

# apt install python-openstackclient

**Note:** Most OpenStack services use an SQL database to store information. The database typically runs on the controller node.

# apt install mariadb-server python-pymysql

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

* Create a [mysqld] section, and set the bind-address key to the management IP address of the controller node to enable access by other nodes via the management network. Set additional keys to enable useful options and the UTF-8 character set:
* **[mysqld]**
* bind-address = 10.0.0.11
* default-storage-engine = innodb
* innodb\_file\_per\_table
* max\_connections = 4096
* collation-server = utf8\_general\_ci
* character-set-server = utf8

## Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/environment-sql-database.html#finalize-installation)

1. Restart the database service:

# service mysql restart

1. Secure the database service by running the mysql\_secure\_installation script. In particular, choose a suitable password for the database root account.

# mysql\_secure\_installation

**NOTE:** OpenStack uses a [message queue](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-message-queue) to coordinate operations and status information among services. The message queue service typically runs on the controller node.

1. Install the package:

# apt install rabbitmq-server

1. Add the openstack user:

# rabbitmqctl add\_user openstack RABBIT\_PASS

Creating user "openstack" ...

Replace RABBIT\_PASS with a suitable password.

1. Permit configuration, write, and read access for the openstack user:

# rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"

Setting permissions for user "openstack" in vhost "/" ...

**NOTE:** The Identity service authentication mechanism for services uses Memcached to cache tokens. The memcached service typically runs on the controller node.

1. Install the packages:

# apt install memcached python-memcache

1. Edit the /etc/memcached.conf file and configure the service to use the management IP address of the controller node. This is to enable access by other nodes via the management network:

Change the existing line that had -l 127.0.0.1.

-l 10.0.0.11

## Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/environment-memcached.html#finalize-installation)

* Restart the Memcached service:

# service memcached restart

Note:  Next step is installing services.

Each service can have one or many endpoints and each endpoint can be one of three types: admin, internal, or public. In a production environment, different endpoint types might reside on separate networks exposed to different types of users for security reasons. For instance, the public API network might be visible from the Internet so customers can manage their clouds. The admin API network might be restricted to operators within the organization that manages cloud infrastructure. The internal API network might be restricted to the hosts that contain OpenStack services.

### Instructions before installing services

* **Installer should be a root user during installing services.**
* **Make sure that in some commands password is chosen by your own for ex: DB\_PASS.**
* **In the documentation DB\_PASS is chosen as password so be attentive while installing.**

# IDENTITY SERVICE INSTALLATION

Keystone is the identity service used by [OpenStack](https://wiki.openstack.org/wiki/OpenStack) for authentication (authN) and high-level authorization (authZ). It currently supports token-based authN and user-service authorization. It has recently been re-architected to allow for expansion to support proxying external services and AuthN/AuthZ mechanisms such as oAuth, SAML and openID in future versions.

The Identity service contains these components:

**Server**

A centralized server provides authentication and authorization services using a RESTful interface.

**Drivers**

Drivers or a service back end are integrated to the centralized server. They are used for accessing identity information in repositories external to OpenStack, and may already exist in the infrastructure where OpenStack is deployed (for example, SQL databases or LDAP servers).

**Modules**

Middleware modules run in the address space of the OpenStack component that is using the Identity service. These modules intercept service requests, extract user credentials, and send them to the centralized server for authorization. The integration between the middleware modules and OpenStack components uses the Python Web Server Gateway Interface.

## Prerequisites[¶](https://docs.openstack.org/newton/install-guide-ubuntu/keystone-install.html#prerequisites)

Note: Should be as a root user

Must create a database and an administration token before configuring keystone.

Use the database access client to connect to the database server

mysql -u root -p

Create the keystone database:

mysql> CREATE DATABASE keystone;

Grant proper access to the keystone database:

Note: Replace KEYSTONE\_DBPASS with a suitable password.

mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' \

IDENTIFIED BY 'KEYSTONE\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' \

IDENTIFIED BY 'KEYSTONE\_DBPASS';

1. Now installing and configuring keystone packages

# apt install keystone

2. Edit the /etc/keystone/keystone.conf file and complete the following actions:

* In the [database] section, configure database access:

Note: Even there existing a connection before just add below that.

Replace KEYSTONE\_DBPASS with the password you chose for the database.

**[database]**

...

connection = mysql+pymysql://keystone:KEYSTONE\_DBPASS@controller/keystone

* In the [token] section, configure the Fernet token provider:

**[token]**

...

provider = fernet

3. Populate the Identity service database:

# su -s /bin/sh -c "keystone-manage db\_sync" keystone

4. Initialize Fernet key repositories:

# keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

# keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone

5. Bootstrap the Identity service:

Replace ADMIN\_PASS with a suitable password for an administrative user.

