Infoblox
CONTROL YOUR NETWORK

Deployment Guide

OpenStack Deployment Guide using Single Server with Infoblox Orchestration

Contents

1 Document Overview

This document explains the deployment steps for OpenStack deployment using the Liberty release on a single Ubuntu 14.04 Server with Heat Orchestration to bring up Infoblox DNS VNFs.

User input Text is shown in **bold**.

Any word in greater than or smaller than sign **<>** is a variable and should be replaced with the choice of the user. For example, if the document says enter <password>, then it means password of your choice. If the document says password without greater-than or smaller-than sign, then it means to write word "password" exactly as it is.

The IP address used in the document is an example and should be replaced by the IP address of the server on which you are installing OpenStack.

2 Configure Repositories and update the packages on Ubuntu

The repository configuration is required on Ubuntu if OS is 14.04 release. The following commands are to be executed as super-user (sudo or root):

apt-get update

apt-get –y dist-upgrade

Once the packages are updated, issue the following command:

add-apt-repository cloud-archive:liberty

Update the packages again:

apt-get update

apt-get –y dist-upgrade

A reboot maybe required if the kernel is updated and can be done by issuing the following command:

reboot now

3. Installation of Support Packages

OpenStack uses a message queue for operations and status information among services. The message queue service typically runs on the controller node. RabbitMQ is one such message queue service. In this guide, we are using RabbitMQ message queue service as it is supported by most distributions.

Install the package using the following command:

apt-get install rabbitmq-server

Add the openstack user:

rabbitmqctl add_user openstack *<password>*

Creating user "openstack" …

…done.

Permit configuration, write, and read access for the openstack user created above:

rabbitmqctl set_permissions openstack '.*' '.*' '.*'
Setting permissions for user "openstack in vhost "/" …
…done.

Installation of MariaDB Server

Install the MariaDB server and related software using the command given below:

apt-get install -y mariadb-server python-pymysql

Create a new file named **mysqld_openstack.cnf** in location **/etc/mysql/conf.d**

and add the following lines:

[mysqld]
bind-address = 0.0.0.0
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8

Restart the mysql service using the following command:

service mysql restart

Edit the following lines in the file **/etc/sysctl.conf to reflect the values as shown below;**

net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0

Apply the updates by issuing the following command:

sysctl -p

Installation of Keystone Service (Identity)

The **Identity** service provides a single point of integration for managing authentication, authorization and service catalog services. It is used by other OpenStack services as a common unified API. When an OpenStack service receives a request from a user, the Identity service is used to verify if the user is authorized to make the request.

The **Identity** service consists of following three components:

- Server: Provides authentication and authorization services using a RESTful API interface.
- Drivers: Integrated to the centralized server. These are used for accessing identity info in repositories external to OpenStack.

- 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.

Prerequisites

Before configuring the OpenStack Identity service, you must create a database and an administration token.

To create the database, complete the following actions:

Use the database access client to connect to the database server as the **root** user:

mysql –u root –p

```
root@tme-os:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 5.5.53-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Create the **keystone** database:

```
mysql> CREATE DATABASE keystone;
Query OK, 1 row affected (0.00 sec)
```

Grant proper access to the keystone database:

```
mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
    ->   IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
    ->   IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)
```

where **Infoblox_1** is the password for keystone service. You can use a different password of your choice.

Exit the database access client using command **quit**.

Generate a random value to use as the administrator token during initial configuration as show below:

```
tme@tme-openstack:~$ openssl rand -hex 10
b582c0060610377e98d4
```

Note: Copy the token that the openssl command generates to a text file. The token appears on the second line, as demonstrated in the above example.

Install and configure components:
To avoid conflicts with other services, disable the keystone service from starting automatically after installation:

echo “manual” > /etc/init/keystone.override

Run the following command to install the packages:

apt-get install keystone apache2 libapache2-mod-wsgi memcached python-memcache

Edit the **/etc/keystone/keystone.conf** file. Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[DEFAULT]
…
admin_token = **ADMIN**
[database]
…
connection = mysql+pymysql://keystone:keystone_dbpass@controller/keystone
[memcache]
…
servers = localhost:11211
[token]
…
provider = uuid
driver = memcache
[revoke]
…
driver = sql

Comment out the following line, if present:

connection = sqlite:////var/lib/keystone/keystone.db

The value of **admin_token** in above config can be left as **ADMIN** or replaced with the value generated from the **openssl rand –hex 10** command**.** The value set for **admin_token** in **keystone.conf** must be used in section 7 for attribute **OS_TOKEN.**

Add a host entry in the **/etc/hosts** file for **controller**. Note: The IP address listed here should match that of your server hosting the controller service. This may be the same IP address for your current server and the following is only an example displaying a configured entry in the hosts file.

```
root@tme-openstack:~# more /etc/hosts
10.60.22.5      controller
```

Populate the Identity service database:

keystone-manage db_sync

```
root@tme-openstack:/etc/keystone# cd
root@tme-openstack:~# keystone-manage db_sync
2017-01-23 11:12:23.135 13067 INFO migrate.versioning.api [-] 43 -> 44...
2017-01-23 11:12:23.305 13067 INFO migrate.versioning.api [-] done
2017-01-23 11:12:23.305 13067 INFO migrate.versioning.api [-] 44 -> 45...
2017-01-23 11:12:23.309 13067 INFO migrate.versioning.api [-] done
2017-01-23 11:12:23.309 13067 INFO migrate.versioning.api [-] 45 -> 46...
2017-01-23 11:12:23.313 13067 INFO migrate.versioning.api [-] done
2017-01-23 11:12:23.313 13067 INFO migrate.versioning.api [-] 46 -> 47...
2017-01-23 11:12:23.316 13067 INFO migrate.versioning.api [-] done
2017-01-23 11:12:23.317 13067 INFO migrate.versioning.api [-] 47 -> 48...
2017-01-23 11:12:23.320 13067 INFO migrate.versioning.api [-] done
```

Installation and configuration of Apache HTTP Server

The following steps describe the installation and configuration of the Apache web server:

Edit the **/etc/apache2/apache.conf** file and add the ServerName line as demonstrated in the example below to set the server name for the controller node:

```
root@tme-openstack:/etc/apache2# more apache2.conf
ServerName controller
# This is the main Apache server configuration file.  It contains the
```

Create the **/etc/apache2/sites-available/wsgi-keystone.conf** file with the following content:

```
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
    <IfVersion >= 2.4>
      ErrorLogFormat "%{cu}t %M"
    </IfVersion>
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </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
    <IfVersion >= 2.4>
      ErrorLogFormat "%{cu}t %M"
    </IfVersion>
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>

</VirtualHost>
```

Enable the identity service virtual hosts by issuing the command below:

```
# ln -s /etc/apache2/sites-available/wsgi-keystone.conf /etc/apache2/sites-enabled
```

Restart the Apache web server:

service apache2 restart

Create the service entity and API endpoints

A temporary authentication token created previously can be used to initialize the service entity and API endpoint for the identity service. In order to do so, use the following steps:

Configure the authentication token:

export OS_TOKEN=<ADMIN_TOKEN>

Note: Replace the value <ADMIN_TOKEN> in the above example with the value set for **admin_token** attribute in **keystone.conf** file.

Configure the endpoint URL:

export OS_URL=http://controller:35357/v3

Configure the Identity API version:

export OS_IDENTITY_API_VERSION=3

run the following command to install openstack client needed for next commands:

apt-get install -y python-openstackclient

Create the service entity for the Identity service:

openstack service create --name keystone --description “OpenStack Identity” identity

```
root@tme-openstack:~# openstack service create --name keystone --description "OPenstack Identity" identity
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OPenstack Identity               |
| enabled     | True                             |
| id          | 1e3d5e7f6ce14444886a91d84da62f9e |
| name        | keystone                         |
| type        | identity                         |
+-------------+----------------------------------+
```

Note: OpenStack generates these IDs dynamically, so different values are seen in the example command output.

Create the Identity service API endpoints:

openstack endpoint create --region RegionOne identity public http://controller:5000/v2.0

```
root@tme-openstack:~# openstack endpoint create --region RegionOne identity public http://10.60.22.5:5000/v2.0
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | a03f38587c844afa8bacd2b51c12b00e |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 1e3d5e7f6ce14444886a91d84da62f9e |
| service_name | keystone                         |
| service_type | identity                         |
| url          | http://10.60.22.5:5000/v2.0      |
+--------------+----------------------------------+
```

Note: Either **ServerName** or **IP address** can be used in the commands. For **ServerName** to be used, this must be resolvable using either DNS resolution or the hosts file.

openstack endpoint create --region RegionOne identity internal

http://controller:5000/v2.0

```
root@tme-openstack:~# openstack endpoint create --region RegionOne identity internal http://10.60.22.5:5000/v2.0

+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 8530ace61b0246ed91058b2e69826d3d |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 1e3d5e7f6ce14444886a91d84da62f9e |
| service_name | keystone                         |
| service_type | identity                         |
| url          | http://10.60.22.5:5000/v2.0      |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne identity admin http://controller:5000/v2.0

```
root@tme-openstack:~# openstack endpoint create --region RegionOne identity admin http://10.60.22.5:35357/v2.0
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | d2b96e1783924564ba1b6676c232e706 |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 1e3d5e7f6ce14444886a91d84da62f9e |
| service_name | keystone                         |
| service_type | identity                         |
| url          | http://10.60.22.5:35357/v2.0     |
+--------------+----------------------------------+
```

Create projects, users, and roles

The authentication service uses a combination of domains, projects (tenants), users and roles.

For this deployment guide, the default domain is used.

Create the admin project:

openstack project create --domain default --description “Admin Project” admin

```
root@tme-openstack:~# openstack project create --domain default --description "Admin Project" admin
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Admin Project                    |
| domain_id   | default                          |
| enabled     | True                             |
| id          | 5ce1983ead234660aa71375a79c942ed |
| is_domain   | False                            |
| name        | admin                            |
| parent_id   | None                             |
+-------------+----------------------------------+
```

Create the admin user:

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

```
root@tme-openstack:~# openstack user create --domain default --password-prompt admin
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | de25b0b5293944059a0f72a7f31cf08a |
| name      | admin                            |
+-----------+----------------------------------+
```

Create the admin role:

openstack role create admin

```
root@tme-openstack:~# openstack role create admin
+-------+----------------------------------+
| Field | Value                            |
+-------+----------------------------------+
| id    | 516d39325458451c81950c7d7dbaa50f |
| name  | admin                            |
+-------+----------------------------------+
```

Add the admin role to the admin project and user

openstack role add --project admin --user admin admin

The above command does not generate any output.

This guide uses a service project that contains a unique user for each service that is added to the environment. Create the service project

openstack project create --domain default --description "Service Project" service

```
root@tme-openstack:~# openstack project create --domain default --description "Service Project" service
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Service Project                  |
| domain_id   | default                          |
| enabled     | True                             |
| id          | 50de7b83f3684882a3a28ac72a96cb04 |
| is_domain   | False                            |
| name        | service                          |
| parent_id   | None                             |
+-------------+----------------------------------+
```

Regular (non-admin) tasks should use an unprivileged project and user. For this guide, the demo project and user are used. Create the demo project:

openstack project create --domain default --description "Demo Project" demo

```
root@tme-openstack:~# openstack project create --domain default --description "Demo Project" demo
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Demo Project                     |
| domain_id   | default                          |
| enabled     | True                             |
| id          | 2900d41573af4a9494ba4f1d9902f6c8 |
| is_domain   | False                            |
| name        | demo                             |
| parent_id   | None                             |
+-------------+----------------------------------+
```

Create the demo user:

openstack user create --domain default --password-prompt demo

```
root@tme-openstack:~# openstack user create --domain default --password-prompt demo
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | 00abb11550674a40ae2290ab513f8ebd |
| name      | demo                             |
+-----------+----------------------------------+
```

Create the user role:

openstack role create user

```
root@tme-openstack:~# openstack role create user
+-------+----------------------------------+
| Field | Value                            |
+-------+----------------------------------+
| id    | bc3472881db141d69de7f17d9de11982 |
| name  | user                             |
+-------+----------------------------------+
```

Add the user role to the demo project and user:

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

Creating the scripts

Create client environment scripts for the admin and demo projects and users. These scripts are going to be used in the guide to load appropriate credentials for client operations.

In the users' home directory (cd ~), edit (or create a new file if one does not exist already) the **admin-openrc.sh** file and add the following content:

export OS_PROJECT_DOMAIN_ID=default
export OS_USER_DOMAIN_ID=default
export OS_PROJECT_NAME=admin
export OS_TENANT_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

Replace ADMIN_PASS with the password you chose for the admin user in the identity service

Still in the users' home directory, edit (or create a new file if one does not exist already) the **demo-openrc.sh** file and add the following content:

export OS_PROJECT_DOMAIN_ID=default
export OS_USER_DOMAIN_ID=default
export OS_PROJECT_NAME=demo
export OS_TENANT_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

Replace demo_pass with the password you chose for the demo user in the identity service configuration.

To run clients as a specific project and user, simply load the associated client environment script prior to running them. For example, load the admin-openrc.sh file to populate environment variables with the location of the identity service and the admin project and user credentials:

source admin-openrc.sh

To verify that the environment variables have been loaded, run the command: **env**

Request an authentication token:

openstack token issue

```
root@tme-openstack:~# source admin-openrc.sh
root@tme-openstack:~# openstack token issue
+------------+----------------------------------+
| Field      | Value                            |
+------------+----------------------------------+
| expires    | 2017-01-26T01:32:35.928312Z      |
| id         | 4bffc4f542ab42bdad27e48636f477cf |
| project_id | 5ce1983ead234660aa71375a79c942ed |
| user_id    | de25b0b5293944059a0f72a7f31cf08a |
+------------+----------------------------------+
```

Add the Image service

OpenStack uses glance (the Image service) to discover, register, and retrieve virtual machine images. The virtual machine images can be stored in a variety of locations, from simple file systems to object-storage systems like OpenStack Object Storage.

This guide describes configuring the Image Service to use the file back end, which uploads and stores in a directory on the controller node hosting the Image service. By default, this directory is /var/lib/glance/images/. The controller node should have at least several gigabytes of space available in this directory.

The OpenStack Image service includes the following components:

- **glance-api:** Is used for the processing of 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:** Used to store image metadata. Common database server types are supported, such as MySQL or SQLite.
- **Storage repository for image files**: Various repository types are supported, including standard file systems, Object Storage, RADOS block devices, HTTP and Amazon S3. Some repository types are limited to read-only usage.

Use the database access client to connect to the database server as the root user:

mysql -u root -p

Create the glance database and grant proper access to the glance database:

CREATE DATABASE glance;

**GRANT ALL PRIVILGES ON glance.* TO 'glance'@'localhost' **

IDENTIFIED BY '<GLANCE_DBPASS>';

**GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' **

IDENTIFIED BY '<GLANCE_DBPASS>';

NOTE: Replace **GLANCE_DBPASS** with a suitable password.

```
root@tme-openstack:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 5.5.53-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye
```

Exit the database access client.
Source the admin credentials to gain access to the admin-only CLI commands:

source admin-openrc.sh

Create the glance user:

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

```
root@tme-openstack:~# openstack user create --domain default --password-prompt glance
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | 553de674d043459d80928cb69879bea1 |
| name      | glance                           |
+-----------+----------------------------------+
```

Add the admin role to the glance user and service project

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

Note: No output is displayed upon running this command.

