**OpenStack+Ceph Deployment Steps**

**cobbler**

http://10.1.4.61/cobbler\_web/system/list

**Clear the chef node which will be installed with fresh operation system.**

Knife node list |grep n091

Knife node delete –y openstack.n091.1

knife client delete -y openstack.n091.1

knife client delete -y openstack.n091.1

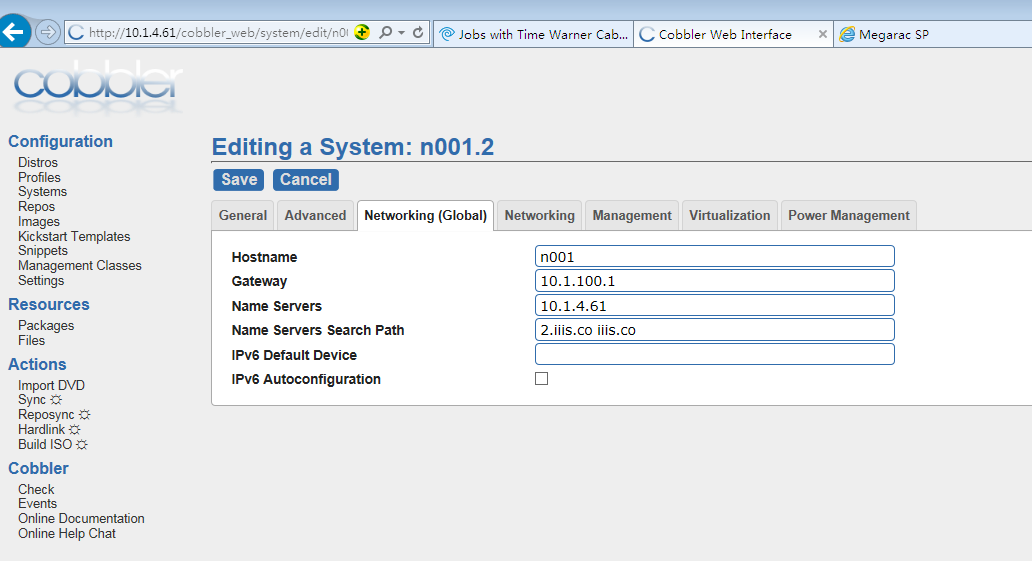
knife wrapper script: remove\_client.sh / remove\_node.sh

**Copy a node, edit it:**

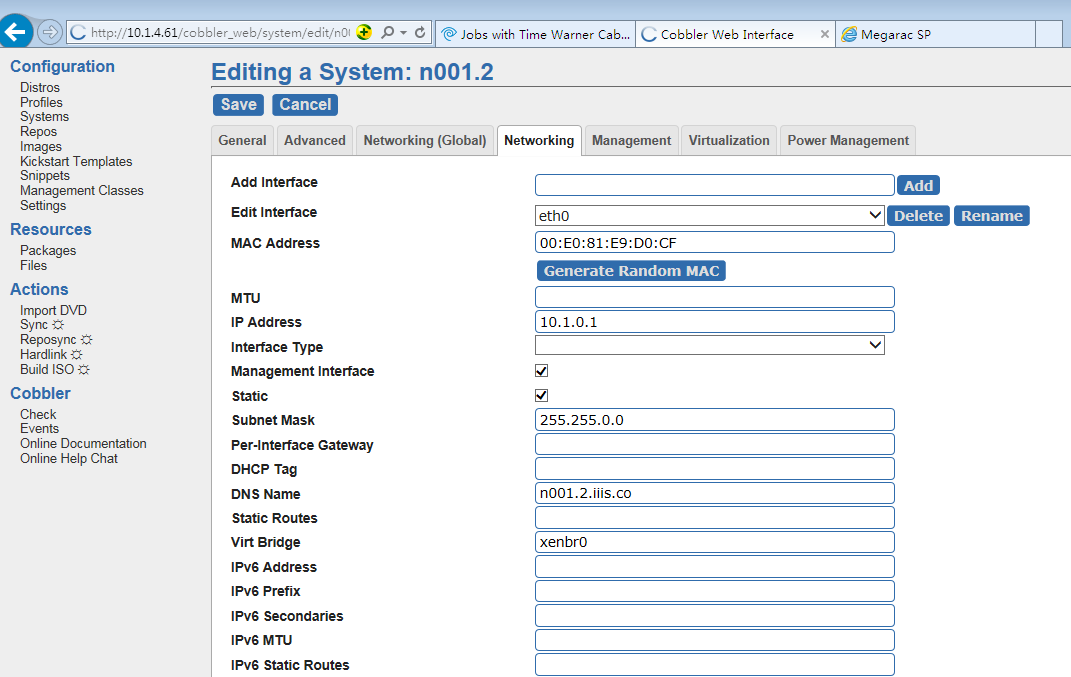
1.Tag: networking(global)