# keystone-manage bootstrap --bootstrap-password ADMIN\_PASS \

--bootstrap-admin-url http://controller:35357/v3/ \

--bootstrap-internal-url http://controller:35357/v3/ \

--bootstrap-public-url http://controller:5000/v3/ \

--bootstrap-region-id RegionOne

## Configure the Apache HTTP server

1. Edit the /etc/apache2/apache2.conf file and configure the ServerName option to reference the controller node:

ServerName controller

## Initialize the installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/keystone-install.html#finalize-the-installation)

1. Restart the Apache service and remove the default SQLite database:

# service apache2 restart

# rm -f /var/lib/keystone/keystone.db

2. Configure the administrative account

Replace ADMIN\_PASS with the password used in the keystone-manage bootstrap

$ export OS\_USERNAME=admin

$ export OS\_PASSWORD=ADMIN\_PASS

$ export OS\_PROJECT\_NAME=admin

$ export OS\_USER\_DOMAIN\_NAME=Default

$ export OS\_PROJECT\_DOMAIN\_NAME=Default

$ export OS\_AUTH\_URL=http://controller:35357/v3

$ export OS\_IDENTITY\_API\_VERSION=3

The authentication service uses a combination of [domains](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-domain), [projects](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-project), [users](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-user), and [roles](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-role).

This guide uses a service project that contains a unique user for each service that you add to your environment. Create the service project:

$ openstack project create --domain default \

--description "Service Project" service

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Service Project |

| domain\_id | default |

| enabled | True |

| id | 24ac7f19cd944f4cba1d77469b2a73ed |

| is\_domain | False |

| name | service |

| parent\_id | default |

+-------------+----------------------------------+

Regular (non-admin) tasks should use an unprivileged project and user. As an example, this guide creates the demoproject and user.

* Create the demo project:

$ openstack project create --domain default \

--description "Demo Project" demo

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Demo Project |

| domain\_id | default |

| enabled | True |

| id | 231ad6e7ebba47d6a1e57e1cc07ae446 |

| is\_domain | False |

| name | demo |

| parent\_id | default |

+-------------+----------------------------------+

NOTE: Do not repeat this step when creating additional users for this project.

Create the demo user:

$ openstack user create --domain default \

--password-prompt demo

User Password:

Repeat User Password:

+---------------------+----------------------------------+

| Field | Value |

+---------------------+----------------------------------+

| domain\_id | default |

| enabled | True |

| id | aeda23aa78f44e859900e22c24817832 |

| name | demo |

| password\_expires\_at | None |

+---------------------+----------------------------------+

* Create the user role:

$ openstack role create user

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | None |

| id | 997ce8d05fc143ac97d83fdfb5998552 |

| name | user |

+-----------+----------------------------------+

* Add the user role to the demo project and user:

$ openstack role add --project demo --user demo user

## VERIFICATION OF KEYSTONE INSTALLATION

1. For security reasons, disable the temporary authentication token mechanism:

Edit the /etc/keystone/keystone-paste.ini file and remove admin\_token\_auth from the [pipeline:public\_api], [pipeline:admin\_api], and [pipeline:api\_v3] sections.

1. Unset the temporary OS\_AUTH\_URL and OS\_PASSWORD environment variable:

$ unset OS\_AUTH\_URL OS\_PASSWORD

3. As the admin user, request an authentication token:

$ openstack --os-auth-url http://controller:35357/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name admin --os-username admin token issue

Password:

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:14:07.056119Z |

| id | gAAAAABWvi7\_B8kKQD9wdXac8MoZiQldmjEO643d-e\_j-XXq9AmIegIbA7UHGPv |

| | atnN21qtOMjCFWX7BReJEQnVOAj3nclRQgAYRsfSU\_MrsuWb4EDtnjU7HEpoBb4 |

| | o6ozsA\_NmFWEpLeKy0uNn\_WeKbAhYygrsmQGA49dclHVnz-OMVLiyM9ws |

| project\_id | 343d245e850143a096806dfaefa9afdc |

| user\_id | ac3377633149401296f6c0d92d79dc16 |

+------------+-----------------------------------------------------------------+

4. As the demo user, request an authentication token:

$ openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name demo --os-username demo token issue

Password:

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:15:39.014479Z |

| id | gAAAAABWvi9bsh7vkiby5BpCCnc-JkbGhm9wH3fabS\_cY7uabOubesi-Me6IGWW |

| | yQqNegDDZ5jw7grI26vvgy1J5nCVwZ\_zFRqPiz\_qhbq29mgbQLglbkq6FQvzBRQ || | JcOzq3uwhzNxszJWmzGC7rJE\_H0A\_a3UFhqv8M4zMRYSbS2YF0MyFmp\_U |

| project\_id | ed0b60bf607743088218b0a533d5943f |

| user\_id | 58126687cbcc4888bfa9ab73a2256f27 |

+------------+-----------------------------------------------------------------+

## Create OpenStack client environment scripts

Create client environment scripts for the admin and demo projects and users.

1. Edit the admin-openrc file and add the following content:

Replace ADMIN\_PASS with the password you chose for the admin user in the Identity service.

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://controller:35357/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

2. Edit the demo-openrc file and add the following content:

Replace DEMO\_PASS with the password you chose for the demo user in the Identity service.