Create the glance service entity:

#**openstack service create --name glance --description “OpenStack Image Service” image**

```
root@tme-openstack:~# openstack service create --name glance --description "Openstack Image Service" im
age
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Openstack Image Service          |
| enabled     | True                             |
| id          | 8e075efab8e946a0a6ae388afc671479 |
| name        | glance                           |
| type        | image                            |
+-------------+----------------------------------+
```

Create the Image service API endpoints:

openstack endpoint create --region RegionOne image public http://controller:9292

openstack endpoint create --region RegionOne image internal http://controller:9292

```
root@tme-openstack:~# openstack endpoint create --region RegionOne image public http://10.60.22.5:9292
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | c8d26f3a297a48aa913bff26c59bc7a9 |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 8e075efab8e946a0a6ae388afc671479 |
| service_name | glance                           |
| service_type | image                            |
| url          | http://10.60.22.5:9292           |
+--------------+----------------------------------+
root@tme-openstack:~#
root@tme-openstack:~# openstack endpoint create --region RegionOne image internal http://10.60.22.5:929
2
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | cc85f56d1ac0432d9a5c14b8108ab8ec |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 8e075efab8e946a0a6ae388afc671479 |
| service_name | glance                           |
| service_type | image                            |
| url          | http://10.60.22.5:9292           |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne image admin http://controller:9292

```
root@tme-openstack:~# openstack endpoint create --region RegionOne image admin http://10.60.22.5:9292
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | c32bd0ae61914539837662afae2febe4 |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 8e075efab8e946a0a6ae388afc671479 |
| service_name | glance                           |
| service_type | image                            |
| url          | http://10.60.22.5:9292           |
+--------------+----------------------------------+
```

Install the packages

apt-get install -y glance python-glanceclient

Edit **/etc/glance/glance-api.conf**. Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[database]
…
connection = mysql+pymysql://glance:glance_dbpass@controller/glance
[keystone_authtoken]
…
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service

username = glance
password = glance_pass
[paste_deploy]
…
flavor = keystone
[glance_store]
…
default_store = file
filesystem_store_datadir = /var/lib/glance/images/

Note: Replace **glance_pass** with the password you chose for the glance user in the identity service. Replace **glance_dbpass** with the password you chose for the Image service database.

Edit **/etc/glance/glance-registry.conf**. Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[database]
connection = mysql+pymysql://glance:Infoblox_1@10.60.31.250/glance

[keystone_authtoken]
auth_uri = http://10.60.31.250:5000
auth_url = http://10.60.31.250:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = <glance_pass>

Note: Replace <**glance_pass>** with the password you chose for the glance user in the identity service and replace the IP addresses for your server.

Populate the Image service database:

su –s /bin/sh –c "glance-manage db_sync" glance

```
root@tme-openstack:/etc/glance# su -s /bin/sh -c "glance-manage db_sync" glance
No handlers could be found for logger "oslo_config.cfg"
2017-01-23 16:05:05.497 7250 INFO migrate.versioning.api [-] 0 -> 1...
2017-01-23 16:05:05.505 7250 INFO glance.db.sqlalchemy.migrate_repo.schema [-] creating table images
2017-01-23 16:05:05.528 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.528 7250 INFO migrate.versioning.api [-] 1 -> 2...
2017-01-23 16:05:05.541 7250 INFO glance.db.sqlalchemy.migrate_repo.schema [-] creating table image_pro
perties
2017-01-23 16:05:05.566 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.566 7250 INFO migrate.versioning.api [-] 2 -> 3...
2017-01-23 16:05:05.589 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.589 7250 INFO migrate.versioning.api [-] 3 -> 4...
2017-01-23 16:05:05.598 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.598 7250 INFO migrate.versioning.api [-] 4 -> 5...
2017-01-23 16:05:05.608 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.608 7250 INFO migrate.versioning.api [-] 5 -> 6...
2017-01-23 16:05:05.628 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.628 7250 INFO migrate.versioning.api [-] 6 -> 7...
2017-01-23 16:05:05.637 7250 INFO migrate.versioning.api [-] done
2017-01-23 16:05:05.638 7250 INFO migrate.versioning.api [-] 7 -> 8...
2017-01-23 16:05:05.651 7250 INFO glance.db.sqlalchemy.migrate_repo.schema [-] creating table image_mem
bers
```

Restart the Image services:

service glance-registry restart

service glance-api restart

Verify Image Service operation

Verify the operation of the Image service using CirrOS, a small Linux image.

In each client environment script, configure the image service client to use API version 2.0:

echo "export OS_IMAGE_API_VERSION=2" | tee -a admin-openrc.sh demo-openrc.sh

Source the admin credentials:

source admin-openrc.sh

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 disk format, bare container format and public visibility so all projects can access it:

glance image-create --name “cirros” --file cirros-0.3.4-x86_64-disk.img --disk-format qcow2 --container-format bare --visibility public --progress

```
root@tme-openstack:~# glance image-create --name "cirros" --file cirros-0.3.4-x86_64-disk.img --disk-fo
rmat qcow2 --container-format bare --visibility public --progress
[=============================>] 100%
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | ee1eca47dc88f4879d8a229cc70a07c6     |
| container_format | bare                                 |
| created_at       | 2017-01-24T00:11:42Z                 |
| disk_format      | qcow2                                |
| id               | 395028c8-f643-4f83-b059-720bef2530b9 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | cirros                               |
| owner            | 5ce1983ead234660aa71375a79c942ed     |
| protected        | False                                |
| size             | 13287936                             |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-01-24T00:11:42Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Confirm the upload of the image and validate attributes:

glance image-list

```
root@tme-openstack:~# glance image-list
+--------------------------------------+--------+
| ID                                   | Name   |
+--------------------------------------+--------+
| 395028c8-f643-4f83-b059-720bef2530b9 | cirros |
+--------------------------------------+--------+
```

Add the Compute service

Before installing and configuring the Compute service, code-named nova, you must create a database, service credentials, and the API endpoints.

Use the database access client to connect to the database server as the root user:

mysql –u root –p

Create the nova database and Grant proper access to the nova database:

CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost'

IDENTIFIED BY '<NOVA_DBPASS>';

GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%'

IDENTIFIED BY '<NOVA_DBPASS>';

Replace **NOVA_DBPASS string** with a suitable password.

Exit the database access client.

```
root@tme-openstack:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 47
Server version: 5.5.53-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye
```

Source the admin credentials:

source admin-openrc.sh

Create the nova user: (choose a password of your choice)

openstack user create --domain default --password-prompt nova

```
root@tme-openstack:~# source admin-openrc.sh
root@tme-openstack:~#
root@tme-openstack:~#
root@tme-openstack:~# openstack user create --domain default --password-prompt nova
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | 810080a5460a4301be496a32fe3ee29b |
| name      | nova                             |
+-----------+----------------------------------+
```

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

```
root@tme-openstack:~# openstack role add --project service --user nova admin
root@tme-openstack:~#
root@tme-openstack:~# openstack service create --name nova --description "Openstack Compute" compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Openstack Compute                |
| enabled     | True                             |
| id          | 062db1f1333840a0ae5b6e619da44073 |
| name        | nova                             |
| type        | compute                          |
+-------------+----------------------------------+
```

Create the Compute service API endpoints:

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

```
root@tme-openstack:~# openstack endpoint create --region RegionOne compute public http://10.60.22.5:877
4/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 7fe7eecf3f51498d826ebcf8317c4bf1        |
| interface    | public                                  |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 062db1f1333840a0ae5b6e619da44073        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://10.60.22.5:8774/v2/%(tenant_id)s |
+--------------+-----------------------------------------+
```

 # openstack endpoint create --region RegionOne compute internal http://controller:8774/v2/%\(tenant_id\)s

```
root@tme-openstack:~# openstack endpoint create --region RegionOne compute internal http://10.60.22.5:8
774/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | ab5ddcccd6084296b4d8188a23428de0        |
| interface    | internal                                |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 062db1f1333840a0ae5b6e619da44073        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://10.60.22.5:8774/v2/%(tenant_id)s |
+--------------+-----------------------------------------+
```

 # openstack endpoint create --region RegionOne compute admin http://controller:8774/v2/%\(tenant_id\)s

```
root@tme-openstack:~# openstack endpoint create --region RegionOne compute admin http://10.60.22.5:8774
/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | d7cedbf559e94fb0a89a40435e1cdf7d        |
| interface    | admin                                   |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 062db1f1333840a0ae5b6e619da44073        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://10.60.22.5:8774/v2/%(tenant_id)s |
+--------------+-----------------------------------------+
```

Install the packages:

apt-get install -y nova-compute sysfsutils nova-api nova-cert nova-conductor nova-consoleauth nova-novncproxy nova-scheduler python-novaclient nova-console

Edit **/etc/nova/nova.conf**. Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):
Note: Replace the values highlighted in red with the appropriate IP addresses and passwords for your configuration.

[DEFAULT]

dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
log_dir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
#verbose=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=osapi_compute,metadata
rpc_backend = rabbit
auth_strategy = keystone
my_ip = 10.60.22.5
vnc_enabled = True
vncserver_listen = 10.60.22.5
vncserver_proxyclient_address = 10.60.22.5
novncproxy_base_url = http://10.60.22.5:6080/vnc_auto.html
network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[database]
connection = mysql+pymysql://nova:Infoblox_1@10.60.22.5/nova

[oslo_messaging_rabbit]
rabbit_host = 10.60.22.5
rabbit_userid = openstack
rabbit_password = <rabbit_pass>

[keystone_authtoken]
auth_uri = http://10.60.22.5:5000
auth_url = http://10.60.22.5:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = nova
password = <password>

[glance]

host = 10.60.22.5

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

[neutron]
service_metadata_proxy = True
metadata_proxy_shared_secret = openstack
url = http://10.60.22.5:9696
admin_auth_url = http://10.60.22.5:35357/v2.0
admin_tenant_name = service
admin_username = neutron
admin_password = <password>

auth_uri = http://10.60.31.250:5000
auth_url = http://10.60.31.250:35357
admin_auth_url = http://10.60.31.250:35357/v2.0
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = <password>

Populate the compute database:

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

```
root@tme-openstack:~# su -s /bin/sh -c "nova-manage db sync" nova
No handlers could be found for logger "oslo_config.cfg"
2017-01-24 10:24:12.408 3007 INFO migrate.versioning.api [-] 215 -> 216...
2017-01-24 10:24:14.115 3007 INFO migrate.versioning.api [-] done
2017-01-24 10:24:14.116 3007 INFO migrate.versioning.api [-] 216 -> 217...
2017-01-24 10:24:14.120 3007 INFO migrate.versioning.api [-] done
2017-01-24 10:24:14.120 3007 INFO migrate.versioning.api [-] 217 -> 218...
2017-01-24 10:24:14.125 3007 INFO migrate.versioning.api [-] done
2017-01-24 10:24:14.125 3007 INFO migrate.versioning.api [-] 218 -> 219...
2017-01-24 10:24:14.129 3007 INFO migrate.versioning.api [-] done
2017-01-24 10:24:14.129 3007 INFO migrate.versioning.api [-] 219 -> 220...
2017-01-24 10:24:14.133 3007 INFO migrate.versioning.api [-] done
```

Restart the compute services:

service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart
service nova-compute restart
service nova-console restart

Remove the SQLlite database file if present as this is generally created by default by Ubuntu:

rm -f /var/lib/nova/nova.sqlite

Verify operation of the compute service

Test the Nova installation using the following commands:

Source the admin credentials to gain access to admin-only CLI commands by issuing the following the command:

source admin-openrc.sh

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

nova service-list

```
root@tme-openstack:~# nova service-list
+----+------------------+---------------+----------+---------+-------+----------------------------+----------
-------+
| Id | Binary           | Host          | Zone     | Status  | State | Updated_at                 | Disabled
Reason |
+----+------------------+---------------+----------+---------+-------+----------------------------+----------
-------+
| 1  | nova-cert        | tme-openstack | internal | enabled | up    | 2017-01-27T19:05:25.000000 | -
       |
| 2  | nova-consoleauth | tme-openstack | internal | enabled | up    | 2017-01-27T19:05:26.000000 | -
       |
| 3  | nova-scheduler   | tme-openstack | internal | enabled | up    | 2017-01-27T19:05:26.000000 | -
       |
| 4  | nova-conductor   | tme-openstack | internal | enabled | up    | 2017-01-27T19:05:26.000000 | -
       |
| 5  | nova-console     | tme-openstack | internal | enabled | up    | 2017-01-27T19:05:25.000000 | -
       |
| 6  | nova-compute     | tme-openstack | nova     | enabled | up    | 2017-01-27T19:05:25.000000 | -
       |
+----+------------------+---------------+----------+---------+-------+----------------------------+----------
-------+
```

nova-manage service list

```
root@tme-openstack:~# nova-manage service list
Binary           Host                                 Zone             Status     State Updated_At
nova-cert        tme-openstack                        internal         enabled    :-)   2017-01-27 19:11:45
nova-consoleauth tme-openstack                        internal         enabled    :-)   2017-01-27 19:11:46
nova-scheduler   tme-openstack                        internal         enabled    :-)   2017-01-27 19:11:46
nova-conductor   tme-openstack                        internal         enabled    :-)   2017-01-27 19:11:46
nova-console     tme-openstack                        internal         enabled    :-)   2017-01-27 19:11:46
nova-compute     tme-openstack                        nova             enabled    :-)   2017-01-27 19:11:45
```

List images in the Image service catalog to verify connectivity with the Image service:

nova image-list

```
root@tme-openstack:~# nova image-list
+--------------------------------------+--------+--------+--------+
| ID                                   | Name   | Status | Server |
+--------------------------------------+--------+--------+--------+
| 395028c8-f643-4f83-b059-720bef2530b9 | cirros | ACTIVE |        |
+--------------------------------------+--------+--------+--------+
```

Add the Networking Service

OpenStack Networking (neutron) allows you to create and attach interface devices managed by other OpenStack services to networks.

It includes the following components:

neutron-server
Accepts and routes API requests to the appropriate OpenStack Networking plug-in for action.

OpenStack Networking plug-ins and agents
Plugs and unplugs ports, creates networks or subnets, and provides IP addressing.

Messaging queue
Used by most OpenStack Networking installations to route information between the neutron-server and various agents.

In order to install the networking service, follow the steps below.

Use the database access client to connect to the database server as the root user:

 # mysql –u root –p

Create the neutron database:

CREATE DATABASE neutron;
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost'
IDENTIFIED BY '<NEUTRON_DBPASS>';

GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%'

IDENTIFIED BY '<NEUTRON_DBPASS>';

Replace **NEUTRON_DBPASS** string with a password of your choice.

```
root@tme-openstack:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 48
Server version: 5.5.53-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE neutron;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye
```

Source the admin credentials to gain access to admin-only CLI commands by issuing the following the command;

source admin-openrc.sh

To create the service credentials, complete these steps:

Create the **neutron** user:

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

Select a password of your choice for user neutron on prompt.

```
root@tme-openstack:~# openstack user create --domain default --password-prompt neutron
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | 985c1d769d544296902621b9ad5f77b0 |
| name      | neutron                          |
+-----------+----------------------------------+
```

Add the **admin** role to the **neutron** user:

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

There is no output for the above command.

Create the **neutron** service entity:

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

```
root@tme-openstack:~# openstack service create --name neutron --description "Openstack Networking" network
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Openstack Networking             |
| enabled     | True                             |
| id          | c9f6b428eb4a498f8bb35c8413df40fa |
| name        | neutron                          |
| type        | network                          |
+-------------+----------------------------------+
```

Create the Networking service API endpoints:

openstack endpoint create --region RegionOne network public http://10.60.22.5:9696

```
root@tme-openstack:~# openstack endpoint create --region RegionOne network public http://10.60.22.5:9696
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | fcd10e54770449c1a831c6e1819be411 |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | c9f6b428eb4a498f8bb35c8413df40fa |
| service_name | neutron                          |
| service_type | network                          |
| url          | http://10.60.22.5:9696           |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne network internal http://10.60.22.5:9696

```
root@tme-openstack:~# openstack endpoint create --region RegionOne network internal http://10.60.22.5:9696
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 11a264064f5a4bdb9b6a1af32bbb3356 |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | c9f6b428eb4a498f8bb35c8413df40fa |
| service_name | neutron                          |
| service_type | network                          |
| url          | http://10.60.22.5:9696           |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne network admin http://10.60.22.5:9696

```
root@tme-openstack:~# openstack endpoint create --region RegionOne network admin http://10.60.22.5:9696
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | e10cb1c261974b44b004547950684586 |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | c9f6b428eb4a498f8bb35c8413df40fa |
| service_name | neutron                          |
| service_type | network                          |
| url          | http://10.60.22.5:9696           |
+--------------+----------------------------------+
```

Install the neutron package:

apt-get install -y neutron-server neutron-plugin-openvswitch neutron-plugin-openvswitch-agent neutron-common neutron-dhcp-agent neutron-l3-agent neutron-metadata-agent openvswitch-switch

Edit **/etc/neutron/neutron.conf**. Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[DEFAULT]
core_plugin = ml2
service_plugins = router
rpc_backend = rabbit
auth_strategy = keystone
notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
nova_url = http://10.60.22.5:8774/v2

allow_overlapping_ips = True

[keystone_authtoken]
auth_uri = http://10.60.22.5:5000
auth_url = http://10.60.22.5:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = <password>

[database]
connection = mysql+pymysql://neutron:<neutron_dbpass>@10.60.22.5/neutron

[oslo_messaging_rabbit]
rabbit_host = 10.60.22.5
rabbit_userid = openstack
rabbit_password = <rabbit_pass>

[nova]
auth_url = http://10.60.22.5:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = nova
password = <nova_pass>

[oslo_concurrency]

lock_path = $state_path/lock

Replace **nova_pass** with the password chosen by you for **nova** user. Replace the **rabbit_pass** for the password you chose earlier for **openstack** user. Replace **neutron_dbpass** with the password chosen for user to create neutron database. Update the IP addresses referenced here as appropriate.

Edit **/etc/neutron/plugins/ml2/ml2_conf.ini.** Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[ml2]
type_drivers = flat,vxlan
tenant_network_types = vxlan
mechanism_drivers = openvswitch,l2population

extension_drivers = port_security

[ml2_type_flat]
flat_networks = External

[ml2_type_vxlan]

vni_ranges = 10000:20000

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

enable_ipset = True

Edit **/etc/neutron/l3_agent.ini.** Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[DEFAULT]

debug = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
use_namespaces = True

external_network_bridge =

Note: The value for the **external_network_bridge** property is purposely left blank.