-- hostname: n001

-- name servers search path: 2.iiis.co



2.Tag: networking

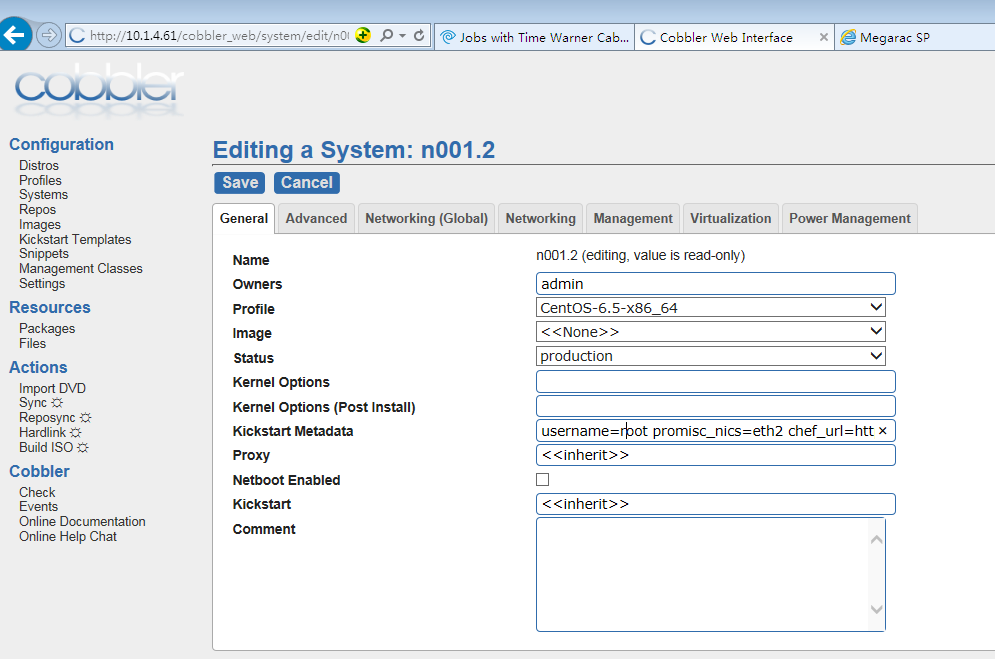


3.Tag: general

Kickstart metadata:

Update Chef\_node\_name

chef\_client\_name=openstack.n001.2 cluster\_databag=openstack\_2 chef\_node\_name=openstack.n001.2



**Check the dchp on cobbler:**

tcpdump -i eth0 port 68

pxe sends dhcp request to cobbler, check the output of tcpdump to debug.

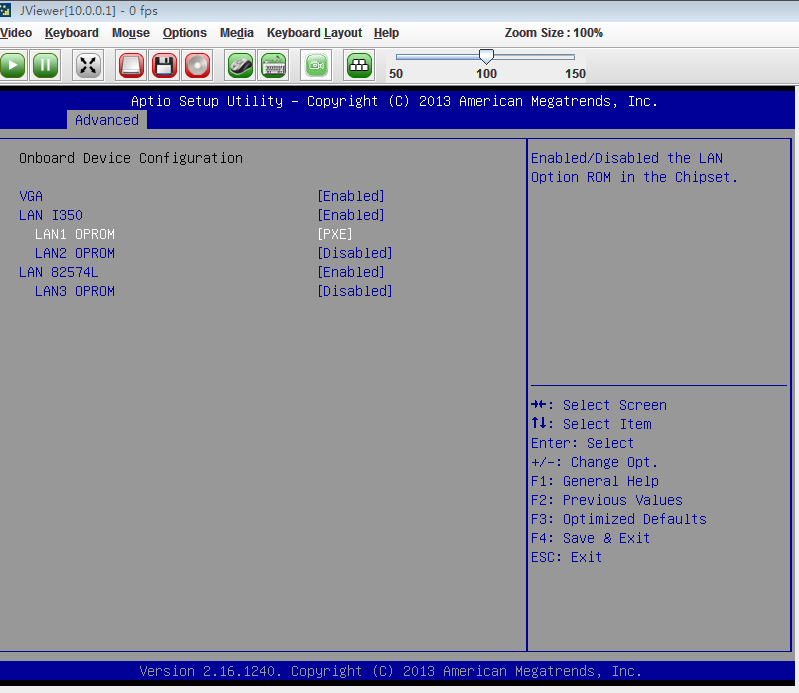
The “MAC address” on networking tag should be aligned with real value.