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=DEMO\_PASS

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

#### Using the scripts

1. Load the admin-openrc file to populate environment variables with the location of the Identity service and the adminproject and user credentials:

$ . admin-openrc

2. Request an authentication token:

$ openstack token issue

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:44:35.659723Z |

| id | gAAAAABWvjYj-Zjfg8WXFaQnUd1DMYTBVrKw4h3fIagi5NoEmh21U72SrRv2trl |

| | JWFYhLi2\_uPR31Igf6A8mH2Rw9kv\_bxNo1jbLNPLGzW\_u5FC7InFqx0yYtTwa1e |

| | eq2b0f6-18KZyQhs7F3teAta143kJEWuNEYET-y7u29y0be1\_64KYkM7E |

| project\_id | 343d245e850143a096806dfaefa9afdc |

| user\_id | ac3377633149401296f6c0d92d79dc16 |

+------------+-----------------------------------------------------------------+

Identity Service installation is completed.

# IMAGE SERVICE INSTALLATION

OpenStack Image (Glance) provides discovery, registration, and delivery services for [disk](https://en.wikipedia.org/wiki/Disk_imaging) and [server images](https://en.wikipedia.org/w/index.php?title=Server_imaging&action=edit&redlink=1). Stored images can be used as a template. It can also be used to store and catalog an unlimited number of backups. The Image Service can store disk and server images in a variety of back-ends, including [Swift](https://en.wikipedia.org/wiki/OpenStack#Swift). The Image Service API provides a standard [REST interface](https://en.wikipedia.org/wiki/Representational_state_transfer) for querying information about disk images and lets clients stream the images to new servers.

Glance adds many enhancements to existing legacy infrastructures. For example, if integrated with VMware, Glance introduces advanced features to the vSphere family such as vMotion, high availability and dynamic resource scheduling (DRS). vMotion is the live migration of a running VM, from one physical server to another, without service interruption. Thus, it enables a dynamic and automated self-optimizing datacentre, allowing hardware maintenance for the underperforming servers without downtimes.

Other OpenStack modules that need to interact with Images, for example [Heat](https://en.wikipedia.org/wiki/OpenStack#Heat), must communicate with the images metadata through Glance. Also, [Nova](https://en.wikipedia.org/wiki/Nova_(software)) can present information about the images, and configure a variation on an image to produce an instance. However, Glance is the only module that can add, delete, share, or duplicate images.

The OpenStack Image service includes the following components:

**Glance-api**

Accepts Image API calls for image discovery, retrieval, and storage.

**Glance-registry**

Stores, processes, and retrieves metadata about images. Metadata includes items such as size and type.

**Database**

Stores image metadata and you can choose your database depending on your preference. Most deployments use MySQL or SQLite.

**Storage repository for image files**

Various repository types are supported including normal file systems (or any filesystem mounted on the glance-api controller node), Object Storage, RADOS block devices, VMware datastore, and HTTP. Note that some repositories will only support read-only usage.

**Metadata definition service**

A common API for vendors, admins, services, and users to meaningfully define their own custom metadata. This metadata can be used on different types of resources like images, artifacts, volumes, flavors, and aggregates. A definition includes the new property’s key, description, constraints, and the resource types which it can be associated with.

**Prerequisites**

 1. Create a database, service credentials, and API endpoints before configuring Image service.

$ mysql -u root -p

Create the glance database:

mysql> CREATE DATABASE glance;

Grant proper access to the glance database:

Replace GLANCE\_DBPASS with a suitable password.

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' \

IDENTIFIED BY 'GLANCE\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' \

IDENTIFIED BY 'GLANCE\_DBPASS';

2. Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

3. To create the service credentials, complete these steps:

* Create the glance user:

$ openstack user create --domain default --password-prompt glance

User Password:

Repeat User Password:

+---------------------+----------------------------------+

| Field | Value |

+---------------------+----------------------------------+

| domain\_id | default |

| enabled | True |

| id | 3f4e777c4062483ab8d9edd7dff829df |

| name | glance |

| password\_expires\_at | None |

+---------------------+----------------------------------+

Add the admin role to the glance user and service project:

$ openstack role add --project service --user glance admin

* Create the glance service entity:

$ openstack service create --name glance \

--description "OpenStack Image" image

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Image |

| enabled | True |

| id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| name | glance |

| type | image |

+-------------+----------------------------------+

4. Create the Image service API endpoints:

$ openstack endpoint create --region RegionOne \

image public http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 340be3625e9b4239a6415d034e98aace |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

image internal http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | a6e4b153c2ae4c919eccfdbb7dceb5d2 |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

image admin http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 0c37ed58103f4300a84ff125a539032d |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

## Install and configure components

1. Install the packages:

# apt install glance

2. Edit the /etc/glance/glance-api.conf file and complete the following actions:

* In the [database] section, configure database access:
* If there is any existing connection just add the new connection below.
* Replace GLANCE\_DBPASS with the password you chose for the Image service database.