Edit **/etc/neutron/dhcp_agent.ini.** Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[DEFAULT]

interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq

use_namespaces = True

Edit **/etc/neutron/metadata_agent.ini.** Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[DEFAULT]

auth_url = http://10.60.22.5:35357
auth_uri = http://10.60.22.5:5000
auth_region = RegionOne
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%
nova_metadata_ip = 127.0.0.1
metadata_proxy_shared_secret = infoblox

Edit **/etc/neutron/plugins/ml2/openvswitch_agent.ini.** Locate the properties referenced below and edit their corresponding values to match the examples provided here (uncomment lines as necessary):

[ovs]

local_ip = 10.60.22.5
enable_tunneling = True
bridge_mappings = External:br-ex

[agent]
tunnel_types = vxlan

l2_population = True

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

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

Restart Neutron services:

service neutron-server restart
service neutron-dhcp-agent restart
service neutron-metadata-agent restart
service neutron-l3-agent restart

Verify the operation of Network Service

source admin-openrc.sh

neutron agent-list

```
+--------------------------------------+--------------------+--------+-------+----------------+---------------------------+
| id                                   | agent_type         | host   | alive | admin_state_up | binary                    |
+--------------------------------------+--------------------+--------+-------+----------------+---------------------------+
| 41f8b6b5-72c4-4ec2-a484-65369d3949cf | Open vSwitch agent | tme-OS | :-)   | True           | neutron-openvswitch-agent |
| 4bce38aa-7d78-41ab-83cb-e9916f015a55 | L3 agent           | tme-OS | :-)   | True           | neutron-l3-agent          |
| b48857d7-f2d7-4b5f-bc09-0897514cd091 | Metadata agent     | tme-OS | :-)   | True           | neutron-metadata-agent    |
| d6d8f61a-7f4d-4a10-a87b-b77d00038c79 | DHCP agent         | tme-OS | :-)   | True           | neutron-dhcp-agent        |
+--------------------------------------+--------------------+--------+-------+----------------+---------------------------+
```

Add bridges as necessary by using the following commands (typically, only these bridges are needed):

ovs-vsctl add-br br-int
ovs-vsctl add-br br-ex
ovs-vsctl add-br br-tun

Once the bridges are added, the Ethernet interface/IP address of the OpenStack node needs to be moved under bridge br-ex and a default route added pointing to br-ex interface. This is needed so that floating-ips can be used and enable routing between the virtual network created in OpenStack and the external network.

Note: Before completing the following, be sure to read through and understand these steps as connectivity may be lost during the process.

Issue the following command on the console:

ifconfig

Verify the Ethernet interface that is being used for the IP connectivity. In the example provided here, it is **eth0**.

Update the **interfaces** file in **/etc/network/** with the following configuration:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual
up ip link set dev eth0 up
down ip link set dev eth0 down

iface br-ex inet static
        address 10.60.22.5
    netmask 255.255.255.0
        mtu 1550
```

Please replace **10.60.22.5** in the above configuration with your controllers IP.

Next, connect to the physical console (keyboard and mouse or virtual console) as the following command is going to result in loss of network connectivity as the servers IP address is being moved to the (virtual/bridging) br-ex interface. Once connected to the servers' console, run the following command (replacing the IP addresses shown in the example with your servers IP address):

ip addr del 10.60.22.5/24 dev eth0 ; ip addr add 10.60.22.5/24 dev br-ex ; ip link set dev br-ex up ; ovs-vsctl add-port br-ex eth0

From console, add default route to point to **br-ex** interface to restore ssh connectivity:

route add default gw 10.60.22.1 br-ex

Launch an ssh connection from your computer to your controller (Ubuntu server) to verify connectivity.

Run the command **ifconfig** and verify that the IP address is now shown under the **br-ex** interface.

```
root@tme-OS:~# ifconfig
br-ex     Link encap:Ethernet  HWaddr 00:50:56:9c:53:f2
          inet addr:10.60.22.5  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::502d:d5ff:feb4:934e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:651 errors:0 dropped:0 overruns:0 frame:0
          TX packets:413 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:58971 (58.9 KB)  TX bytes:57086 (57.0 KB)

eth0      Link encap:Ethernet  HWaddr 00:50:56:9c:53:f2
          inet6 addr: fe80::250:56ff:fe9c:53f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:463155 errors:0 dropped:11 overruns:0 frame:0
```

TX packets:394399 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:184681019 (184.6 MB) TX bytes:36966282 (36.9 MB)

Note: The above settings are not persistent. If the server is rebooted, these commands may need to be run again to restore network connectivity.

Add the Dashboard

The OpenStack Dashboard, also known as horizon is a web interface that enables cloud administrators and users to manage various OpenStack resources and services.

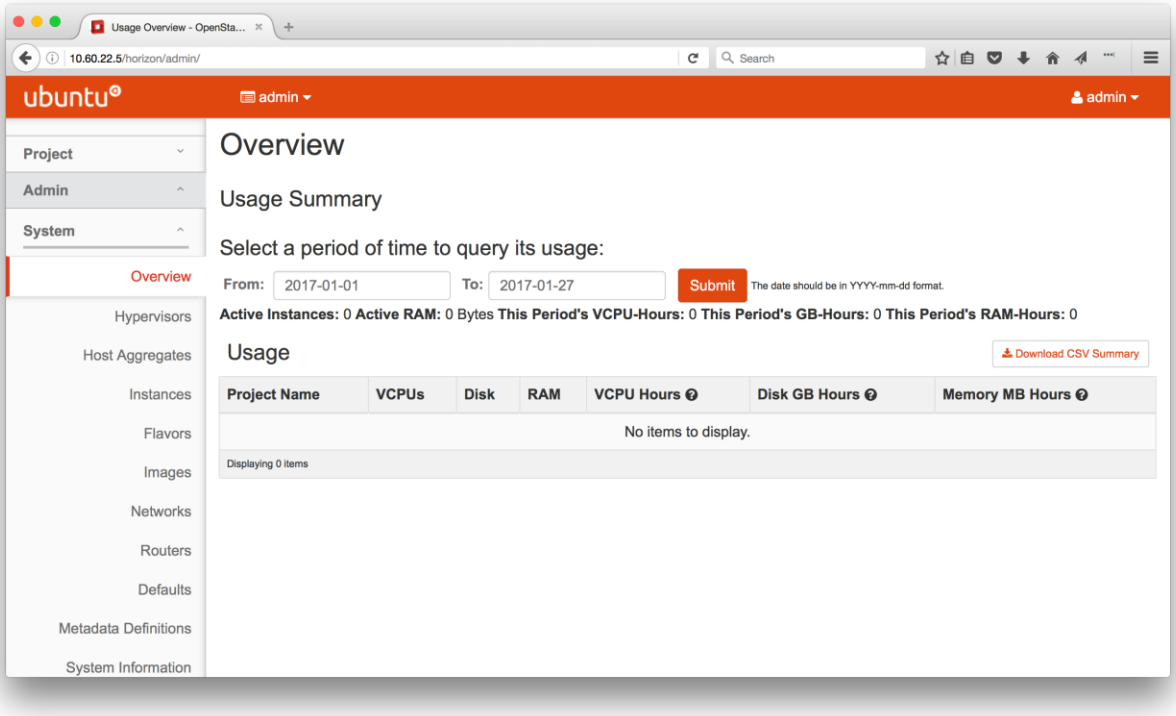
Install OpenStack Web UI using the following command:

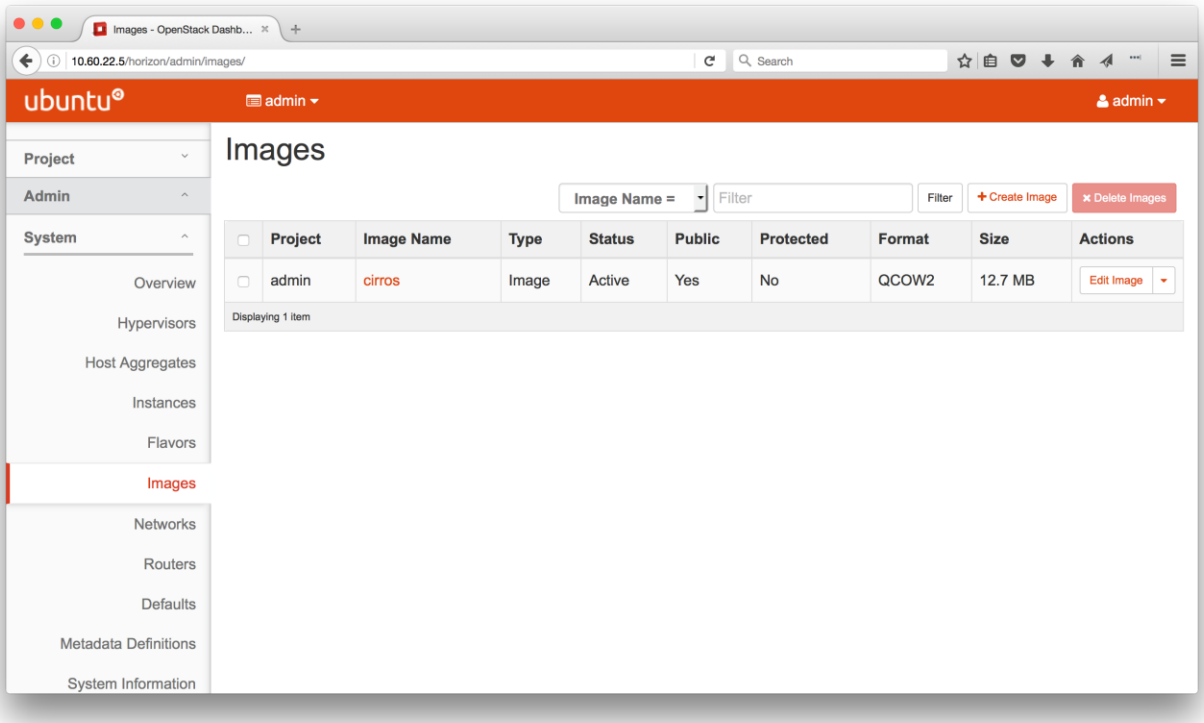
apt-get install -y openstack-dashboard

After installing login using the following credentials

URL : http://<ip-address>/horizon
Username: admin
Password: <password>

Use the password above that you created for user **admin**. Replace the <ip-address> with the IP of your OpenStack server.





Add the Orchestration Service (HEAT)

The Orchestration service provides a template-based orchestration for describing a cloud application by running OpenStack API calls to generate running cloud applications.

The Orchestration service consists of the following components:

heat command-line client
A CLI that communicates with the heat-api to run AWS CloudFormation APIs. End developers can directly use the Orchestration REST API.

heat-api component
An OpenStack-native REST API that processes API requests by sending them to the heat-engine over Remote Procedure Call (RPC).

heat-api-cfn component
An AWS Query API that is compatible with AWS CloudFormation. It processes API requests by sending them to the heat-engine over RPC.

heat-engine
Orchestrates the launching of templates and provides events back to the API consumer.

Complete the following the steps to install the orchestration service on the OpenStack server.

Create database for heat by logging in as admin user to the SQL database:

mysql –u root –p

Create the heat database:

CREATE DATABASE heat;

Grant proper access:

GRANT ALL PRIVILIGES ON heat.* TO ‘heat’@’localhost’

IDENTIFIED BY '<HEAT_DBPASS>';

GRANT ALL PRIVILIGES ON heat.* TO ‘heat’@’%’

IDENTIFIED BY '<HEAT_DBPASS>';

Note: Replace **<HEAT_DBPASS>** with a password of your choice.

```
root@tme-openstack:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11665
Server version: 5.5.54-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE heat;
ERROR 1007 (HY000): Can't create database 'heat'; database exists
MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'localhost'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'%'
    -> IDENTIFIED BY 'Infoblox_1';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye
```

Source the admin credentials to gain access to admin-only CLI commands by issuing the following the command:

source admin-openrc.sh

To create the service credentials, complete these steps:

Create the heat user:

openstack user create --domain default --password-prompt heat

Choose a password for user **heat**.

```
root@tme-openstack:~# source admin-openrc.sh
root@tme-openstack:~# openstack user create --domain default --password-prompt heat
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | default                          |
| enabled   | True                             |
| id        | 5fd095d3e2974a9bbce67dff81fda2a1 |
| name      | heat                             |
+-----------+----------------------------------+
```

Add the admin role to the heat user:

openstack role add --project service --user heat admin

```
root@tme-openstack:~# openstack role add --project service --user heat admin
root@tme-openstack:~#
```

Note: No output is displayed upon successful completion for above command.

Create the heat and heat-cfn service entities:

openstack service create --name heat --description "Orchestration" orchestration

```
root@tme-openstack:~# openstack service create --name heat --description "Orchestration" orchestration
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Orchestration                    |
| enabled     | True                             |
| id          | 79a24a3751fc49ad8e5a6fd31031bf17 |
| name        | heat                             |
| type        | orchestration                    |
+-------------+----------------------------------+
```

openstack service create --name heat-cfn --description "Orchestration" cloudformation

```
root@tme-openstack:~# openstack service create --name heat-cfn --description "Orchestration" cloudformation
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Orchestration                    |
| enabled     | True                             |
| id          | 64f2ec7b84e5446ca6b82495085c976c |
| name        | heat-cfn                         |
| type        | cloudformation                   |
+-------------+----------------------------------+
```

Create the Orchestration service API endpoints:

openstack endpoint create --region RegionOne orchestration public http://10.60.22.5:8004/v1/%\(tenant_id\)s

```
root@tme-openstack:~# openstack endpoint create --region RegionOne orchestration public http://10.60.22.5:8004/v1
/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 16780a08611241c490bfe98f6aed32a5        |
| interface    | public                                  |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 79a24a3751fc49ad8e5a6fd31031bf17        |
| service_name | heat                                    |
| service_type | orchestration                           |
| url          | http://10.60.22.5:8004/v1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

openstack endpoint create --region RegionOne orchestration internal http://10.60.22.5:8004/v1/%\(tenant_id\)s

```
root@tme-openstack:~# openstack endpoint create --region RegionOne orchestration internal http://10.60.22.5:8004/
v1/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 6ddb16e0c84342c084e8884672496982        |
| interface    | internal                                |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 79a24a3751fc49ad8e5a6fd31031bf17        |
| service_name | heat                                    |
| service_type | orchestration                           |
| url          | http://10.60.22.5:8004/v1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

openstack endpoint create --region RegionOne orchestration admin http://10.60.22.5:8004/v1/%\(tenant_id\)s

```
root@tme-openstack:~# openstack endpoint create --region RegionOne orchestration admin http://10.60.22.5:8004/v1/
%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 7b9f5344ae51493bb70040e89af3878c        |
| interface    | admin                                   |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 79a24a3751fc49ad8e5a6fd31031bf17        |
| service_name | heat                                    |
| service_type | orchestration                           |
| url          | http://10.60.22.5:8004/v1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

openstack endpoint create --region RegionOne cloudformation public http://10.60.22.5:8000/v1

```
root@tme-openstack:~# openstack endpoint create --region RegionOne cloudformation public http://10.60.22.5:8000/v
1
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | ef154fe337104709b4b638ae8daf2711 |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 64f2ec7b84e5446ca6b82495085c976c |
| service_name | heat-cfn                         |
| service_type | cloudformation                   |
| url          | http://10.60.22.5:8000/v1        |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne cloudformation internal http://10.60.22.5:8000/v1

```
root@tme-openstack:~# openstack endpoint create --region RegionOne cloudformation internal http://10.60.22.5:8000
/v1
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | f556cc81d3874d1ab0c13d6dae775f47 |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 64f2ec7b84e5446ca6b82495085c976c |
| service_name | heat-cfn                         |
| service_type | cloudformation                   |
| url          | http://10.60.22.5:8000/v1        |
+--------------+----------------------------------+
```

openstack endpoint create --region RegionOne cloudformation admin http://10.60.22.5:8000/v1

```
root@tme-openstack:~# openstack endpoint create --region RegionOne cloudformation admin http://10.60.22.5:8000/v1
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 07fd6e43458a4f0e87585a8ead4f8f1a |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 64f2ec7b84e5446ca6b82495085c976c |
| service_name | heat-cfn                         |
| service_type | cloudformation                   |
| url          | http://10.60.22.5:8000/v1        |
+--------------+----------------------------------+
```

Create the **heat** domain that contains projects and users for stacks:

openstack domain create --description "Stack projects and users" heat

```
root@tme-openstack:~# openstack domain create --description "Stack projects and users" heat
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Stack projects and users         |
| enabled     | True                             |
| id          | 97be581da3394f13a048185c6c54e1a4 |
| name        | heat                             |
+-------------+----------------------------------+
```

Create the **heat_domain_admin** user to manage projects and **users** in the heat domain:

openstack user create --domain heat --password-prompt heat_domain_admin

Choose a password of your choice.

```
root@tme-openstack:~# openstack user create --domain heat --password-prompt heat_domain_admin
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | 97be581da3394f13a048185c6c54e1a4 |
| enabled   | True                             |
| id        | 6dbf485cb285400296acfe28c60432fa |
| name      | heat_domain_admin                |
+-----------+----------------------------------+
```

Add the **admin** role to the **heat_domain_admin** user in the **heat** domain **t**o enable administrative stack management privileges by the **heat_domain_admin** user:

openstack role add --domain heat --user heat_domain_admin admin

```
root@tme-openstack:~# openstack role add --domain heat --user heat_domain_admin admin
root@tme-openstack:~#
```

Note: No output is displayed for successful completion of the above command.

Create the **heat_stack_owner** role:

openstack role create heat_stack_owner

```
root@tme-openstack:~# openstack role create heat_stack_owner
+-------+----------------------------------+
| Field | Value                            |
+-------+----------------------------------+
| id    | 30306e370a7e4d5389c7bc69ed56f29c |
| name  | heat_stack_owner                 |
+-------+----------------------------------+
```

Add the **heat_stack_owner** role to the demo project and user to enable stack management by the demo user:

openstack role add --project demo --user demo heat_stack_owner

```
root@tme-openstack:~# openstack role add --project demo --user demo heat_stack_owner
root@tme-openstack:~#
```

Note: This command has no output.

Create the heat_stack_user role:

openstack role create heat_stack_user

Note: No output is displayed for successful completion of the above command.

