在CentOS7上安装OpenStack(Stein版)

原创 胡嚞衎 2019-09-29 17:32:21 4338 收藏 22 版权

分类专栏： Linux运维 文章标签： OpenStack Stein S版 CentOS 7

Linux运维 专栏收录该内容

文章目录

一、 前言

1. 在阅读本文前，确保您已经会使用VMware Workstation安装CentOS7。
2. 本文只所以会提到用VMware Workstation安装CentOS7是为了确保您和我的环境保持一致，否则您在安装OpenStack过程中可能会遇到麻烦。
3. 在安装OpenStack时有些操作必须使用root用户，有些可以使用其他用户；本文为了方便，均在root用户下执行命令。
4. 本文参考官方文档

二、准备虚拟机模板

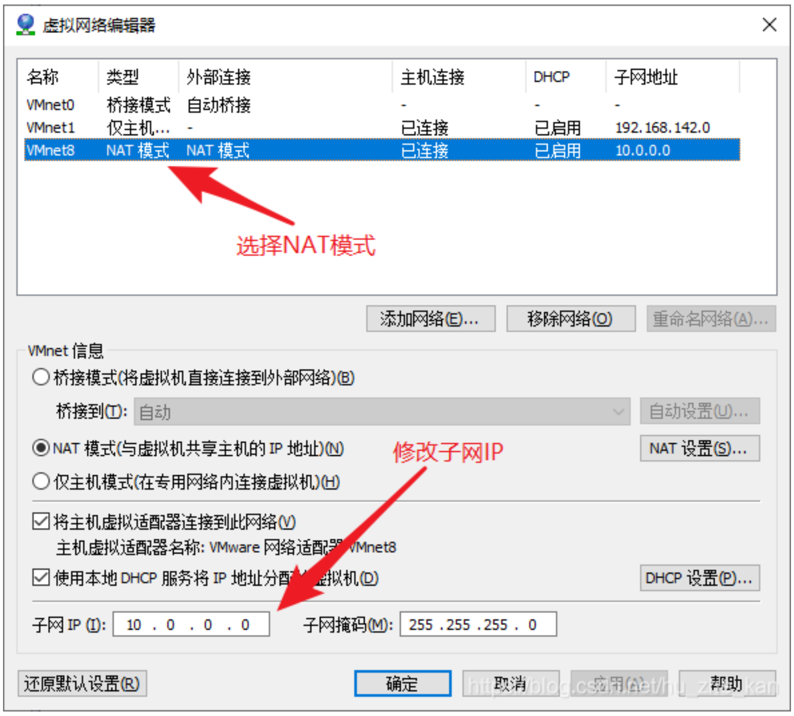
2.1、安装VMware

1. 准备一台PC机，并安装Windows 10操作系统。
2. 安装VMware Workstation 14 Pro

2.2、设置虚拟机网络

注意： 此步骤是在VMware上设置

- 修改NAT模式子网IP为：10.0.0.0
  编辑 -> 虚拟网络编辑器



虚拟网络编辑器

名称	类型	外部连接	主机连接	DHCP	子网地址
VMnet0	桥接模式	自动桥接	-	-	-
VMnet1	仅主机...	-	已连接	已启用	192.168.142.0
VMnet8	NAT 模式	NAT 模式	已连接	已启用	10.0.0.0

选择NAT模式

VMnet 信息

○ 桥接模式(将虚拟机直接连接到外部网络)(B)

桥接到(T): 自动

● NAT 模式(与虚拟机共享主机的 IP 地址)(N)

○ 仅主机模式(在专用网络内连接虚拟机)(H)

☑ 将主机虚拟适配器连接到此网络(V)

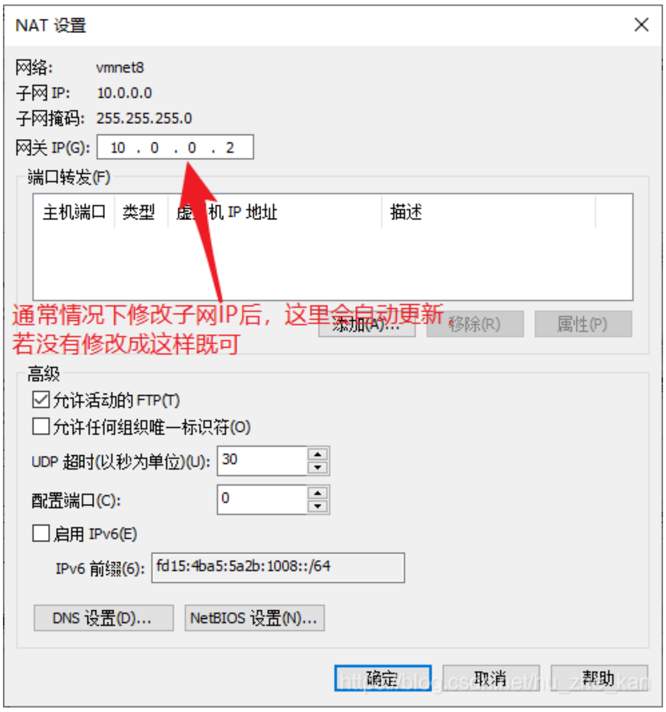
主机虚拟适配器名称: VMware 网络适配器 VMnet8

☑ 使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)