**[database]**

...

connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance

In the [keystone\_authtoken] and [paste\_deploy] sections, configure Identity service access:

Replace GLANCE\_PASS with the password you chose for the glance user in the Identity service.

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = glance

password = GLANCE\_PASS

**[paste\_deploy]**

...

flavor = keystone

In the [glance\_store] section, configure the local file system store and location of image files:

**[glance\_store]**

...

stores = file,http

default\_store = file

filesystem\_store\_datadir = /var/lib/glance/images/

3. Edit the /etc/glance/glance-registry.conf file and complete the following actions:

* In the [database] section, configure database access:
* Replace GLANCE\_DBPASS with the password you chose for the Image service database.
* If there is any existing connection just add the new connection below.

**[database]**

...

connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance

In the [keystone\_authtoken] and [paste\_deploy] sections, configure Identity service access:

Replace GLANCE\_PASS with the password you chose for the glance user in the Identity service.

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = glance

password = GLANCE\_PASS

**[paste\_deploy]**

...

flavor = keystone

NOTE: Comment out or remove any other options in the [keystone\_authtoken] section.

4. Populate the Image service database:

# su -s /bin/sh -c "glance-manage db\_sync" glance

Note: Ignore any deprecation messages in this output.

### Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/glance-install.html#finalize-installation)

1. Restart the Image services:

# service glance-registry restart

# service glance-api restart

# Verify operation

Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

Download the source image:

$ wget http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86\_64-disk.img

Upload the image to the Image service using the [QCOW2](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-qemu-copy-on-write-2-qcow2) disk format, [bare](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-bare) container format, and public visibility so all projects can access it:

$ openstack image create "cirros" \

--file cirros-0.3.4-x86\_64-disk.img \

--disk-format qcow2 --container-format bare \

--public

+------------------+------------------------------------------------------+

| Field | Value |

+------------------+------------------------------------------------------+

| checksum | 133eae9fb1c98f45894a4e60d8736619 |

| container\_format | bare |

| created\_at | 2015-03-26T16:52:10Z |

| disk\_format | qcow2 |

| file | /v2/images/cc5c6982-4910-471e-b864-1098015901b5/file |

| id | cc5c6982-4910-471e-b864-1098015901b5 |

| min\_disk | 0 |

| min\_ram | 0 |

| name | cirros |

| owner | ae7a98326b9c455588edd2656d723b9d |

| protected | False |

| schema | /v2/schemas/image |

| size | 13200896 |

| status | active |

| tags | |

| updated\_at | 2015-03-26T16:52:10Z |

| virtual\_size | None |

| visibility | public |

+------------------+------------------------------------------------------+

Confirm upload of the image and validate attributes:

$ openstack image list

+--------------------------------------+--------+--------+

| ID | Name | Status |

+--------------------------------------+--------+--------+

| 38047887-61a7-41ea-9b49-27987d5e8bb9 | cirros | active |

+--------------------------------------+--------+--------+

# INSTALLING COMPUTE SERVICE

OpenStack Compute (Nova) is a cloud computing fabric controller, which is the main part of an [IaaS](https://en.wikipedia.org/wiki/IaaS" \o "IaaS) system. It is designed to manage and automate pools of computer resources and can work with widely available virtualization technologies, as well as [bare metal](https://en.wikipedia.org/wiki/Bare_metal) and [high-performance computing](https://en.wikipedia.org/wiki/High-performance_computing) (HPC) configurations. [KVM](https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine), [VMware](https://en.wikipedia.org/wiki/VMware), and [Xen](https://en.wikipedia.org/wiki/Xen" \o "Xen) are available choices for [hypervisor](https://en.wikipedia.org/wiki/Hypervisor) technology (virtual machine monitor), together with [Hyper-V](https://en.wikipedia.org/wiki/Hyper-V) and Linux container technology such as [LXC](https://en.wikipedia.org/wiki/LXC).

## Prerequisites

$ mysql -u root -p

1. Create the nova\_api and nova databases:

mysql> CREATE DATABASE nova\_api;

mysql> CREATE DATABASE nova;

Grant proper access to the databases:

Replace NOVA\_DBPASS with a suitable password.

mysql> GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'localhost' \

IDENTIFIED BY 'NOVA\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'%' \

IDENTIFIED BY 'NOVA\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' \

IDENTIFIED BY 'NOVA\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' \

IDENTIFIED BY 'NOVA\_DBPASS';

2. Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

3. To create the service credentials, complete these steps:

* Create the nova user:

$ openstack user create --domain default \

--password-prompt nova

User Password:

Repeat User Password:

+---------------------+----------------------------------+

| Field | Value |

+---------------------+----------------------------------+

| domain\_id | default |

| enabled | True |

| id | 8a7dbf5279404537b1c7b86c033620fe |

| name | nova |

| password\_expires\_at | None |

+---------------------+----------------------------------+

* Add the admin role to the nova user:

$ openstack role add --project service --user nova admin

* Create the nova service entity:

$ openstack service create --name nova \

--description "OpenStack Compute" compute

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Compute |

| enabled | True |

| id | 060d59eac51b4594815603d75a00aba2 |

| name | nova |

| type | compute |

+-------------+----------------------------------+

4. Create the Compute service API endpoints:

openstack endpoint create --region RegionOne \

compute public http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | 3c1caa473bfe4390a11e7177894bcc7b |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 060d59eac51b4594815603d75a00aba2 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

$ openstack endpoint create --region RegionOne \

compute internal http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | e3c918de680746a586eac1f2d9bc10ab |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 060d59eac51b4594815603d75a00aba2 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

$ openstack endpoint create --region RegionOne \

compute admin http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | 38f7af91666a47cfb97b4dc790b94424 |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 060d59eac51b4594815603d75a00aba2 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

## Install and configure components

1. Install the packages:

# apt install nova-api nova-conductor nova-consoleauth \

nova-novncproxy nova-scheduler

2. Edit the /etc/nova/nova.conf file and complete the following actions:

* In the [api\_database] and [database] sections, configure database access:
* Replace NOVA\_DBPASS with the password you chose for the Compute databases.

**[api\_database]**

...

connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova\_api

**[database]**

...

connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova

* In the [DEFAULT] section, configure RabbitMQ message queue access:
* Replace RABBIT\_PASS with the password you chose for the openstack account in RabbitMQ.

**[DEFAULT]**

...

transport\_url = rabbit://openstack:RABBIT\_PASS@controller

* In the [DEFAULT] and [keystone\_authtoken] sections, configure Identity service access:
* Replace NOVA\_PASS with the password you chose for the nova user in the Identity service.

**[DEFAULT]**

...

auth\_strategy = keystone

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = nova

password = NOVA\_PASS

Note: Comment out or remove any other options in the [keystone\_authtoken] section.

* In the [DEFAULT] section, configure the my\_ip option to use the management interface IP address of the controller node:

**[DEFAULT]**

...

my\_ip = 10.0.0.11

* In the [DEFAULT] section, enable support for the Networking service:

**[DEFAULT]**

...

use\_neutron = True

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

* In the [vnc] section, configure the VNC proxy to use the management interface IP address of the controller node:

**[vnc]**

...

vncserver\_listen = $my\_ip

vncserver\_proxyclient\_address = $my\_ip

* In the [glance] section, configure the location of the Image service API:

**[glance]**

...

api\_servers = http://controller:9292

* In the [oslo\_concurrency] section, configure the lock path:

**[oslo\_concurrency]**

...

lock\_path = /var/lib/nova/tmp

* Due to a packaging bug, remove the log-dir option from the [DEFAULT] section.

3. Populate the Compute databases:

# su -s /bin/sh -c "nova-manage api\_db sync" nova

# su -s /bin/sh -c "nova-manage db sync" nova

## Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/nova-controller-install.html#finalize-installation)

* Restart the Compute services:

# service nova-api restart

# service nova-consoleauth restart

# service nova-scheduler restart

# service nova-conductor restart

# service nova-novncproxy restart

# Install and configure a compute node

1. Install the packages:

# apt install nova-compute

2. Edit the /etc/nova/nova.conf file and complete the following actions:

* In the [DEFAULT] section, configure RabbitMQ message queue access:
* Replace RABBIT\_PASS with the password you chose for the openstack account in RabbitMQ.

**[DEFAULT]**

...

transport\_url = rabbit://openstack:RABBIT\_PASS@controller

* In the [DEFAULT] and [keystone\_authtoken] sections, configure Identity service access:
* Replace NOVA\_PASS with the password you chose for the nova user in the Identity service.

**[DEFAULT]**

...

auth\_strategy = keystone

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = nova

password = NOVA\_PASS

* In the [DEFAULT] section, configure the my\_ip option:
* Replace MANAGEMENT\_INTERFACE\_IP\_ADDRESS with the IP address of the management network interface on your compute node.

**[DEFAULT]**

...

my\_ip = MANAGEMENT\_INTERFACE\_IP\_ADDRESS

* In the [DEFAULT] section, enable support for the Networking service:

**[DEFAULT]**

...

use\_neutron = True

firewall\_driver = nova.virt.firewall.NoopFirewallDrive

* In the [vnc] section, enable and configure remote console access:

**[vnc]**

...

enabled = True

vncserver\_listen = 0.0.0.0

vncserver\_proxyclient\_address = $my\_ip

novncproxy\_base\_url = http://controller:6080/vnc\_auto.html

* In the [glance] section, configure the location of the Image service API:

**[glance]**

...

api\_servers = http://controller:9292

## Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/nova-compute-install.html#finalize-installation)

1. Determine whether your compute node supports hardware acceleration for virtual machines:

$ egrep -c '(vmx|svm)' /proc/cpuinfo

If this command returns a value of one or greater, your compute node supports hardware acceleration which typically requires no additional configuration.

If this command returns a value of zero, your compute node does not support hardware acceleration and you must configure libvirt to use QEMU instead of KVM.