**BMC**

10.0.0.1

Launch console, Reset the machine

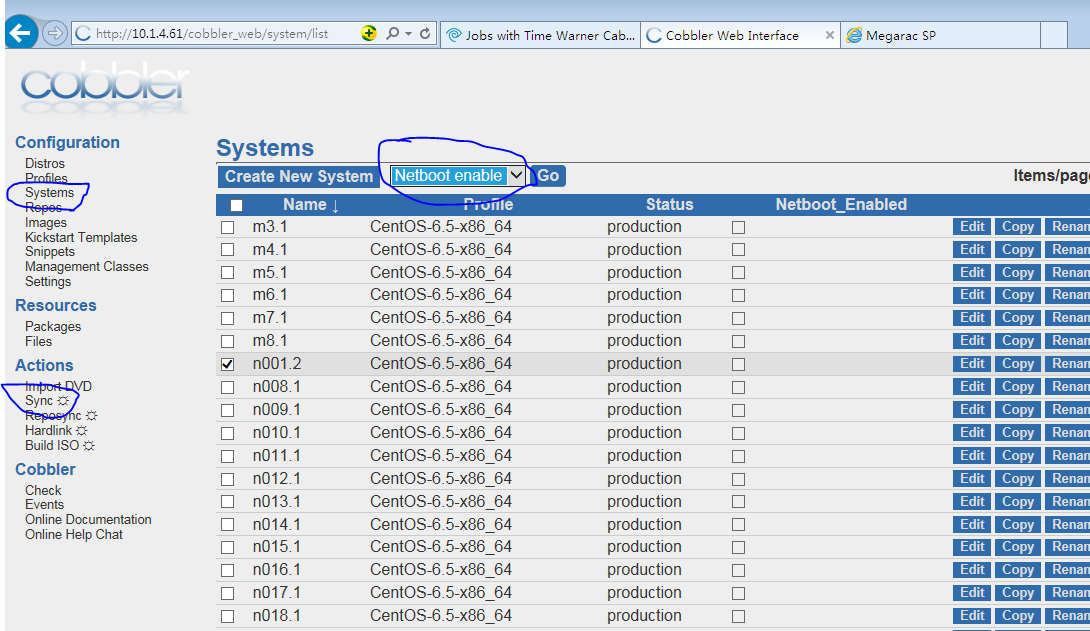
F2 enter into BIOS



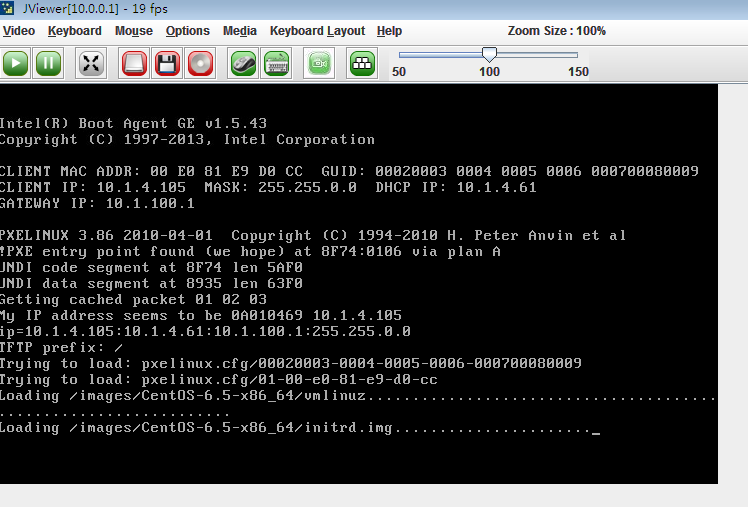
Disable unused NIC BOOT

Save and exit BIOS.