修改子网IP

子网 IP (I): 10 . 0 . 0 . 0 子网掩码(M): 255 .255 .255 . 0

- 查看NAT模式网关
  虚拟网络编辑器 -> NAT设置



NAT 设置

网络: vmnet8

子网 IP: 10.0.0.0

子网掩码: 255.255.255.0

网关 IP(G): 10 . 0 . 0 . 2

端口转发(F)

主机端口	类型	虚拟机 IP 地址	描述

通常情况下修改子网IP后，这里会自动更新
若没有修改成这样既可

高级

☑ 允许活动的 FTP(T)

☐ 允许任何组织唯一标识符(O)

UDP 超时(以秒为单位)(U): 30

配置端口(C): 0

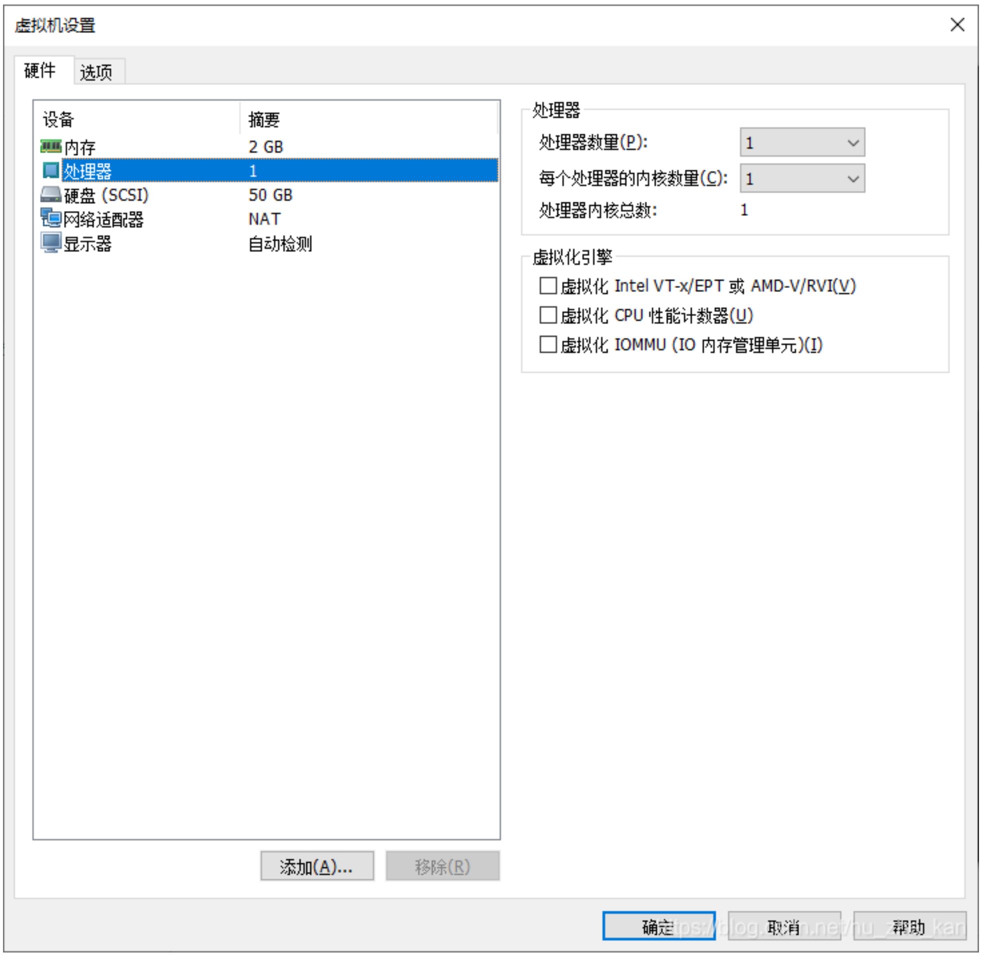
☐ 启用 IPv6(E)

IPv6 前缀(6): fd15:4ba5:5a2b:1008::/64

2.3、创建虚拟机

2.3.1、在VMware中新建虚拟机

创建一个虚拟机，设置如下：



虚拟机设置

设备	摘要
内存	2 GB
处理器	1
硬盘 (SCSI)	50 GB
网络适配器	NAT
显示器	自动检测

处理器

处理器数量(P): 1

每个处理器的内核数量(C): 1

处理器内核总数: 1

虚拟化引擎

□虚拟化 Intel VT-x/EPT 或 AMD-V/RVI(V)

