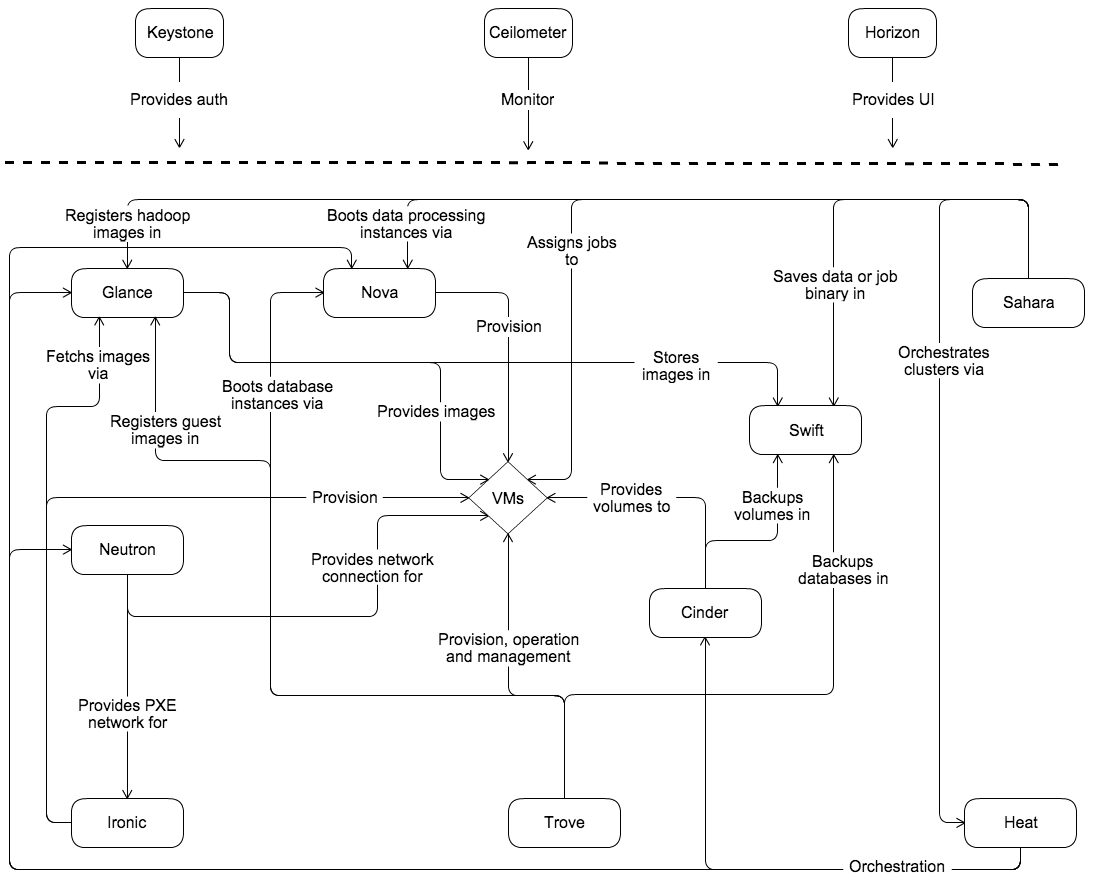
