OpenStack Icehouse

安装手册

# 目录

1 目录 1

2 目的 1

3 逻辑架构 2

4 物理架构 2

5 负载均衡haproxy、高可用keepalived搭建 2

6 mysql(MariaDB)-同步复制集群搭建 3

7 ceph搭建 3

8 安装控制节点 3

8.1 rabbitMq 4

8.2 Keystone 5

8.3 Glance 6

8.4 nova 8

8.5 neutron 11

8.6 dashboard 14

8.7 cinder 14

9 安装计算节点 16

9.1 Nova 16

9.2 neutron 18

10 测试 19

11 备注 20

# 目的

* 在openstack环境中，方便管理员搭建高可用的openstack环境
* 在项目实战中，能够灵活使用各个组件如haproxy、keepalived、rabbitmq.

# 逻辑架构

操作系统为ubuntu14.04.1,64位、Openstack版本为icehouse



# 物理架构

各机器的IP设置、网络规划，请参考《NewtouchX环境.docx》

（注：新的OpenStack物理节点不断添加中）



# 负载均衡haproxy、高可用keepalived搭建

请参考《HAProxy and Keepalived安装指导.docx》



实施建议：先搭建ceph、mariadb集群、控制节点、计算节点、最后做haproxy、keepalived

# mysql(MariaDB)-同步复制集群搭建

请参考《NewtouchX-OpenStack-Mysql(MariaDB)安装指南.docx》



# ceph搭建

请参考《NewtouchX-openstack-ceph集群部署.docx》



# 安装控制节点

控制节点两台，首先调整/etc/hosts配置，将所有的节点（控制节点、计算节点）配置上去

vi /etc/hosts

192.168.203.20 haproxy

192.168.203.21 haproxy1

192.168.203 22 haproxy2

192.168.203.23 controller1

192.168.203.25 controller2

192.168.203.41 compute1

192.168.203.42 compute2

192.168.203.43 compute3

192.168.203.44 compute4

192.168.203.45 compute5

192.168.203.46 compute6

192.168.203.47 compute7

192.168.203.48 compute8

192.168.203.49 compute9

192.168.203.50 compute10

192.168.203.51 compute11

192.168.203.52 compute12

192.168.203.53 compute13

192.168.203.54 compute14

192.168.203.11 ceph1

192.168.203.12 ceph2

192.168.203.15 ceph3

添加读权限：

# chmod 644 /etc/hosts

## RabbitMq

* 安装rabbitmq

apt-get install ntp rabbitmq-server

* 两台安装好后，先停止服务

service rabbitmq-server stop

* 同步cookie

scp /var/lib/rabbitmq/.erlang.cookie [root@192.168.203.25:/var/lib/rabbitmq/.erlang.cookie](mailto:root@192.168.203.25:/var/lib/rabbitmq/.erlang.cookie)

然后两台机器均重启rabbitmq服务:service rabbitmq-server start

备注:如果不同scp覆盖，注意两台机器的.erlang.cookie权限和所属用户、用户组须保持一致

* 在第二个节点将rabbitmq加入集群

rabbitmqctl stop\_app

rabbitmqctl join\_cluster rabbit@controller1

rabbitmqctl start\_app

* 两台机器上均可以查看集群状态

rabbitmqctl cluster\_status

Cluster status of node rabbit@controller1 ...

[{nodes,[{disc,[rabbit@controller1,rabbit@controller2]}]},

{running\_nodes,[rabbit@controller1,rabbit@controller2]},

{partitions,[]}]

...done.

* 镜像队列配置

设置以字母开始的队列都被镜像到集群中的所有节点上

rabbitmqctl set\_policy ha-all "^ [a-zA-Z]\." '{"ha-mode":"all"}'

*备注:其实在两个节点做好集群，还没有做mirrored queue镜像队列时，在slave节点上用命令“rabbitmqctl list\_queues”查看集群队列，其实查看到的是master节点上queue，假如停止master上的rabbitmq-server服务，在slave节点上就查看不到了。*

*做mirrored queue队列的作用就在于：定义policy策略，让slave能够把master上queue同步到slave本地，当master服务down了slave能够替代master提供服务。*