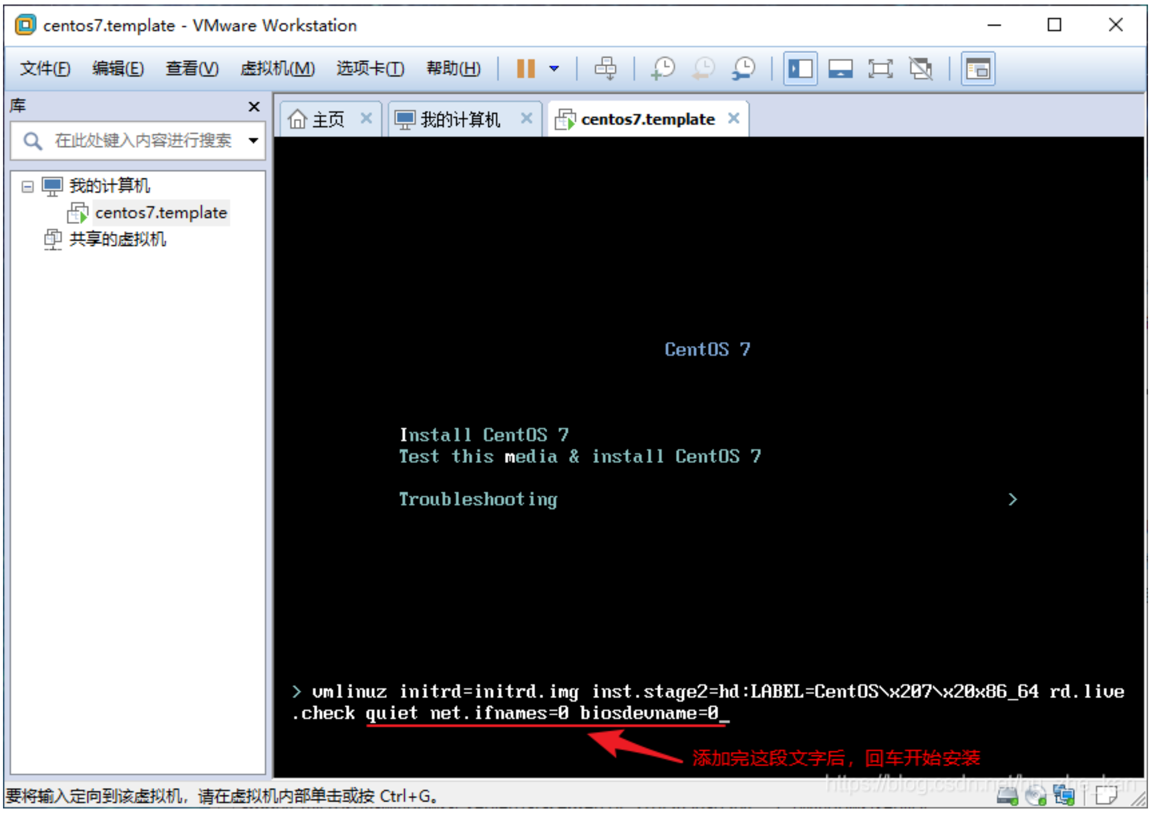
□虚拟化 CPU 性能计数器(U)

□虚拟化 IOMMU (IO 内存管理单元)(I)

2.3.2、安装CentOS 7

1. 设置网卡名称
OpenStack的很多服务都用到网卡名称，因此将这里的网卡名称统一修改为eth0避免不必要的麻烦。点击开机此虚拟机后进入安装系统界面，按

Tab键输入net.ifnames=0 biosdevname=0(如下图)



```
CentOS 7

Install CentOS 7
Test this media & install CentOS 7

Troubleshooting                                          >

> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=CentOS\x207\x20x86_64 rd.live
.check quiet net.ifnames=0 biosdevname=0_
```

添加完这段文字后，回车开始安装

要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

2. 设置时区
   选择东八区

2.4、配置模板机

输入root用户及密码，登录服务器后执行下列命令

```
# 1、修改主机名
echo 'centos7.template' > /etc/hostname

# 2、配置网卡
cat >/etc/sysconfig/network-scripts/ifcfg-eth0<<EOF
TYPE=Ethernet
BOOTPROTO=static
NAME=eth0
DEVICE=eth0
ONBOOT=yes
IPADDR=10.0.0.6
NETMASK=255.255.255.0
GATEWAY=10.0.0.2
EOF
systemctl restart network.service

# 3、关闭防火墙
systemctl stop firewalld.service
systemctl disable firewalld.service

# 4、关闭SeLinux
sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
grep -n 'SELINUX='  /etc/selinux/config

# 5、安装常用的软件包
yum install -y vim net-tools wget lrzsz tree screen lsof tcpdump nmap mlocate

# 6、命令提示符颜色
echo "PS1='[\[\e[31m\]\u\[\e[m\]@\[\e[36m\]\H\[\e[33m\] \W\[\e[m\]]\[\e[35m\]\\$ \[\e[m\]'" >>/etc/bashrc
source /etc/bashrc