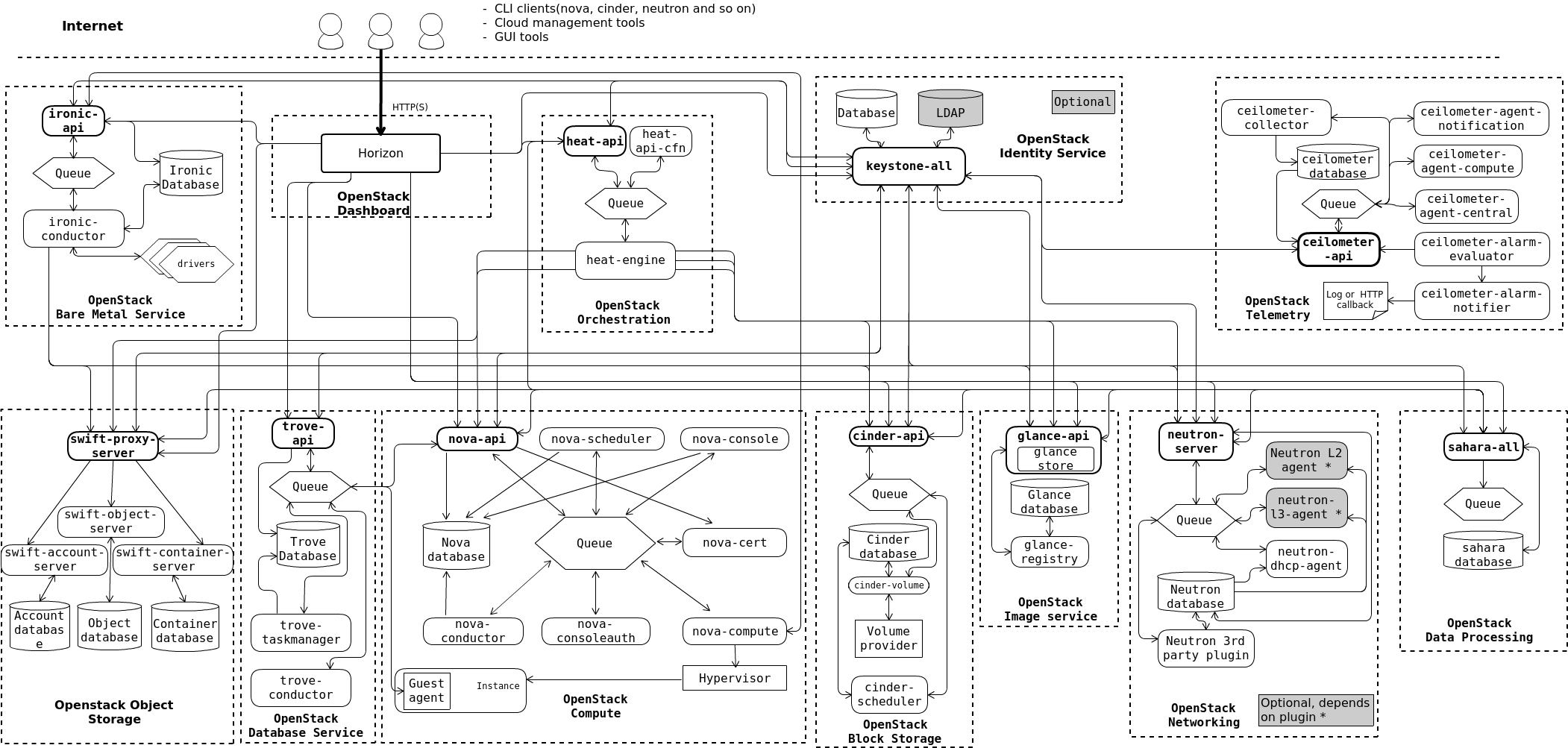
# 逻辑架构

