**安装配置手册**

1. 部署环境

1.操作系统：Ubuntu 14.04

2.Openstack版本：juno

3.服务器配置: 最少4G，1个CPU

二．安装前部署

1.ntp安装

apt-get install ntpd

vim /etc/ntp.conf

server cn.pool.ntp.org iburst

restrict -4 default kod notrap nomodify

restrict -6 default kod notrap nomodify

2.源更新

a. vim /etc/apt/sources.list

deb http://mirrors.aliyun.com/ubuntu/ utopic main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ utopic-security main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ utopic-updates main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ utopic-proposed main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ utopic-backports main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ utopic main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ utopic-security main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ utopic-updates main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ utopic-proposed main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ utopic-backports main restricted universe multiverse

b.apt-get update

c.apt-get install ubuntu-cloud-keyring

d.echo "deb http://ubuntu-cloud.archive.canonical.com/ubuntu" \

"trusty-updates/juno main" > /etc/apt/sources.list.d/cloudarchive-juno.list

e.apt-get update && apt-get dist-upgrade –y

注意:必须重启机器

1. 安装数据库

a. apt-get install mariadb-server python-mysqldb –y

b. vim /etc/mysql/my.cnf

bind-address = 0.0.0.0

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

c. mysql\_secure\_installation (修改root密码，以及安全设置)

d. service mysql restart

1. 安装消息队列服务

apt-get install rabbitmq-server –y

rabbitmqctl change\_password guest guest

rabbitmqctl status | grep rabbit

vim /etc/rabbitmq/rabbitmq.config

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

service rabbitmq-server restart

1. 安装验证服务

1.数据库设置和配置文件配置

mysql -u root -p111111

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' \

IDENTIFIED BY '111111';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' \

IDENTIFIED BY '111111';

flush privileges;

apt-get install keystone python-keystoneclient –y

[DEFAULT]

admin\_token = 86d1a11c3ff724044538

verbose = True

[database]

connection = mysql://keystone:111111@controller/keystone

[token]

provider = keystone.token.providers.uuid.Provider

driver = keystone.token.persistence.backends.sql.Token

[revoke]

driver = keystone.contrib.revoke.backends.sql.Revoke

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

service keystone restart

rm -f /var/lib/keystone/keystone.db

(crontab -l -u keystone 2>&1 | grep -q token\_flush) || \

echo '@hourly /usr/bin/keystone-manage token\_flush >/var/log/keystone/keystone-tokenflush.log 2>&1' \

>> /var/spool/cron/crontabs/keystone

2.新建账户，创建api

export OS\_SERVICE\_TOKEN=86d1a11c3ff724044538

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

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

keystone user-create --name admin --pass 111111 --email tianyk@putao.com

keystone role-create --name admin

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

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

keystone user-create --name demo --tenant demo --pass 111111 --email tianyk@putao.com

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://controller:5000/v2.0 \

--internalurl http://controller:5000/v2.0 \

--adminurl http://controller:35357/v2.0 \

--region regionOne

3.验证用户

unset OS\_SERVICE\_TOKEN OS\_SERVICE\_ENDPOINT

keystone --os-tenant-name admin --os-username admin --os-password 111111 \

--os-auth-url http://controller:35357/v2.0 token-get

4.创建验证文件

vim admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=111111

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

1. 安装镜像服务

1.配置数据库，创建api,修改配置文件

mysql -u root -p111111

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' \

IDENTIFIED BY '111111';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' \

IDENTIFIED BY '111111';

flush privileges;

source admin-openrc.sh

keystone user-create --name glance --pass 111111

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://controller:9292 \

--internalurl http://controller:9292 \

--adminurl http://controller:9292 \

--region regionOne

apt-get install glance python-glanceclient -y

/etc/glance/glance-api.conf

[database]

connection = mysql://glance:111111@controller/glance

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = glance

admin\_password = 111111

[paste\_deploy]

flavor = keystone

[glance\_store]

default\_store = file

filesystem\_store\_datadir = /var/lib/glance/images/

[DEFAULT]

notification\_driver = noopverbose = True

/etc/glance/glance-registry.conf

[database]

connection = mysql://glance:111111@controller/glance

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = glance

admin\_password = 111111

[paste\_deploy]

flavor = keystone

[DEFAULT]

notification\_driver = noopverbose = True

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

service glance-registry restart

service glance-api restart

rm -f /var/lib/glance/glance.sqlite

2.新建镜像（下载或者上传镜像文件到本机）

glance image-create --name "debian-8.0.0" --file debian-8.0.0-openstack-amd64.qcow2 \

--disk-format qcow2 --container-format bare --is-public True –progress

1. 安装计算服务

1.配置数据库，创建api接口，修改配置文件

mysql -u root -p111111

CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' \

IDENTIFIED BY '111111';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' \

IDENTIFIED BY '111111';

flush privileges;

source admin-openrc.sh

keystone user-create --name nova --pass 111111

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://controller:8774/v2/%\(tenant\_id\)s \

