Glance Chef Cookbook

1. Objective

The objective here is to automate the deployment of an OpenStack Image (Glance) cluster using Chef and Vagrant. In this scenario, a 3 node Glance cluster will be deployed and configured.

2. Cookbook references

A complete OpenStack cookbook has been used, from which we will be able to download the Glance Chef cookbook. This OpenStack cookbook comprises components such as:
OpenStack Block Storage

- OpenStack Compute
- OpenStack Dashboard
- OpenStack Identity
- OpenStack Image
- OpenStack Network
- OpenStack Object Storage

Each of the OpenStack services has its own cookbook and can be used separately
OpenStack Cookbook URL:
https://github.com/stackforge/openstack-chef-repo.git

In this documentation, we will focus on the **Glance cookbook** only:
https://github.com/stackforge/cookbook-openstack-image.git

Important:
Depending on the cookbook you choose, cookbook configuration may be relatively different.

3. Dependencies & Pre-requisites

a. Database (MySQL)

Glance requires a database access to persist Glance images properties.

On MySQL server environment:

i. Set up firewall

Make sure database ports are open for incoming requests (3306 port for MySQL)

On Ubuntu you can disable firewall with this command:

sudo ufw disable

Or open a specific port by issuing:

sudo ufw allow 3306

ii. Configure MySQL endpoint

Ensure MySQL bind ip address will be accessible from outside (we should be able to connect to MySQL server from Keystone environment by issuing this command mysql –h mysql_host –u mysql_username –p
and enter mysql_user_pwd)

Verify and modify, if required, MySQL bind-address attribute in /etc/mysql/my.cnf :
bind-address = 10.125.0.15

Restart MySQL:

/etc/init.d/mysqld restart

iii. Create Glance database and grant access to Glance user

- **Connect to MySQL**

Connect to the database by running the command:

mysql –u mysql_username –p

- **Create Glance database**

mysql> CREATE DATABASE glance;

- **Encrypt mysql glance password (if required)**

```
select password('glance');
+-------------------------------------------+
| password('glance')                        |
+-------------------------------------------+
| *CC67CAF178CB9A07D756302E0BBFA3B0165DFD49 |
+-------------------------------------------+
```

- **Grant access to Glance user**

mysql>GRANT ALL PRIVILEGES ON *.* TO 'glance'@'%' IDENTIFIED BY PASSWORD 'encrypted_password' WITH GRANT OPTION;

mysql>FLUSH PRIVILEGES;

4. RabbitMQ pre-requisites

You need a RabbitMQ endpoint as Glance uses it as messaging system.
Check RabbitMQ documentation for further information.

5. Keystone pre-requisites

Cf. OpenStack Identity Keystone Chef Cookbook

6. CookBooks Import

Import OpenStack Image:

git clone https://github.com/stackforge/cookbook-openstack-image.git openstack-image

Edit openstack-image/attributes/default.rb file:

- Add the database endpoint:

```
default["openstack"]["db"]["image"]["host"] = "10.125.0.15"
```

- Change the image registry network interface:

```
default["openstack"]["image"]["registry"]["bind_interface"] = nil
```

to

```
default["openstack"]["image"]["registry"]["bind_interface"] = "eth1"
```

(eth1 network interface is used on our case to retrieve VMs external ip addresses, but it can be changed to eth0 for other deployments)

- Change the image API network interface:

```
default["openstack"]["image"]["api"]["bind_interface"] = nil
```

to

```
default["openstack"]["image"]["api"]["bind_interface"] =
"eth1"
```

(eth1 network interface is used on our case to retrieve VMs external ip addresses, but it can be changed to eth0 for other deployments)

- Add the Image registry host endpoint:

```
default['openstack']['endpoints']['image-registry']['host'] =
node["network"]["interfaces"]["eth1"]["addresses"].select
{|address, data| data["family"] == "inet" }.first.first
```

- Add Keystone attributes (change ip address and ports according to your own settings):

```
#Keystone attributes
default['openstack']['endpoints']['identity-admin']['host'] =
"10.125.0.11"
default['openstack']['endpoints']['identity-admin']['scheme'] =
```

```
"http"
default['openstack']['endpoints']['identity-admin']['port'] =
"35357"

default['openstack']['endpoints']['identity-api']['host'] =
"10.125.0.11"
default['openstack']['endpoints']['identity-api']['scheme'] = "http"
default['openstack']['endpoints']['identity-api']['port'] = "5000"
```

- Add Glance endpoints registered against Keystone : (change Glance ip address and port according to your Keystone settings):

```
#Glance Endpoint (Keystone Service Endpoint)
default['openstack']['endpoints']['image-api']['host'] =
"10.125.0.11"
default['openstack']['endpoints']['image-api']['port'] = "9292"
```

- Modify RabbitMQ attributes according to your RabbitMQ configuration:

```
default["openstack"]["image"]["rabbit"]["username"] = "guest"
default["openstack"]["image"]["rabbit"]["vhost"] = "/"
default["openstack"]["image"]["rabbit"]["port"] = 5672
default["openstack"]["image"]["rabbit"]["host"] = "10.125.0.19"
```