```

执行完毕上述代码后，关闭该虚拟机。

三、准备一台控制节点虚拟机

3.1、配置虚拟机

1. 克隆一台虚拟机命名为controller
   OpenStack 主机名不能修改，一改就认为该计算机节点挂掉了。主机名非常重要，安装完OpenStack后千万不可擅自修改。

2. 修改配置
   将虚拟机内存修改为4GB，CPU设置为2核。

3. 修改IP

```
sed -i 's/10.0.0.6/10.0.0.11/g' /etc/sysconfig/network-scripts/ifcfg-eth0
systemctl restart network.service
```

4. 修改主机名

```
echo 'controller' > /etc/hostname  # reboot生效
hostname controller  # 退出session, 重新进入即可生效
```

5. 修改hosts文件

```
cat >>/etc/hosts<<EOF

# controller
10.0.0.11    controller

# compute
10.0.0.12    compute
EOF
```

3.2、安装基础服务

1. 启用OpenStack库

```
yum install -y centos-release-openstack-stein
```

2. 安装 OpenStack 客户端

```
yum install -y python-openstackclient
```

3. 安装时间同步服务

```
# 1、安装软件包
yum install -y chrony

# 2、允许其他节点可以连接到控制节点的 chrony 后台进程
echo 'allow 10.0.0.0/24' >> /etc/chrony.conf

# 3、启动 NTP 服务并将其配置为随系统启动
systemctl enable chronyd.service
systemctl start chronyd.service

# 4、设置时区
timedatectl set-timezone Asia/Shanghai

# 5、查询时间
timedatectl status
```

4. 安装MariaDB

```
# 1、安装软件包
yum install -y mariadb mariadb-server MySQL-python

# 2、配置
vim /etc/my.cnf.d/mariadb-server.cnf #在mysqld模块下放入一下几行
default-storage-engine = innodb
innodb_file_per_table = on
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8

# 3、启动数据库服务，并将其配置为开机自启
systemctl start mariadb.service
systemctl enable mariadb.service

# 4、对数据库进行安全加固(设置root用户密码)
mysql_secure_installation
```

5. 安装Memcache

```
# 1、安装软件包
yum install -y memcached python-memcached


# 2、修改监听ip
sed -i 's/127.0.0.1/0.0.0.0/' /etc/sysconfig/memcached

# 3、启动并加入开机自启
systemctl start memcached.service
systemctl enable memcached.service

#4、测试
printf "set foo 0 0 3\r\nbar\r\n"|nc controller 11211  # 添加数据
printf "get foo\r\n"|nc controller 11211  # 获取数据,在计算节点上也测试下
```

6. 安装消息队列

```
# 1、安装
yum install -y rabbitmq-server

# 2、启动
systemctl enable rabbitmq-server.service
systemctl start rabbitmq-server.service

# 3、创建用户
rabbitmqctl add_user openstack openstack

# 4、授权
rabbitmqctl set_permissions openstack ".*" ".*" ".*"

# 5、启用web管理界面
rabbitmq-plugins list  # 查看rabbitmq有哪些插件
rabbitmq-plugins enable rabbitmq_management  # 启用web管理界面

# 6、浏览器上登录
# 在浏览器上输入http://10.0.0.11:15672/
# 用户名、密码均为：guest（第一次登录必须使用该用户密码）

# 7、在浏览器上为刚创建的openstack更新Tags为：administrator
# 点击Admin -> 点击Users列表中的openstack ->在Update this user中输入两次openstack作为密码(密码必须写，因此我们写原密码)，Ta
```

四、控制节点安装Keystone

本节介绍如何在控制器节点上安装和配置代号为Keystone的OpenStack身份服务。
为了实现可伸缩性，此配置部署了Fernet令牌和Apache HTTP服务器来处理请求。

4.1、安装前提

```
# 为keystone创建数据库并授权
-- 1、登录数据库管理系统
mysql -uroot -p

-- 2、创建数据库
create database keystone;

-- 3、创建用户并授权
grant all privileges on keystone.* to keystone_user@controller identified by 'keystone_pass';

-- 4、刷新权限
flush privileges;

-- 5、退出该session
quit;
```

4.2、安装及配置

1. 安装软件包

```
yum install -y openstack-keystone httpd mod_wsgi
```

2. 修改配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/keystone/keystone.conf

# 2、修改模块如下, vim /etc/keystone/keystone.conf
[database]
connection = mysql+pymysql://keystone_user:keystone_pass@controller/keystone

[token]
provider = fernet
```

3. 同步数据库

```
su -s /bin/sh -c "keystone-manage db_sync" keystone
```

4. 初始化Fernet密钥存储库

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

5. 创建keystone管理员

```
keystone-manage bootstrap --bootstrap-password admin_pass \
  --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
```

4.3、配置并启动Apache HTTP server

```
# 1、配置ServerName
sed -i '/#ServerName/aServerName controller:80' /etc/httpd/conf/httpd.conf

```