*具体信息可参考:* [*http://blog.sina.com.cn/s/blog\_959491260101m6ql.html*](http://blog.sina.com.cn/s/blog_959491260101m6ql.html)

* 增加配置

在两台安装rabbitmq的节点上添加配置：

vi /etc/rabbitmq/rabbitmq.config

*[{rabbit, [{loopback\_users, []}]}].*

然后分别重启两台rabbitmq-server服务

## Keystone

* 配置环境变量

vi /root/keystone/creds

*export OS\_SERVICE\_TOKEN=RDC*

*export OS\_SERVICE\_ENDPOINT=http://192.168.203.20:35357/v2.0*

*export OS\_USERNAME=admin*

*export OS\_PASSWORD=admin*

*export OS\_TENANT\_NAME=admin*

*export OS\_AUTH\_URL=http://192.168.203.20:35357/v2.0*

source /root/keystone/creds

* 安装keystone并修改相关配置

（两台控制节点配置保持一致）

apt-get install keystone python-mysqldb

vi /etc/keystone/keystone.conf

*[database]*

*connection = mysql://keystone:Service123@haproxy/keystone*

*[DEFAULT]*

*admin\_token=RDC*

在[DEFAULT]部分添加rabbitmq高可用配置：

*rabbit\_hosts=192.168.203.23:5672,192.168.203.25:5672*

*rabbit\_ha\_queues=true*

*备注:haproxy为两台负载均衡器上对外提供的虚拟IP对应的主机名*

* 创建keystone数据库用户,并授权

mysql -u root -p

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'Service123';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'Service123';

exit

* 初始化keystone数据库

su -s /bin/sh -c "keystone-manage db\_sync" keystone

* Keystone安装好后，其相关目录文件权限可能不是keystone用户的，下面要修改为Keystone用户权限：

chown -R keystone:keystone /etc/keystone

chown -R keystone:keystone /var/lib/keystone/

chown -R keystone:keystone /var/log/keystone/

* 重启keystone服务

service keystone restart

* 同步证书文件

将192.168.203.23上的keystone证书传输到192.168.203.25

scp -r /etc/keystone/ssl root@192.168.203.25:/etc/keystone

另外一台机器keystone重启:service keystone restart

如果需要可以到日志文件里面查看是否正常,日志文件路径在配置文件里已指定

* 创建keystone用户/服务/租户

keystone user-create --name=admin --pass=admin

keystone role-create --name=admin

keystone tenant-create --name=admin --description="Admin Tenant"

keystone user-role-add --user=admin --tenant=admin --role=admin

keystone user-role-add --user=admin --role=\_member\_ --tenant=admin

keystone user-create --name=demo --pass=keystone

keystone tenant-create --name=demo --description="Demo Tenant"

keystone user-role-add --user=demo --role=\_member\_ --tenant=demo

keystone tenant-create --name=service --description="Service Tenant"

keystone service-create --name=keystone --type=identity \

--description="OpenStack Identity"

keystone endpoint-create --service-id=$(keystone service-list | awk '/ identity / {print $2}') --publicurl=http://haproxy:5000/v2.0 --internalurl=http://haproxy:5000/v2.0 --adminurl=http://haproxy:35357/v2.0

* 将keystone的服务配置到haproxy1、haproxy2中(5000端口服务、35357端口服务)

请参考《HAProxy and Keepalived安装指导.docx》的配置

## Glance

* 安装glance服务并修改相关配置

apt-get install glance python-glanceclient

vi /etc/glance/glance-api.conf

*[DEFAULT]*

*default\_store = rbd*

*show\_image\_direct\_url = True*

*rbd\_store\_ceph\_conf = /etc/ceph/ceph.conf*

*rbd\_store\_pool = images*

*rbd\_store\_chunk\_size = 8*

*rabbit\_host = 192.168.203.20*

*rabbit\_port = 5672*

*[database]*

*#sqlite\_db = /var/lib/glance/glance.sqlite #将该行注释*

*connection = mysql://glance:Service123@haproxy/glance*

*[keystone\_authtoken]*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = glance*