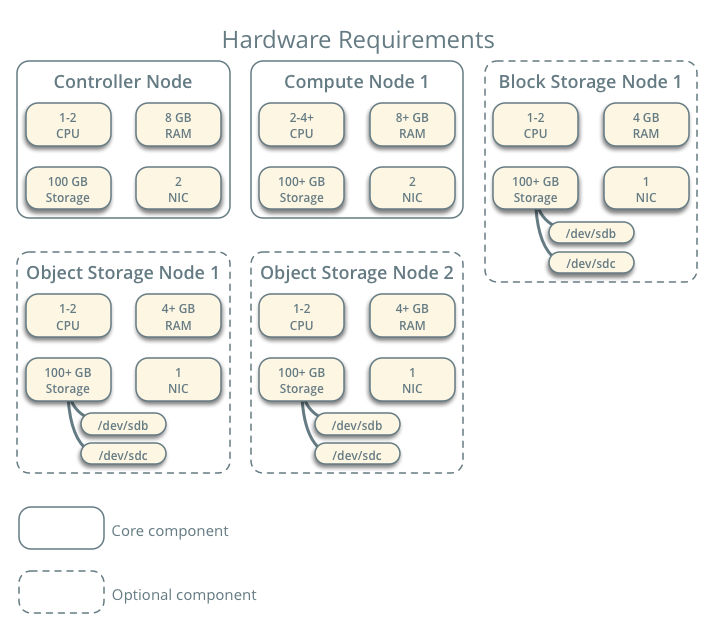




# 架构设计

参考: https://docs.openstack.org/arch-design

本地测试

* 本地测试最小资源需求，参考材料：https://docs.openstack.org/install-guide/overview.html
  + 

# 配置

## 准备基础资源和环境

创建虚拟机或者使用物理服务器。

以虚拟机为例，最低需求，1CPU，2核，50G存储，双网卡，一个桥接，一个NAT，这是最低配置的建议，如果低于这个配置，很多组件将无法安装使用。

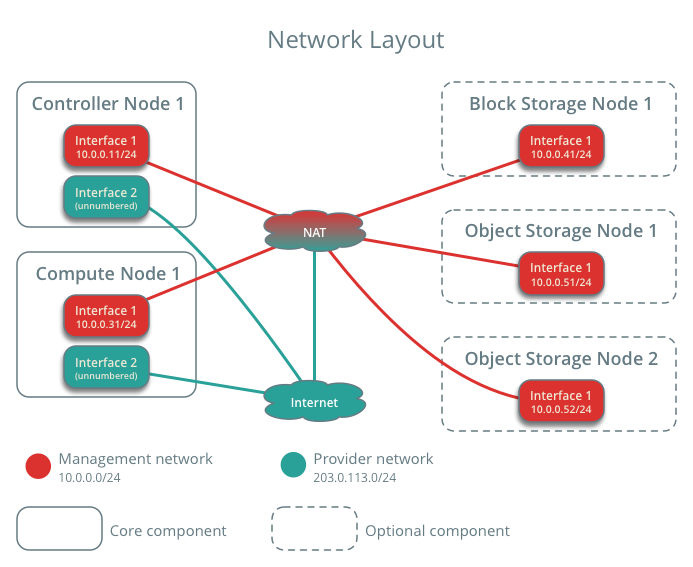
按照该最低配置，部署后的内容，不包含存储节点，不包含独立的网络节点，只有一个控制节点和计算节点。

物理服务器，也是双网卡，需要额外的交换机配置，通过vlan区分不同的网络即可。

安装系统可以使用centos7 64bit，最小化安装，关闭kdump，网卡配置IP，第一个网卡配置为管理网络使用的网卡，第二个网卡配置为业务网络使用的网卡。两个网卡的IP地址不同网段，不同vlan即可。

务必保证虚拟机或服务器可以访问到外网，因为安装需要访问外网下载安装包。

网卡样例示意图：



### 网卡配置

配置文件路径：/etc/sysconfig/network-scripts/ifcfg-INTERFACE\_NAME

检查以下配置：

DEVICE=INTERFACE\_NAME

TYPE=Ethernet

ONBOOT="yes"

BOOTPROTO="none"

这里需要学习过最低水准为RHCSA课程的人员进行操作。因为网卡名字是随机的，这里容易出现理解误差导致操作失败。

网卡中其他配置，如果没有明确的需求，请勿随意改动，IP地址务必自定义配置。请不要自动分配。

控制节点和计算节点都需要进行网卡配置。

### 配置hosts文件

文件路径：/etc/hosts

配置样例：

#127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

#::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

#controller

192.168.32.132 controller

#compute

192.168.32.133 compute

这里的IP地址是管理网络的IP地址。

控制节点和计算节点都需要进行一样的配置。

### 配置DNS

# vi /etc/resolv.conf

# Generated by NetworkManager

nameserver 114.114.114.114

nameserver 8.8.8.8

控制节点和计算节点都需要进行一样的配置。

### 验证网络连通性

控制节点：

# ping -c 4 docs.openstack.org

# ping -c 4 compute

计算节点：

# ping -c 4 openstack.org

# ping -c 4 controller

务必确保都可以ping通。

### 防火墙

# iptables -F

# systemctl stop firewalld

如果只是简单的进行使用测试，可以选择关闭防火墙。控制节点和计算节点都配置。

### NTP时钟

**控制节点：**

# yum install chrony

编辑配置文件 /etc/chrony.conf

server NTP\_SERVER iburst

默认情况下，控制节点就是NTP服务器，给其他节点提供时间同步，这里可以选择外部的NTP时钟源。

配置接入客户端的许可，允许其他节点向本节点申请时间同步。

# Allow NTP client access from local network.

#allow 192.168.0.0/16

allow 192.168.32.0/24

启动服务：

# systemctl enable chronyd.service

# systemctl start chronyd.service

验证

chronyc sources

**计算节点：**

# yum install chrony

编辑配置文件 /etc/chrony.conf

server controller iburst

启动服务

# systemctl enable chronyd.service

# systemctl start chronyd.service

验证

chronyc sources

### 添加OpenStack资源仓库

# yum install centos-release-openstack-rocky

在控制节点和计算节点都执行。

备注：如果出现安装失败，建议将base源换成国内的，例如华为和阿里的。

### 更新软件包

# yum upgrade

在控制节点和计算节点都执行。

### 重启节点

# reboot

在控制节点和计算节点都执行。

### 安装OpenStack客户端

# yum install python-openstackclient

在控制节点和计算节点都执行。

### 安装OpenStack的SELinux管理包

# yum install openstack-selinux

在控制节点和计算节点都执行。

### 安装数据库

只在控制节点操作。

# yum install mariadb mariadb-server python2-PyMySQL

编辑配置文件

vi /etc/my.cnf.d/openstack.cnf

[mysqld]