* Edit the [libvirt] section in the /etc/nova/nova-compute.conf file as follows:

**[libvirt]**

...

virt\_type = qemu

2. Restart the Compute service:

# service nova-compute restart

## Verify operation

1. Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

2. List service components to verify successful launch and registration of each process:

$ openstack compute service list

+----+--------------------+------------+----------+---------+-------+----------------------------+

| Id | Binary | Host | Zone | Status | State | Updated At |

+----+--------------------+------------+----------+---------+-------+----------------------------+

| 1 | nova-consoleauth | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 2 | nova-scheduler | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 3 | nova-conductor | controller | internal | enabled | up | 2016-02-09T23:11:16.000000 |

| 4 | nova-compute | compute1 | nova | enabled | up | 2016-02-09T23:11:20.000000 |

+----+--------------------+------------+----------+---------+-------+----------------------------+

# INSTALLING NETWORK SERVICE

OpenStack Networking (Neutron) is a system for managing networks and [IP addresses](https://en.wikipedia.org/wiki/IP_address). OpenStack Networking ensures the network is not a bottleneck or limiting factor in a cloud deployment,] and gives users self-service ability, even over network configurations.

## Prerequisites[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install.html#prerequisites)

Before you configure the OpenStack Networking (neutron) service, you must create a database, service credentials, and API endpoints.

* Access client to connect to the database server

$ mysql -u root -p

* Create the neutron database:

mysql> CREATE DATABASE neutron;

* Grant proper access to the neutron database, replacing NEUTRON\_DBPASS with a suitable password:

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' \

IDENTIFIED BY 'NEUTRON\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' \

IDENTIFIED BY 'NEUTRON\_DBPASS';

2. Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

3. To create the service credentials, complete these steps:

* Create the neutron user:
* $ openstack user create --domain default --password-prompt neutron
* User Password:
* Repeat User Password:
* +---------------------+----------------------------------+
* | Field | Value |
* +---------------------+----------------------------------+
* | domain\_id | default |
* | enabled | True |
* | id | 319f34694728440eb8ffcb27b6dd8b8a |
* | name | neutron |
* | password\_expires\_at | None |
* +---------------------+----------------------------------+
* Add the admin role to the neutron user:

$ openstack role add --project service --user neutron admin

* Create the neutron service entity:

$ openstack service create --name neutron \

--description "OpenStack Networking" network

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Networking |

| enabled | True |

| id | f71529314dab4a4d8eca427e701d209e |

| name | neutron |

| type | network |

+-------------+----------------------------------+

4. Create the Networking service API endpoints:

$ openstack endpoint create --region RegionOne \

network public http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 85d80a6d02fc4b7683f611d7fc1493a3 |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | f71529314dab4a4d8eca427e701d209e |

| service\_name | neutron |

| service\_type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

network internal http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 09753b537ac74422a68d2d791cf3714f |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | f71529314dab4a4d8eca427e701d209e |

| service\_name | neutron |

| service\_type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

network admin http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 1ee14289c9374dffb5db92a5c112fc4e |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | f71529314dab4a4d8eca427e701d209e |

| service\_name | neutron |

| service\_type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

## Configure networking options[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install.html#configure-networking-options)

You can deploy the Networking service using one of two architectures represented by options 1 and 2.

Option 1 deploys the simplest possible architecture that only supports attaching instances to provider (external) networks. No self-service (private) networks, routers, or floating IP addresses. Only the admin or other privileged user can manage provider networks.

Option 2 augments option 1 with layer-3 services that support attaching instances to self-service networks. The demo or other unprivileged user can manage self-service networks including routers that provide connectivity between self-service and provider networks. Additionally, floating IP addresses provide connectivity to instances using self-service networks from external networks such as the Internet.

We used chose option1 since it is simple.

# Networking Option 1: Provider networks

## Install the components[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install-option1.html#install-the-components)

# apt install neutron-server neutron-plugin-ml2 \

neutron-linuxbridge-agent neutron-dhcp-agent \

neutron-metadata-agent

## Configure the server component[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install-option1.html#configure-the-server-component)

The Networking server component configuration includes the database, authentication mechanism, message queue, topology change notifications, and plug-in.

Edit the /etc/neutron/neutron.conf file and complete the following actions:

* In the [database] section, configure database access:
* Replace NEUTRON\_DBPASS with the password you chose for the database.

**[database]**

...

connection = mysql+pymysql://neutron:NEUTRON\_DBPASS@controller/neutron

* In the [DEFAULT] and [keystone\_authtoken] sections, configure Identity service access:
* Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

**[DEFAULT]**

...

auth\_strategy = keystone

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = neutron

password = NEUTRON\_PASS

* In the [DEFAULT] and [nova] sections, configure Networking to notify Compute of network topology changes:
* Replace NOVA\_PASS with the password you chose for the nova user in the Identity service.