**Set “netboot” on cobbler for node**



Don’t forget “sync” the setting.



Operation System is installed now.

**OS preparation**

(The shell script uses BASH\_ARGV, we can only use bash as interpreter)

1. Update yum repo

**./repo\_update.sh 91**

1. Update knife node’s environment

# knife node show openstack.n091.2

Node Name: openstack.n091.2

Environment: \_default

FQDN: n091

IP: 10.1.0.91

Run List:

Roles:

Recipes:

Platform: centos 6.5

Tags:

**$ ./update\_environment.sh 91**

[10.1.0.91] Executing task 'chef\_client'

[10.1.0.91] run: chef-client -E tsinghua-ceph

[10.1.0.91] out: Starting Chef Client, version 11.8.0

[10.1.0.91] out: resolving cookbooks for run list: []

[10.1.0.91] out: Synchronizing Cookbooks:

[10.1.0.91] out: Compiling Cookbooks...

[10.1.0.91] out: Converging 0 resources

[10.1.0.91] out: Chef Client finished, 0 resources updated

Done.

Disconnecting from 10.1.0.91... done.

# knife node show openstack.n091.2

Node Name: openstack.n091.2

Environment: tsinghua-ceph

FQDN: n091

IP: 10.1.0.91

Run List:

Roles:

Recipes:

Platform: centos 6.5

Tags:

1. Install mallox NIC driver

**./ install\_nic\_driver.sh 91**

1. Prepare /var partition, reboot OS

**./install\_mount\_var.sh 91**

1. Remove varvol volume

**./install\_varvol.sh 91**

1. check the result varvol.sh script

**./check\_nic\_var.sh 91**

1. eth\_order.sh

adjust the eth\* order

1. Add compute role

**./ add\_compute\_role.sh 91**

1. Run chef-client on target node

**./update\_environment 91**

All the above script make use of Python module “Fabric”.

fab –u root –p root –H 10.1.0.91,10.1.0.92,10.1.0.93 -P repo\_update

fab –u root –p root –H 10.1.0.91,10.1.0.92,10.1.0.93 -P chef\_client

10.1.0.1~75: need to run install\_varvol.sh.

10.1.0.76~125: need to run all the above scripts.

**Install Openstack packages**

1. **Set environment**

Copy an environment，modify the IPs：

Haproxy ipaddress

Db/queue

Modify the name of environment:

"name": "tsinghua-ceph",

16 "ceph": {

17 "keystone environment": "tsinghua-ceph",

1. **Erlang\_solution baseurl**

"override\_attributes": {

"yum": {

"centos": {

"repo\_host": "10.1.4.65"

},

"erlang\_solutions": {

"baseurl": "http://10.1.4.65/erlang\_solutions/centos/6/$basearch",

"gpgkey": "http://10.1.4.65/erlang\_solutions/erlang\_solutions.asc"

}

1. **Create node**

Knife node create openstack.n005.2

// modify the environment from “\_default” to “Tsinghua-ceph”

Knife node edit

Knife node run\_list add openstack.n005.2 “role[base]”

1. **Update yum repository**
2. **Install db/queue node**

Install HA, run chef-client 3 times

Install controller node, when encounter glance issue, go back to HA node to run chef-client,then run chef-client on controller node again. Controller node is finished now.

Install network-worker node

Install compute node

1. Install first compute node

The first installed compute node will be “ssh\_keygen” node, the ssh stuff file of root/nova account on this node will be copied to all compute nodes.

All the compute node can ssh each other without password.

