KeyStone Chef Cookbook

1. Objective

The objective here is to automate the deployment of an OpenStack Identity (Keystone) cluster using Chef and Vagrant. In this scenario, a 3 node Keystone 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 Keystone 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 **Keystone cookbook** only:
https://github.com/stackforge/cookbook-openstack-identity.git

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

3. Database Configuration pre-requisite (focus on MySQL)

Keystone requires a database access to persist Keystone entries such as users, tenants, services and endpoints.

On MySQL server environment:

a. 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

b. 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 :
For example: bind-address = 10.125.0.15 *(MySQL ip address)*

Restart MySQL:

/etc/init.d/mysqld restart

c. Create Keystone database and Grant Access to Keystone user

Connect to the database by running the command:

mysql –u mysql_username –p

Create keystone database:

mysql> CREATE DATABASE keystone;

Grant access to keystone user for remote access:

Enter mysql_user_name password when prompted

Run the following command (don't forget to change user credentials accordingly):

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

    IDENTIFIED BY PASSWORD 'keystone_password'

    WITH GRANT OPTION;

FLUSH PRIVILEGES;
```

Keystone MySQL credentials: In this cookbook, the keystone MySQL user password is defined in the Keystone chef cookbook.

It can be specified either by a databag entry or in the case that we don't want to use databags, a method returns non-encrypted login/password for each OpenStack service.

Both login and password will have the same value as the service name.
For example: for Keystone service: credentials will be keystone/keystone.
In this scenario, this "clear login/password" authentication will be used by setting the following property to true:

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

Otherwise, you need to configure databags accordingly (this won't be described in this documentation)

Important:

If you get this error message when flushing privileges:

ERROR 1372 (HY000): Password hash should be a 41-digit hexadecimal number

when running the grant command it's because the password used is not encrypted.

To fix it, run the MySQL command with your mysql user password that needs to be encrypted:

```
mysql> select password('keystone_password');
```

Copy the encrypted value and replace your non encrypted password by the new one in your grant command.

Then run:

```
mysql> FLUSH PRIVILEGES;
```

Exit MySQL command line tool

4. Import Cookbooks for Keystone

On Chef server:

Go to the cookbook directory on the chef server:

Downloads all dependencies related to Keystone (execute commands in the same order as they appear)

Import apt:

git clone https://github.com/opscode-cookbooks/apt.git

sudo knife cookbook upload apt

Import open-ssl:

git clone https://github.com/opscode-cookbooks/openssl.git

sudo knife cookbook upload openssl

Import build-essential:

git clone https://github.com/opscode-cookbooks/build-essential.git

sudo knife cookbook upload build-essential

Import mysql:

git clone https://github.com/opscode-cookbooks/mysql.git

sudo knife cookbook upload mysql

Import aws: (related to database dependency)

git clone https://github.com/opscode-cookbooks/aws.git

sudo knife cookbook upload aws

Import postgreSQL: (related to database dependency)

git clone https://github.com/opscode-cookbooks/postgresql.git

sudo knife cookbook upload postgresql

Import xfs: (related to database dependency)

git clone https://github.com/opscode-cookbooks/xfs.git

sudo knife cookbook upload xfs

Import database:

git clone https://github.com/opscode-cookbooks/database.git

sudo knife cookbook upload database

Import sysctl:

git clone https://github.com/onehealth-cookbooks/sysctl.git

sudo knife cookbook upload sysctl

Import yum:

git clone https://github.com/opscode-cookbooks/yum.git

sudo knife cookbook upload yum

Import osops-utils:

git clone https://github.com/rcbops-cookbooks/osops-utils.git

sudo knife cookbook upload osops-utils

Import OpenStack common (mandatory to run OpenStack Identity cookbook):

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

sudo knife cookbook upload openstack-common

Import OpenStack-Identity (Keystone):

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

Edit the openstack-identity/attributes/default.rb file and add the following parameters:

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

    default["openstack"]["db"]["identity"]["host"] = "10.125.0.15"

    default["openstack"]["endpoints"]["identity-admin"]["host"] =
node["network"]["interfaces"]["eth1"]["addresses"].select {|address, data|
data["family"] == "inet" }.first.first
     default["openstack"]["endpoints"]["identity-admin"]["port"] = "35357"
     default["openstack"]["endpoints"]["identity-admin"]["scheme"] = "http"

     default["openstack"]["endpoints"]["identity-api"]["host"] =
node["network"]["interfaces"]["eth1"]["addresses"].select {|address, data|
data["family"] == "inet" }.first.first
     default["openstack"]["endpoints"]["identity-api"]["port"] = "5000"
     default["openstack"]["endpoints"]["identity-api"]["scheme"] = "http"
```

Change the line:

```
     default["openstack"]["identity"]["bind_interface"] = "lo"
          by
     default["openstack"]["identity"]["bind_interface"] = "eth1"
```

Developer Mode attributes:

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

This parameter has been set to true to user clear login/password credentials as authentication mechanism. Is false, we would have needed to set up databags.