bind-address = 192.168.32.132

default-storage-engine = innodb

innodb\_file\_per\_table = on

max\_connections = 4096

collation-server = utf8\_general\_ci

character-set-server = utf8

192.168.32.132是指控制节点的管理网络的网卡上的IP

启动服务

# systemctl enable mariadb.service

# systemctl start mariadb.service

保护数据库

# mysql\_secure\_installation

用密码的方式保护数据库

### 安装配置消息队列

只在控制节点操作。

安装

# yum install rabbitmq-server

开启服务

# systemctl enable rabbitmq-server.service

# systemctl start rabbitmq-server.service

添加用户

# rabbitmqctl add\_user openstack qwe123456

qwe123456是rabbitmq的密码。

配置权限

# rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"

### 安装Memcached

只安装在控制节点

安装

# yum install memcached python-memcached

修改配置文件

vi /etc/sysconfig/memcached

OPTIONS="-l 127.0.0.1,::1,controller"

添加关于控制节点的信息，让其他节点可以通过管理网络访问控制节点的服务。

启动服务

# systemctl enable memcached.service

# systemctl start memcached.service

### 安装etcd

只安装在控制节点

安装

# yum install etcd

修改配置文件

# vi /etc/etcd/etcd.conf

#[Member]

ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"

ETCD\_LISTEN\_PEER\_URLS="http://10.0.0.11:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://10.0.0.11:2379"

ETCD\_NAME="controller"

#[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.0.0.11:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.0.0.11:2379"

ETCD\_INITIAL\_CLUSTER="controller=http://10.0.0.11:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster-01"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

10.0.0.11是控制节点管理网络的IP。

启动服务

# systemctl enable etcd

# systemctl start etcd

## 安装OpenStack服务

### 部署Identity service

在控制节点操作

配置数据库：

$ mysql -u root -p

CREATE DATABASE keystone;

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

IDENTIFIED BY 'qwe123456';

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

IDENTIFIED BY 'qwe123456';

exit

安装

# yum install openstack-keystone httpd mod\_wsgi

编辑配置文件,添加配置

vi /etc/keystone/keystone.conf

[database]

connection = mysql+pymysql://keystone:qwe123456@controller/keystone

[token]

provider = fernet

链接填充数据库

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

初始化密钥

# keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

# keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone

引导服务

# keystone-manage bootstrap --bootstrap-password qwe123456 \

--bootstrap-admin-url http://controller:5000/v3/ \

--bootstrap-internal-url http://controller:5000/v3/ \

--bootstrap-public-url http://controller:5000/v3/ \

--bootstrap-region-id RegionOne

配置apache http服务

vi /etc/httpd/conf/httpd.conf

ServerName controller

创建文件链接

# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/

启动服务

# systemctl enable httpd.service

# systemctl start httpd.service

配置管理员环境变量

$ export OS\_USERNAME=admin

$ export OS\_PASSWORD=qwe123456

$ export OS\_PROJECT\_NAME=admin

$ export OS\_USER\_DOMAIN\_NAME=Default

$ export OS\_PROJECT\_DOMAIN\_NAME=Default

$ export OS\_AUTH\_URL=http://controller:5000/v3

$ export OS\_IDENTITY\_API\_VERSION=3

这里可以写成一个文件。source命令执行以下。

配置OpenStack的域名

$ openstack domain create --description "An Example Domain" example

配置域名下归属的project

$ openstack project create --domain default \

--description "Service Project" service

$ openstack project create --domain default \

--description "Demo Project" myproject

配置用户

$ openstack user create --domain default \

--password-prompt myuser

创建角色

$ openstack role create myrole

给用户添加角色

$ openstack role add --project myproject --user myuser myrole

验证

取消环境变量中的密码

$ unset OS\_AUTH\_URL OS\_PASSWORD

验证admin用户

$ openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name admin --os-username admin token issue

验证其他用户

$ openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name myproject --os-username myuser token issue

编辑环境变量文件

$ vi admin-openrc

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=qwe123456

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

$ vi demo-openrc

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=myproject

export OS\_USERNAME=myuser

export OS\_PASSWORD=qwe123456

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

应用环境变量

$ . admin-openrc

查看认证的token

$ openstack token issue

### 部署Image service

在控制节点操作。

配置数据库

$ mysql -u root -p

CREATE DATABASE glance;

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

IDENTIFIED BY 'qwe123456';