```

# 2、连接keystone配置文件
ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/

# 3、启动并加入开机自启动
systemctl start httpd.service
systemctl enable httpd.service

# 4、配置管理员账号环境变量
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
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
```

4.4、创建域、项目、用户和角色

1. 创建域

```
# 创建一个域示例，不创建也可以，Keystone已存在一个域：default
# openstack domain create --description "An Example Domain" example
```

2. 创建服务项目

```
# 供glance、placement、nova和neutron等组件使用
openstack project create --domain default --description "Service Project" service
```

3. 创建常规（非管理员）任务应使用无特权的项目和用户

```
# 1、创建项目
openstack project create --domain default --description "Demo Project" myproject

# 2、创建用户
openstack user create --domain default --password myuser_pass myuser

# 3、创建角色
openstack role create myrole

# 4、把用户和角色添加到项目
openstack role add --project myproject --user myuser myrole
```

4.6、验证身份及密码

1. 删除临时环境变量OS_AUTH_URL、OS_PASSWORD

```
unset OS_AUTH_URL OS_PASSWORD
```

2. 验证admin,密码为：admin_pass

```
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
```

3. 验证myuser，密码为：myuser_pass

```
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
```

4.5、创建客户端环境变量脚本

由于临时环境变量只存在本session，每次session断开或重新打开一个session临时变量都会失效，因此将这些环境变量写入脚本中，需要用到时执行下脚本即可。

1. 创建脚本

```
# 1、进入家目录
cd ~

# 2、创建admin用户的OpenStack客户端环境变量脚本
cat >admin-openrc<<EOF
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
EOF

# 3、创建myuser用户的OpenStack客户端环境变量脚本
cat >demo-openrc<<EOF
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=myproject
export OS_USERNAME=myuser
export OS_PASSWORD=myuser_pass
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
EOF

```

2. 验证脚本

```
# 1、加载环境变量
cd ~
. admin-openrc

# 2、请求验证token
openstack token issue
```

能看到返回的token就说明安装成功，具体看官方实例

五、控制节点安装Glance

5.1、安装前提

1. 为Glance建库并授权

```
create database glance;
grant all privileges on glance.* to glance_user@controller identified by 'glance_pass';
flush privileges;
quit;
```

2. 获取keystone管理员凭据

```
. admin-openrc
```

3. 创建Glance服务凭证

```
# 1、 创建glance用户
openstack user create --domain default --password glance_pass glance

# 2、将glance用户加入到service项目并授予admin(管理员)角色
openstack role add --project service --user glance admin


```

```
# 3、创建glance服务实体
openstack service create --name glance --description "OpenStack Image" image
```

4. 创建Glance服务API端点

```
# 1、创建共有Glance服务API端点
openstack endpoint create --region RegionOne image public http://controller:9292

# 2、创建私有Glance服务API端点
openstack endpoint create --region RegionOne image internal http://controller:9292

# 3、创建管理Glance服务API端点
openstack endpoint create --region RegionOne image admin http://controller:9292
```

5.2、安装及配置

1. 安装软件包

```
yum install -y openstack-glance
```

2. 修改glance-api.conf配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/glance/glance-api.conf

# 2、修改模板如下，vim /etc/glance/glance-api.conf
[database]
connection = mysql+pymysql://glance_user:glance_pass@controller/glance

[glance_store]
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/

[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 = glance_pass

[paste_deploy]
flavor = keystone
```

3. 修改glance-registry.conf配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/glance/glance-registry.conf

# 2、修改模块如下，vim /etc/glance/glance-registry.conf
[database]
connection = mysql+pymysql://glance_user:glance_pass@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 = glance_pass

[paste_deploy]
flavor = keystone
```

4. 同步数据

```
su -s /bin/sh -c "glance-manage db_sync" glance
```

5.3、启动并加入开启自启

```
systemctl start openstack-glance-api.service openstack-glance-registry.service
systemctl enable openstack-glance-api.service openstack-glance-registry.service
```

5.4、上传镜像

1. 下载镜像

```
cd ~
wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
```

2. 将刚下载的镜像上传到glance

```
# 1、获取keystone管理员凭据
cd ~
. admin-openrc

# 2、上传镜像
cd ~
openstack image create "cirros" \
  --file cirros-0.4.0-x86_64-disk.img \
  --disk-format qcow2 --container-format bare \
  --public

# 3、查看上传结果
openstack image list
```

六、控制节点安装Placement

Placement组件从n版引入，p版强制用户使用，该组件的主要作用是参与 nova-scheduler 选择目标主机的调度流程中，负责跟踪记录 Resource Provider 的 Inventory 和 Usage，并使用不同的 Resource Classes 来划分资源类型，使用不同的 Resource Traits 来标记资源特征。

6.1、安装前提

1. 为Placement建库并授权

```
create database placement;
grant all privileges on placement.* to 'placement_user'@'controller' identified by 'placement_pass';
flush privileges;
quit;
```

2. 获取Keystone管理员凭据