```
root@tme-openstack:~# openstack role create heat_stack_user
+-------+----------------------------------+
| Field | Value                            |
+-------+----------------------------------+
| id    | 8aa9a17efa74470aac818af5f058cd7b |
| name  | heat_stack_user                  |
+-------+----------------------------------+
```

Install the orchestration packages:

apt-get install heat-api heat-api-cfn heat-engine python-heatclient

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

[DEFAULT]
rpc_backend = rabbit
heat_metadata_server_url = http://10.60.22.5:8000
heat_waitcondition_server_url = http://10.60.22.5:8000/v1/waitcondition
stack_domain_admin = heat_domain_admin
stack_domain_admin_password = <password>
stack_user_domain_name = heat

[database]
connection = mysql+pymysql://heat:Infoblox_1@10.60.22.5/heat

[keystone_authtoken]
auth_uri = http://10.60.22.5:5000
auth_url = http://10.60.22.5:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = heat
password = <password>

[trustee]
auth_plugin = password
auth_url = http://10.60.22.5:35357
username = heat
password = <password>

user_domain_id = default

[clients_keystone]
auth_uri = http://10.60.22.5:5000

[ec2authtoken]
auth_uri = http://10.60.22.5:5000/v3

Note: Replace **<password>** with the appropriate password for the respective users.

In the example above, **10.60.22.5** is the IP address used for the OpenStack controller node. Replace this with the appropriate IP address for your server.

Populate the Orchestration database:

su -s /bin/sh -c "heat-manage db_sync" heat

```
root@tme-openstack:/etc/heat# su -s /bin/sh -c "heat-manage db_sync" heat
2017-01-27 15:32:37.462 26543 INFO migrate.versioning.api [-] 27 -> 28...
2017-01-27 15:32:37.522 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.522 26543 INFO migrate.versioning.api [-] 28 -> 29...
2017-01-27 15:32:37.553 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.554 26543 INFO migrate.versioning.api [-] 29 -> 30...
2017-01-27 15:32:37.681 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.681 26543 INFO migrate.versioning.api [-] 30 -> 31...
2017-01-27 15:32:37.713 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.714 26543 INFO migrate.versioning.api [-] 31 -> 32...
2017-01-27 15:32:37.790 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.791 26543 INFO migrate.versioning.api [-] 32 -> 33...
2017-01-27 15:32:37.872 26543 INFO migrate.versioning.api [-] done
2017-01-27 15:32:37.872 26543 INFO migrate.versioning.api [-] 33 -> 34...
2017-01-27 15:32:37.892 26543 INFO migrate.versioning.api [-] done
```

Restart the Orchestration services:

service heat-api restart
service heat-api-cfn restart
service heat-engine restart

By default, the Ubuntu packages create an SQLite database.

Because the configuration demonstrated in this guide uses a SQL database server, you can remove the SQLite database file:

rm -f /var/lib/heat/heat.sqlite

Verify Heat Service

To verify the operation of heat service, perform the following steps:

Source the admin credentials to gain access to admin-only CLI commands by issuing the following the command:

source admin-openrc.sh

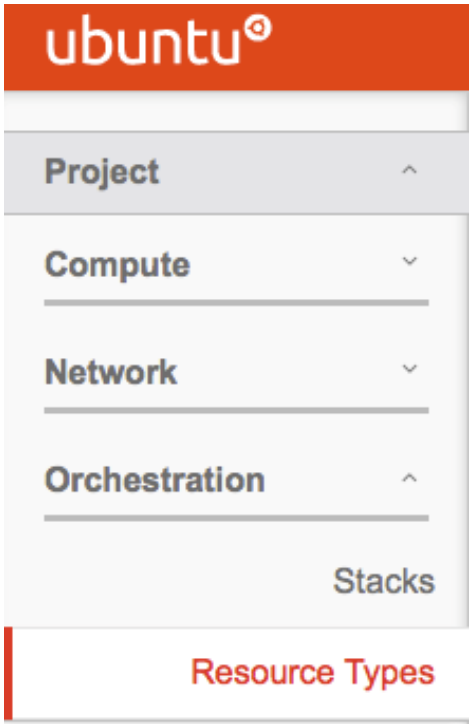
List service components to verify successful launch and registration of each process:

heat service-list