*admin\_password = glance*

*[paste\_deploy]*

*flavor=keystone*

*#注: 如果不存储在ceph上，存储在controller上，则为如下的配置*

*default\_store = file*

*#how\_image\_direct\_url = True*

*#bd\_store\_ceph\_conf = /etc/ceph/ceph.conf*

*#bd\_store\_pool = images*

*#bd\_store\_chunk\_size = 8*

vi /etc/glance/glance-registry.conf

*[database]*

*connection = mysql://glance:Service123@haproxy/glance*

*[keystone\_authtoken]*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = glance*

*admin\_password = glance*

*[paste\_deploy]*

*flavor=keystone*

* 创建glance数据库用户,并授权

mysql -u root -p

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY 'Service123';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'Service123';

exit

* 初始化glance数据库

su -s /bin/sh -c "glance-manage db\_sync" glance

* 创建keystone用户/服务/租户

keystone user-create --name=glance --pass=glance

keystone user-role-add --user=glance --tenant=service --role=admin

keystone service-create --name=glance --type=image --description="OpenStack Image Service"

keystone endpoint-create --service-id=$(keystone service-list | awk '/ image / {print $2}') --publicurl=http://haproxy:9292 --internalurl=http://haproxy:9292 --adminurl=http://haproxy:9292

* 将glance的服务配置到haproxy1、haproxy2中(9292端口服务)
* Glance安装好后，其相关目录文件权限可能不是Glance用户的，下面要修改为Glance用户权限：

chown -R glance:glance /etc/glance

chown -R glance:glance /var/lib/glance/

chown -R glance:glance /var/log/glance/

* 重启服务

service glance-registry restart

service glance-api restart

* 验证

#glance index

## nova

* 安装nova

apt-get install nova-api nova-cert nova-conductor nova-consoleauth \

nova-novncproxy nova-scheduler python-novaclient python-guestfs

* 修改nova相关配置

vi /etc/nova/nova.conf

*[DEFAULT]*

*#*

*api\_rate\_limit = False*

*wait\_soft\_reboot\_seconds = 1*

*resume\_guests\_state\_on\_host\_boot = True*

*allow\_resize\_to\_same\_host = True*

*allow\_migrate\_to\_same\_host = True*

*#*

*metadata\_host = 192.168.203.20*

*metadata\_listen = 0.0.0.0*

*metadata\_listen\_port = 8775*

*metadata\_manager = nova.api.manager.MetadataManager*

*metadata\_port = 8775*

*metadata\_workers = 2*

*#*

*service\_neutron\_metadata\_proxy = true*

*neutron\_metadata\_proxy\_shared\_secret = helloworld*

*#*

*dhcpbridge\_flagfile=/etc/nova/nova.conf*

*dhcpbridge=/usr/bin/nova-dhcpbridge*

*logdir=/var/log/nova*

*state\_path=/var/lib/nova*

*lock\_path=/var/lock/nova*

*force\_dhcp\_release=True*

*iscsi\_helper=tgtadm*

*libvirt\_use\_virtio\_for\_bridges=True*

*connection\_type=libvirt*

*root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf*

*verbose=True*

*ec2\_private\_dns\_show\_ip=True*

*api\_paste\_config=/etc/nova/api-paste.ini*

*volumes\_path=/var/lib/nova/volumes*

*enabled\_apis=ec2,osapi\_compute,metadata*

*rpc\_backend = rabbit*

*#rabbit\_host = controller1*

*rabbit\_hosts = 192.168.203.23:5672,192.168.203.25:5672*

*rabbit\_ha\_queues=true*

*my\_ip = 192.168.203.23*

*vnc\_enabled = True*

*vncserver\_listen = 0.0.0.0*

*vncserver\_proxyclient\_address = 192.168.203.23*

*novncproxy\_base\_url = http://192.168.203.20:6080/vnc\_auto.html*

*auth\_strategy=keystone*

*glance\_host = 192.168.203.20*

*network\_api\_class = nova.network.neutronv2.api.API*

*neutron\_url = http://192.168.203.20:9696*

*neutron\_auth\_strategy = keystone*

*neutron\_admin\_tenant\_name = service*

*neutron\_admin\_username = neutron*

*neutron\_admin\_password = neutron*

*neutron\_admin\_auth\_url = http://192.168.203.20:35357/v2.0*

*linuxnet\_interface\_driver = nova.network.linux\_net.LinuxBridgeInterfaceDriver*