- If you don't want to use data bags to store/manage users passwords, add the following line to your default.rb file: (This is acceptable for testing purpose)

```
default["openstack"]["developer_mode"] = true
```

- Finally upload the openstack-image cookbook to chef server:

sudo knife cookbook import openstack-image

7. Prepare your Vagrant file

On the vagrant machine:

Create a Vagrant file with the following command:

vagrant init

```
Vagrant.require_plugin('vagrant-hostmanager')

domain = 'dell.com'

# Define the Glance cluster
nodes = [
  { :hostname => 'glance1', :ip => '10.125.0.12', :box => 'ubuntu1204-chef'},
  { :hostname => 'glance2', :ip => '10.125.0.13', :box => 'ubuntu1204-chef'},
  { :hostname => 'glance3', :ip => '10.125.0.14', :box => 'ubuntu1204-chef'}
]


VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|

  nodes.each do |node|
       config.vm.define node[:hostname] do |node_config|
       # configure the box, hostname and networking
       node_config.vm.box = node[:box]
       node_config.vm.hostname = node[:hostname] + '.' + domain
       node_config.vm.network :private_network, ip: node[:ip]

       node_config.vm.provision :chef_client do |chef|
          chef.chef_server_url = 'https://10.125.0.10'
          chef.validation_key_path = "chef-validator.pem"
          chef.add_recipe "apt"
          chef.add_recipe "mysql::client"
          chef.add_recipe "openstack-common"
          chef.add_recipe "openstack-image::registry"
          chef.add_recipe "openstack-image::api"
          chef.add_recipe "openstack-image::identity_registration"
       end
  end
 end
end
```

The chef client section needs to be cautiously configured:

In Vagrantfile, verify that all the following parameters are correctly assigned and chef recipes indicated in the same order as follows:

chef.chef_server_url = 'chef_server_url': Assign the chef server URL; make sure you can access it from the Vagrant machine.

chef.validation_key_path = "validation_key_path": For this specific attribute, copy the chef server validation key (chef-validator.pem for example) into your Vagrant machine and change the key path accordingly.

chef.add_recipe "apt": Run the apt recipe to make sure we will be able to run the latest apt packages later on

chef.add_recipe "mysql::client": mysql::client recipe needs to be run to install MySQL client before running keystone cookbook

chef.add_recipe "openstack-common ": Chef recipe to be run before performing any of the OpenStack component installation. Installs and configures common openstack recipes

chef.add_recipe " openstack-image::registry": Installs packages required for a Glance registry server

chef.add_recipe " openstack-image::api": Installs packages required for a Glance api server

chef.add_recipe " openstack-image::identity_registration": Registers Glance endpoints and service with Keystone

Save file. Then run:

vagrant up

The VM is provisioned and keystone is installed.

8. Glance Cookbook Overview

- **Registry recipe:**
  - Install Python packages
  - Install Glance packages
  - Restart Glance registry service
  - Create /etc/glance/glance-registry.conf from template
  - Synchronize database (glance-manage db_sync)
  - Create /etc/glance/glance-registry-paste.ini from template
  - Set Glance directories and files permissions
  - Restart Glance registry service
- **Api recipe:**
  - Installs python keystone packages
  - Upgrade Glance packages
  - Create /etc/glance/policy.json from template
  - Restart Glance API service
  - Create /etc/glance/glance-api.conf from template
  - Set Glance directories and files permissions
  - Create /etc/glance/glance-api-paste.ini from template
    - Restart Glance API service
  - Create /etc/glance/glance-cache.conf from template
    - Restart Glance API service
  - Create /etc/glance/glance-cache-paste.ini from template
    - Restart Glance API service
  - Create /etc/glance/glance-scrubber.conf from template
  - Configure glance-cache-pruner (to be run every 30min)
  - Configure glance-cache-cleaner (to be run once a day)
  - Create /etc/glance/glance-scrubber-paste.ini from template
  - Upload list of images if configured
- **Identity Registration recipe:**
  - On Keystone:
    - Register Glance Service
    - Register Glance Endpoint
    - Register Service Tenant
    - Register Glance User
    - Grant Admin role to Glance user for Service Tenant
  - If entries already exist, you will get logs that entries already exist but it will not throw any errors.

9. Cleanup

When you provision your Vagrant virtual machine with Chef server, it creates a new Chef "node" entry and Chef "client" entry on the Chef server, using the hostname of the machine. After you tear down your guest machine, you must explicitly delete these entries from the Chef server before you provision a new one with Chef server.

To do so, go to the chef server and run the following commands:

knife client delete glance_vm_host

knife node delete glance_vm_host

If you forget to do so, you will get the following error when Vagrant tries to provision the keystone VM with Chef client. Delete client and node entries for each cluster node.

```
INFO: HTTP Request Returned 409 Conflict: Client already exists
================================================================================
Chef encountered an error attempting to create the client "glanceehost1"
================================================================================
Authorization Error:
--------------------
Your validation client is not authorized to create the client for this node (HTTP 403)

Possible Causes:
----------------
* There may already be a client named "glancehost1"
* Your validation client (chef-validator) may have misconfigured authorization permissions.
```