```
root@tme-openstack:~# heat service-list
+---------------+-------------+--------------------------------------+---------------+--------+----------------------------+--------+
| hostname      | binary      | engine_id                            | host          | topic  | updated_at                 | status |
+---------------+-------------+--------------------------------------+---------------+--------+----------------------------+--------+
| tme-openstack | heat-engine | 03a85161-8291-4136-b944-9932e0405aa0 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 14547753-2e39-43d9-81be-291bcabc16ba | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 17762ef9-0ef7-48d4-b2a5-2df7cb99c7ad | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 19caa53f-92d6-4049-816d-9a2f1b5c4e52 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 1d6d2bb4-d369-42bf-9b30-749ce23c121a | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 1dde42e9-7c87-4fe8-a8ef-26e7014a8594 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 1e73c1ca-0fc3-4709-b1b5-15a8eb305abc | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 22685ca6-921a-40ec-a07a-74289cd9dc65 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 26e0f35c-cf14-43dd-bb35-7d16c0dca804 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 29852d3f-1950-4bfa-b4c1-e58453b80b5f | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 2cad62ef-d192-4af5-9f05-c241fb49a348 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 45ad8497-fe9e-4e9e-9c35-618f04100cce | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 5a5a6ad3-9030-4e12-87a6-2fff27042ae1 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 614048ce-7b35-40d8-80a1-2005adffb679 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 6e477d7c-92ff-453c-b65a-928344badb32 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 83a33416-000b-4f83-8175-bb0d04b508ff | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 8b818b53-9815-468f-b127-b8f7f2896f5e | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | 9b3490c2-107c-4b6d-9df2-6fd0e2b3907c | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | a2cf55fe-89a5-45e8-9873-7c440cec200c | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | b5ae02a3-1614-42e0-b50e-6ea49f6cde0d | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | c3cb1304-c582-49aa-9807-937e7da3c229 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | cbcec6de-0458-4556-84d0-ddacccb20898 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | cd63f258-7cee-404d-88f4-9d669c4ae5f5 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | d7231314-16ac-407e-88e3-30738332c9db | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | df9b78b3-2dc9-4a99-8e7f-4d4b1cab57fc | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | e12c17f5-7f44-44b6-b262-15cf96886d9b | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | e2ffba5e-faac-4633-a37b-fa2e9872cfc9 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | ee005c46-638b-48c0-a266-eee26f5819eb | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | ee731c7f-d29e-4287-a3a1-45664292bbb6 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | f776c7e6-c6a1-4328-a6fd-c18c314bad5c | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | fc0a1f00-40b6-4727-9b63-278eb6133741 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
| tme-openstack | heat-engine | ffa70525-ffa2-4ade-a1e3-6ecd7dc08982 | tme-openstack | engine | 2017-02-13T20:34:11.000000 | up     |
+---------------+-------------+--------------------------------------+---------------+--------+----------------------------+--------+
```

Your OpenStack environment now includes Orchestration.

The Horizon Dashboard should now also show the **Orchestration** section in the UI.



Installation of Infoblox Heat Resources

To start the installation of the Infoblox Heat resources, make sure you have “pip” module installed. To install pip, use the following command:

apt-get install python-pip

Use the following command to install Infoblox Heat resources on the controller node:

pip install heat-infoblox==2.0.0

Internet connectivity and working DNS resolution from the Controller node is required in order for this to complete successfully.

Edit **/etc/heat/heat.conf** and complete the following actions:

Add the following line under **[Default]** section:

plugin_dirs = /usr/local/lib/python2.7/dist-packages/heat_infoblox,/usr/lib64/heat,/usr/lib/heat

Add the following section as shown before:

[oslo_concurrency]
lock_path = /home/heat-admin/directory_for_locks

Restart the Orchestration services to apply the above changes in Heat:

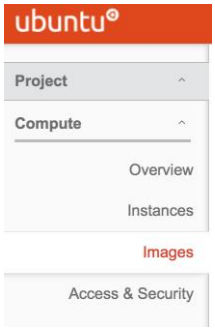
service heat-api restart
service heat-api-cfn restart
service heat-engine restart

The Infoblox Heat resources can be seen from Horizon Dashboard, under **Orchestration** > **Resource Types**

Infoblox::Grid::AnycastLoopback	Infoblox	Grid	AnycastLoopback
Infoblox::Grid::Bgp	Infoblox	Grid	Bgp
Infoblox::Grid::BgpNeighbor	Infoblox	Grid	BgpNeighbor
Infoblox::Grid::HaPair	Infoblox	Grid	HaPair
Infoblox::Grid::Member	Infoblox	Grid	Member
Infoblox::Grid::NameServerGroupMember	Infoblox	Grid	NameServerGroupMember
Infoblox::Grid::Ospf	Infoblox	Grid	Ospf
Infoblox::NetMRI::Job	Infoblox	NetMRI	Job
Infoblox::NetMRI::ManagedResource	Infoblox	NetMRI	ManagedResource

Uploading the vNIOS Image

To upload a vNIOS image, log onto the horizon Dashboard and click on **Images** under **Compute** section.



Click on Create Image on the right-hand side of the page to open the **Create An Image** pop-up window.

In our example, we name the fields as follows:

Name – vnios-802-1420

Description - NIOS image 8.x for 1420 appliance (optional)

Image Source – Select **Image File**

Image File – Click on **Choose File** and select the image to be uploaded

Format – Select QCOW2 – QEMU Emulator

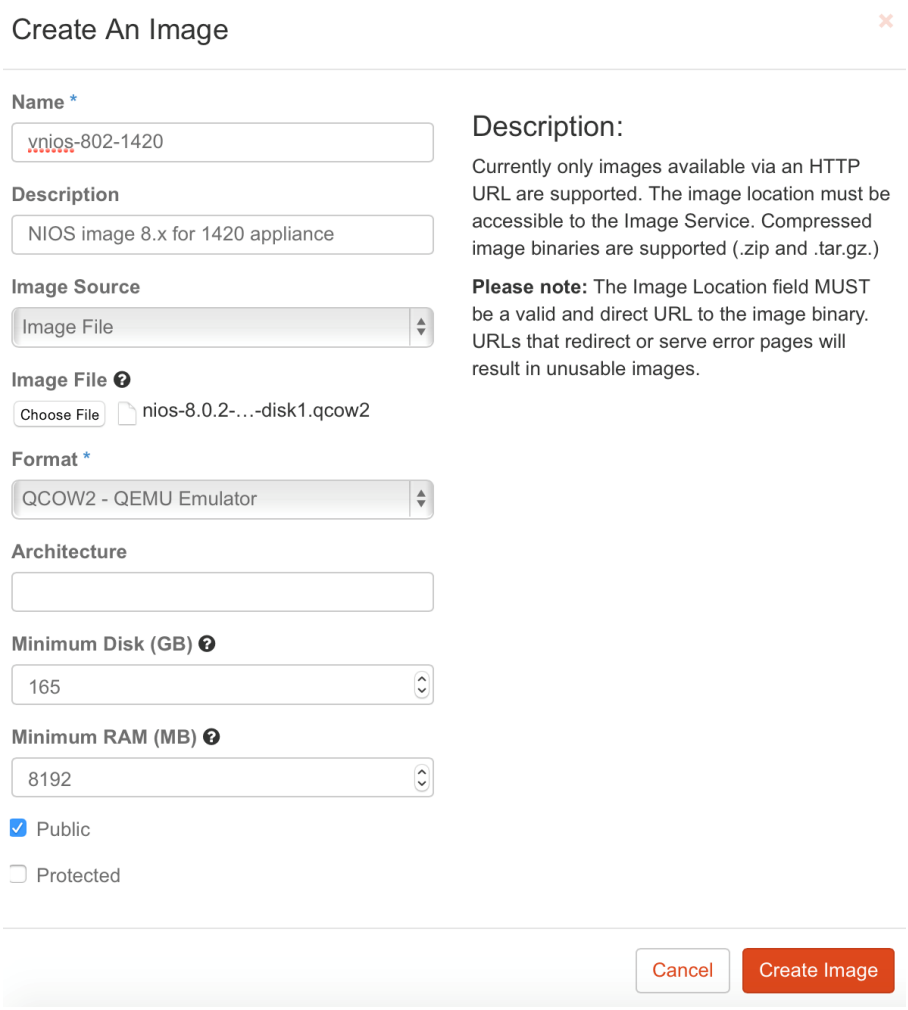
Minimum Disk (GB) – Select 165 GB

Minimum Ram (MB) – Select 8192 MB

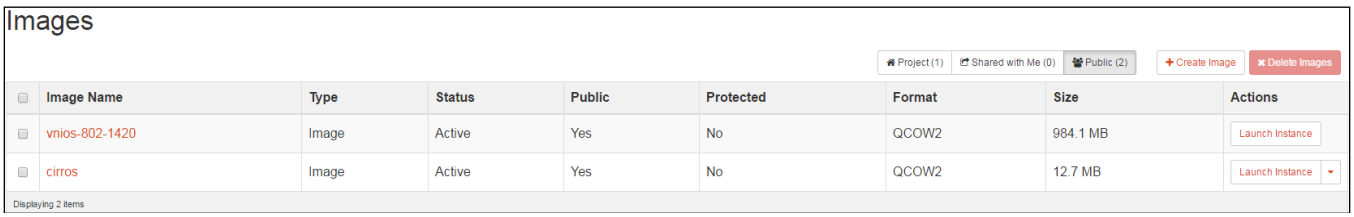
Public – Check this box

Click on **Create Image** to start the upload of the image.

Note: The image files are available in the **Downloads** section of the Infoblox Support Portal (https://support.infoblox.com/). The download type is **vNIOS for KVM**.