*firewall\_driver = nova.virt.firewall.NoopFirewallDriver*

*security\_group\_api = neutron*

*live\_migration\_bandwidth=0*

*live\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_LIVE*

*[libvirt]*

*libvirt\_images\_type=rbd*

*libvirt\_images\_rbd\_pool=volumes*

*libvirt\_images\_rbd\_ceph\_conf=/etc/ceph/ceph.conf*

*libvirt\_inject\_password=false*

*libvirt\_inject\_key=false*

*libvirt\_inject\_partition=-2*

*[database]*

*connection = mysql://nova:Service123@192.168.203.20/nova*

*[keystone\_authtoken]*

*auth\_uri = http://192.168.203.20:5000*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = nova*

*admin\_password = nova*

vi /etc/nova/nova-compute.conf

*[DEFAULT]*

*compute\_driver=libvirt.LibvirtDriver*

*[libvirt]*

*virt\_type=kvm*

*（注：如果是在物理机环境下搭建OpenStack，则virt\_type=kvm，如果在虚拟机上搭建OpenStack，则virt\_type=qemu）*

rm /var/lib/nova/nova.sqlite

* 修改libvirtd配置

apt-get install libvirt-bin

vi /etc/libvirt/libvirtd.conf

*listen\_tls = 0*

*listen\_tcp = 1*

*auth\_tcp = "none"*

vi /etc/default/libvirt-bin

*libvirtd\_opts="-d -l"*

/etc/init.d/libvirt-bin restart

* 创建nova用户以及授权

mysql -u root -p

CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY 'Service123';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'Service123';

exit

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

* 创建keystone用户/服务/租户

keystone user-create --name=nova --pass=nova

keystone user-role-add --user=nova --tenant=service --role=admin

keystone service-create --name=nova --type=compute \

--description="OpenStack Compute"

keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ compute / {print $2}') \

--publicurl=http://haproxy:8774/v2/%\(tenant\_id\)s \

--internalurl=http://haproxy:8774/v2/%\(tenant\_id\)s \

--adminurl=http://haproxy:8774/v2/%\(tenant\_id\)s

* 将控制节点的nova服务配置到haproxy(8774端口服务、8775端口服务、vnc:6080端口服务)
* Nova安装好后，其相关目录文件权限可能不是nova用户的，下面要修改为nova用户权限：

chown -R nova:nova /etc/nova

chown -R nova:nova /var/lib/nova/

chown -R nova:nova /var/log/nova/

* 重启nova服务

service nova-api restart

service nova-cert restart

service nova-consoleauth restart

service nova-scheduler restart

service nova-conductor restart

service nova-novncproxy restart

nova list

*+----+------+--------+------------+-------------+----------+*

*| ID | Name | Status | Task State | Power State | Networks |*

*+----+------+--------+------------+-------------+----------+*

*+----+------+--------+------------+-------------+----------+*

## neutron

* 安装neutron服务

apt-get install neutron-server neutron-plugin-ml2 neutron-plugin-ml2 neutron-plugin-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent neutron-lbaas-agent

* 创建neutron数据库用户

mysql -u root -p

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY 'Service123';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY 'Service123';

quit

* 创建keystone用户/服务/租户

keystone user-create --name neutron --pass neutron

keystone user-role-add --user neutron --tenant service --role admin

keystone service-create --name neutron --type network --description "OpenStack Networking"

keystone endpoint-create \

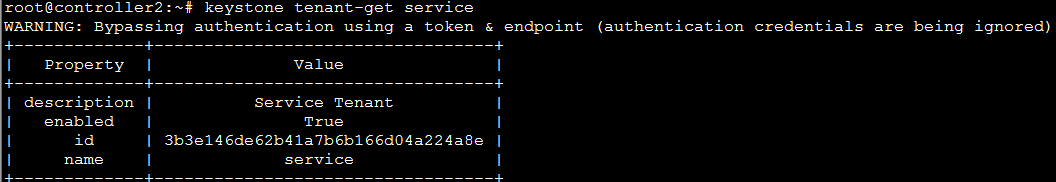
--service-id $(keystone service-list | awk '/ network / {print $2}') \

--publicurl http://haproxy:9696 \

--adminurl http://haproxy:9696 \

--internalurl http://haproxy:9696

* 获取service租户的id,用于增加到配置文件,如下:



* 修改neutron相关配置

vi /etc/neutron/neutron.conf

*[DEFAULT]*

*core\_plugin = ml2*

*service\_plugins = router,neutron.services.loadbalancer.plugin.LoadBalancerPlugin*

*api\_paste\_config = api-paste.ini*

*auth\_strategy = keystone*

*allow\_overlapping\_ips = True*

*notify\_nova\_on\_port\_status\_changes = True*

*notify\_nova\_on\_port\_data\_changes = True*

*nova\_url = http://192.168.203.20:8774/v2*

*nova\_admin\_username = nova*

*nova\_admin\_tenant\_id = 3b3e146de62b41a7b6b166d04a224a8e*

*nova\_admin\_password = nova*

*nova\_admin\_auth\_url =* [*http://192.168.203.20:35357/v2.0*](http://192.168.203.20:35357/v2.0)

*control\_exchange = neutron*

*rabbit\_hosts = 192.168.203.23:5672,192.168.203.25:5672*

*rabbit\_ha\_queues = true*

*[keystone\_authtoken]*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = neutron*

*admin\_password = neutron*

*signing\_dir = $state\_path/keystone-signing*

*[database]*

*connection = mysql://neutron:Service123@192.168.203.20/neutron*

vi /etc/neutron/plugins/ml2/ml2\_conf.ini

*[ml2]*

*tenant\_network\_types = vlan*

*mechanism\_drivers = linuxbridge*

*[ml2\_type\_flat]*

*flat\_networks = physnet1*

*[ml2\_type\_vlan]*

*network\_vlan\_ranges = physnet1:1000:2999*

vi /etc/neutron/l3\_agent.ini

*[DEFAULT]*

*interface\_driver = neutron.agent.linux.interface.BridgeInterfaceDriver*

*use\_namespaces = True*

*router\_id =*

*gateway\_external\_network\_id =*

*external\_network\_bridge =*

vi /etc/neutron/dhcp\_agent.ini

*[DEFAULT]*

*interface\_driver = neutron.agent.linux.interface.BridgeInterfaceDriver*

*use\_namespaces = True*

*enable\_isolated\_metadata = True*

*enable\_metadata\_network = True*

vi /etc/neutron/lbaas\_agent.ini

*interface\_driver = neutron.agent.linux.interface.BridgeInterfaceDriver*

*device\_driver =neutron.services.loadbalancer.drivers.haproxy.namespace\_driver.HaproxyNSDriver*

vi /etc/neutron/plugins/linuxbridge/linuxbridge\_conf.ini

*tenant\_network\_type = vlan*

*network\_vlan\_ranges = physnet1:1000:2999*

*physical\_interface\_mappings = physnet1:p33p1*

*firewall\_driver = neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver*

*enable\_security\_group = True*

vi /etc/neutron/metadata\_agent.ini

*auth\_url = http://192.168.203.20:5000/v2.0*

*auth\_region = RegionOne*

*admin\_tenant\_name = service*

*admin\_user = neutron*

*admin\_password = neutron*

*nova\_metadata\_ip = 192.168.203.20*

*nova\_metadata\_port = 8775*

*metadata\_proxy\_shared\_secret = helloworld*

*metadata\_workers = 2*

* 将neutorn相关的服务配置到haproxy(9696端口)
* Neutron安装好后，其相关目录文件权限可能不是Neutron用户的，下面要修改为Neutron用户权限：

chown -R neutron:neutron /etc/neutron

chown -R neutron:neutron /var/lib/neutron/

chown -R neutron:neutron /var/log/neutron/

* 重启neutron服务

service neutron-plugin-linuxbridge-agent restart

service neutron-l3-agent restart

service neutron-metadata-agent restart

service neutron-dhcp-agent restart

service neutron-server restart

neutron net-list

neutron agent-list

## dashboard

* 安装dashboard平台

apt-get install apache2 memcached libapache2-mod-wsgi openstack-dashboard

apt-get remove --purge openstack-dashboard-ubuntu-theme

* 将dashboard的web访问端口配置到haproxy(80端口)