**[DEFAULT]**

...

notify\_nova\_on\_port\_status\_changes = True

notify\_nova\_on\_port\_data\_changes = True

**[nova]**

...

auth\_url = http://controller:35357

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

region\_name = RegionOne

project\_name = service

username = nova

password = NOVA\_PASS

## Configure the Modular Layer 2 (ML2) plug-in

The ML2 plug-in uses the Linux bridge mechanism to build layer-2 (bridging and switching) virtual networking infrastructure for instances.

* Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file and complete the following actions:
  + In the [ml2] section, enable flat and VLAN networks:

**[ml2]**

...

type\_drivers = flat,vlan

* In the [ml2] section, disable self-service networks:

**[ml2]**

...

tenant\_network\_types =

* In the [ml2] section, enable the Linux bridge mechanism:

**[ml2]**

...

mechanism\_drivers = linuxbridge

* In the [ml2] section, enable the port security extension driver:

**[ml2]**

...

extension\_drivers = port\_security

* In the [ml2\_type\_flat] section, configure the provider virtual network as a flat network:

**[ml2\_type\_flat]**

...

flat\_networks = provider

* In the [securitygroup] section, enable [ipset](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html" \l "term-ipset) to increase efficiency of security group rules:

**[securitygroup]**

...

enable\_ipset = True

# Configure the Linux bridge agent

The Linux bridge agent builds layer-2 (bridging and switching) virtual networking infrastructure for instances and handles security groups.

* Edit the /etc/neutron/plugins/ml2/linuxbridge\_agent.ini file and complete the following actions:
  + In the [linux\_bridge] section, map the provider virtual network to the provider physical network interface:
  + Replace PROVIDER\_INTERFACE\_NAME with the name of the underlying provider physical network interface. See [Host networking](https://docs.openstack.org/newton/install-guide-ubuntu/environment-networking.html#environment-networking) for more information.

**[linux\_bridge]**

physical\_interface\_mappings = provider:PROVIDER\_INTERFACE\_NAME

* In the [vxlan] section, disable VXLAN overlay networks:

**[vxlan]**

enable\_vxlan = False

* In the [securitygroup] section, enable security groups and configure the Linux bridge [iptables](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html" \l "term-iptables) firewall driver:

**[securitygroup]**

...

enable\_security\_group = True

firewall\_driver = neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver

## Configure the DHCP agent

Edit the /etc/neutron/dhcp\_agent.ini file and complete the following actions:

* In the [DEFAULT] section, configure the Linux bridge interface driver, Dnsmasq DHCP driver, and enable isolated metadata so instances on provider networks can access metadata over the network:

**[DEFAULT]**

...

interface\_driver = neutron.agent.linux.interface.BridgeInterfaceDriver

dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq

enable\_isolated\_metadata = True

We have completed the configuring network artitecture1.

# Configure the metadata agent[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install.html#configure-the-metadata-agent)

The [metadata agent](https://docs.openstack.org/newton/install-guide-ubuntu/common/glossary.html#term-metadata-agent) provides configuration information such as credentials to instances.

* Edit the /etc/neutron/metadata\_agent.ini file and complete the following actions:
  + In the [DEFAULT] section, configure the metadata host and shared secret:
  + Replace METADATA\_SECRET with a suitable secret for the metadata proxy.

**[DEFAULT]**

...

nova\_metadata\_ip = controller

metadata\_proxy\_shared\_secret = METADATA\_SECRET

## Configure the Compute service to use the Networking service[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-controller-install.html#configure-the-compute-service-to-use-the-networking-service)

* Edit the /etc/nova/nova.conf file and perform the following actions:
  + In the [neutron] section, configure access parameters, enable the metadata proxy, and configure the secret:
  + Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.
  + Replace METADATA\_SECRET with the secret you chose for the metadata proxy.

**[neutron]**

...

url = http://controller:9696

auth\_url = http://controller:35357

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

region\_name = RegionOne

project\_name = service

username = neutron

password = NEUTRON\_PASS

service\_metadata\_proxy = True

metadata\_proxy\_shared\_secret = METADATA\_SECRET

## Finalize installation

1. Populate the database:

# 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

2. Restart the Compute API service:

# service nova-api restart

3. Restart the Networking services.

# 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 and configure compute node

## Install the components

# apt install neutron-linuxbridge-agent

## Configure the common component

2. Edit the /etc/neutron/neutron.conf file and complete the following actions:

* In the [database] section, comment out any connection options because compute nodes do not directly access the database.
* In the [DEFAULT] section, configure RabbitMQ message queue access:
* Replace RABBIT\_PASS with the password you chose for the openstack account in RabbitMQ.