Wait until the **Status** of the Image shows as **Active** under **Images → Public.** Once the status shows as active, it can be used.

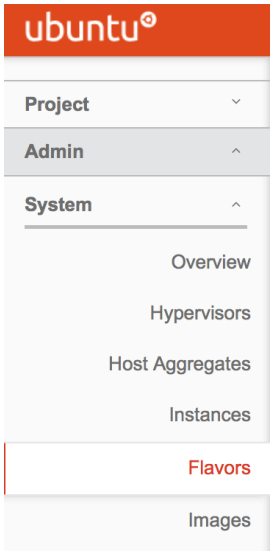


Creating a Flavor

In OpenStack, flavors define the compute, memory, and storage capacity of nova computing instances. To put it simply, a flavor is an available hardware configuration for a server. It defines the size of a virtual server that can be launched.

To create a Flavor to be used with Infoblox appliance, please refer to the Infoblox Support site (https://support.infoblox.com/) as it is based on the size and type of appliance being used.

To create a flavor in OpenStack, click on Flavors under **Admin→ System** in the Horizon Dashboard.



Click on **Create Flavor.**

Fill the fields under **Flavor Information** Tab as follows:

Name – vNIOS-1

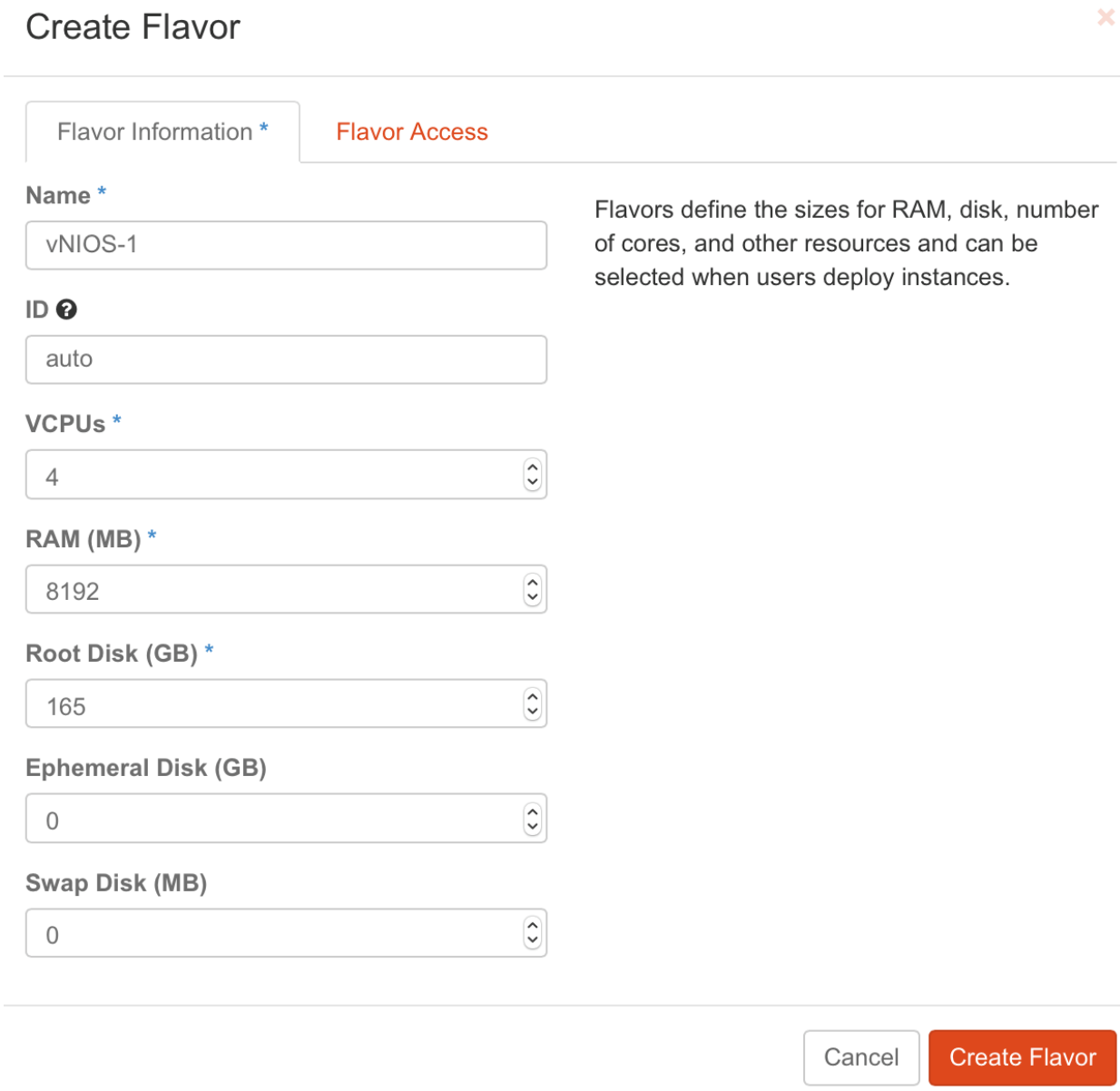
VCPUs- 4

RAM (MB) – 8192

Root Disk (GB) – 165

Ephemeral Disk (GB) – 0

Swap Disk (MB) – 0

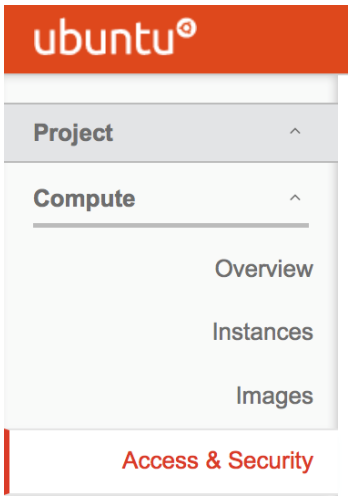


Click **Create Flavor**.

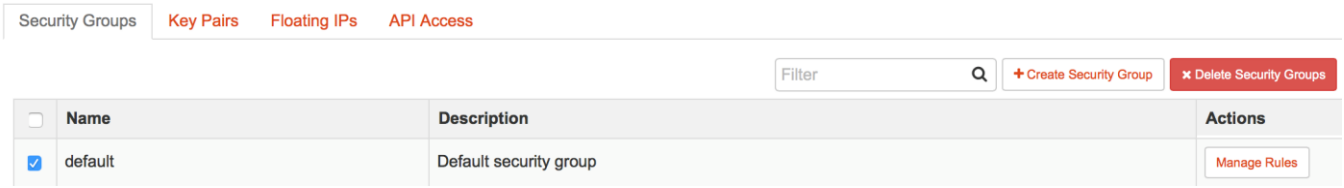
Creating a Security Group

In our example, we are using a Security Group named **default**. In order to access the Grid and be able to use DNS in the Grid, a few rules need to be added. To access the Security Group, click on:

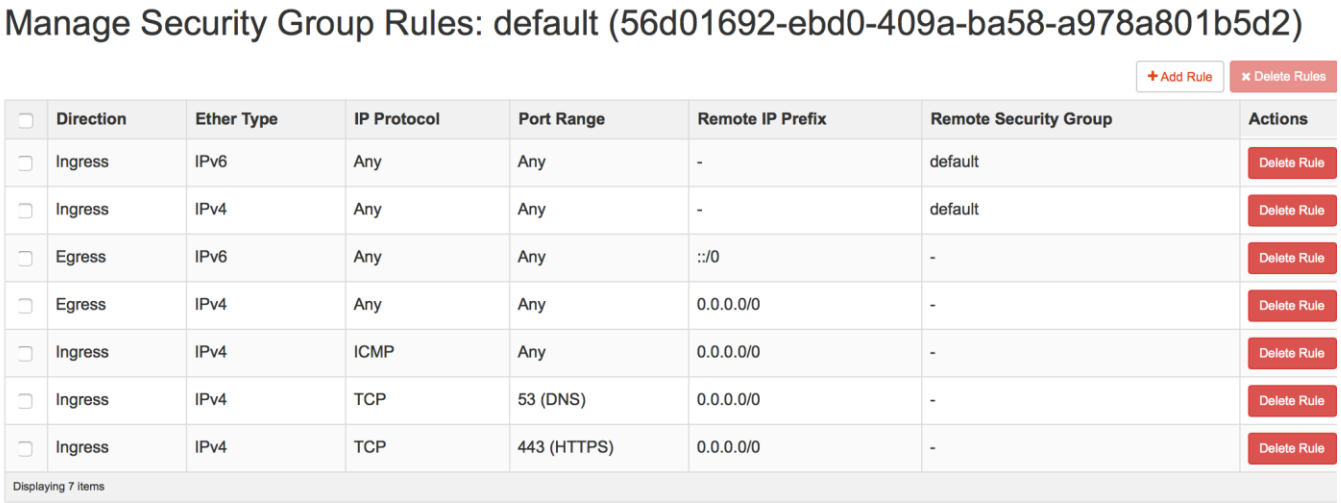
Project → Compute → Access & Security.



Click on the checkbox next to **default** and then **Manage Rules.**



Use **Add Rule** to reflect the settings as shown in screenshot below:

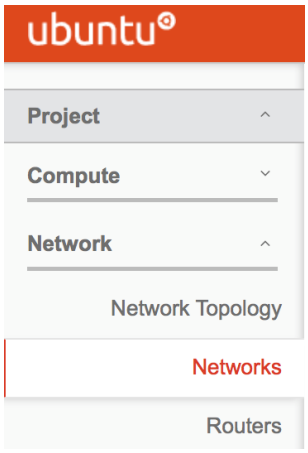


Note: The security rules at a minimum must pass ICMP, DNS , SSH and HTTPS traffic.

Creating Networks

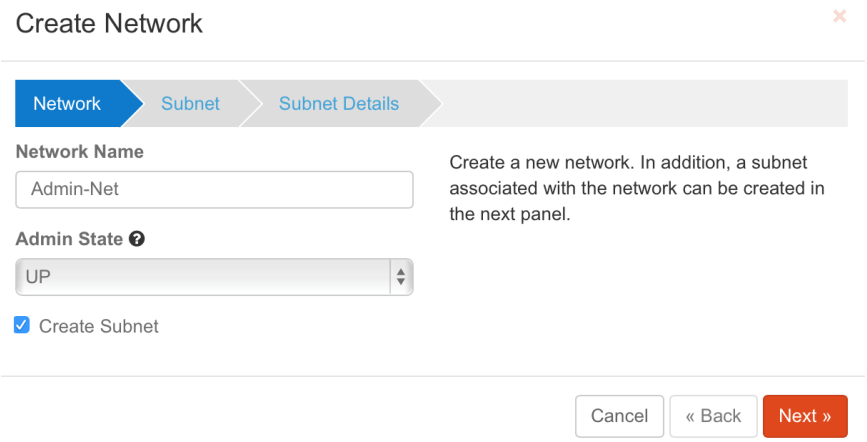
In our example, we have two internal networks named **Admin-Net** and **lan1-net** to be used with Infoblox Appliances. The third network (named **External** in this example) is used to provide floating IPs to the appliances which enable external connectivity.

To create the two internal networks, click on **Project → Network → Networks** from Horizon Dashboard.



Click **Create Network**.

Type **Admin-Net** in the **Network Name** field.



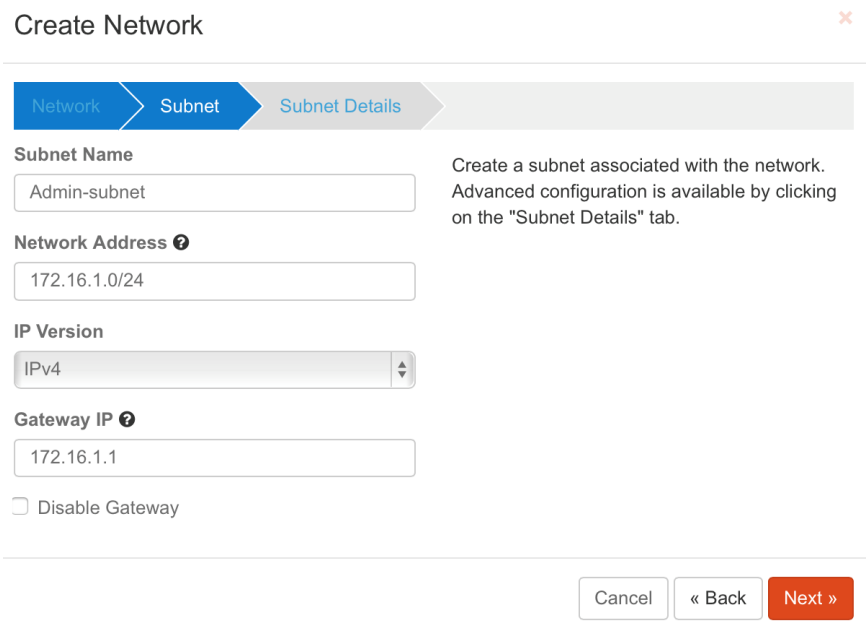
Click **Next**.

Type **Admin-subnet** in the **Subnet Name** field.

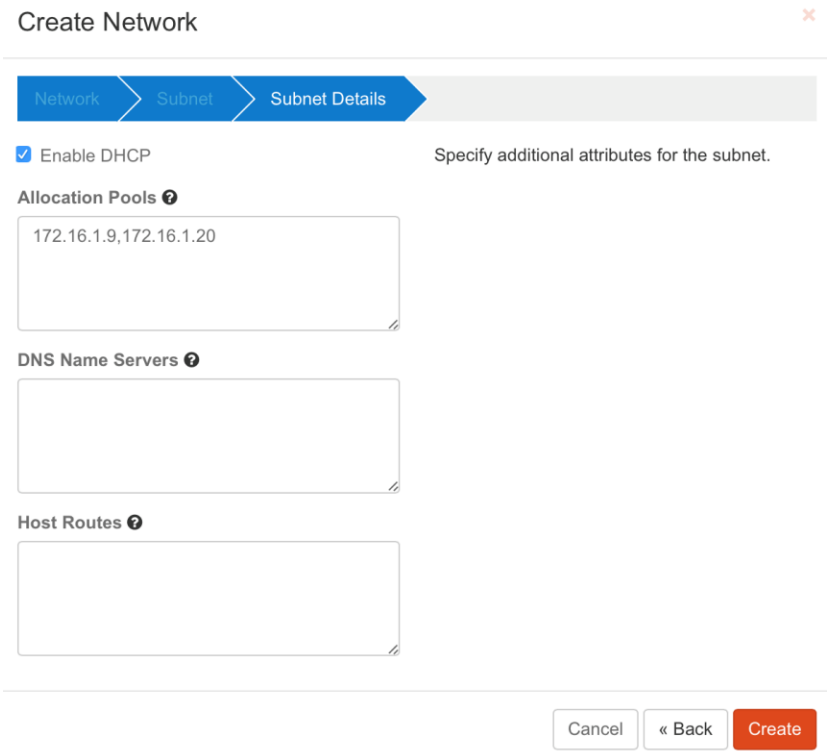
Type **172.16.1.0/24** in the **Network Address** field.

Type **172.16.1.1** in the **Gateway IP** field.

Click **Next**.



Type **172.16.1.9,172.16.1.20** in **Allocation Pools**.



Click **Create.**

Create the other internal network named **lan1-net** networks in same fashion. In our example, we have used **192.168.153.0/24** as the subnet for **lan1-net** so that the lan1-subnet is configured as shown in the screenshot below,

lan1-net subnet details:

The **External** Network is created from command line of the controller node by using the command **neutron net-create**. Example: (Some values are case sensitive)

> **neutron net-create --shared --router:external --provider:network_type flat --provider:physical_network External External**

This command creates a public network named External,setting the network type to **flat**.

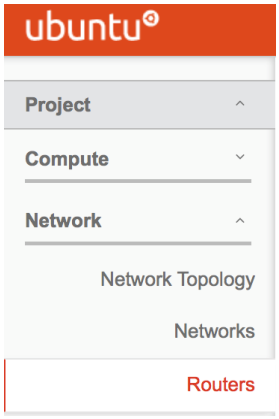
After creating the network, use the **neutron subnet-create** command to create the Floating IP range and subnet. Example:

> **neutron subnet-create --gateway 10.60.31.1 --allocation-pool start=10.60.31.50,end=10.60.31.200 External 10.60.31.0/24**

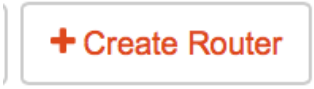
The above command example creates the subnet 10.60.31.0/24 and a floating IP range from 10.60.31.50 to 10.60.31.200.

Creating Router

The Router in our example provides connectivity to the Infoblox appliances externally. To create the Router, Go to **Project > Network > Routers**.

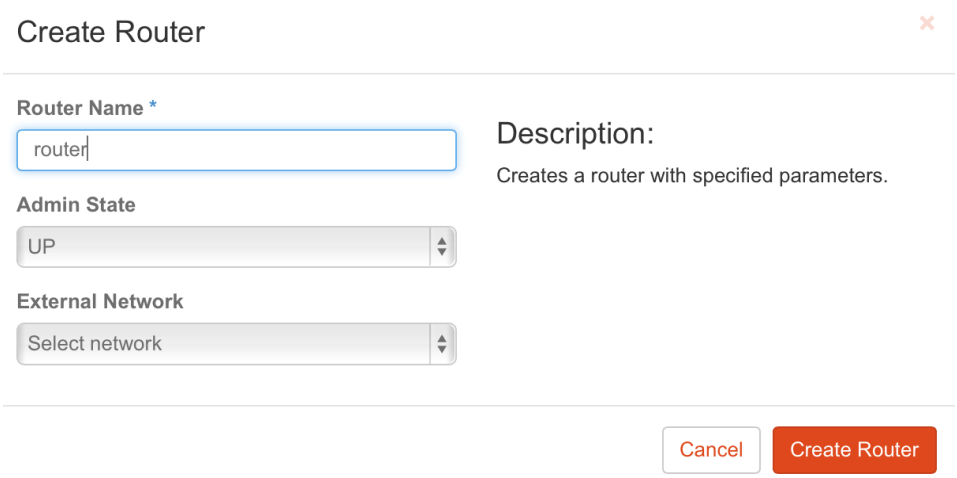


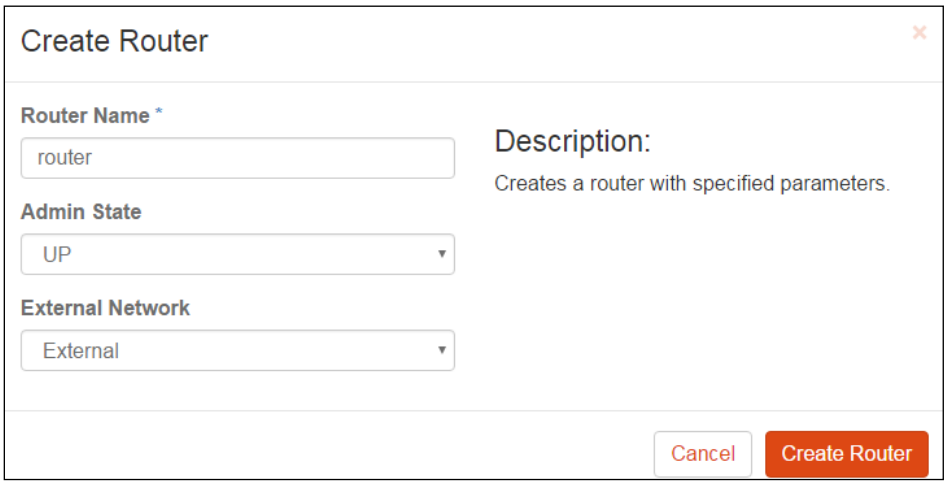
Click **Create Router**.



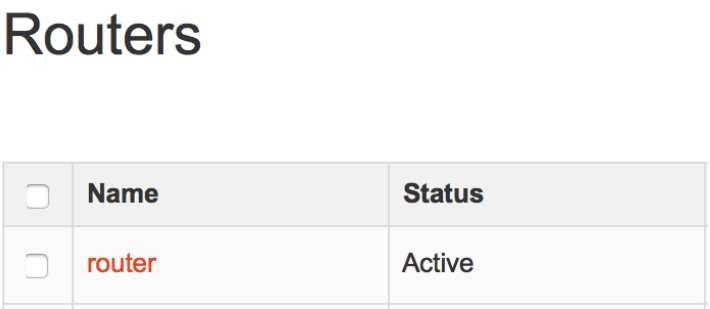
Type **router** in the **Router Name** field.

Click **Create Router**.

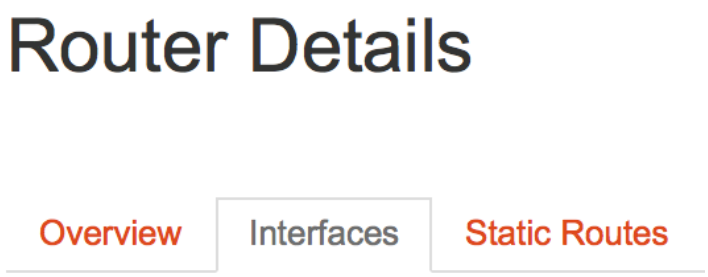




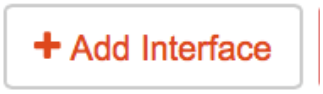
Click on the newly created **router**.



Click on the **Interfaces** tab.

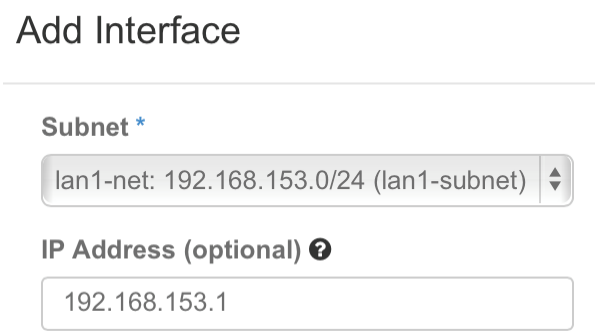


Click **Add Interface.**



Select **Subnet** as **lan1-net.**

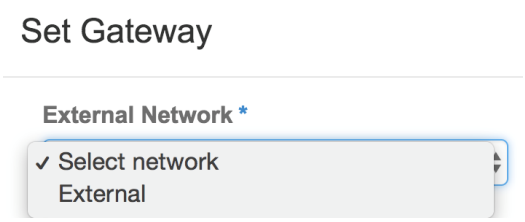
Type **192.168.153.1** in the **IP Address (optional)** field.



Click **Add interface**.

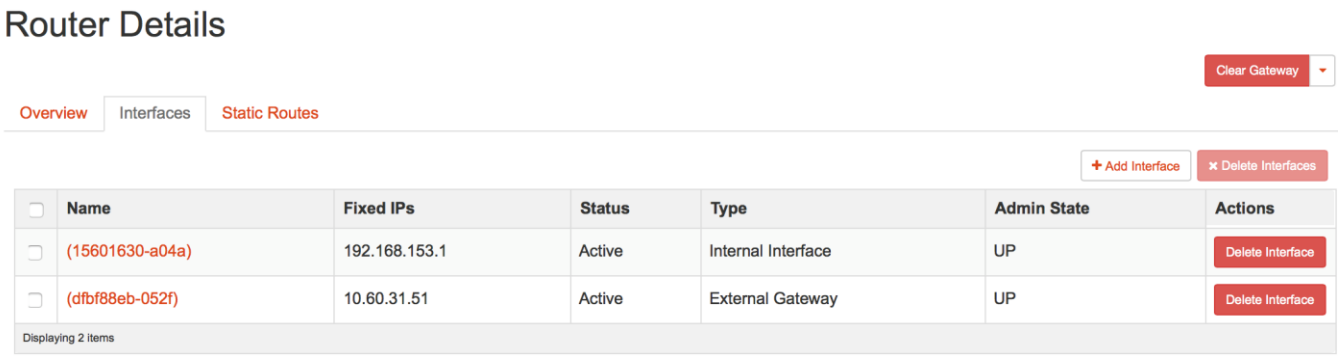
To add External interface, click on **set gateway**

Under **External Network**, select interface **External**



Click **Set Gateway**

Once both interfaces have been added, the router configuration is going to look like the screenshot below:

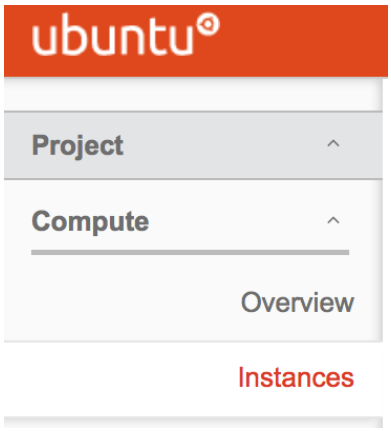


Manually spinning up Infoblox Grid Master

Now is the time to manually spin up an Infoblox appliance and make it a Grid Master.

Note: Large numbers of connections are established to the database for the following steps. If these exceed the maximum allowed by the mysql server, the Horizon web interface may start to throw errors and become extremely slow to respond, to the point of being unable to load certain screens. If this happens, refer to the Troubleshooting section found at the end of this guide for potential solutions.

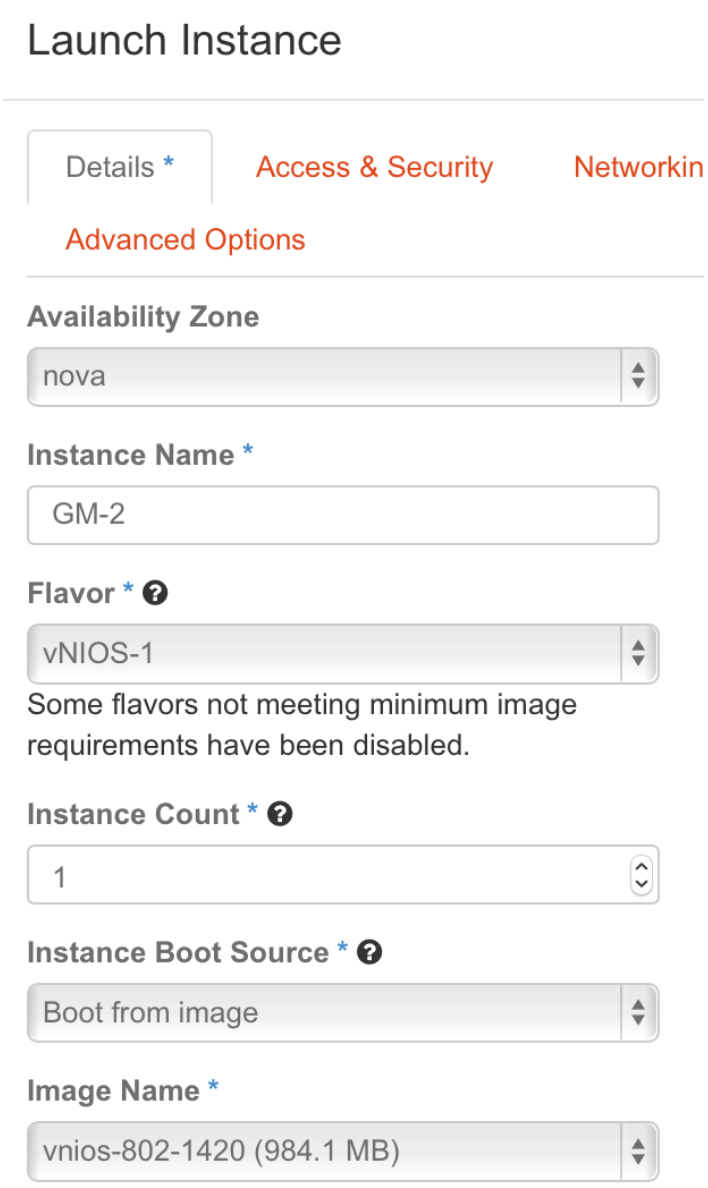
Go to **Project → Compute → Instances**.



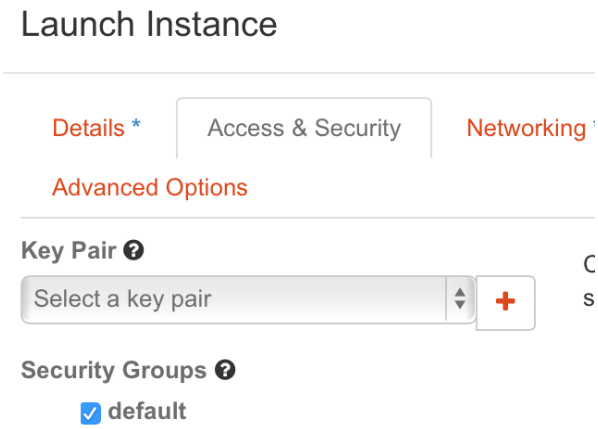
Click **Launch Instance**.

In the **Launch** Instance window, fill in the values under the **Details** tab.

In our example, we have Instance Name as **GM-2, vNIOS-1** as **Flavor** and **Instance Boot Source** set to **Boot from image.** The **Image Name** is **vnios-802-1420**.

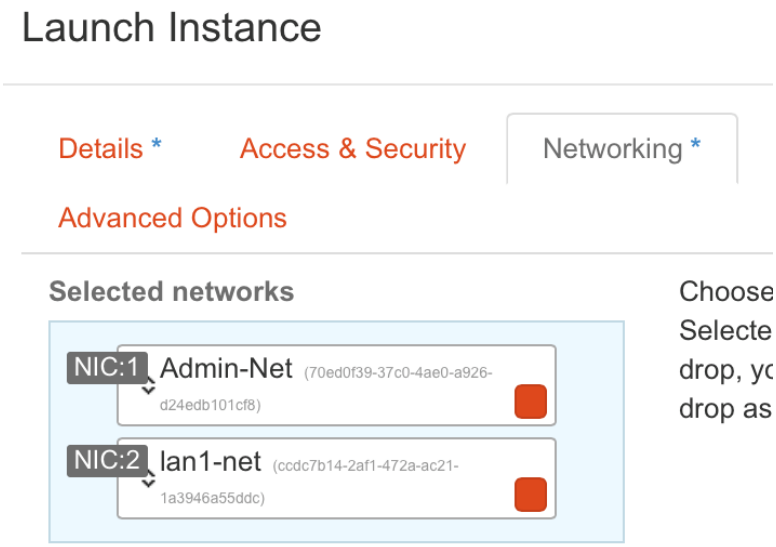


Click on the **Access & Security** tab and select **default**.



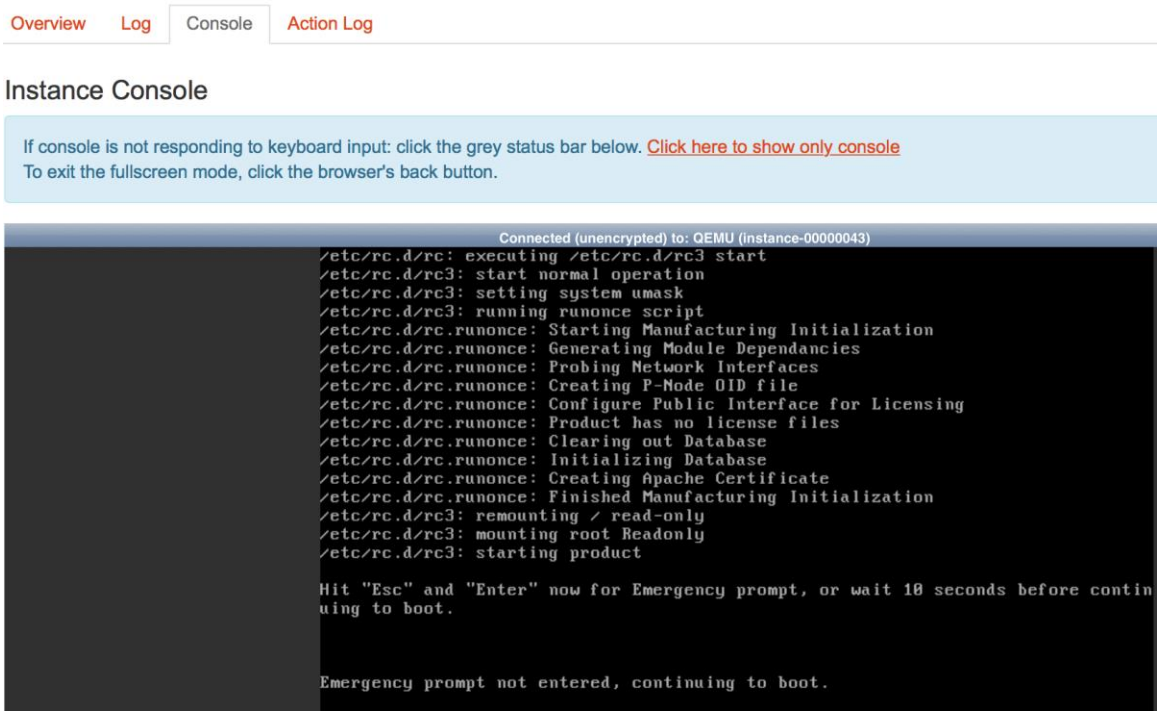
Under the **Networking** Tab, move the following two networks from **Available networks** to **Selected networks**:

Admin-Net
lan1-net



Click **Launch**.

Once the Instance **Status** is **Active,** click on the hyperlink for your instance's name and switch to the **Console** tab. In the Instance Console window, you will see the Infoblox appliance booting up.

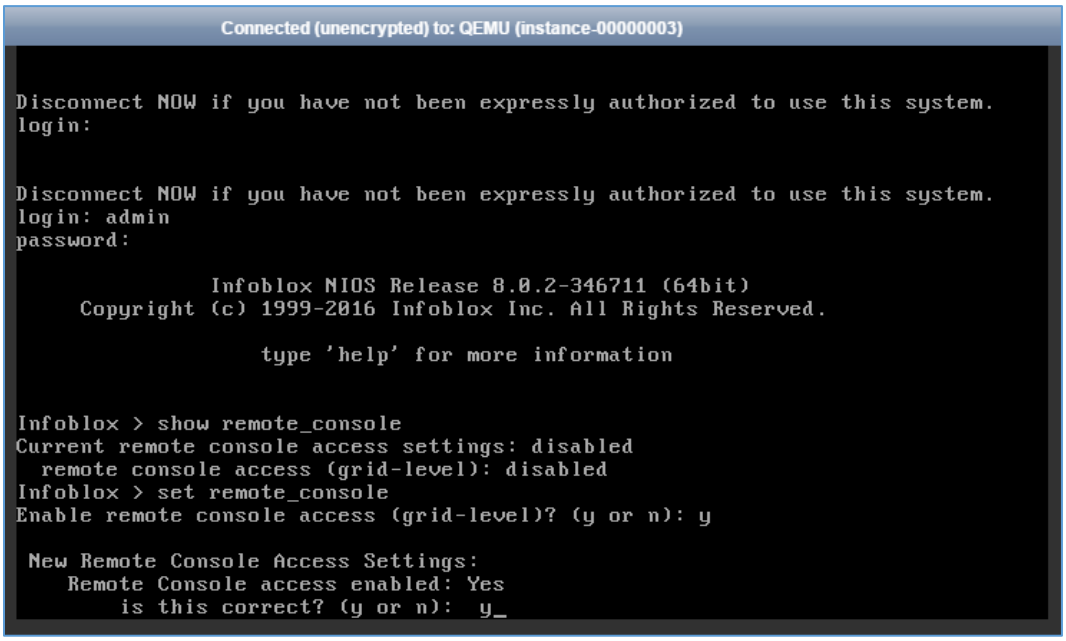


Login to the appliance using the default credentials: **admin/infoblox**