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

IDENTIFIED BY 'qwe123456';

exit

应用环境变量

$ . admin-openrc

创建glance用户

$ openstack user create --domain default --password-prompt glance

添加角色

$ openstack role add --project service --user glance admin

创建glance服务

$ openstack service create --name glance \

--description "OpenStack Image" image

创建glance服务API endpoint

$ openstack endpoint create --region RegionOne \

image public <http://controller:9292>

$ openstack endpoint create --region RegionOne \

image internal http://controller:9292

$ openstack endpoint create --region RegionOne \

image admin <http://controller:9292>

安装glance

# yum install openstack-glance

修改配置文件

$ vi /etc/glance/glance-api.conf

[database]

connection = mysql+pymysql://glance:qwe123456@controller/glance

[keystone\_authtoken]

www\_authenticate\_uri = http://controller:5000

auth\_url = http://controller:5000

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = glance

password = qwe123456

[paste\_deploy]

flavor = keystone

[glance\_store]

# ...

stores = file,http

default\_store = file

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

在S版本，这个配置文件和组件将被撤掉

$ vi /etc/glance/glance-registry.conf

[database]

connection = mysql+pymysql://glance:qwe123456@controller/glance

[keystone\_authtoken]

www\_authenticate\_uri = http://controller:5000

auth\_url = http://controller:5000

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = glance

password = qwe123456

[paste\_deploy]

flavor = keystone

填写数据到数据库中

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

启动服务

# systemctl enable openstack-glance-api.service \

openstack-glance-registry.service

# systemctl start openstack-glance-api.service \

openstack-glance-registry.service

### 部署Compute service

#### 在控制节点操作

配置数据库

$ mysql -u root -p

CREATE DATABASE nova\_api;

CREATE DATABASE nova;

CREATE DATABASE nova\_cell0;

CREATE DATABASE placement;

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

IDENTIFIED BY 'qwe123456';

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

IDENTIFIED BY ' qwe123456';

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

IDENTIFIED BY ' qwe123456';

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

IDENTIFIED BY ' qwe123456';

GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'localhost' \

IDENTIFIED BY ' qwe123456';

GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'%' \

IDENTIFIED BY ' qwe123456';

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

IDENTIFIED BY ' qwe123456';

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

IDENTIFIED BY ' qwe123456';

应用环境变量

$ . admin-openrc

创建nova用户

$ openstack user create --domain default --password-prompt nova

添加角色

$ openstack role add --project service --user nova admin

创建服务

$ openstack service create --name nova \

--description "OpenStack Compute" compute

创建API endpoint

$ openstack endpoint create --region RegionOne \

compute public <http://controller:8774/v2.1>

$ openstack endpoint create --region RegionOne \

compute internal <http://controller:8774/v2.1>

$ openstack endpoint create --region RegionOne \

compute admin <http://controller:8774/v2.1>

$ openstack user create --domain default --password-prompt placement

$ openstack role add --project service --user placement admin

$ openstack service create --name placement \

--description "Placement API" placement

$ openstack endpoint create --region RegionOne \

placement public <http://controller:8778>

$ openstack endpoint create --region RegionOne \

placement internal <http://controller:8778>

$ openstack endpoint create --region RegionOne \

placement admin <http://controller:8778>

安装nova

# yum install openstack-nova-api openstack-nova-conductor \

openstack-nova-console openstack-nova-novncproxy \

openstack-nova-scheduler openstack-nova-placement-api

编辑配置文件

$ vi /etc/nova/nova.conf

[DEFAULT]

enabled\_apis = osapi\_compute,metadata

transport\_url = rabbit://openstack:qwe123456@controller

my\_ip = 192.168.32.132

use\_neutron = true

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

[api]

auth\_strategy = keystone

[api\_database]

connection = mysql+pymysql://nova:qwe123456@controller/nova\_api

[database]

connection = mysql+pymysql://nova:qwe123456@controller/nova

[devices]

[ephemeral\_storage\_encryption]

[filter\_scheduler]

[glance]

api\_servers = http://controller:9292

[keystone\_authtoken]

auth\_url = http://controller:5000/v3

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = nova

password = qwe123456

[oslo\_concurrency]

lock\_path = /var/lib/nova/tmp

[placement]

region\_name = RegionOne

project\_domain\_name = Default

project\_name = service

auth\_type = password

user\_domain\_name = Default

auth\_url = http://controller:5000/v3

username = placement

password = qwe123456

[placement\_database]

connection = mysql+pymysql://placement:qwe123456@controller/placement

[vnc]

enabled = true

server\_listen = $my\_ip

server\_proxyclient\_address = $my\_ip

$ vi /etc/httpd/conf.d/00-nova-placement-api.conf

<Directory /usr/bin>

<IfVersion >= 2.4>

Require all granted

</IfVersion>

<IfVersion < 2.4>

Order allow,deny

Allow from all

</IfVersion>

</Directory>

重启http服务

# systemctl restart httpd

同步数据

# su -s /bin/sh -c "nova-manage api\_db sync" nova

# su -s /bin/sh -c "nova-manage cell\_v2 map\_cell0" nova

# su -s /bin/sh -c "nova-manage cell\_v2 create\_cell --name=cell1 --verbose" nova

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

查看注册的结果

# su -s /bin/sh -c "nova-manage cell\_v2 list\_cells" nova

启动服务

# systemctl enable openstack-nova-api.service \

openstack-nova-consoleauth openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