## cinder

* 安装cinder服务

apt-get install cinder-api cinder-scheduler cinder-volume cinder-backup

* 编辑cinder配置文件

vi /etc/cinder/cinder.conf

*[DEFAULT]*

*rootwrap\_config = /etc/cinder/rootwrap.conf*

*api\_paste\_confg = /etc/cinder/api-paste.ini*

*iscsi\_helper = tgtadm*

*volume\_name\_template = volume-%s*

*volume\_group = cinder-volumes*

*verbose = True*

*auth\_strategy = keystone*

*state\_path = /var/lib/cinder*

*lock\_path = /var/lock/cinder*

*volumes\_dir = /var/lib/cinder/volumes*

*glance\_host = 192.168.203.20*

*volume\_driver=cinder.volume.drivers.rbd.RBDDriver*

*rbd\_pool=volumes*

*glance\_api\_version= 2*

*rbd\_ceph\_conf=/etc/ceph/ceph.conf*

*[database]*

*connection = mysql://cinder:Service123@192.168.203.20/cinder*

*[keystone\_authtoken]*

*auth\_uri = http://192.168.203.20:5000*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = cinder*

*admin\_password = cinder*

* 创建cinder数据库用户

mysql -u root -p

CREATE DATABASE cinder;

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY 'Service123';

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY 'Service123';

exit

* 初始化cinder数据库

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

* 创建keystone用户/服务/租户

keystone user-create --name=cinder --pass=cinder

keystone user-role-add --user=cinder --tenant=service --role=admin

keystone service-create --name=cinder --type=volume --description="OpenStack Block Storage"

keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ volume / {print $2}') \

--publicurl=http://haproxy:8776/v1/%\(tenant\_id\)s \

--internalurl=http://haproxy:8776/v1/%\(tenant\_id\)s \

--adminurl=http://haproxy:8776/v1/%\(tenant\_id\)s

keystone service-create --name=cinderv2 --type=volumev2 --description="OpenStack Block Storage v2"

keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ volumev2 / {print $2}') \

--publicurl=http://haproxy:8776/v2/%\(tenant\_id\)s \

--internalurl=http://haproxy:8776/v2/%\(tenant\_id\)s \

--adminurl=http://haproxy:8776/v2/%\(tenant\_id\)s

* 将cinder的相关服务配置到haproxy(8776端口)
* Cinder安装好后，其相关目录文件权限可能不是Cinder用户的，下面要修改为Cinder用户权限：

