

Cinder Chef Cookbook

1. Objective

The objective here is to automate the deployment of an OpenStack Block Storage (Cinder) cluster using Chef and Vagrant. In this scenario, a 3 node Cinder 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 Cinder Chef cookbook. This OpenStack cookbook comprises components such as:

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

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 **Cinder cookbook** only:

<https://github.com/stackforge/cookbook-openstack-block-storage.git>

Important:

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

3. Dependencies & Pre-requisites

a. Database (MySQL)

Cinder requires a database access to persist volumes, backups and snapshots 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 Cinder database and grant access to Cinder user

- **Connect to MySQL**

Connect to the database by running the command:

mysql -u mysql_username -p

- **Create Cinder database**

`mysql> CREATE DATABASE cinder;`

- **Encrypt mysql cinder password (if required) (*credentials used are then cinder/cinder*)**

```
mysql> select password('cinder');
+-----+
| password('cinder') |
+-----+
| *028F8298C041368BA08A280AA8D1EF895CB68D5C |
+-----+
```

- **Grant access to Cinder user**

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

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 documentation

https://github.com/openstack-titanium-community/Identity/blob/master/configuration/KeyStone_Chef_Cookbook.pdf

6. Glance pre-requisites

Cf OpenStack Image Chef Cookbook documentation

https://github.com/openstack-titanium-community/Image/blob/master/configuration/Glance_Chef_Cookbook.pdf

7. CookBooks Import

On top of the other cookbooks you may already have imported for Keystone and Glance (cf Keystone and Glance documentation) you need to import the Openstack Block Storage, python and selinux cookbooks:

Import selinux:

```
git clone https://github.com/opscode-cookbooks/selinux.git
knife cookbook import selinux
```

Import python:

```
git clone https://github.com/opscode-cookbooks/python.git
knife cookbook import python
```

Import OpenStack Block Storage:

```
git clone https://github.com/stackforge/cookbook-openstack-block-storage.git openstack-block-storage
```

Edit openstack-block-storage attributes:

- Add the database endpoint (change database ip address accordingly):

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

- Add Glance API endpoints attributes:

```
default['openstack']['endpoints']['image-api']['host'] = "10.125.0.12"
default['openstack']['endpoints']['image-api']['scheme'] = "http"
default['openstack']['endpoints']['image-api']['port'] = "9292"
```

- Add Keystone admin & API endpoints attributes:

```
#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 Cinder Endpoint on Keystone (must point to Keystone ip address):

```
#Cinder Endpoint (Keystone)
default['openstack']['endpoints']['volume-api']['host'] =
node["network"]["interfaces"]["eth1"]["addresses"].select {|address,
data| data["family"] == "inet" }.first.first
```

- Edit RabbitMQ attributes and change values according to your RabbitMQ settings:

```
default["openstack"]["block-storage"]["rabbit"]["username"] = "guest"
default["openstack"]["block-storage"]["rabbit"]["vhost"] = "/"
default["openstack"]["block-storage"]["rabbit"]["port"] = 5672
default["openstack"]["block-storage"]["rabbit"]["host"] = "10.125.0.19"
default["openstack"]["block-storage"]["rabbit"]["ha"] = false
```

Important:

In the template used to build cinder.conf (templates/default/cinder.conf.erb), node["ipaddress"] is used to retrieve the node ip address. By default, eth0 is used as network interface and the node ip address is retrieved from this latter.

Unlike the way to change network interface bindings in other cookbooks, there is no way to change it from the attributes file. Then, depending on how network interfaces are configured, we may not get the correct ip address for this parameter. As a conclusion, in the case of Cinder installation and configuration, eth0 must be configured to return ip addresses accessible from outside.

/etc/cinder/cinder.conf

```
my_ip=<%= node["ipaddress"] %>
```

8. 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 Cinder cluster
nodes = [
  { :hostname => 'cinder1', :ip => '10.125.0.16', :box => 'ubuntu1204-chef'},
  { :hostname => 'cinder2', :ip => '10.125.0.17', :box => 'ubuntu1204-chef'},
  { :hostname => 'cinder3', :ip => '10.125.0.18', :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 "python"
        chef.add_recipe "mysql::client"
        chef.add_recipe "openstack-common"
        chef.add_recipe "openstack-block-storage::cinder-common"
        chef.add_recipe "openstack-block-storage::api"
        chef.add_recipe "openstack-block-storage::identity_registration"
        chef.add_recipe "openstack-block-storage::volume"
        chef.add_recipe "openstack-block-storage::scheduler"
      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 "python": This python package has to be installed to run the scheduler recipe as it requires pip packages included in python cookbook

chef.add_recipe "mysql::client": mysql::client recipe needs to be run to install MySQL client before running Cinder 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-block-storage::cinder-common": Defines the common pieces of repeated code from the other recipes

chef.add_recipe " openstack-block-storage::api ": Installs the cinder-api, sets up the cinder database, and cinder service/user/endpoints in Keystone.

chef.add_recipe " openstack-block-storage::identity_registration ": Registers cinder service/user/endpoints in keystone.

chef.add_recipe " openstack-block-storage::volume": Installs the cinder-volume service and sets up the iscsi helper

chef.add_recipe " openstack-block-storage::scheduler": Installs the cinder-scheduler service

Save file. Then run:

vagrant up

Our Cinder cluster is deployed and Cinder is installed.

9. Cinder Cookbook Overview

- **Cinder-common recipe:**
 - Install Cinder common packages
 - Create /etc/cinder/cinder.conf from template
 - Set Cinder directories and files permissions
- **Api recipe:**
 - Installs Cinder api packages
 - Synchronize database (cinder-manage db sync)
 - Create /etc/cinder/api-paste.ini from template
 - Restart Cinder API service
 - Create /etc/cinder/policy.json from template
 - Restart Cinder API service
- **Identity Registration recipe:**
 - On Keystone:
 - Register Cinder Volume Service
 - Register Cinder Volume Endpoint
 - Register Cinder Volume User
 - Grant Admin role to Cinder Volume user for Service Tenant
 - If entries already exist, you will get logs that entries already exist but it will not throw any errors.
- **Scheduler recipe:**
 - Install database python packages
 - Install and start cinder-scheduler service
- **Volume recipe:**
 - Install Cinder Volume packages
 - Configure Netapp and NFS if required
 - Install Cinder NFS packages
 - Restart cinder-volume and iscsi target services
 - Create /etc/tgt/targets.conf from template
 - Restart Cinder iscsi target service

10. 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 cinder_vm_host
```

```
knife node delete cinder_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 "cinderhost1"
=====
```

```
=====
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 "cinderhost1"
- * Your validation client (chef-validator) may have misconfigured authorization permissions.