# systemctl start openstack-nova-api.service \

openstack-nova-consoleauth openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

#### 在计算节点进行操作

安装

# yum install openstack-nova-compute

修改配置文件(192.168.32.132是控制节点的管理IP)

$ vi /etc/nova/nova.conf

[DEFAULT]

enabled\_apis = osapi\_compute,metadata

transport\_url = rabbit://openstack:qwe123456@controller

my\_ip = 192.168.32.132

use\_neutron = true

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

[api]

auth\_strategy = keystone

[keystone\_authtoken]

auth\_url = http://controller:5000/v3

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = Default

user\_domain\_name = Default

project\_name = service

username = nova

password = qwe123456

[vnc]

enabled = true

server\_listen = 0.0.0.0

server\_proxyclient\_address = $my\_ip

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

[glance]

api\_servers = <http://controller:9292>

[oslo\_concurrency]

lock\_path = /var/lib/nova/tmp

[placement]

region\_name = RegionOne

project\_domain\_name = Default

project\_name = service

auth\_type = password

user\_domain\_name = Default

auth\_url = http://controller:5000/v3

username = placement

password = qwe123456

检查节点是否支持硬件加速

$ egrep -c '(vmx|svm)' /proc/cpuinfo

如果返回的数值是0，编辑配置文件

$ vi /etc/nova/nova.conf

[libvirt]

virt\_type = qemu

如果返回的数值不是0，无需做任何操作。

启动服务

# systemctl enable libvirtd.service openstack-nova-compute.service

# systemctl start libvirtd.service openstack-nova-compute.service

备注：openstack-nova-compute.service可能会启动失败，在控制节点关闭防火墙服务即可。

# systemctl stop firewalld.service

# systemctl disable firewalld.service

在控制节点进行如下操作，进行计算节点的发现。

$ . admin-openrc

$ openstack compute service list --service nova-compute

# su -s /bin/sh -c "nova-manage cell\_v2 discover\_hosts --verbose" nova

编辑配置文件

$ vi /etc/nova/nova.conf

[scheduler]

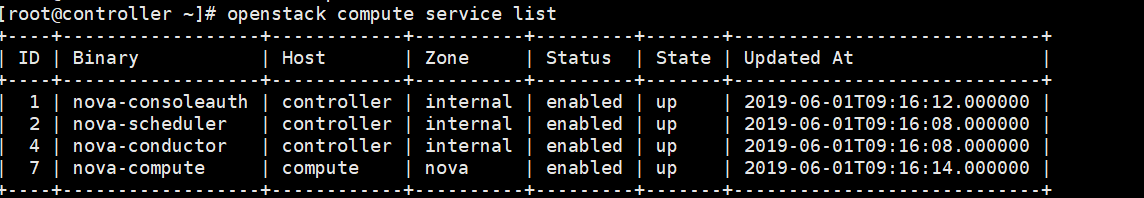
discover\_hosts\_in\_cells\_interval = 300

验证，在控制节点进行操作

$ . admin-openrc

$ openstack compute service list

可以看到类似图中所示的服务节点状态



$ openstack catalog list

可以看到已经安装的组件

$ openstack image list

可以看到注册的镜像

# nova-status upgrade check

检查现有组件状态

### 部署Networking service

#### 在控制节点操作

配置数据库

# mysql -u root -p

CREATE DATABASE neutron;

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

IDENTIFIED BY 'qwe123456';

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

IDENTIFIED BY ' qwe123456';

exit

应用环境变量

$ . admin-openrc

创建服务

$ openstack user create --domain default --password-prompt neutron

$ openstack role add --project service --user neutron admin

$ openstack service create --name neutron \

--description "OpenStack Networking" network

$ openstack endpoint create --region RegionOne \

network public <http://controller:9696>

$ openstack endpoint create --region RegionOne \

network internal <http://controller:9696>

$ openstack endpoint create --region RegionOne \

network admin <http://controller:9696>

网络类型，此处选择了Networking Option 2: Self-service networks

在控制节点操作

安装网络组件

# yum install openstack-neutron openstack-neutron-ml2 \

openstack-neutron-linuxbridge ebtables

编辑配置文件

$ vi /etc/neutron/neutron.conf

[database]

connection = mysql+pymysql://neutron:qwe123456@controller/neutron

[DEFAULT]