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

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

--region regionOne

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

nova-novncproxy nova-scheduler python-novaclient –y

/etc/nova/nova.conf

[database]

connection = mysql://nova:111111@controller/nova

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = controller

rabbit\_password = guest

auth\_strategy = keystone

my\_ip = 172.20.0.154

vncserver\_listen = 172.20.0.154

vncserver\_proxyclient\_address = 172.20.0.154

verbose = True

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = nova

admin\_password = 111111

[glance]

host = controller

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

service nova-api restart

service nova-cert restart

service nova-consoleauth restart

service nova-scheduler restart

service nova-conductor restart

service nova-novncproxy restart

rm -f /var/lib/nova/nova.sqlite

至此controller配置了一个阶段，现在需要配置计算节点服务器，请查看network-compute配置手册

2.计算节点配置完成之后验证

**nova service-list**

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

| Id | Binary | Host | Zone | Status | State | Updated\_at | Disabled Reason |

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

| 1 | nova-cert | controller | internal | enabled | up | 2015-06-23T03:24:51.000000 | - |

| 2 | nova-consoleauth | controller | internal | enabled | up | 2015-06-23T03:24:51.000000 | - |

| 3 | nova-scheduler | controller | internal | enabled | up | 2015-06-23T03:24:48.000000 | - |

| 4 | nova-conductor | controller | internal | enabled | up | 2015-06-23T03:24:55.000000 | - |

| 6 | nova-compute | network-compute | nova | enabled | up | 2015-06-23T03:24:50.000000 | None |

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

nova image-list

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

| ID | Name | Status | Server |

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

| 07afc1c5-a303-4d89-a7e1-6c8519da78c0 | debian-8.0.0 | ACTIVE | |

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

1. 安装网络服务
2. 配置数据库，创建api接口，修改配置文件

mysql -u root -p111111

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' \

IDENTIFIED BY '111111';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' \

IDENTIFIED BY '111111';

flush privileges;

source admin-openrc.sh

keystone user-create --name neutron --pass 111111

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://controller:9696 \

--adminurl http://controller:9696 \

--internalurl http://controller:9696 \

--region regionOne

apt-get install neutron-server neutron-plugin-ml2 python-neutronclient -y

/etc/neutron/neutron.conf

[database]

connection = mysql://neutron:111111@controller/neutron

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = controller

rabbit\_password = guest

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

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

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

nova\_url = http://controller:8774/v2

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

nova\_region\_name = regionOne

nova\_admin\_username = nova

nova\_admin\_tenant\_id = 111111

nova\_admin\_password = 111111

verbose = True

nova\_admin\_tenant\_id查看方法如下:

####source admin-openrc.sh

####keystone tenant-get service

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = 111111

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

[ml2]

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

/etc/nova/nova.conf

[DEFAULT]

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

security\_group\_api = neutron

linuxnet\_interface\_driver = nova.network.linux\_net.LinuxOVSInterfaceDriver

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

[neutron]

url = http://controller:9696

auth\_strategy = keystone

admin\_auth\_url = http://controller:35357/v2.0

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = 111111

su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf \

--config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade juno" neutron

service nova-api restart

service nova-scheduler restart

service nova-conductor restart

service neutron-server restart

1. 验证是否安装成功网络服务

neutron ext-list

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

| alias | name |

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

| security-group | security-group |

| l3\_agent\_scheduler | L3 Agent Scheduler |

| ext-gw-mode | Neutron L3 Configurable external gateway mode |

| binding | Port Binding |

| provider | Provider Network |

| agent | agent |

| quotas | Quota management support |

| dhcp\_agent\_scheduler | DHCP Agent Scheduler |

| l3-ha | HA Router extension |

| multi-provider | Multi Provider Network |

| external-net | Neutron external network |

| router | Neutron L3 Router |

| allowed-address-pairs | Allowed Address Pairs |

| extraroute | Neutron Extra Route |

| extra\_dhcp\_opt | Neutron Extra DHCP opts |

| dvr | Distributed Virtual Router |

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

网络服务控制节点安装完成，配置网络节点，网络节点和计算节点我安装在一台机器上

1. 配置结束网络节点后，元数据配置需要在/etc/nova/nova.conf里面配置两行代码，并且重启nova-api

[neutron]

service\_metadata\_proxy = True

metadata\_proxy\_shared\_secret = *111111*

1. 添加虚拟网络，路由

外网网段一配置

neutron net-create ext-net --router:external True \

--provider:physical\_network external --provider:network\_type flat

neutron subnet-create ext-net --name ext-subnet \

--allocation-pool start=122.226.100.151,end=122.226.100.158 \

--disable-dhcp --gateway 122.226.100.145 122.226.100.144/28

neutron net-create demo-net

neutron subnet-create demo-net --name demo-subnet \

--gateway 172.16.0.1 172.16.0.0/24

neutron router-create demo-router

neutron router-interface-add demo-router demo-subnet

neutron router-gateway-set demo-router ext-net

外网网段二配置

neutron net-create end-net --router:external True \

--provider:physical\_network endternal --provider:network\_type flat

neutron subnet-create end-net --name end-subnet \

--allocation-pool start=172.20.0.180,end=172.20.0.200 \

--disable-dhcp --gateway 172.20.0.1 172.20.0.0/24

neutron net-create putao-net

neutron subnet-create putao-net --name putao-subnet \

--gateway 172.17.0.1 172.17.0.0/24

neutron router-create putao-router

neutron router-interface-add putao-router putao-subnet

neutron router-gateway-set putao-router end-net

1. 安装web服务

apt-get install openstack-dashboard apache2 \

libapache2-mod-wsgi memcached python-memcache -y

/etc/openstack-dashboard/local\_settings.py

OPENSTACK\_HOST = "controller"

service apache2 restart

service memcached restart