```
login: admin
password:

              Infoblox NIOS Release 8.0.2-346711 (64bit)
     Copyright (c) 1999-2016 Infoblox Inc. All Rights Reserved.

                  type 'help' for more information


Infoblox >
```

Enable Remote Console (SSH) access

SSH access is not enabled by default. To enable this, run the command **set remote_console** and type **y** at the confirmation prompts.



Set temporary license keys

Type the command **set temp_license**. At the **Select license** prompt, type the number 2 and press enter. Accept the confirmation prompts to complete the installation of the license. Once the DNS, DHCP and Grid licenses have been installed, run the **set temp_license** command again, typing the number 8 at the **Select license** prompt to install the vNIOS license (reguired in order for the server to fully start). Accept any confirmation prompts that are displayed to complete the installation. **set temp_license**

```
Infoblox > set temp_license

  1. DNSone (DNS, DHCP)
  2. DNSone with Grid (DNS, DHCP, Grid)
  3. Network Services for Voice (DHCP, Grid)
  4. Add DNS Server license
  5. Add DHCP Server license
  6. Add Grid license
  7. Add Microsoft management license
  8. Add vNIOS license
```

Note: After the **vNIOS license is installed,** the appliance will restart. Once the restart completes, you will need to log back in again to complete the next steps.

Update the network settings

Use the following command from appliance console to update the network settings (Note: We must use the IP address assigned by OpenStack to the appliance.):

set network

Once entered, the appliance will set the IP that was provided by OpenStack DHCP as the default. In the example shown here, this is **192.168.153.29**.

Hit **Enter** to accept the default IP address.

Hit **Enter** to accept default netmask (255.255.255.0 in our example).

Hit **Enter** to accept default gateway.

Type **n** for option **Become grid member** and press **Enter**.

Type **y** to accept the settings and at the second confirmation prompt, pressing **Enter** after each.

Note: The server will restart to apply the IP address as a static configuration in place of using DHCP.

Associate Floating IP to Infoblox Appliance

To access the UI externally, we need to assign a floating IP to the newly created Infoblox appliance that is to be configured as a Grid Master.

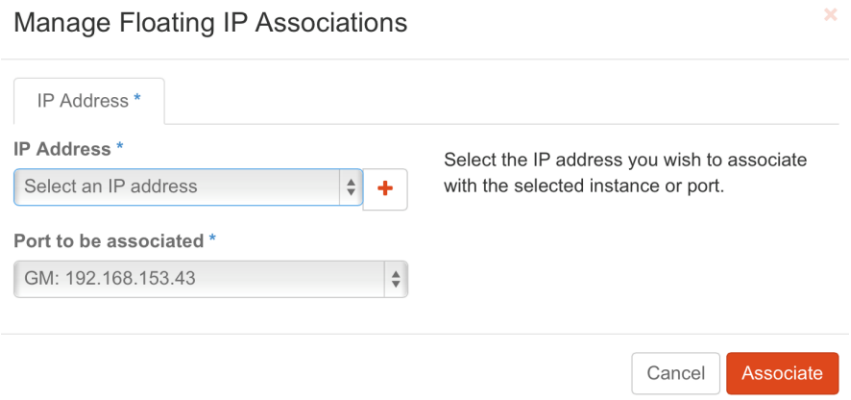
Go to **Project → Compute → Instances**.

Select the instance that you need to associate the floating IP with. In our example, it is named GM.

GM	vnios-802-1420	Admin-Net 172.16.1.10 lan1-net 192.168.153.43	vNIOS-1	ubuntu-kp	Active	nova	None	Running	1 day, 9 hours	Create Snapshot

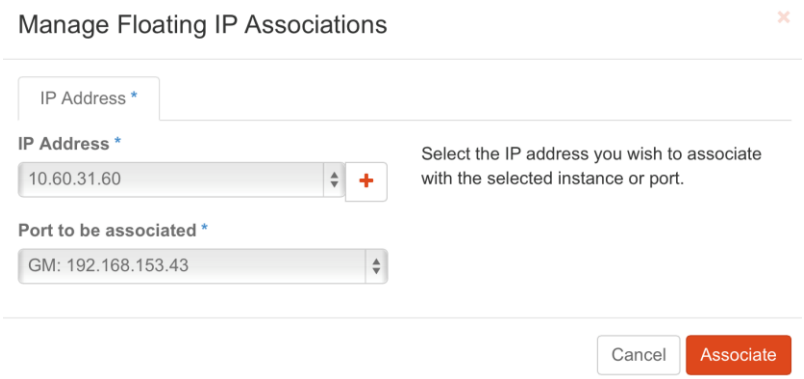
Click on the drop-down link next to **Create Snapshot**.

Click **Associate Floating IP**.



Click **+**.

Click **Allocate IP**.



In our example, 10.60.31.60 is the floating IP that was allocated.

Click **Associate** to complete the association.

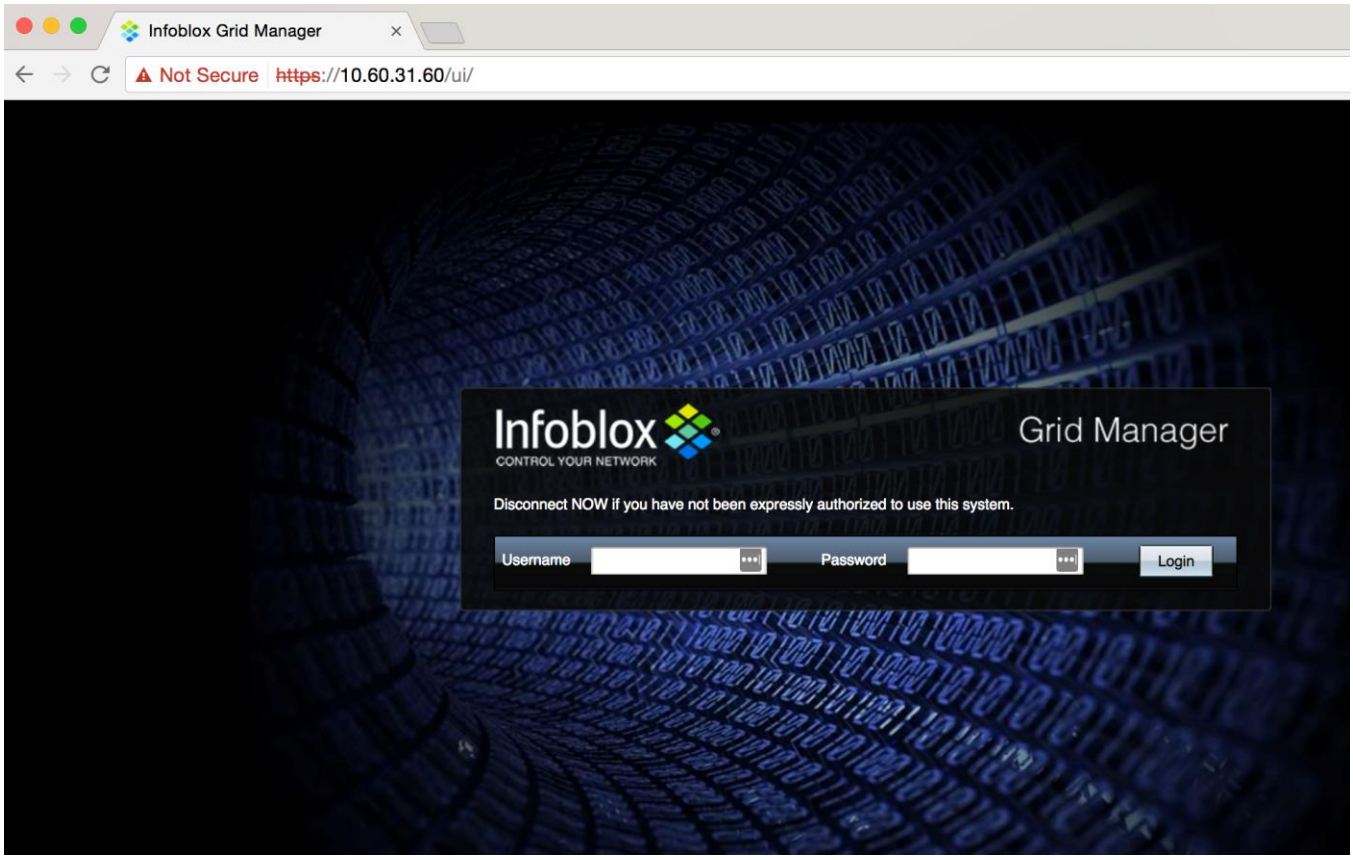
Instance Name	Image Name	IP Address
GM	vnios-802-1420	Admin-Net 172.16.1.10 lan1-net 192.168.153.43 Floating IPs: 10.60.31.60

Infoblox Grid Manager GUI

Access the UI of the Infoblox member using its floating IP and configure it as Grid Master. In our example, we are going to access it using the following url:

https://10.60.31.60

Note: This must be an https connection as http access is not enabled by default. Use the default credentials to login (**admin/infoblox**).



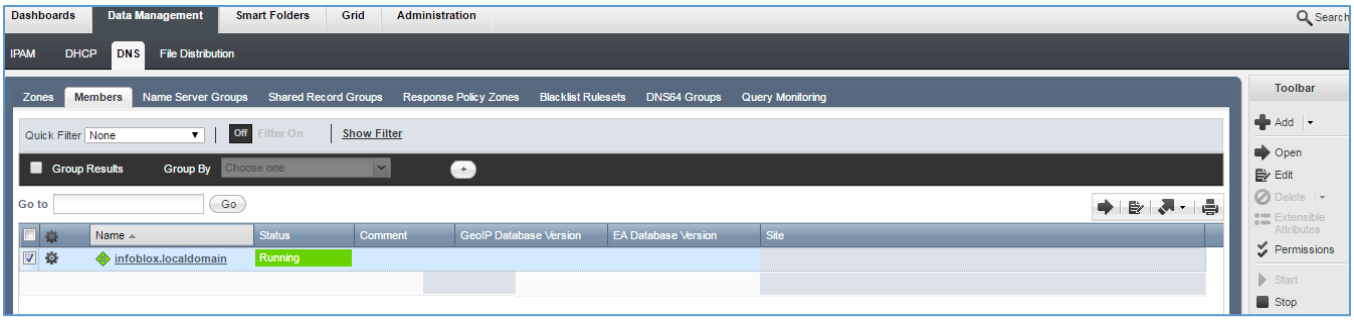
Refer to the NIOS administrator guide for additional details on configuring your Infoblox server.

Start the DNS service.

For the DNS queries to work, the DNS service must be started. To start the DNS service:

1. Login to the Infoblox Grid Manager GUI if not already logged in.
2. Navigate to the **Data Management → DNS → Members** tab.
3. Enable the check box next to the name of your Infoblox server.
4. Click on the Start button in the toolbar on the right-hand side of the page.
5. Click on the Yes button.
6. Click on the (refresh) button found at the bottom of the page to verify that the DNS service starts successfully (shows the status as Running and in green).

Note: The service may take up to a minute to start (though usually is faster than that) and will show an error or warning state until that process completes.



Name Server Group

Create a Name Server (NS) Group with the name **default** and add your Grid Master as a Grid primary name server. In the next section, we demonstrate how to leverage orchestration to automate the creation of an Infoblox vNIOS appliance and join it to your Grid. As part of this process, this new server will automatically be added to this NS Group.

Orchestrating the creation of an Infoblox Grid Member

This section describes the steps needed to orchestrate a member appliance and join it to an existing Grid.

In our example, the Grid Master is @ 10.60.31.60.

Install git software by issuing the following command

sudo apt-get install git

Issue the following command to get yaml scripts from github

git clone https://github.com/infobloxopen/engcloud

Go to **~/engcloud/grid-templates**

Create the file write_env.sh using the following command:

vi write_env.sh

Edit write_env.sh, populating the file with the following:

```
#!/bin/bash

FIP=$1
VIP=$2
FIP_ID=$3

if [[ -z $FIP || -z $VIP || -z $FIP_ID ]]; then
        echo "Usage: $0 FIP VIP FIP_ID"
        echo "Try 'neutron floatingip-list' to find that info for FIP that goes with the VIP"
        exit 1
fi

source grid-lib.sh

write_env $FIP $VIP $FIP_ID
```

Save the file.

Update the permissions on write_env.sh:

chmod +x write_env.sh

Issue the following commands:

source admin-openrc.sh

neutron floatingip-list

```
root@tme-openstack:~# neutron floatingip-list
+--------------------------------------+------------------+---------------------+--------------------------------------+
| id                                   | fixed_ip_address | floating_ip_address | port_id                              |
+--------------------------------------+------------------+---------------------+--------------------------------------+
| 01a9f228-3829-4bb6-96e4-14fb5d3f2bfb | 192.168.153.43   | 10.60.31.60         | 0cfa075e-34b7-4345-9ed6-ae5756ca4276 |
| 29156c13-d4dd-4525-b940-198c471fe0ab |                  | 10.60.31.59         |                                      |
| 2df564d9-5717-4f2f-9f16-b50bd0546067 | 192.168.153.41   | 10.60.31.57         | ebd30a8b-15ed-480e-8f15-2eec377603e3 |
| b4c22c5b-31df-43b2-9517-360e571bb109 | 192.168.153.42   | 10.60.31.58         | 383bf69f-3144-4c05-a757-cb26cd3b1018 |
| ecaccd01-52f7-4135-8391-b7366f300429 | 192.168.153.29   | 10.60.31.52         | 7cf5c6bc-a68c-4102-82f6-66c2b60091d3 |
+--------------------------------------+------------------+---------------------+--------------------------------------+
```

The next step is to download the certificate from the Grid Master. In the following example, we are getting this from the Grid Master (GM) which has the floating-ip of 10.60.31.60.

./write_env.sh 10.60.31.60 192.168.153.43 01a9f228-3829-4bb6-96e4-14fb5d3f2bfb

Where **10.60.31.60** is the floating-ip.

192.168.153.43 is the fixed-ip-address of LAN port.

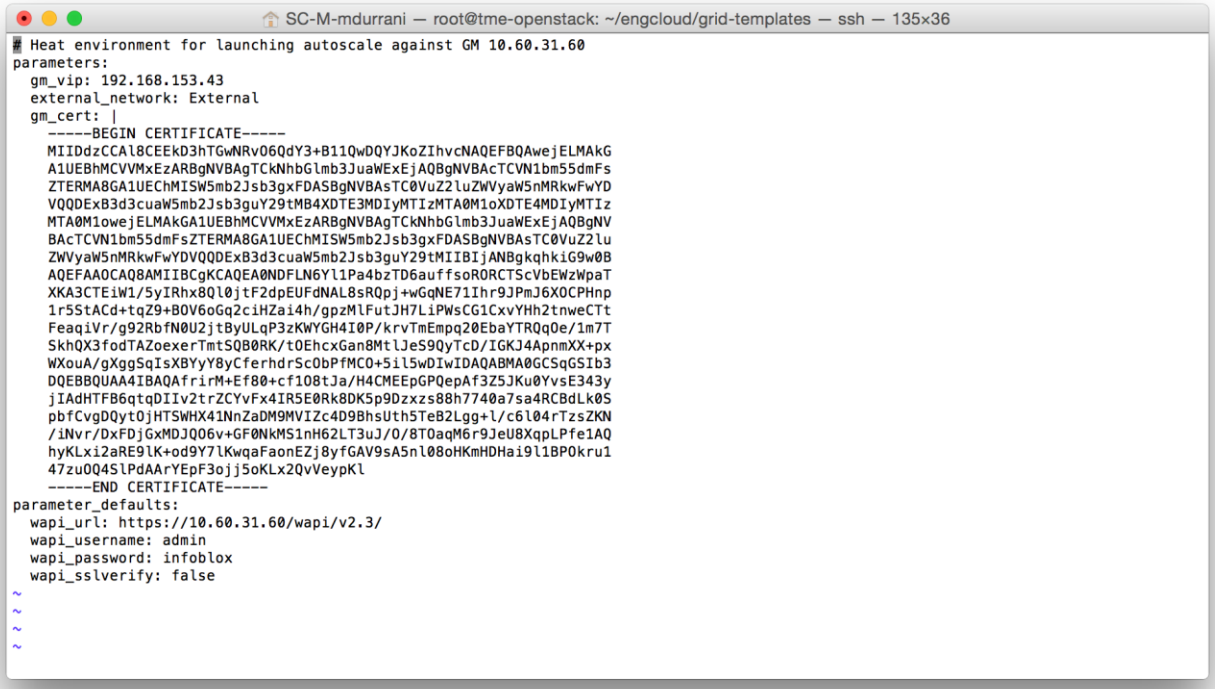
01a9f228-3829-4bb6-96e4-14fb5d3f2bfb is the id taken from the neutron floatingip-list command output for that appliance. Example:

./write_env.sh 10.60.31.60 192.168.153.43 01a9f228-3829-4bb6-96e4-14fb5d3f2bfb
Thu Feb 23 11:39:02 PST 2017: Downloading certificate from 10.60.31.60 for use in member join...

Thu Feb 23 11:39:02 PST 2017: Done

Next issue the following command. In our example 10.60.31.60 is the appliance floating ip address.

vi gm-10.60.31.60-env.yaml



Close the file once done reviewing it.

In our example the LAN network is named **lan1-net** and has subnet of **192.168.153.0/24, hence in member.yaml file, it needs to be reflected as shown below:**

```
# injection via user_data
  lan1_port:
    type: OS::Neutron::Port
    properties:
      network: lan1-net
      security_groups: [ default ]

# the MGMT interface configuration is not yet supported

 networks: [{network: Admin-net }, {port: { get_resource: lan1_port}} ]
```

The strings for network and for security groups must match with the names used by you in the OpenStack setup. In our example, we used Admin-net so it should match in networks section.

Our working member.yaml file is given below for reference:

```
heat_template_version: 2014-10-16
description: An Infoblox Grid Member
parameters:
  external_network:
    type: string
    description: the external network for floating IP allocations
    default: public-138-net
  model:
    type: string
```

```
    description: vNIOS Model
    default: IB-VM-810
  flavor:
    type: string
    description: vNIOS Flavor
    default: vnios-810.55
  image:
    type: string
    description: vNIOS Flavor
    default: nios-7.3.0-314102-55G-810
  wapi_url:
    type: string
    description: the URL to access the GM WAPI from the Heat engine
  wapi_username:
    type: string
    description: the username for the WAPI access
  wapi_password:
    type: string
    description: the username for the WAPI access
  wapi_sslverify:
    type: string
    description: the value for SSL Verify (true/false/certificate path)
    default: false
  gm_vip:
    type: string
    description: the VIP of the GM, to be used by members for joining the grid
  gm_cert:
    type: string
    description: the GM certificate contents
resources:
  host_name:
    type: OS::Heat::RandomString
    properties:
      length: 12
      sequence: lowercase

# We pre-allocate the port for LAN1, so that we have the IP address already for
# injection via user_data
  lan1_port:
    type: OS::Neutron::Port
    properties:
      network: lan1-net
      security_groups: [ default ]

# Each member needs a floating IP so Ceilometer can poll the member for QPS.
  floating_ip:
    type: OS::Neutron::FloatingIP
    properties:
      floating_network: { get_param: external_network }
      port_id: { get_resource: lan1_port }

  grid_member:
    type: Infoblox::Grid::Member
```

```
    properties:
      connection: {url: {get_param: wapi_url}, username: {get_param: wapi_username}, password:
{get_param: wapi_pa
ssword}, sslverify: {get_param: wapi_sslverify}}
      name: { list_join: [ '.', [{ list_join: [ '-', [ 'member-dns', { get_resource: host_name } ]] },
'localdomai
n' ]] }
      model: { get_param: model }
# the MGMT interface configuration is not yet supported
      LAN1: { get_resource: lan1_port }
      dns: { enable: True }
      temp_licenses: ["vnios", "dns", "enterprise", "rpz"]
      gm_ip: { get_param: gm_vip }
      gm_certificate: { get_param: gm_cert }
      remote_console_enabled: true
      admin_password: infoblox

  grid_member_ns_group_entry:
    type: Infoblox::Grid::NameServerGroupMember
    properties:
      connection: {url: {get_param: wapi_url}, username: {get_param: wapi_username}, password:
{get_param: wapi_pa
ssword}, sslverify: {get_param: wapi_sslverify}}
      group_name: default
      member_role: grid_secondary
      member_server: { name: { get_attr: [grid_member, name] } }

  server:
    type: OS::Nova::Server
    properties:
      name: { list_join: [ '-', [ 'member-dns', { get_resource: host_name } ]] }
      flavor: { get_param: flavor }
      image: { get_param: image }
      networks: [{network: Admin-Net }, {port: { get_resource: lan1_port}} ]
      config_drive: true
      user_data_format: RAW
      user_data: { get_attr: [grid_member, user_data] }
```

Next, we need to get images on the OpenStack node by executing the following command:

source admin-openrc.sh

nova image-list

```
root@tme-openstack:~/engcloud/grid-templates# nova image-list
+--------------------------------------+------------------+--------+--------+
| ID                                   | Name             | Status | Server |
+--------------------------------------+------------------+--------+--------+
| ef27ebc9-8421-461b-b4da-e28831890b8d | Ubuntu-Desktop   | ACTIVE |        |
| f67fcefa-2bb7-4a92-bfd4-3633f51a3bd6 | Windows-2012     | ACTIVE |        |
| 87f93b16-3088-46f5-af63-88be3239ba4e | centos-7         | ACTIVE |        |
| 9c98f189-f19e-4e6a-8cf0-4a14835a43e1 | cirros           | ACTIVE |        |
| c4958971-66cf-4d24-a5ea-8820060ebd0f | ubuntu-cloud     | ACTIVE |        |
| bd139188-6514-4c2a-bfbf-2465f6628a38 | ubuntu-cloud-img | ACTIVE |        |
| 35f25ba7-c527-453f-9d96-1680d411a66f | ubuntu-image     | ACTIVE |        |
| f35fbc13-a878-4a4e-a32a-797ba2dee297 | vNIOS-7.10       | ACTIVE |        |
| a5225c0f-e27e-40cf-a32d-5a75022741c6 | vNIOS-736        | ACTIVE |        |
| 97486dc4-effb-44bf-b8e2-e5d3e17fcfb6 | vnios-802-1420   | ACTIVE |        |
| 6162684b-0298-4bfd-bc40-e1bfe3d31e0f | windows-server   | ACTIVE |        |
+--------------------------------------+------------------+--------+--------+
```

Issue the following command to view flavors:

nova flavor-list

```
root@tme-openstack:~/engcloud/grid-templates# nova flavor-list
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID                                   | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1                                    | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2                                    | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3                                    | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4                                    | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 4db439fd-6018-4dd4-996b-c6e3124e3e51 | vNIOS-1   | 8192      | 165  | 0         |      | 4     | 1.0         | True      |
| 5                                    | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
```

Next, execute the following command from the location **/engcloud/grid-templates** to automatically instantiate a member appliance to join the Grid, be added to the **default** nameserver group and act as a DNS server:

heat stack-create -e gm-10.60.31.60-env.yaml -P"flavor=vNIOS-1;image=97486dc4-effb-44bf-b8e2-e5d3e17fcfb6;model=IB-VM-1420" -f member.yaml member-lan

In the above command:

10.60.31.60 is the Grid Master Floating IP address.

vNIOS-1 is the previously created flavor.

The image used here is **vnios-802-1420** and has the id **97486dc4-effb-44bf-b8e2-e5d3e17fcfb6**.

Model of the Grid member is **IB-VM-1420**.

