OPENSTACK MITAKA

JOSÉ GARCÍA ROLDÁN

2º ASIR

Indice

Introducción

Documentación para el proyecto fin de ciclo de Administración de sistemas informáticos en red en realizado en el IES Gonzalo Nazareno.

El proyecto consiste en la instalación de Openstack Mitaka sobre dos nodos virtuales kvm con Ubuntu server 14.04 LTS.

En esta documentación se detallara paso a paso la instalación y configuración de cada uno de los servicios que componen OpenStack (keystone, glance, nova, neutron, cinder y horizon).

Entorno

Maquinas virtuales

Se compondrá de dos máquinas virtuales kvm con Ubuntu server 14.04 LTS:

```
# virsh list
 Id    Name                           State
----------------------------------------------------
 2     odin                           running
 3     thor                           running
```

Características:

Nombre	Función	RAM	Cores	IPs
odin	controlador	4,5 GB	2	10.0.0.21 192.168.1.36
thor	computo	2,5 GB	2	10.0.0.22

Servicios por nodos:

odin:

- Keystone
- Glance
- Nova-controller
- Neutron
- Horizon
- Cinder

thor:

- Nova-compute

Redes

Red interna: red aislada para el funcionamiento y las comunicaciones internas entre los nodos que componen el entorno del cloud de OpenStack. Esta red será utilizada para las comunicaciones de los diferentes servcios (keystone, glance, nova, neutron, cinder).

Red externa: red utilizada para la instalación de los componentes y para el acceso desde el exterior.

Pre-configuración

Habilitamos el repositorio de Openstack

```
# apt-get install software-properties-common

# add-apt-repository cloud-archive:mitaka

# apt-get update && apt-get dist-upgrade
```

Instalamos el cliente de Openstack:

```
# apt-get install python-openstackclient
```

MariaDB

Instalación y configuración de MariaDB

Instalar los paquetes necesarios en el nodo controlador (ragnar):

```
# apt-get install mariadb-server python-pymysql
```

Crear y editar el fichero de configuración de mariadb para openstack:

```
# nano /etc/mysql/conf.d/openstack.cnf

[mysqld]


bind-address = 10.0.0.11

default-storage-engine = innodb

innodb_file_per_table

collation-server = utf8_general_ci

character-set-server = utf8
```

Reiniciar el servicio:

```
# service mysql restart
```

RabbitMQ

RabbitMQ es un software de negociación de mensajes. Implementa el estandar AMQP (Advanced Message Queuing Protocol).

El proyecto RabbitMQ consta de diferentes partes:

- El servidor de intercambio RabbitMQ en sí mismo.
- Pasarelas para los protocolos HTTP, XMPP y STOMP.
- Bibliotecas de clientes para Java y el framework .NET. (Bibliotecas similares para otros lenguajes se encuentran disponibles por parte de otros proveedores).
- El plugin Shovel (pala) que se encarga de copiar (replicar) mensajes desde un corredor de mensajes a otros.

Instalación y configuración

Instalar RabbitMQ en el nodo controllador (ragnar):

```
# apt-get install rabbitmq-server
```

Añadir el usuario de openstack:

```
# rabbitmqctl add_user openstack asd1234
```

Permitir el acceso a la configuración, lectura y escritura, al usuario openstack:

```
# rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

Keystone

Keystone es el servicio de identidad utilizado por OpenStack para la autenticación, este sistema de autenticación se realiza mediante proyectos asignados a los usuarios. Soporta varias formas de autenticación: tokens, usuario y contraseña.

Instalación y configuración

Creamos la base de datos keystone y el usuario y le otorgamos los privilegios correspondientes:

```
# mysql -u root -p

MariaDB [(none)]> create database keystone;

Query OK, 1 row affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO
'keystone'@'localhost' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.02 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO
'keystone'@'%' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> exit

Bye
```

Deshabilitamos el inicio automático del servicio:

```
# echo "manual" > /etc/init/keystone.override
```

Instalamos los paquetes necesarios:

```
# apt-get install keystone apache2 libapache2-mod-wsgi
```

Generamos un token para el usuraio de administración:

```
# openssl rand -hex 10

46d65a3892395c17c1d7
```

Editamos el fichero de configuración de keystone:

```
# nano /etc/keystone/keystone.conf

[DEFAULT]

admin_token = 46d65a3892395c17c1d7

[database]

connection = mysql+pymysql://keystone:asd1234@odin/keystone

[token]

provider = fernet
```

Poblamos la base de datos de keystone:

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

Comprobamos que se han creado las tablas correctamente:

```
MariaDB [(none)]> use keystone
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [keystone]> show tables;
+------------------------+
| Tables_in_keystone     |
+------------------------+
| access_token           |
| assignment             |
| config_register        |
| consumer               |
| credential             |
| domain                 |
| endpoint               |
| endpoint_group         |
| federated_user         |
| federation_protocol    |
| group                  |
| id_mapping             |
| identity_provider      |
| idp_remote_ids         |
| implied_role           |
| local_user             |
| mapping                |
| migrate_version        |
| password               |
| policy                 |
| policy_association     |
| project                |
| project_endpoint       |
| project_endpoint_group |
| region                 |
| request_token          |
| revocation_event       |
| role                   |
| sensitive_config       |
| service                |
| service_provider       |
| token                  |
| trust                  |
| trust_role             |
| user                   |
| user_group_membership  |
| whitelisted_config     |
+------------------------+
37 rows in set (0.01 sec)
```

Inicializamos fernet keys:

```
# keystone-manage fernet_setup --keystone-user keystone
--keystone-group keystone
```

Configuración del servicio de apache:

Editar el fichero de configuración de apache y añadimos el ServerName:

```
# nano /etc/apache2/apache2.conf

ServerName ragnar
```

Crear los virtualhost con el siguiente contenido:

```
# nano /etc/apache2/sites-available/wsgi-keystone.conf

Listen 5000

Listen 35357

<VirtualHost *:5000>

    WSGIDaemonProcess keystone-public processes=5 threads=1
user=keystone group=keystone display-name=%{GROUP}

    WSGIProcessGroup keystone-public

    WSGIScriptAlias / /usr/bin/keystone-wsgi-public

    WSGIApplicationGroup %{GLOBAL}

    WSGIPassAuthorization On

    ErrorLogFormat "%{cu}t %M"

    ErrorLog /var/log/apache2/keystone.log

    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>

        Require all granted

    </Directory>

</VirtualHost>

<VirtualHost *:35357>

    WSGIDaemonProcess keystone-admin processes=5 threads=1
user=keystone group=keystone display-name=%{GROUP}

    WSGIProcessGroup keystone-admin

    WSGIScriptAlias / /usr/bin/keystone-wsgi-admin

    WSGIApplicationGroup %{GLOBAL}

    WSGIPassAuthorization On

    ErrorLogFormat "%{cu}t %M"

    ErrorLog /var/log/apache2/keystone.log

    CustomLog /var/log/apache2/keystone_access.log combined
    <Directory /usr/bin>

        Require all granted

    </Directory>

</VirtualHost>
```

Reiniciamos el servicio:

```
# service apache2 restart
```

Crear el servicio de entidad y la API endpoints

Exportar las variables de entorno necesarias:

```
# export OS_TOKEN=46d65a3892395c17c1d7

# export OS_URL=http://odin:35357/v3

# export OS_IDENTITY_API_VERSION=3
```

Crear el servicio de entidad para keystone:

```
# openstack service create --name keystone --description
"OpenStack Identity" identity

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

| Field       | Value                            |

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

| description | OpenStack Identity               |

| enabled     | True                             |

| id          | 690cf9b8e98648968e0f7a674dcadd0c |

| name        | keystone                         |

| type        | identity                         |

+-------------+----------------------------------+
```

Crear la API endpoints para keystone:

```
# openstack endpoint create --region RegionOne identity public
http://odin:5000/v3
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 1f9bead13ac44112b471a5d0d1db185e |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 690cf9b8e98648968e0f7a674dcadd0c |
| service_name | keystone                         |
| service_type | identity                         |
| url          | http://odin:5000/v3              |
+--------------+----------------------------------+


# openstack endpoint create --region RegionOne identity internal
http://odin:5000/v3
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 752ccfa5250443989b75ca89e63e651a |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 690cf9b8e98648968e0f7a674dcadd0c |
| service_name | keystone                         |
| service_type | identity                         |
| url          | http://odin:5000/v3              |
+--------------+----------------------------------+
# openstack endpoint create --region RegionOne identity admin
http://odin:35357/v3

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

| Field        | Value                            |

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

| enabled      | True                             |

| id           | 62cbe72872e442b2ae28cbe0e06d2ca2 |

| interface    | admin                            |

| region       | RegionOne                        |

| region_id    | RegionOne                        |

| service_id   | 690cf9b8e98648968e0f7a674dcadd0c |

| service_name | keystone                         |

| service_type | identity                         |

| url          | http://odin:35357/v3             |

+--------------+----------------------------------+
```

Creamos los dominios, projectos, usuarios y roles

Creamos el dominio por defecto:

```
# openstack domain create --description "Default Domain" default

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

| Field       | Value                            |

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

| description | Default Domain                   |

| enabled     | True                             |

| id          | ade03a091f6d44e28240b9e11716a127 |

| name        | default                          |

+-------------+----------------------------------+
```

Creamos un proyecto para la administración, un usuario y un rol para las operaciones de administración:

- Creamos el proyecto de administración:

```
# openstack project create --domain default --description "Admin
Project" admin

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

| Field       | Value                            |

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

| description | Admin Project                    |

| domain_id   | ade03a091f6d44e28240b9e11716a127 |

| enabled     | True                             |

| id          | b5902b7f2fac44d2bfec9f71422340fb |

| is_domain   | False                            |

| name        | admin                            |

| parent_id   | ade03a091f6d44e28240b9e11716a127 |

+-------------+----------------------------------+
```

- Creamos el usuario de administración:

```
# openstack user create --domain default --password-prompt admin

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

| Field     | Value                            |

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

| domain_id | ade03a091f6d44e28240b9e11716a127 |

| enabled   | True                             |

| id        | a3f4b00c29da42ae8d3621d09ccc0628 |

| name      | admin                            |

+-----------+----------------------------------+
```

- Creamos el rol de administración:

```
# openstack role create admin
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | None                             |
| id        | 413da2909bb541a1967010c4367356b9 |
| name      | admin                            |
+-----------+----------------------------------+
```

- Añadimos el rol de administración al proyecto y el usuraio de administración:

```
# openstack role add --project admin --user admin admin
```

Crear un proyecto servicio para todos los servicios que utilizara un usuario único para cada uno que se creará cunado instalemos el servicio:

```
# openstack project create --domain default --description "Service
Project" service
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Service Project                  |
| domain_id   | ade03a091f6d44e28240b9e11716a127 |
| enabled     | True                             |
| id          | 51728412bc1644b0889135bb95fc4e47 |
| is_domain   | False                            |
| name        | service                          |
| parent_id   | ade03a091f6d44e28240b9e11716a127 |
+-------------+----------------------------------+
```

Crear un script para exportar las variables de entorno del usuario de administración para luego utilizarlo para instalar el resto de componentes:

```
# nano admin-openrc
#!/bin/bash
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_PASSWORD=asd1234
export OS_AUTH_URL=http://ragnar:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

Glance

Glance es el componente de openstack que se encarga de almacenar las imágenes (plantillas) que posteriormente serán utilizadas para crear las instancias. También almacena las copias de seguridad (snapshots) de las instancias.

Instalación y configuración

Crear la base de datos glance y el usuario y le otorgamos los privilegios correspondientes:

```
# mysql -u root -p
MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.01 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO
'glance'@'localhost' IDENTIFIED BY 'asd1234';
Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%'
IDENTIFIED BY 'asd1234';
Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> flush privileges;
```

Exportar las credenciales de administración:

```
# . admin-openrc
```

Crear las credenciales del servicio:

- Crear el usuario:

```
# openstack user create --domain default --password-prompt glance
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | ade03a091f6d44e28240b9e11716a127 |
| enabled   | True                             |
| id        | 33cc92df9eab46fcb90f7a1935db77dd |
| name      | glance                           |
+-----------+----------------------------------+
```

- Añadir el rol admin al usuario y proyecto del servicio glance:

```
# openstack role add --project service --user glance admin
```

- Crear el servicio de imágenes:

```
# openstack service create --name glance --description "OpenStack
Image" image

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

| Field       | Value                            |

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

| description | OpenStack Image                  |

| enabled     | True                             |

| id          | 14f68c6bc28e45f6a4ee8eaa8f6898b2 |

| name        | glance                           |

| type        | image                            |

+-------------+----------------------------------+
```

Crear la API endpoints:

```
# openstack endpoint create --region RegionOne image public
http://odin:9292

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

| Field        | Value                            |

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

| enabled      | True                             |

| id           | 311500406ef24075a79c7df2573578f4 |

| interface    | public                           |

| region       | RegionOne                        |

| region_id    | RegionOne                        |

| service_id   | 14f68c6bc28e45f6a4ee8eaa8f6898b2 |

| service_name | glance                           |

| service_type | image                            |

| url          | http://odin:9292                 |

+--------------+----------------------------------+
# openstack endpoint create --region RegionOne image internal
http://odin:9292
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 4d1b8725e4794ee59470e4d3243dcd4f |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 14f68c6bc28e45f6a4ee8eaa8f6898b2 |
| service_name | glance                           |
| service_type | image                            |
| url          | http://odin:9292                 |
+--------------+----------------------------------+
# openstack endpoint create --region RegionOne image admin
http://odin:9292
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 3e7655f1d4a44ddda082143972f8d5f7 |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 14f68c6bc28e45f6a4ee8eaa8f6898b2 |
| service_name | glance                           |
| service_type | image                            |
| url          | http://odin:9292                 |
+--------------+----------------------------------+
```

Instalar los paquetes necesarios:

```
# apt-get install glance
```

Editamos el fichero glance-api.conf :

```
# nano /etc/glance/glance-api.conf
[database]
connection = mysql+pymysql://glance:asd1234@odin/glance

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = glance
password = asd1234

[paste_deploy]
flavor = keystone

[glance_store]
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```

Editamos el fichero glance-registry.conf:

```
# nano /etc/glance/glance-registry.conf
[database]
connection = mysql+pymysql://glance:asd1234@odin/glance

[keystone_authtoken]

auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = glance
password = asd1234

[paste_deploy]
flavor = keystone
```

Poblamos la base de datos del servicio de imágenes:

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```

Reiniciamos los servicios de glance:

```
# service glance-registry restart
# service glance-api restart
```

Probamos a subir una imagen:

```
#openstack image create "cirros" --file cirros-0.3.4-x86_64-
disk.img --disk-format qcow2 --container-format bare --public
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | ee1eca47dc88f4879d8a229cc70a07c6                     |
| container_format | bare                                                 |
| created_at       | 2016-06-08T18:49:55Z                                 |
| disk_format      | qcow2                                                |
| file             | /v2/images/997241d0-1b85-4acb-9861-ed6265b23c96/file |
| id               | 997241d0-1b85-4acb-9861-ed6265b23c96                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | cirros                                               |
| owner            | b5902b7f2fac44d2bfec9f71422340fb                     |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 13287936                                             |
| status           | active                                               |
| tags             |                                                      |
| updated_at       | 2016-06-08T18:49:55Z                                 |
| virtual_size     | None                                                 |
| visibility       | public                                               |
+------------------+------------------------------------------------------+
```

Comprobamos que se ha subido la imagen:

```
# openstack image list
+--------------------------------------+--------+--------+
| ID                                   | Name   | Status |
+--------------------------------------+--------+--------+
| 997241d0-1b85-4acb-9861-ed6265b23c96 | cirros | active |
+--------------------------------------+--------+--------+
```

Nova

Nova es un controlador de estructura cloud computing, que es la parte principal de un sistema de IaaS. Se puede implementar sobre varios sistemas de virtualización como Xen y KVM, también es posible utilizar contenedores como LXC. En este caso obtaremos por utilizar KVM para la virtualización de las instancias.

Instalación y configuración nodo controlador

Crear la base de datos nova y nova_api y los usuario y le otorgamos los privilegios correspondientes:

```
# mysql -u root -p

MariaDB [(none)]> CREATE DATABASE nova_api;

Query OK, 1 row affected (0.00 sec)


MariaDB [(none)]> CREATE DATABASE nova;

Query OK, 1 row affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO
'nova'@'localhost' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%'
IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO
'nova'@'localhost' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%'
IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)
```

Exportar las credenciales de administración:

```
# . admin-openrc
```

Crear las credenciales del servicio:

- Crear el usuario:

```
# openstack user create --domain default --password-prompt nova
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | ade03a091f6d44e28240b9e11716a127 |
| enabled   | True                             |
| id        | 899c4ae6bb32404f9a3d75f432336722 |
| name      | nova                             |
+-----------+----------------------------------+
```

- Añadir el rol admin al usuario y proyecto del servicio nova:

```
# openstack role add --project service --user nova admin
```

- Crear el servicio de entidad:

```
# openstack service create --name nova --description "OpenStack
Compute" compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Compute                |
| enabled     | True                             |
| id          | 2dece252f61b4052b31235ade22eaa83 |
| name        | nova                             |
| type        | compute                          |
+-------------+----------------------------------+
```

Crear la API endpoints del servicio de computo:

```
# openstack endpoint create --region RegionOne compute public
http://odin:8774/v2.1/%\(tenant_id\)s

+--------------+--------------------------------------+
| Field        | Value                                |
+--------------+--------------------------------------+
| enabled      | True                                 |
| id           | 2c98c91001854fd5b16f3fa93078e9eb     |
| interface    | public                               |
| region       | RegionOne                            |
| region_id    | RegionOne                            |
| service_id   | 2dece252f61b4052b31235ade22eaa83     |
| service_name | nova                                 |
| service_type | compute                              |
| url          | http://odin:8774/v2.1/%(tenant_id)s |
+--------------+--------------------------------------+
# openstack endpoint create --region RegionOne compute internal
http://odin:8774/v2.1/%\(tenant_id\)s

+--------------+--------------------------------------+
| Field        | Value                                |
+--------------+--------------------------------------+
| enabled      | True                                 |
| id           | bdf94e5dc006488c9caf6865e264fb02     |
| interface    | internal                             |
| region       | RegionOne                            |
| region_id    | RegionOne                            |
| service_id   | 2dece252f61b4052b31235ade22eaa83     |
| service_name | nova                                 |
| service_type | compute                              |
| url          | http://odin:8774/v2.1/%(tenant_id)s |
+--------------+--------------------------------------+
# openstack endpoint create --region RegionOne compute admin
http://odin:8774/v2.1/%\(tenant_id\)s

+--------------+--------------------------------------+
| Field        | Value                                |
+--------------+--------------------------------------+
| enabled      | True                                 |
| id           | 3b8d7d239c7c46cbb43290f2fb092107     |
| interface    | admin                                |
| region       | RegionOne                            |
| region_id    | RegionOne                            |
| service_id   | 2dece252f61b4052b31235ade22eaa83     |
| service_name | nova                                 |
| service_type | compute                              |
| url          | http://odin:8774/v2.1/%(tenant_id)s |
+--------------+--------------------------------------+
```

Instalar los paquetes necesarios:

```
# apt-get install nova-api nova-conductor nova-consoleauth nova-
novncproxy nova-scheduler
```

Editamos el fichero de configuración /etc/nova/nova.conf:

- Añadir lo siguiente para activar la api de metadatos y computo:

```
[DEFAULT]
enabled_apis = osapi_compute,metadata
```

- Añadir lo siguiente para configurar los conectores de la base de datos:

```
[api_database]
connection = mysql+pymysql://nova:asd1234@odin/nova_api


[database]
connection = mysql+pymysql://nova:asd1234@odin/nova
```

- Añadir lo siguiente para configurar RabbitMQ:

```
[DEFAULT]
rpc_backend = rabbit

[oslo_messaging_rabbit]
rabbit_host = odin
rabbit_userid = openstack
rabbit_password = asd1234
```

- Añadir lo siguiente para configurar keystone:

```
[DEFAULT]
auth_strategy = keystone

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = asd1234
```

- Añadir lo siguiente para configurar la ip:

```
[DEFAULT]
my_ip = 10.0.0.21
```

- Añadir lo siguiente para configurar el servicio de red:

```
[DEFAULT]
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

- Añadir lo siguiente para configurar para configurar el proxy de VNC para utilizar la dirección IP de gestión del nodo controlador:

```
[vnc]

vncserver_listen = $my_ip

vncserver_proxyclient_address = $my_ip
```

- Añadir lo siguiente para configurar la localización del servicio de imagenes:

```
[glance]

api_servers = http://odin:9292
```

- Añadir lo siguiente para configurar la ruta de bloqueo:

```
[oslo_concurrency]

lock_path = /var/lib/nova/tmp
```

Poblamos las bases de datos del servicio de computo:

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

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

Reiniciamos los servicios:

```
# service nova-api restart

# service nova-consoleauth restart

# service nova-scheduler restart

# service nova-conductor restart

# service nova-novncproxy restart
```

Instalación y configuración nodo de computo

Instalar los paquetes necesarios:

```
# apt-get install nova-compute
```

Editamos el fichero de configuración /etc/nova/nova.conf:

- Añadir lo siguiente para configurar RabbitMQ:

```
[DEFAULT]
rpc_backend = rabbit

[oslo_messaging_rabbit]
rabbit_host = odin
rabbit_userid = openstack
rabbit_password = asd1234
```

- Añadir lo siguiente para configurar keystone:

```
[DEFAULT]
auth_strategy = keystone

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = asd1234
```

- Añadir lo siguiente para configurar la ip:

```
[DEFAULT]
my_ip = 10.0.0.22
```

- Añadir lo siguiente para configurar el servicio de red:

```
[DEFAULT]
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

- Añadir lo siguiente para configurar para configurar el proxy de VNC para utilizar la dirección IP de geestión del nodo controlador:

```
[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://odin:6080/vnc_auto.html
```

- Añadir lo siguiente para configurar la localización del servicio de imagenes:

```
[glance]
api_servers = http://odin:9292
```

- Añadir lo siguiente para configurar la ruta de bloqueo:

```
[oslo_concurrency]
lock_path = /var/lib/nova/tmp
```

Reiniciar el servicio:

```
# service nova-compute restart
```

Listamos los componentes de servicio de computo para verificar el funcionamiento y el registro de cada proceso:

```
# openstack compute service list
+----+------------------+------+----------+---------+-------+----------------------------+
| Id | Binary           | Host | Zone     | Status  | State | Updated At                 |
+----+------------------+------+----------+---------+-------+----------------------------+
|  1 | nova-consoleauth | odin | internal | enabled | up    | 2016-06-09T21:36:24.000000 |
|  2 | nova-scheduler   | odin | internal | enabled | up    | 2016-06-09T21:36:21.000000 |
|  3 | nova-conductor   | odin | internal | enabled | up    | 2016-06-09T21:36:24.000000 |
|  4 | nova-compute     | thor | nova     | enabled | up    | 2016-06-09T21:36:27.000000 |
+----+------------------+------+----------+---------+-------+----------------------------+
```

Neutron

Neutron es un sistema para la gestión de redes y direcciones IP. Asegura que la red no presente el problema del cuello de botella o el factor limitante en un despliegue en la nube y ofrece a los usuarios un autoservicio real, incluso a través de sus configuraciones de red.

Neutron gestiona todas las facetas de redes para la infraestructura de red virtual y los aspectos de la capa de acceso de la infraestructura de red física. Neutron permite a los usuarios crear topologías de redes virtuales avanzadas que pueden incluir servicios tales como cortafuegos, balanceadores de carga, etc.

Instalación y configuración nodo controlador

Crear la base de datos neutron y el usuario y le otorgamos los privilegios correspondientes:

```
# mysql -u root -p

MariaDB [(none)]> CREATE DATABASE neutron;

Query OK, 1 row affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO
'neutron'@'localhost' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO
'neutron'@'%' IDENTIFIED BY 'asd1234';

Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)
```

Exportar las credenciales de administración:

```
# . admin-openrc
```

Crear las credenciales del servicio:

- Crear el usuario:

```
# openstack user create --domain default --password-prompt neutron

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

| Field     | Value                            |

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

| domain_id | ade03a091f6d44e28240b9e11716a127 |

| enabled   | True                             |

| id        | 912b8e4fabf547168f4a0d84eb6eb466 |

| name      | neutron                          |

+-----------+----------------------------------+
```

- Añadir el rol admin al usuario y proyecto del servicio neutron:

```
# openstack role add --project service --user neutron admin
```

- Crear el servicio de entidad de neutron:

```
# openstack service create --name neutron --description "OpenStack
Networking" network

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

| Field       | Value                            |

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

| description | OpenStack Networking             |

| enabled     | True                             |

| id          | 7b296719e7f743f4a268ab46999067a6 |

| name        | neutron                          |

| type        | network                          |

+-------------+----------------------------------+
```

Crear la API endpoints del servicio de red:

```
# openstack endpoint create --region RegionOne network public
http://odin:9696

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

| Field        | Value                            |

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

| enabled      | True                             |

| id           | 653f71d6424146eca3cf968d83c9bf69 |

| interface    | public                           |

| region       | RegionOne                        |

| region_id    | RegionOne                        |

| service_id   | 7b296719e7f743f4a268ab46999067a6 |

| service_name | neutron                          |

| service_type | network                          |

| url          | http://odin:9696                 |

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

# openstack endpoint create --region RegionOne network internal
http://odin:9696

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

| Field        | Value                            |

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

| enabled      | True                             |

| id           | f906da04d96e43a78a1a3b4796ef222c |

| interface    | internal                         |

| region       | RegionOne                        |

| region_id    | RegionOne                        |

| service_id   | 7b296719e7f743f4a268ab46999067a6 |

| service_name | neutron                          |

| service_type | network                          |

| url          | http://odin:9696                 |

+--------------+----------------------------------+
# openstack endpoint create --region RegionOne network admin
http://odin:9696

+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | bf61670050134425aa22fd8487e0263f |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 7b296719e7f743f4a268ab46999067a6 |
| service_name | neutron                          |
| service_type | network                          |
| url          | http://odin:9696                 |
+--------------+----------------------------------+
```

Instalar los paquetes necesarios:

```
# apt-get install neutron-server neutron-plugin-ml2 neutron-
linuxbridge-agent neutron-l3-agent neutron-dhcp-agent neutron-
metadata-agent
```

Editar el fichero de configuración neutron.conf:

```
# nano /etc/neutron/neutron.conf
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
rpc_backend = rabbit
auth_strategy = keystone
notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
global_physnet_mtu = 9000

[database]
connection = mysql+pymysql://neutron:asd1234@odin/neutron

[oslo_messaging_rabbit]
rabbit_host = odin
rabbit_userid = openstack
rabbit_password = asd1234

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = asd1234
```

```
[nova]
auth_url = http://odin:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = nova
password = asd1234
```

Editar el fichero ml2_conf.ini:

```
# nano /etc/neutron/plugins/ml2/ml2_conf.ini
[ml2]
type_drivers = flat,vxlan
tenant_network_types = vxlan
mechanism_drivers = linuxbridge,l2population
extension_drivers = port_security
path_mtu = 9000

[ml2_type_flat]
flat_networks = provider

[ml2_type_vxlan]
vni_ranges = 1:1000

[securitygroup]
enable_ipset = True
```

Editar el fichero linuxbridge_agent.ini:

```
# nano /etc/neutron/plugins/ml2/linuxbridge_agent.ini
[DEFAULT]
[linux_bridge]
physical_interface_mappings = provider:eth1


[securitygroup]
enable_security_group = True
firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallDriver


[vxlan]
enable_vxlan = true
local_ip = 10.0.0.21
l2_population = True
```

Editar el fichero l3_agent.ini:

```
# nano /etc/neutron/l3_agent.ini
[DEFAULT]
interface_driver =
neutron.agent.linux.interface.BridgeInterfaceDriver
external_network_bridge =
```

Editar el fichero dhcp_agent.ini:

```
# nano /etc/neutron/dhcp_agent.ini
[DEFAULT]
interface_driver =
neutron.agent.linux.interface.BridgeInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = True
```

Editar el fichero metadata_agent.ini:

```
# nano /etc/neutron/metadata_agent.ini

[DEFAULT]

nova_metadata_ip = odin

metadata_proxy_shared_secret = asd1234
```

Editamos el fichero nova.conf:

```
# nano /etc/nova/nova.conf

[neutron]

url = http://odin:9696

auth_url = http://odin:35357

auth_type = password

project_domain_name = default

user_domain_name = default

region_name = RegionOne

project_name = service

username = neutron

password = asd1234


service_metadata_proxy = True

metadata_proxy_shared_secret = asd1234
```

Poblamos la base de datos neutron:

```
# 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
```

Reiniciamos los servicios:

```
# service nova-api restart

# service neutron-server restart

# service neutron-linuxbridge-agent restart

# service neutron-dhcp-agent restart

# service neutron-metadata-agent restart

# service neutron-l3-agent restart
```

Instalación y configuración nodo computo

Instalamos los paquetes necesarios:

```
# apt-get install neutron-linuxbridge-agent
```

Editamos el fichero neutron.conf (comentar todas las opciones de conexión con la base de datos):

```
# nano /etc/neutron/neutron.conf
[DEFAULT]
rpc_backend = rabbit
auth_strategy = keystone

[oslo_messaging_rabbit]
rabbit_host = odin
rabbit_userid = openstack
rabbit_password = asd1234