```
cd ~
. admin-openrc
```

3. 创建Placement服务凭证

```
# 1、 创建placement用户,密码设置为: placement_pass
openstack user create --domain default --password placement_pass placement

# 2、将管理员角色添加都placement用户和service项目中
openstack role add --project service --user placement admin

# 3、创建placement服务实体
openstack service create --name placement --description "Placement API" placement
```

4. 创建Placement服务API端点

```
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
```

6.2、安装及配置

1. 安装软件包

```
yum install -y openstack-placement-api
```

2. 修改placement.conf配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/placement/placement.conf

# 2、修改模块如下, vim /etc/placement/placement.conf
[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 = placement
password = placement_pass

[placement_database]
connection = mysql+pymysql://placement_user:placement_pass@controller/placement
```

3. 同步数据库

```
su -s /bin/sh -c "placement-manage db sync" placement
```

4. 允许其他组件访问Placement API

```
# 1、修改Apache HTTP server配置
cat >>/etc/httpd/conf.d/00-placement-api.conf<<EOF

<Directory /usr/bin>
   <IfVersion >= 2.4>
      Require all granted
   </IfVersion>
   <IfVersion < 2.4>
      Order allow,deny
      Allow from all
   </IfVersion>
</Directory>
EOF

# 2、重启Apache HTTP server使之生效
systemctl restart httpd
```

6.3、检查Placement安装结果

```
placement-status upgrade check
```

七、控制节点安装Nova

7.1、安装前提

1. 为Nova建库并授权

```
# 1、建库
create database nova_api;
create database nova;
create database nova_cell0;

# 2、授权
grant all privileges on nova_api.* to 'nova_user'@'controller' identified by 'nova_pass';
grant all privileges on nova.* to 'nova_user'@'controller' identified by 'nova_pass';
grant all privileges on nova_cell0.* to 'nova_user'@'controller' identified by 'nova_pass';

# 3、刷新权限
flush privileges;
```

2. 获取Keystone管理员凭证

```
cd ~
. admin-openrc
```

3. 创建Nova服务凭证

```
# 1、 创建nova用户
openstack user create --domain default --password nova_pass nova

# 2、将管理员角色添加都nova用户和service项目中
openstack role add --project service --user nova admin

# 3、创建nova服务实体
openstack service create --name nova --description "OpenStack Compute" compute
```

4. 创建Nova服务API端点

```
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
```

7.2、安装及配置

1. 安装软件包

```
yum install -y openstack-nova-api openstack-nova-conductor openstack-nova-novncproxy openstack-nova-scheduler
```

2. 编辑nova.conf配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/nova/nova.conf

# 2、修改模块如下，vim /etc/nova/nova.conf
[DEFAULT]
enabled_apis = osapi_compute,metadata
transport_url = rabbit://openstack:openstack@controller
my_ip = 10.0.0.11
use_neutron = true
```

```
firewall_driver = nova.virt.firewall.NoopFirewallDriver
rpc_backend=rabbit

[api]
auth_strategy = keystone

[api_database]
connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova_api

[database]
connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova

[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 = nova_pass

[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 = placement_pass

[vnc]
enabled = true
server_listen = $my_ip
server_proxyclient_address = $my_ip
```

3. 同步nova-api数据库

```
su -s /bin/sh -c "nova-manage api_db sync" nova
```

4. 注册cell0数据库

```
su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova
```

5. 创建cell1原件

```
su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 --verbose" nova
```

6. 同步nova数据库

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

7. 验证novacell0和cell1注册情况