Node Name: openstack.n007.2

Environment: tsinghua-ceph

FQDN: n007

IP: 10.1.0.7

Run List: role[base], role[os-compute-worker], role[centos-base]

Roles: base, os-compute-worker, os-base, centos-base

Recipes: yum, net, openstack-common, openstack-common::logging, openstack-common::set\_endpoints\_by\_interface, openstack-common::sysctl, openstack-compute::compute, openssh::passwordless, openldap::auth, mountnfs, sudo

Platform: centos 6.5

Tags: ssh\_keygen

**Chef roles**

HA role:

'role[base],role[os-ha],role[centos-base]'

controller role:

role[base], role[os-identity], role[os-dashboard], role[os-image],role[os-compute-controller], role[os-network-server], role[os-block-storage-controller],role[centos-base]

compute role:

role[base], role[os-compute-worker], role[centos-base]

db/queue role:

role[base], role[os-ops-database], role[os-ops-messaging], role[centos-base]

network role:

role[base], role[os-network-worker],role[centos-base]

ceph monitor: run chef-client 3 times

role[base], role[os-compute-worker], role[centos-base], recipe[bridge], role[ceph-mon], role[ceph-osd], recipe[ceph::openstack\_config\_mon], recipe[openstack-compute::compute-config-ceph]

ceph: run chef-client 2 times

role[base], role[os-compute-worker], role[centos-base], recipe[bridge], role[ceph-osd], recipe[openstack-compute::compute-config-ceph]

ceph-controller:

role[base], role[os-identity], role[os-dashboard], role[os-image], role[os-compute-controller], role[os-network-server], role[os-block-storage-controller], role[centos-base], recipe[yum], recipe[yum-epel], recipe[net], role[os-block-storage-volume], recipe[openstack-block-storage::cinder-config-ceph], recipe[openstack-image::glance-config-ceph], recipe[openstack-object-storage::swift-config-ceph], recipe[openstack-object-storage::swiftclient-patch]

**Connect LDAP to OpenStack**

1. **Add LDAP setting to chef environment**

Knife environment edit Tsinghua-ceph

"identity": {

"identity": {

"backend": "ldap"

},

"ldap": {

"url": "ldap://10.1.0.201",

"user": "cn=Manager,dc=iiis,dc=co",

"password": "iiis123456",

"suffix": "dc=iiis,dc=co",

"use\_dumb\_member": "False",

"page\_size": 0,

"alias\_dereferencing": "default",

"query\_scope": "one",

"user\_tree\_dn": "ou=users,dc=iiis,dc=co",

"user\_objectclass": "organizationalPerson",

"user\_id\_attribute": "cn",

"user\_name\_attribute": "cn",

"user\_mail\_attribute": "email",

"user\_pass\_attribute": "userPassword",

"user\_domain\_id\_attribute": "businessCategory",

"user\_enabled\_mask": 0,

"user\_enabled\_default": "True",

"user\_attribute\_ignore": "tenant\_id,tenants",

"user\_allow\_create": "False",

"user\_allow\_update": "False",

"user\_allow\_delete": "False",

"user\_enabled\_emulation": "False",

"tenant\_objectclass": "groupOfNames",

"tenant\_id\_attribute": "cn",

"tenant\_member\_attribute": "member",

"tenant\_name\_attribute": "ou",

"tenant\_desc\_attribute": "description",

"tenant\_enabled\_attribute": "enabled",

"tenant\_domain\_id\_attribute": "businessCategory",

"tenant\_allow\_create": "False",

"tenant\_allow\_update": "False",

"tenant\_allow\_delete": "False",

"tenant\_enabled\_emulation": "False",

"role\_objectclass": "organizationalRole",

"role\_id\_attribute": "cn",

"role\_name\_attribute": "ou",

"role\_member\_attribute": "roleOccupant",

"role\_allow\_create": "False",

"role\_allow\_update": "False",

"role\_allow\_delete": "False",

"group\_objectclass": "groupOfNames",

"group\_id\_attribute": "cn",

"group\_name\_attribute": "ou",

"group\_member\_attribute": "member",

"group\_desc\_attribute": "description",

"group\_domain\_id\_attribute": "businessCategory",

"group\_allow\_create": "False",

"group\_allow\_update": "False",

"group\_allow\_delete": "False"

},

1. **Modify keystone source code**

def \_ldap\_res\_to\_model(self, res):

obj = self.model(id=self.\_dn\_to\_id(res[0]))

# LDAP attribute names may be returned in a different case than

# they are defined in the mapping, so we need to check for keys

# in a case-insensitive way. We use the case specified in the

# mapping for the model to ensure we have a predictable way of

# retrieving values later.

lower\_res = dict((k.lower(), v) for k, v in six.iteritems(res[1]))

for k in obj.known\_keys:

if k in self.attribute\_ignore:

continue

try:

**#v = lower\_res[self.attribute\_mapping.get(k, k).lower()]**

**v = lower\_res[self.attribute\_mapping.get(k, k)]**

except KeyError:

pass

else:

try:

obj[k] = v[0]

except IndexError:

obj[k] = None

return obj

def check\_allow\_create(self):

if not self.allow\_create:

action = \_('LDAP %s create') % self.options\_name

raise exception.ForbiddenAction(action=action)

def check\_allow\_update(self):

"/usr/lib/python2.6/site-packages/keystone/common/ldap/core.py" [Modified] line 497 of 1030 --48%-- col 62

1. **Restart keystone service**

service openstack-keystone restart

You can login on with admin account now.

1. **Run inittenant.py to create tenant**
2. **Create ext-net**

neutron net-create ext-net --router:external=True --shared

neutron subnet-create ext-net --allocation-pool start='10.2.3.0',end='10.2.4.254' --gateway='10.2.100.1' --enable\_dhcp=True --name ext-subnet --dns-nameserver 8.8.8.8 10.2.0.0/16

**6 . run initnet.sh （the value of OS\_AUTH\_URL and ext\_net\_id should be updated.）**

OpenStack + Ceph deployment is finished.

**Restore**

1. Import glance image

import\_glance\_image.sh

1. Import backup vm as glance image

import\_glance\_vm.sh

1. Import flavor

import\_flavor.sh

1. Import keypair

import\_keypair.sh

**Modify OpenStack Web UI IP address**

1. Change VIP in chef environment

"haproxy": {

"incoming\_address": "10.1.1.250"

},

"keepalived": {

"vip": {

"ipaddress": "10.1.1.250",

"interface": "eth0"

}

},

…

"endpoints": {

"db": {

"host": "10.1.0.76"

},

"mq": {

"host": "10.1.0.76"

},

"compute-api": {

"host": "10.1.1.250",

"scheme": "http",

"port": "8774",

"path": "/v2/%(tenant\_id)s"

},

…

1. Delete all endpoint

Keystone endpoint-delete

All the API will get endpoint info while getting token, the endpoint info contain the IP address of Web UI. The “keystone catalog” is sql backend, which will not be updated automated by chef cookbook. We should delete all the endpoint, the chef cookbook will register then again with updated Web UI IP.

1. Run chef-client on HA/controller node, then all the compute node
2. New Web UI takes effect now.

**Change IP allocation pool**

In RDO Icehouse release, the “neutron subnet-update” command cann’t work well. In order to change the allocation pool, we can update it in database directly.

# mysql -u root –p

mysql> use neutron

mysql> show tables;

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

| Tables\_in\_neutron |

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

| agents |

| allowedaddresspairs |

| dnsnameservers |

| externalnetworks |

| extradhcpopts |

| floatingips |

| ipallocationpools |

| ipallocations |

| ipavailabilityranges |

| ml2\_flat\_allocations |

| ml2\_gre\_allocations |

| ml2\_gre\_endpoints |

| ml2\_network\_segments |

| ml2\_port\_bindings |

| ml2\_vlan\_allocations |

| networkdhcpagentbindings |

| networks |

| ports |

| quotas |

| routerl3agentbindings |

| routerroutes |

| routers |

| securitygroupportbindings |

| securitygrouprules |

| securitygroups |

| subnetroutes |

| subnets |

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

27 rows in set (0.00 sec)

mysql> select \* from ipallocationpools where subnet\_id='ffa19acb-ff61-43a8-9d84-288cf8857f5c';

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

| id | subnet\_id | first\_ip | last\_ip |

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

| ec2576a9-f48d-4ac9-bcd5-b88808c700d0 | ffa19acb-ff61-43a8-9d84-288cf8857f5c | 10.2.1.1 | 10.2.4.254 |

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

1 row in set (0.00 sec)

mysql> update ipallocationpools set last\_ip='10.2.4.254' where subnet\_id='ffa19acb-ff61-43a8-9d84-288cf8857f5c';

TO-DO:

When shrink the allocation pool, the IP outside the pool can still be allocated. There may other tables which should be updated too.