```
root@tme-openstack:~/engcloud/grid-templates# heat stack-create -e gm-10.60.31.60-env.yaml -P"flavor=vNIOS-1;image=
97486dc4-effb-44bf-b8e2-e5d3e17fcfb6;model=IB-VM-1420" -f member.yaml member-lan
+--------------------------------------+------------+-------------------+----------------------+--------------+
| id                                   | stack_name | stack_status      | creation_time        | updated_time |
+--------------------------------------+------------+-------------------+----------------------+--------------+
| 018e8eb5-ded2-4002-b4e6-21b32b6fd0f1 | member-3   | CREATE_COMPLETE   | 2017-02-17T19:06:28 | None         |
| cb9fa6aa-61bd-43f2-86d2-69d5e7defde8 | member-4   | CREATE_COMPLETE   | 2017-02-17T22:31:17 | None         |
| bceec2c0-fd10-42a7-9b27-bfb16b29cff8 | member-lan | CREATE_IN_PROGRESS | 2017-02-23T22:39:58 | None         |
+--------------------------------------+------------+-------------------+----------------------+--------------+
```

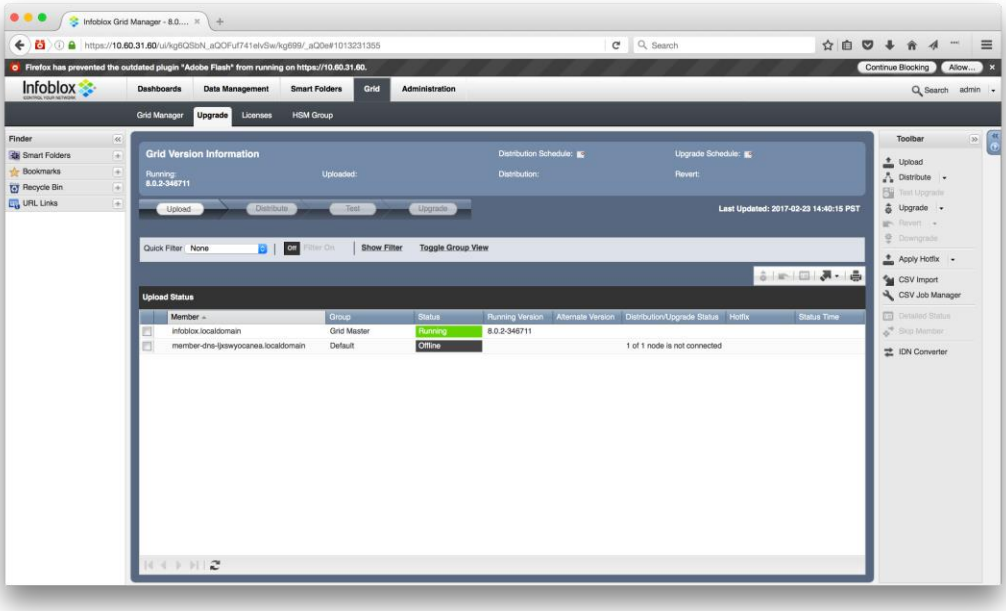
To see if the stack was created, execute the following command:

heat stack-list

```
root@tme-openstack:~/engcloud/grid-templates# heat stack-list
+--------------------------------------+------------+-----------------+----------------------+--------------+
| id                                   | stack_name | stack_status    | creation_time        | updated_time |
+--------------------------------------+------------+-----------------+----------------------+--------------+
| 018e8eb5-ded2-4002-b4e6-21b32b6fd0f1 | member-3   | CREATE_COMPLETE | 2017-02-17T19:06:28 | None         |
| cb9fa6aa-61bd-43f2-86d2-69d5e7defde8 | member-4   | CREATE_COMPLETE | 2017-02-17T22:31:17 | None         |
| bceec2c0-fd10-42a7-9b27-bfb16b29cff8 | member-lan | CREATE_COMPLETE | 2017-02-23T22:39:58 | None         |
+--------------------------------------+------------+-----------------+----------------------+--------------+
```

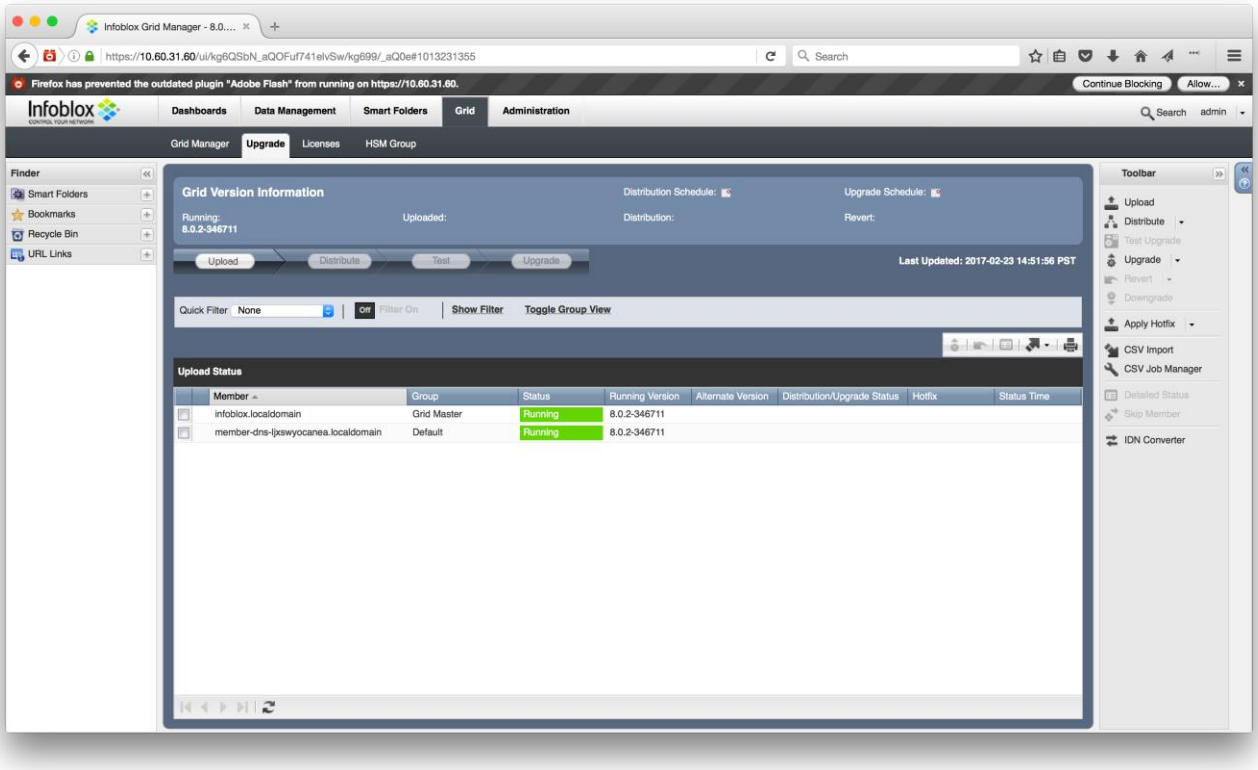
Now, in the Infoblox Grid Manager GUI, we can see the newly created Grid member has been setup and will have a status pf offline. In our example, the Grid Manager GUI is reached using the following URL:

https://10.60.31.60/

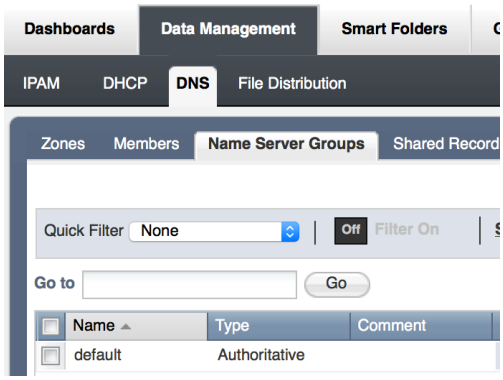


It takes about 8 to 10 minutes for this IB-VM-1420 to synchronize with the Grid Master and multiple reboots will take place during this time.

Once it has completed synchronizing with the Grid, the status should update to Running (you may need to refresh the display to view this).



Go to **Data Management → DNS → Name Server Groups**.



Select the **default NS** Group and click **Edit** from **Toolbar.**

Verify that the newly created **member-lan is now** part of the NS Group.

This concludes our guide, which shows how to create OpenStack on a single server and how to automatically instantiate members using yaml scripts and heat resources.

Troubleshooting

Editing files using the VI command does not work as expected

Cause: Your installation may use a different version of vi which has a different behavior.

Solution: An alternative editor, such as vim, can be used. To install vim in Ubuntu, run the following command:

```
# sudo apt-get install vim
```

Environment variables have been set by MySQL does not appear to be using them

Cause: The file **/etc/mysql/my.cnf** (may be located in different locations depending on your build) contains default environment variables which may be conflicting with the ones you are setting.

Solution: Review the my.cnf file for any settings which may be creating a conflict.

Seeing the following error when running the command **keystone-manage db_sync**:

CRITICAL keystone [-] NoSuchModuleError: Can't load plugin: sqlalchemy.dialects:mysql.pymsql

Cause: Required packages are not installed

Solution: Run the following commands to install additional packages:

sudo pip install rfc3986

sudo pip install positional

The command "apt-get install -y python-openstackclient" fails.

The following error is displayed: IOError: [Errno 13] Permission denied: '/var/log/keystone/keystone-manage.log'

In keystone-manage.log, you may see the following error:

2017-03-15 14:57:52.058 14066 CRITICAL keystone [-] DBConnectionError: (pymysql.err.OperationalError) (2003, 'Can\'t connect to MySQL server on \'controller\' ((1045, u"Access denied for user \'keystone\'@\'controller\' (using password: YES)"))')

Cause: There may be an error in the /etc/keystone/keystone.conf file.

Solution: Verify that the configuration is set correctly. In keystone.conf, the connection line includes the user name and password that is used for the connection. If there is a typo or other error with these credentials, that can cause this type of failure.

Example: **connection = mysql+pymysql://keystone:keystone_dbpass@controller/keystone**

In the above example, “keystone” is the user name and if this is set incorrectly, will be displayed in the error output. “keystone_dbpass” is the password used for the connection. If this is set incorrectly, you will also encounter this error but no specific message will be provided to indicate this.

The command **keystone-manage db_sync** fails with the following error:

2017-03-20 11:44:45.589 3673 ERROR keystone NoSuchModuleError: Can't load plugin: sqlalchemy.dialects:mysql.pymsql

Cause: There may be an error in /etc/keystone/keystone.conf.

Solution: Verify that the connection string in keystone.conf is set correctly. Example:

```
connection = mysql+pymysql://keystone:keystone_dbpass@controller/keystone
```

If **mysql+pymysql** is not set correctly, you will see this error. A common typo is to omit one of the 'y's in pymysql.

The command service apache2 restart fails with the following error:

```
* Restarting web server apache2

[Mon Mar 20 12:05:53.723559 2017] [core:error] [pid 4177:tid 140483295143808] (EAI 2)Name or service not known: AH00547: Could not resolve host name *.5000 -- ignoring!
```

Cause: There may be an error in the Apache configuration file (/etc/apache2/apache2.conf).

Solution: Review the configurations set in apache2.conf. In the error message displayed above, the cause was a dot being used in place of the required semi-colon for the line **<VirtualHost *:5000>**.

The command "su -s /bin/sh -c "glance-manage db_sync" glance" fails with the error:

2017-03-20 14:27:39.378 9941 WARNING oslo_db.sqlalchemy.engines [-] SQL connection failed. 1 attempts left.
2017-03-20 14:27:49.389 9941 CRITICAL glance [-] DBConnectionError: (pymysql.err.OperationalError) (2003, 'Can\'t connect to MySQL server on \'controller\' ((1045, u"Access denied for user \'glance\'@\'controller\' (using password: YES)"))')

Cause: There was an error when granting privileges for the glance user to the glance database in MySQL

Solution: Run the **GRANT ALL PRIVILEGES ON glance** commands again, verifying that everything is entered correctly before executing them. Example:

```
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY 'Infoblox_1';
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'Infoblox_1';
```

In the above examples, Infoblox_1 is the password that the glance user will be using. Be sure that this is set correctly, and that no quotation marks (single quotes) are missing. The password for the glance user is set when this user is created. As a reference, the command provided previously for creating this user was:

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

The command nova image-list fails with the following errors:

```
2017-03-20 16:09:18.985 15151 ERROR nova.api.openstack.extensions DBConnectionError:
(pymysql.err.OperationalError) (2003, 'Can\'t connect to MySQL server on \'10.60.160.130\' ((1045, u"Access denied
for user \'nova\'@\'controller\' (using password: YES)"))')
2017-03-20 16:09:18.985 15151 ERROR nova.api.openstack.extensions
2017-03-20 16:09:18.988 15151 INFO nova.api.openstack.wsgi [req-4ef19906-d008-49dd-950b-21913fa704d9
f75b1ff14195461a951c9c7cef7a6139 efc64fc33cb74b739a56cb30c9a609ae - - -] HTTP exception thrown:
Unexpected API Error. Please report this at http://bugs.launchpad.net/nova/ and attach the Nova API log if possible.
<class 'oslo_db.exception.DBConnectionError'>
2017-03-20 16:09:18.989 15151 INFO nova.osapi_compute.wsgi.server [req-4ef19906-d008-49dd-950b-
21913fa704d9 f75b1ff14195461a951c9c7cef7a6139 efc64fc33cb74b739a56cb30c9a609ae - - -] 10.60.160.130 "GET
/v2/efc64fc33cb74b739a56cb30c9a609ae/os-services HTTP/1.1" status: 500 len: 440 time: 100.1201141
```

Cause: /etc/glance/glance-registry.conf has not been created or has an error.

Solution. Verify that glance-registry.conf has been created and is configured correctly, including passwords and IP addresses.

Slow performance/unusable Horizon web UI

Cause: In some environments, the Horizon web UI may report numerous errors when loading certain pages, such as System → Instances, slow to respond or even completely unusable. One frequent cause of this is due to the large number of connections to the database that are required for some operations to complete and which may exceed the maximum number of connections allowed to the database.

Solution: With a mysql database, you will want to complete the following steps to temporarily resolve this issue:

1. Open a terminal window or command prompt on your computer which has access to the MySQL server (such as your controller/Ubuntu server).
2. Login to your MySQL server: **mysql -u root -p**
3. Verify the current connections limit: **show variables like "max_connections";**

   ```
   MariaDB [(none)]> show variables like "max_connections";
   +-----------------+-------+
   | Variable_name   | Value |
   +-----------------+-------+
   | max_connections | 200   |
   +-----------------+-------+
   1 row in set (0.00 sec)
   ```
4. Increase the connection limit to a suitable number: **set global max_connections = 300;**

Note: This change will take effect immediately but is temporary and will be lost the next time MySQL is restarted. Memory usage may also increase and care should be used to verify that you do not exceed the amount of available memory on your server. It may also take time for any existing connections to be processed or expire. Refer to the MySQL documentation for steps to make this change permanent if desired. This may involve updating a configuration file such as **/etc/my.cnf** (depending on your operating system and database version).