```
su -s /bin/sh -c "nova-manage cell_v2 list_cells" nova
```

7.3、启动并加入开机自启

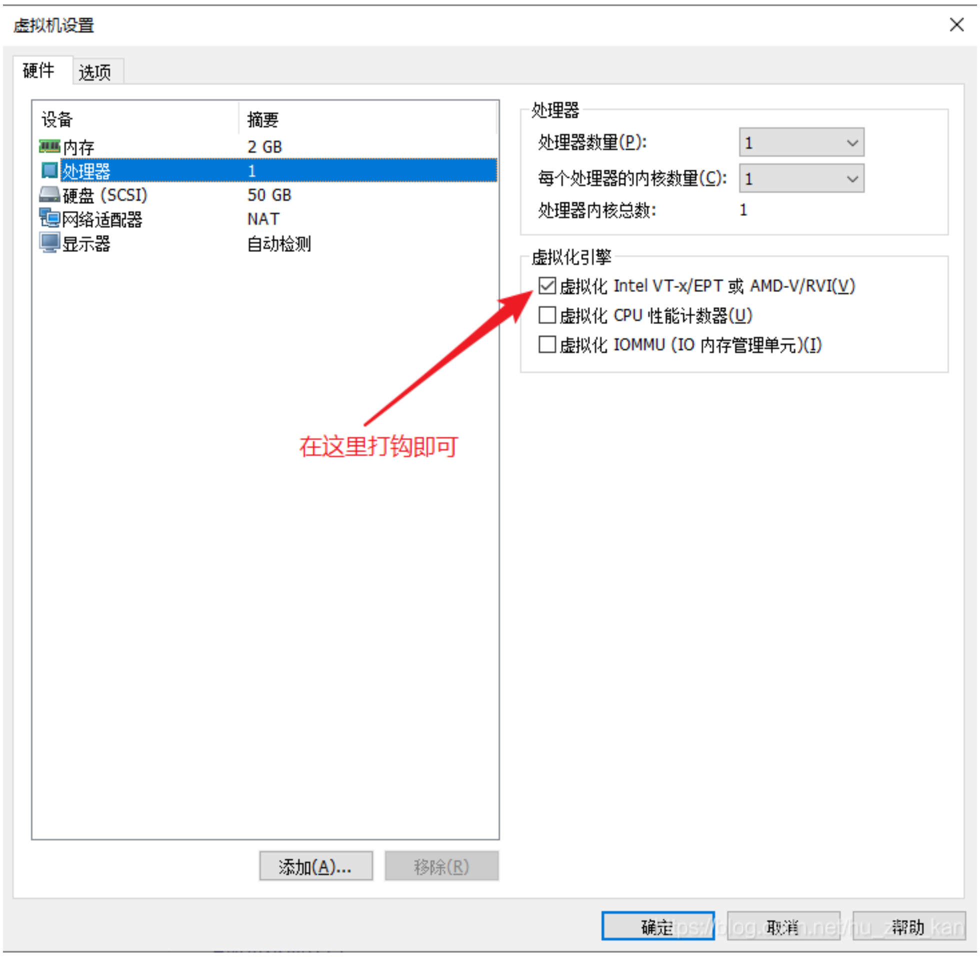
```
systemctl start openstack-nova-api.service openstack-nova-scheduler.service \
  openstack-nova-conductor.service openstack-nova-novncproxy.service

systemctl enable openstack-nova-api.service openstack-nova-scheduler.service \
  openstack-nova-conductor.service openstack-nova-novncproxy.service
```

八、准备一台计算节点虚拟机

8.1、配置虚拟机

1. 克隆一台虚拟机名为compute
2. 开启CPU支持虚拟化



设备	摘要
内存	2 GB
处理器	1
硬盘 (SCSI)	50 GB
网络适配器	NAT
显示器	自动检测

处理器
处理器数量(P): 1
每个处理器的内核数量(C): 1
处理器内核总数: 1

虚拟化引擎
☑ 虚拟化 Intel VT-x/EPT 或 AMD-V/RVI(V)
☐ 虚拟化 CPU 性能计数器(U)
☐ 虚拟化 IOMMU (IO 内存管理单元)(I)

在这里打钩即可

然后开启虚拟机，继续配置。

3. 修改IP

```
sed -i 's/10.0.0.6/10.0.0.12/g' /etc/sysconfig/network-scripts/ifcfg-eth0
systemctl restart network.service
```

4. 修改主机名

```
echo 'compute' > /etc/hostname
hostname compute
exit  #退出该session重新进入
```

5. 修改hosts文件

```
cat >>/etc/hosts<<EOF

# controller
10.0.0.11    controller
```

```
# compute
10.0.0.12    compute
EOF
```

8.2、安装基础服务

1. 启用OpenStack库

```
yum install -y centos-release-openstack-stein
```

2. 安装 OpenStack 客户端

```
yum install -y python-openstackclient
```

3. 时间同步

```
# 1、安装软件包
yum install -y chrony

# 2、将时间同步服务器修改为controller节点
sed -i '/^server/d' /etc/chrony.conf
sed -i '2aserver controller iburst' /etc/chrony.conf

# 3、启动 NTP 服务并将其配置为随系统启动
systemctl enable chronyd.service
systemctl start chronyd.service

# 4、设置时区
timedatectl set-timezone Asia/Shanghai

# 5、查看时间同步源
chronyc sources

# 6、查看时间是否正确
timedatectl status
```

九、计算节点安装Nova

9.1、安装及配置

1. 安装软件包

```
yum install -y openstack-nova-compute
```

2. 检查是否支持虚拟化

```
egrep -c '(vmx|svm)' /proc/cpuinfo  # 结果大于等于1,支持
```

3. 编辑nova.conf配置文件

```
# 1、备份原文件
sed -i.default -e '/^#/d' -e '/^$/d' /etc/nova/nova.conf

# 2、修改模块如下, vim /etc/nova/nova.conf
[DEFAULT]
enabled_apis = osapi_compute,metadata
transport_url = rabbit://openstack:openstack@controller
my_ip = 10.0.0.12
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 = nova_pass

[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

[libvirt]
virt_type = qemu
```

9.2、启动并加入开机自启

```
systemctl start libvirtd.service openstack-nova-compute.service
systemctl enable libvirtd.service openstack-nova-compute.service
```

9.3、在控制节点上添加计算节点

1. 取得keystone管理员凭据