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = true

transport\_url = rabbit://openstack:qwe123456@controller

auth\_strategy = keystone

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

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

[keystone\_authtoken]

www\_authenticate\_uri = http://controller:5000

auth\_url = http://controller:5000

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = neutron

password = qwe123456

[nova]

auth\_url = http://controller:5000

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

region\_name = RegionOne

project\_name = service

username = nova

password = qwe123456

[oslo\_concurrency]

lock\_path = /var/lib/neutron/tmp

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

[ml2]

type\_drivers = flat,vlan,vxlan

tenant\_network\_types = vxlan

mechanism\_drivers = linuxbridge,l2population

extension\_drivers = port\_security

[ml2\_type\_flat]

flat\_networks = provider

[ml2\_type\_vxlan]

vni\_ranges = 1:1000

[securitygroup]

enable\_ipset = true

ens34是指业务网络的网卡，192.168.32.132是指管理网络的IP。

$ vi /etc/neutron/plugins/ml2/linuxbridge\_agent.ini

[linux\_bridge]

physical\_interface\_mappings = provider:ens34

[vxlan]

enable\_vxlan = true

local\_ip = 192.168.32.132

l2\_population = true

[securitygroup]

enable\_security\_group = true

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

$ vi /etc/sysctl.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

$ vi /etc/neutron/l3\_agent.ini

[DEFAULT]

interface\_driver = linuxbridge

$ vi /etc/neutron/dhcp\_agent.ini

[DEFAULT]

interface\_driver = linuxbridge

dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq

enable\_isolated\_metadata = true

$ vi /etc/neutron/metadata\_agent.ini

[DEFAULT]

nova\_metadata\_host = controller

metadata\_proxy\_shared\_secret = qwe123456

$ vi /etc/nova/nova.conf

[neutron]

url = http://controller:9696

auth\_url = http://controller:5000

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

region\_name = RegionOne

project\_name = service

username = neutron

password = qwe123456

service\_metadata\_proxy = true

metadata\_proxy\_shared\_secret = qwe123456

链接文件

# ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

同步数据库数据

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

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

启动服务

# systemctl restart openstack-nova-api.service

# systemctl enable neutron-server.service \

neutron-linuxbridge-agent.service neutron-dhcp-agent.service \

neutron-metadata-agent.service

# systemctl start neutron-server.service \

neutron-linuxbridge-agent.service neutron-dhcp-agent.service \

neutron-metadata-agent.service

# systemctl enable neutron-l3-agent.service

# systemctl start neutron-l3-agent.service

#### 在计算节点操作

安装

# yum install openstack-neutron-linuxbridge ebtables ipset

$ vi /etc/neutron/neutron.conf

[DEFAULT]

transport\_url = rabbit://openstack:qwe123456@controller

auth\_strategy = keystone

[keystone\_authtoken]

www\_authenticate\_uri = http://controller:5000

auth\_url = http://controller:5000

memcached\_servers = controller:11211

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = neutron

password = qwe123456

[oslo\_concurrency]

lock\_path = /var/lib/neutron/tmp

ens34是业务网络的网卡，192.167.32.133是管理网络的IP

$ vi /etc/neutron/plugins/ml2/linuxbridge\_agent.ini

[linux\_bridge]

physical\_interface\_mappings = provider:ens34

[vxlan]

enable\_vxlan = true

local\_ip = 192.167.32.133

l2\_population = true

[securitygroup]

enable\_security\_group = true

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

$ vi /etc/sysctl.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

$ vi /etc/nova/nova.conf

[neutron]

url = http://controller:9696

auth\_url = http://controller:5000

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

region\_name = RegionOne

project\_name = service

username = neutron

password = qwe123456

启动服务

# systemctl restart openstack-nova-compute.service

# systemctl enable neutron-linuxbridge-agent.service

# systemctl start neutron-linuxbridge-agent.service

### 部署Dashboard

在控制节点操作

# yum install openstack-dashboard

编辑配置文件

$ vi /etc/openstack-dashboard/local\_settings

OPENSTACK\_HOST = "controller"

ALLOWED\_HOSTS = ['\*']

SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': 'controller:11211',

}

}

需要注释掉原有的配置。

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "Default"

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

按照配置文件默认即可

TIME\_ZONE = "utc"