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = asd1234
```

Editamos el fichero linuxbridge_agent.ini:

```
# nano /etc/neutron/plugins/ml2/linuxbridge_agent.ini
[DEFAULT]
[linux_bridge]
physical_interface_mappings = provider:eth1


[securitygroup]
firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
enable_security_group = True


[vxlan]
enable_vxlan = True
local_ip = 10.0.0.22
l2_population = True
```

Editamos el fichero nova.conf:

```
# nano /etc/nova/nova.conf
[neutron]
url = http://odin:9696
auth_url = http://odin:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = asd1234
```

Reiniciamos los servicios:

```
# service nova-compute restart
# service neutron-openvswitch-agent restart
```

Listamos los agentes en el nodo controllador para comprobar el funcionamiento:

```
# neutron agent-list
+--------------------------------------+--------------------+------+-------------------+-------
+----------------+---------------------------+
| id                                   | agent_type         | host | availability_zone | alive |
admin_state_up | binary                    |
+--------------------------------------+--------------------+------+-------------------+-------
+----------------+---------------------------+
| 24c182db-631c-401f-bd42-587f6fc4555d | Linux bridge agent | thor |                   | :-)   |
True           | neutron-linuxbridge-agent |
| 25b8245d-4798-40fd-b7dd-3b277266d948 | L3 agent           | odin | nova              | :-)   |
True           | neutron-l3-agent          |
| 8f0a1b97-8090-4ab6-a234-3ea5a830619b | Linux bridge agent | odin |                   | :-)   |
True           | neutron-linuxbridge-agent |
| 95c4b902-b149-47c8-b82a-f178c792c3c1 | DHCP agent         | odin | nova              | :-)   |
True           | neutron-dhcp-agent        |
| c4167f92-3c08-41d1-9610-16c352e4bf86 | Metadata agent     | odin |                   | :-)   |
True           | neutron-metadata-agent    |
+--------------------------------------+--------------------+------+-------------------+-------
+----------------+---------------------------+
```

Horizon

Horizon proporciona a los administradores y usuarios una interfaz gráfica web para gestionar los diferentes recursos y servicios basados en la nube. Horizon es sólo una forma de interactuar con los recursos de OpenStack.

Instalación y configuración

Instalamos los paquetes necesarios:

```
# apt-get install openstack-dashboard

# apt-get remove --purge openstack-dashboard-ubuntu-theme
```

Editamos el fichero /etc/openstack-dashboard/local_settings.py:

```
# nano /etc/openstack-dashboard/local_settings.py

OPENSTACK_HOST = "odin"

ALLOWED_HOSTS = ['*', ]

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'


CACHES = {

    'default': {

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

         'LOCATION': 'odin:11211',

    }

}

OPENSTACK_HOST = "odin"

OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST

OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True

OPENSTACK_API_VERSIONS = {

    "identity": 3,

    "image": 2,

    "volume": 2,

    "compute": 2,
}
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

Reiniciamos el servicio de apache:

```
# service apache2 restart
```

Cinder

Cinder proporciona dispositivos de almacenamiento a nivel de bloque persistentes para usar con instancias de OpenStack Compute. a gestión Snapshot ofrece una potente funcionalidad para realizar copias de seguridad de los datos guardados en volúmenes de almacenamiento en bloque.

Instalación y configuración

Crear la base de datos nova y nova_api y los usuario y le otorgamos los privilegios correspondientes:

```
# mysql -u root -p
MariaDB [(none)]> CREATE DATABASE cinder;
Query OK, 1 row affected (0.07 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 
'cinder'@'localhost' IDENTIFIED BY 'asd1234';
Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' 
IDENTIFIED BY 'asd1234';
Query OK, 0 rows affected (0.00 sec)


MariaDB [(none)]> flush privileges;
```

Exportar las credenciales de administración:

```
# . admin-openrc
```

Crear las credenciales del servicio:

- Crear el usuario:

```
# openstack user create --domain default --password-prompt cinder
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | ade03a091f6d44e28240b9e11716a127 |
| enabled   | True                             |
| id        | 37932a3ea5224e4ea8bf54ce720497ed |
| name      | cinder                           |
+-----------+----------------------------------+
```

- Añadir el rol admin al usuario y proyecto del servicio cinder:

```
# openstack role add --project service --user cinder admin
```

- Crear el servicio de entidad de cinder y cinderv2:

```
# openstack service create --name cinder --description "OpenStack
Block Storage" volume
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Block Storage          |
| enabled     | True                             |
| id          | 6451109add1a4456abddc1ccfc653d98 |
| name        | cinder                           |
| type        | volume                           |
+-------------+----------------------------------+
# openstack service create --name cinderv2 --description
"OpenStack Block Storage" volumev2
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Block Storage          |
| enabled     | True                             |
| id          | 5fe35fa88765416497b3b94296f56d30 |
| name        | cinderv2                         |
| type        | volumev2                         |
+-------------+----------------------------------+
```