**[DEFAULT]**

...

transport\_url = rabbit://openstack:RABBIT\_PASS@controller

* In the [DEFAULT] and [keystone\_authtoken] sections, configure Identity service access:
* Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

**[DEFAULT]**

...

auth\_strategy = keystone

**[keystone\_authtoken]**

...

auth\_uri = http://controller:5000

auth\_url = http://controller:35357

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = neutron

password = NEUTRON\_PASS

## Configure the Compute service to use the Networking service[¶](https://docs.openstack.org/newton/install-guide-ubuntu/neutron-compute-install.html#configure-the-compute-service-to-use-the-networking-service)

* Edit the /etc/nova/nova.conf file and complete the following actions:
  + In the [neutron] section, configure access parameters:
  + Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

**[neutron]**

...

url = http://controller:9696

auth\_url = http://controller:35357

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

region\_name = RegionOne

project\_name = service

username = neutron

password = NEUTRON\_PASS

## Finalize installation

1. Restart the Compute service:

# service nova-compute restart

2. Restart the Linux bridge agent:

# service neutron-linuxbridge-agent restart

## Verify operation

1. Source the admin credentials to gain access to admin-only CLI commands:

$ . admin-openrc

2. List loaded extensions to verify successful launch of the neutron-server process:

$ neutron ext-list

+---------------------------+-----------------------------------------------+

| alias | name |

+---------------------------+-----------------------------------------------+

| default-subnetpools | Default Subnetpools |

| network-ip-availability | Network IP Availability |

| network\_availability\_zone | Network Availability Zone |

| auto-allocated-topology | Auto Allocated Topology Services |

| ext-gw-mode | Neutron L3 Configurable external gateway mode |

| binding | Port Binding |

| agent | agent |

| subnet\_allocation | Subnet Allocation |

| l3\_agent\_scheduler | L3 Agent Scheduler |

| tag | Tag support |

| external-net | Neutron external network |

| net-mtu | Network MTU |

| availability\_zone | Availability Zone |

| quotas | Quota management support |

| l3-ha | HA Router extension |

| flavors | Neutron Service Flavors |

| provider | Provider Network |

| multi-provider | Multi Provider Network |

| address-scope | Address scope |

| extraroute | Neutron Extra Route |

| timestamp\_core | Time Stamp Fields addition for core resources |

| router | Neutron L3 Router |

| extra\_dhcp\_opt | Neutron Extra DHCP opts |

| dns-integration | DNS Integration |

| security-group | security-group |

| dhcp\_agent\_scheduler | DHCP Agent Scheduler |

| router\_availability\_zone | Router Availability Zone |

| rbac-policies | RBAC Policies |

| standard-attr-description | standard-attr-description |

| port-security | Port Security |

| allowed-address-pairs | Allowed Address Pairs |

| dvr | Distributed Virtual Router |

+---------------------------+-----------------------------------------------+

$ openstack network agent list

+--------------------------------------+--------------------+------------+-------------------+-------+-------+---------------------------+

| ID | Agent Type | Host | Availability Zone | Alive | State | Binary |

+--------------------------------------+--------------------+------------+-------------------+-------+-------+---------------------------+

| 0400c2f6-4d3b-44bc-89fa-99093432f3bf | Metadata agent | controller | None | True | UP | neutron-metadata-agent |

| 83cf853d-a2f2-450a-99d7-e9c6fc08f4c3 | DHCP agent | controller | nova | True | UP | neutron-dhcp-agent |

| ec302e51-6101-43cf-9f19-88a78613cbee | Linux bridge agent | compute | None | True | UP | neutron-linuxbridge-agent |

| fcb9bc6e-22b1-43bc-9054-272dd517d025 | Linux bridge agent | controller | None | True | UP | neutron-linuxbridge-agent |

# INSTALLING DASHBOARD

The Dashboard (horizon) is a web interface that enables cloud administrators and users to manage various OpenStack resources and services

The only core service required by the dashboard is the Identity service. You can use the dashboard in combination with other services, such as Image service, Compute, and Networking. You can also use the dashboard in environments with stand-alone services such as Object Storage.

1. Install the packages:

# apt install openstack-dashboard

2. Edit the /etc/openstack-dashboard/local\_settings.py file and complete the following actions:

* Configure the dashboard to use OpenStack services on the controller node:

OPENSTACK\_HOST = "controller"

* Allow all hosts to access the dashboard:

ALLOWED\_HOSTS = ['\*', ]

* Configure the memcached session storage service:

SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': 'controller:11211',

}

}

* Enable the Identity API version 3:

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

* Enable support for domains:

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

* Configure API versions:

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

* Configure default as the default domain for users that you create via the dashboard:

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "default"

* Configure user as the default role for users that you create via the dashboard:

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

* If you chose networking option 1, disable support for layer-3 networking services:

OPENSTACK\_NEUTRON\_NETWORK = {

...

'enable\_router': False,

'enable\_quotas': False,

'enable\_ipv6': False,

'enable\_distributed\_router': False,

'enable\_ha\_router': False,

'enable\_lb': False,

'enable\_firewall': False,

'enable\_vpn': False,

'enable\_fip\_topology\_check': False,

}

* Optionally, configure the time zone:

TIME\_ZONE = "TIME\_ZONE"

## Finalize installation[¶](https://docs.openstack.org/newton/install-guide-ubuntu/horizon-install.html#finalize-installation)

* Reload the web server configuration:

# service apache2 reload

## VERIFICATION

Access the dashboard using a web browser at http://controller/horizon.