正常默认即可

$ vi /etc/httpd/conf.d/openstack-dashboard.conf

WSGIApplicationGroup %{GLOBAL}

启动服务

# systemctl restart httpd.service memcached.service

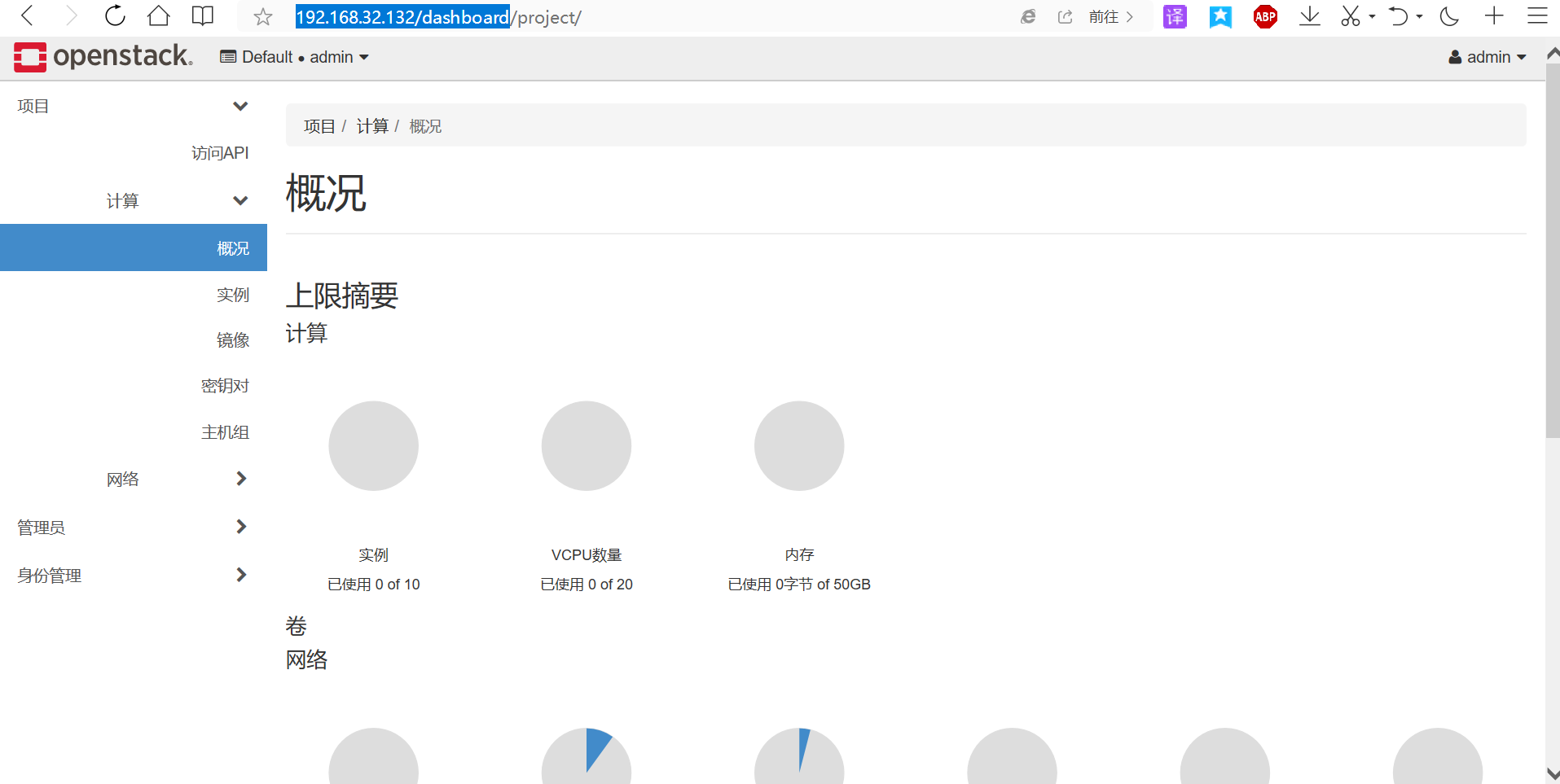
### 访问和使用

访问<http://192.168.32.132/dashboard>

域是default

用户是admin或者demo，密码是qwe123456

效果如图



部署以后，当前是没有存储可以使用的，需要部署cinder：https://docs.openstack.org/cinder/rocky/install/ 总体参考: https://docs.openstack.org/install-guide/openstack-services.html#minimal-deployment-for-rocky

### 部署Block Storage service

因为需要一个额外的存储节点，需要100G存储空间，以便进行存储使用的操作，当前本地资源不足，这部分暂时不做。

实验暂停。

如果没有存储节点，ECS（弹性伸缩实例，即云主机）是无法正常发放的。

安装结束后，参考<https://docs.openstack.org/install-guide/launch-instance.html> 进行测试

# 附录

1. 挂起虚拟机，但是重新开机以后，IP访问不到了。

答：重启虚拟机的网络服务即可，这是Vmware workstation的一个bug。