Crear la API endpoints del servicio de cinder:

```
# openstack endpoint create --region RegionOne volume public
http://odin:8776/v1/%\(tenant_id\)s
+--------------+-----------------------------------+
| Field        | Value                             |
+--------------+-----------------------------------+
| enabled      | True                              |
| id           | bde0e074fe104cd6bd03bd33583b92bf  |
| interface    | public                            |
| region       | RegionOne                         |
| region_id    | RegionOne                         |
| service_id   | 6451109add1a4456abddc1ccfc653d98  |
| service_name | cinder                            |
| service_type | volume                            |
| url          | http://odin:8776/v1/%(tenant_id)s |
+--------------+-----------------------------------+
# openstack endpoint create --region RegionOne volume internal
http://odin:8776/v1/%\(tenant_id\)s
+--------------+-----------------------------------+
| Field        | Value                             |
+--------------+-----------------------------------+
| enabled      | True                              |
| id           | 54b84d97a8f14720ac5160cf13e677d5  |
| interface    | internal                          |
| region       | RegionOne                         |
| region_id    | RegionOne                         |
| service_id   | 6451109add1a4456abddc1ccfc653d98  |
| service_name | cinder                            |
| service_type | volume                            |
| url          | http://odin:8776/v1/%(tenant_id)s |
+--------------+-----------------------------------+
# openstack endpoint create --region RegionOne volume admin
http://odin:8776/v1/%\(tenant_id\)s
+--------------+-----------------------------------+
| Field        | Value                             |
+--------------+-----------------------------------+
| enabled      | True                              |
| id           | 25cee74f11c24e8da7ef7cdfea974b73  |
| interface    | admin                             |
| region       | RegionOne                         |
| region_id    | RegionOne                         |
| service_id   | 6451109add1a4456abddc1ccfc653d98  |
| service_name | cinder                            |
| service_type | volume                            |
| url          | http://odin:8776/v1/%(tenant_id)s |
+--------------+-----------------------------------+
# openstack endpoint create --region RegionOne volumev2 public
http://odin:8776/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 901b7b22bd7e409fa499b70fbe987bed        |
| interface    | public                                  |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 5fe35fa88765416497b3b94296f56d30        |
| service_name | cinderv2                                |
| service_type | volumev2                                |
| url          | http://odin:8776/v2/%(tenant_id)s       |
+--------------+-----------------------------------------+
# openstack endpoint create --region RegionOne volumev2 internal
http://odin:8776/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | f95b3c45324d4065bc77b1707e720182        |
| interface    | internal                                |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 5fe35fa88765416497b3b94296f56d30        |
| service_name | cinderv2                                |
| service_type | volumev2                                |
| url          | http://odin:8776/v2/%(tenant_id)s       |
+--------------+-----------------------------------------+
# openstack endpoint create --region RegionOne volumev2 admin
http://odin:8776/v2/%\(tenant_id\)s
+--------------+-----------------------------------+
| Field        | Value                             |
+--------------+-----------------------------------+
| enabled      | True                              |
| id           | 2a963f8c6d42456ea3a3cf53a1139e63  |
| interface    | admin                             |
| region       | RegionOne                         |
| region_id    | RegionOne                         |
| service_id   | 5fe35fa88765416497b3b94296f56d30  |
| service_name | cinderv2                          |
| service_type | volumev2                          |
| url          | http://odin:8776/v2/%(tenant_id)s |
+--------------+-----------------------------------+
```

Instalar los paquetes necesarios:

```
# apt-get install cinder-api cinder-scheduler cinder-volume lvm2
```

Editamos el fichero cinder.conf:

```
# nano /etc/cinder/cinder.conf
[database]
connection = mysql+pymysql://cinder:asd1234@odin/cinder


[DEFAULT]
rpc_backend = rabbit
auth_strategy = keystone
my_ip = 10.0.0.21
enabled_backends = lvm
glance_api_servers = http://odin:9292
```

```
[oslo_messaging_rabbit]
rabbit_host = odin
rabbit_userid = openstack
rabbit_password = asd1234

[keystone_authtoken]
auth_uri = http://odin:5000
auth_url = http://odin:35357
memcached_servers = odin:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = asd1234

[oslo_concurrency]
lock_path = /var/lib/cinder/tmp

[lvm]
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
volume_group = cinder-volumes
iscsi_protocol = iscsi
iscsi_helper = tgtadm
```

Poblamos la base de datos cinder:

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

Creamos un volumen físico en /dev/sdb:

```
# pvcreate /dev/vdb
```

Creamos el grupo de volumenes cinder-volumes:

```
# vgcreate cinder-volumes /dev/vdb
```

Editamos el fichero /etc/lvm/lvm.conf:

```
# nano /etc/lvm/lvm.conf

devices {

filter = [ "a/vdb/", "r/.*/"]
```

Reiniciamos los servicios:

```
# service cinder-scheduler restart

# service cinder-api restart

# service tgt restart

# service cinder-volume restart
```

Configuramos el nodo de computo para usar almacenamiento de bloques:

```
# nano /etc/nova/nova.conf

[cinder]

os_region_name = RegionOne
```

Reiniciamos el servicio:

```
# service nova-api restart
```