Database endpoint attributes:

```
default["openstack"]["db"]["identity"]["host"] = "mysql_host" (MySQL hostname)
```

This attribute defines where MySQL (or any other database) is installed.
This attribute might need to be changed or defined somewhere else during the complete OpenStack set up.

Identity Admin Endpoint attributes:

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

It retrieves dynamically keystone node ip address from eth1 network interface

```
default["openstack"]["endpoints"]["identity-admin"]["port"] =
"keystone_admin_port"  (port: 35357)
```

```
default["openstack"]["endpoints"]["identity-admin"]["scheme"] =
"http_url_prefix" (https or https)
```

These 3 attributes are used to build the Keystone Admin endpoint as follows:

http_prefix:// keystone_host:keystone_admin_port/v2.0

Example: http://10.125.0.11:35357/v2.0

Identity API Endpoint attributes:

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

It retrieves dynamically keystone node ip address from eth1 network interface

```
 default["openstack"]["endpoints"]["identity-api"]["port"] = "keystone_api_port"
(port: 5000)
```

```
default["openstack"]["endpoints"]["identity-api"]["scheme"] = "http_url_prefix"
(https or https)
```

These 3 attributes are used to build the Keystone API endpoint as follows:

http_prefix://keystone_host:keystone_api_port/v2.0
Example: http://10.125.0.11:5000/v2.0

Bind Endpoint attributes:

```
default["openstack"]["identity"]["bind_interface"] = "eth1"
```

This is a key attribute since it will be used typically to allow remote connections and is designed to point to a valid network interface.

We must remove the "lo" value which basically points to the localhost interface and set it to "eth1" interface.

From this attribute, Keystone cookbook will be able to retrieve dynamically the Keystone VM ip address.

Save and finally upload OpenStack Identity cookbook by issuing:

sudo knife cookbook upload openstack-identity

Run the following command to verify that all chef cookbooks have been loaded to chef server:

sudo knife cookbook list

Important:

If you need to modify files in your cookbook directory (attributes, recipes, templates…) whereas your cookbook has been already uploaded on the server don't forget to delete it (on server) and upload it again in order to take into account modifications. To do so, after cookbook modifications run the following commands:

sudo knife cookbook delete cookbook_name

sudo knife cookbook upload cookbook_name

5. Vagrant file Configuration

On the vagrant machine:

Create a Vagrant file with the following command:

vagrant init

Edit Vagrantfile as follows:

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

domain = 'dell.com'

# Defines the Keystone cluster
nodes = [
  { :hostname => 'keystone1', :ip => '10.125.0.11', :box =>
'ubuntu1204-chef'},
  { :hostname => 'keystone2', :ip => '10.125.0.12', :box =>
'ubuntu1204-chef'},
  { :hostname => 'keystone3', :ip => '10.125.0.13', :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-identity::server"
           chef.add_recipe "openstack-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-identity::server": Keystone installation and configuration chef recipe

chef.add_recipe " openstack-identity::registration": Add users, tenants, roles, services and endpoints

Save file. Then run:

vagrant up

The VM is provisioned and keystone is installed.

6. Keystone cookbook overview

- Create "keystone" MySQL database and keystone MySQL user
- Install python-mysqldb package
- Install Keystone packages
- Install Keystone
- Remove existing db : /var/lib/keystone/keystone.db
- Synchronize database (keystone-manage db)
- Replace /etc/keystone/keystone.conf with keystone values defined in keystone/attributes/default.rb
- Manage logging (/etc/keystone/logging.conf)
- Register tenants
    - admin
    - service
- Register roles
    - admin
    - KeystoneAdmin
    - KeystoneServiceAdmin
    - Member
- Register users
    - admin
    - monitoring
- Assign roles to users
- Register Service
    - Keystone Identity Service (name: keystone, type: identity)
- Register Keystone Endpoint
- Create EC2 credentials for each user

7. 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 keystone_vm_host

knife node delete keystone_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

```
INFO: HTTP Request Returned 409 Conflict: Client already exists
================================================================================
Chef encountered an error attempting to create the client "keystonehost"
================================================================================
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 "keystonehost"
* Your validation client (chef-validator) may have misconfigured authorization permissions.
```