chown -R cinder:cinder /etc/cinder

chown -R cinder:cinder /var/lib/cinder/

chown -R cinder:cinder /var/log/cinder/

* 重启cinder服务

service cinder-scheduler restart

service cinder-api restart

service cinder-volume restart

* 测试cinder

cinder create --display-name myVolume 1

cinder delete myVolume

## 安装ceph客户端

在OpenStack所有节点上安装Ceph客户端

安装：

apt-get install ceph-common python-ceph

从ceph集群节点上拷贝ceph的配置文件：

scp -r ceph@ceph1:/etc/ceph /etc/

# 安装计算节点

计算节点可以安装若干台，首先调整/etc/hosts配置，将所有的节点(控制节点、计算节点)配置上去

/etc/hosts的配置与控制节点的一样。

## Nova

* 安装nova

apt-get install nova-compute-kvm python-guestfs

* 修改nova配置

vi /etc/nova/nova-compute.conf

*virt\_type=kvm*

vi /etc/nova/nova.conf

*[DEFAULT]*

*wait\_soft\_reboot\_seconds = 1*

*resume\_guests\_state\_on\_host\_boot = True*

*allow\_resize\_to\_same\_host = True*

*allow\_migrate\_to\_same\_host = True*

*#*

*dhcpbridge\_flagfile=/etc/nova/nova.conf*

*dhcpbridge=/usr/bin/nova-dhcpbridge*

*logdir=/var/log/nova*

*state\_path=/var/lib/nova*

*lock\_path=/var/lock/nova*

*force\_dhcp\_release=True*

*iscsi\_helper=tgtadm*

*libvirt\_use\_virtio\_for\_bridges=True*

*connection\_type=libvirt*

*root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf*

*verbose=True*

*ec2\_private\_dns\_show\_ip=True*

*api\_paste\_config=/etc/nova/api-paste.ini*

*volumes\_path=/var/lib/nova/volumes*

*enabled\_apis=ec2,osapi\_compute,metadata*

*rpc\_backend = rabbit*

*#rabbit\_host = controller1*

*rabbit\_hosts = 192.168.203.23:5672,192.168.203.25:5672*

*rabbit\_ha\_queues = true*

*my\_ip = 192.168.203.41*

*vnc\_enabled = True*

*vncserver\_listen = 0.0.0.0*

*vncserver\_proxyclient\_address = 192.168.203.41*

*novncproxy\_base\_url = http://192.168.203.20:6080/vnc\_auto.html*

*auth\_strategy=keystone*

*glance\_host = 192.168.203.20*

*network\_api\_class = nova.network.neutronv2.api.API*

*neutron\_url = http://192.168.203.20:9696*

*neutron\_auth\_strategy = keystone*

*neutron\_admin\_tenant\_name = service*

*neutron\_admin\_username = neutron*

*neutron\_admin\_password = neutorn*

*neutron\_admin\_auth\_url = http://192.168.203.20:35357/v2.0*

*linuxnet\_interface\_driver = nova.network.linux\_net.LinuxBridgeInterfaceDriver*

*firewall\_driver = nova.virt.firewall.NoopFirewallDriver*

*security\_group\_api = neutron*

*service\_neutron\_metadata\_proxy = true*

*neutron\_metadata\_proxy\_shared\_secret = helloworld*

*live\_migration\_bandwidth=0*

*live\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_LIVE*

*[libvirt]*

*libvirt\_images\_type=rbd*

*libvirt\_images\_rbd\_pool=volumes*

*libvirt\_images\_rbd\_ceph\_conf=/etc/ceph/ceph.conf*

*libvirt\_inject\_password=false*

*libvirt\_inject\_key=false*

*libvirt\_inject\_partition=-2*

*[database]*

*connection = mysql://nova:Service123@192.168.203.20/nova*

*[keystone\_authtoken]*

*auth\_uri = http://192.168.203.20:5000*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = nova*

*admin\_password = nova*

* 修改libvirtd配置

vi /etc/libvirt/libvirtd.conf

*listen\_tls = 0*

*listen\_tcp = 1*

*auth\_tcp = "none"*

vi /etc/default/libvirt-bin

*libvirtd\_opts="-d -l"*

* 修改nova相关文件的用户、用户组

*chown -R nova:nova /etc/nova*

*chown -R nova:nova /var/log/nova*

*chown -R nova:nova /var/lib/nova*

/etc/init.d/libvirt-bin restart

service nova-compute restart

## neutron

* 安装neutron

apt-get install neutron-plugin-linuxbridge-agent

* 修改neutron配置

vi /etc/neutron/neutron.conf

*[DEFAULT]*

*core\_plugin = ml2*

*service\_plugins = router,neutron.services.loadbalancer.plugin.LoadBalancerPlugin*

*rabbit\_hosts = 192.168.203.23:5672,192.168.203.25:5672*

*rabbit\_ha\_queues = true*

*allow\_overlapping\_ips = True*

*notify\_nova\_on\_port\_status\_changes = True*

*notify\_nova\_on\_port\_data\_changes = True*

*nova\_url = http://192.168.203.20:8774/v2*

*nova\_admin\_username = nova*

*nova\_admin\_tenant\_id = 3b3e146de62b41a7b6b166d04a224a8e*

*nova\_admin\_password = nova*

*nova\_admin\_auth\_url = http://192.168.203.20:35357/v2.0*

*[database]*

*connection = mysql://neutron:Service123@haproxy/neutron*

*[keystone\_authtoken]*

*auth\_host = 192.168.203.20*

*auth\_port = 35357*

*auth\_protocol = http*

*admin\_tenant\_name = service*

*admin\_user = neutron*

*admin\_password =neutorn*

*signing\_dir = $state\_path/keystone-signing*

vi /etc/neutron/plugins/linuxbridge/linuxbridge\_conf.ini

*tenant\_network\_type = vlan*

*network\_vlan\_ranges = physnet1:1000:2999*

*physical\_interface\_mappings = physnet1:p33p1*

*firewall\_driver = neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver*

*enable\_security\_group = True*

* 修改neutron相关文件的用户组、用户

*chown -R neutron:neutron /etc/neutron*

*chown -R neutron:neutron /var/log/neutron*

*chown -R neutron:neutron /var/lib/neutron*

service neutron-plugin-linuxbridge-agent restart

## 安装ceph客户端

在OpenStack所有节点上安装Ceph客户端

安装：

apt-get install ceph-common python-ceph

从ceph集群节点上拷贝ceph的配置文件：

scp -r ceph@ceph1:/etc/ceph /etc/

## 9.4 /etc/hosts添加读权限

# chmod 644 /etc/hosts

# 测试

* 下载Cirros测试镜像

wget <https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img>

* 将镜像加到入glance

glance image-create --name=cirros-0.3.0-x86\_64 --disk-format=raw --container-format=bare --is-public=True < cirros-0.3.0-x86\_64-disk.img

*备注:日志在/var/log/glance/api.log*

* 查看镜像

glance index

*ID Name Disk Format Container Format Size*

*------------------------------------ ------------------------------ -------------------- -------------------- --------------*

*65a8edf6-40cf-4da4-9d61-373ad7b2894c CentOS-6.4.img\_virt\_sysprep raw bare 10737418240*

*在各ceph节点查看:* *rbd ls -p images -l*

* 创建网络,公网使用eth0

neutron net-create ext\_net --provider:network\_type flat --provider:physical\_network physnet0 --router:external=True

neutron subnet-create --allocation-pool start=192.168.202.10,end=192.168.202.250 ext\_net 192.168.0.0/16 --gateway\_ip 192.168.0.6

neutron net-create net1

neutron subnet-create net1 11.11.11.0/24

neutron router-create router1

neutron router-interface-add 2cb7dd72-c41b-4169-ad21-aeba6146b7e8 22612924-3c32-479a-bd40-75febf27a5af

neutron router-gateway-set 2cb7dd72-c41b-4169-ad21-aeba6146b7e8 99736500-8dab-484e-af42-712400ca5a23

* 从镜像创建卷

cinder create --image-id 65a8edf6-40cf-4da4-9d61-373ad7b2894c --display-name v1 20

* 创建实例

nova boot --flavor 1 --block-device-mapping vda=350e6366-260b-486c-aba6-7cb18a8ca70d --nic net-id=507c6bed-9252-4af1-85ea-8afc6f9db476 test1

# 备注

配置nova用户双向ssh无密码rsa登录

在两台机器上执行:

usermod -d /var/lib/nova -s /bin/bash nova

passwd nova

在controller1执行:

su - nova

ssh-genkey -t rsa

ssh-copy-id -I .ssh/id\_rsa.pub nova@compute1

测试:

ssh -v nova@compute1

在compute1执行:

su - nova

ssh-genkey -t rsa

ssh-copy-id -I .ssh/id\_rsa.pub nova@controller1

测试:

ssh -v nova@controller1

解决volume从glance创建并引导的问题,如下:

/usr/lib/python2.7/dist-packages/cinder# vi image/glance.py

*output[attr] = getattr(image, attr)*

修改为:

*output[attr] = getattr(image, attr, None)*

参见:

<https://review.openstack.org/#/c/88089/>

live-migration时,提示cpu不兼容,提示如下:

 Instance launched has CPU info:  
{"vendor": "Intel", "model": "core2duo", "arch": "x86\_64", "features": ["lahf\_lm", "rdtscp", "lahf\_lm", "rdtscp", "lm", "nx", "syscall", "ssse3", "monitor", "pni", "sse2", "sse", "fxsr", "mmx", "clflush", "pse36", "pat", "cmov", "mca", "pge", "mtrr", "sep", "apic", "cx8", "mce", "pae", "msr", "tsc", "pse", "de", "vme", "fpu"], "topology": {"cores": 1, "threads": 1, "sockets": 1}}  
ERROR nova.virt.libvirt.driver [req-1ee12afe-c0ec-4d82-a6a5-3280407ca8e1 67ac350f0a164433ac342b4960300341 a3ace7f03db4449db2b57f95fadc94ea] CPU doesn't have compatibility.

XML error: CPU feature `lahf\_lm' specified more than once

解决如下:

<https://review.openstack.org/#/c/100632/>