```
cd ~
. admin-openrc
```

2. 添加计算节点到cell 数据库

```
openstack compute service list --service nova-compute
```

3. 发现计算节点

```
# 手动发现
su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --verbose" nova

# 定期主动发现
# 1、修改/etc/nova/nova.conf配置文件
[scheduler]
discover_hosts_in_cells_interval=300

# 2、重启nova服务
systemctl restart openstack-nova-api.service openstack-nova-scheduler.service \
  openstack-nova-conductor.service openstack-nova-novncproxy.service
```

十、控制节点安装Neutron

10.1、安装前提

1. 建库并授权

```
create database neutron;
grant all privileges on neutron.* to 'neutron_user'@'controller' identified by 'neutron_pass';
flush privileges;
quit;
```

2. 获取Keystone管理员凭证

```
cd ~
. admin-openrc
```

3. 创建Neutron服务凭证

```
openstack user create --domain default --password neutron_pass neutron
openstack role add --project service --user neutron admin
openstack service create --name neutron --description "OpenStack Networking" network
```

4. 创建Neutron服务API端点

```
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
```

10.2、安装及配置

1. 安装软件

```
yum install -y openstack-neutron openstack-neutron-ml2 \
  openstack-neutron-linuxbridge ebtables
```

2. 编辑neutron.conf配置文件

```
# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d' /etc/neutron/neutron.conf

# 2、修改模块如下，vim /etc/neutron/neutron.conf
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = true
transport_url = rabbit://openstack:openstack@controller
auth_strategy = keystone
notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true

[database]
connection = mysql+pymysql://neutron_user:neutron_pass@controller/neutron

[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 = neutron_pass
```

```
[oslo_concurrency]
lock_path = /var/lib/neutron/tmp

[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 = nova_pass
```

3. 配置模块化第2层（ML2）插件

```
# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d' /etc/neutron/plugins/ml2/ml2_conf.ini

# 2、修改模块如下，vim /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
```

4. 配置Linux桥代理

```
# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d' /etc/neutron/plugins/ml2/linuxbridge_agent.ini

# 2、修改模块如下，vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini
[linux_bridge]
physical_interface_mappings = provider:eth0

[vxlan]
enable_vxlan = false

[securitygroup]
enable_security_group = true
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

5. 配置DHCP代理

```
# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d' /etc/neutron/dhcp_agent.ini

# 2、修改模块如下，vim /etc/neutron/dhcp_agent.ini
[DEFAULT]
interface_driver = linuxbridge
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = true
```

6. 配置元数据代理

```
# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d' /etc/neutron/metadata_agent.ini
```

```

# 2、修改模块如下, vim /etc/neutron/metadata_agent.ini
[DEFAULT]
nova_metadata_host = controller
metadata_proxy_shared_secret = metadata_secret
```

7. 配置/etc/nova/nova.conf文件neutron模块

```
[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 = neutron_pass
service_metadata_proxy = true
metadata_proxy_shared_secret = metadata_secret
```

8. 创建网络服务初始化脚本需要的软连接

```
ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini
```

9. 同步数据库

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

10.3、启动

1. 重启Compute API服务

```
systemctl restart openstack-nova-api.service
```

2. 启动网络服务并开启自启

```
systemctl start neutron-server.service \
  neutron-linuxbridge-agent.service \
  neutron-dhcp-agent.service \
  neutron-metadata-agent.service

systemctl enable neutron-server.service \
  neutron-linuxbridge-agent.service \
  neutron-dhcp-agent.service \
  neutron-metadata-agent.service
```

十一、计算节点安装Neutron

1. 安装软件

```
yum install -y openstack-neutron-linuxbridge ebtables ipset
```

2. 编辑neutron.conf配置文件

```
# 1、备份原文件并删除注释
sed  -i.default-e '/^#/d' -e '/^$/d' /etc/neutron/neutron.conf

```

```
# 2、修改模块如下, vim /etc/neutron/neutron.conf
[DEFAULT]
transport_url = rabbit://openstack:openstack@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 = neutron_pass

[oslo_concurrency]
lock_path = /var/lib/neutron/tmp
```

3. 配置Linux桥代理

```

# 1、备份原文件并删除注释
sed -i.bak -e '/^#/d' -e '/^$/d' /etc/neutron/plugins/ml2/linuxbridge_agent.ini

# 2、修改模块如下, vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini
[linux_bridge]
physical_interface_mappings = provider:eth0

[vxlan]
enable_vxlan = false

[securitygroup]
enable_security_group = true
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

4. 确保您的Linux操作系统内核支持网桥过滤器

```

# 1、添加配置
cat >>/etc/sysctl.conf<<EOF
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
EOF

# 2、启用
modprobe br_netfilter

# 3、生效
 sysctl -p
```

5. 编辑/etc/nova/nova.conf文件

```

# 1、备份原文件并删除注释
sed -i.default -e '/^#/d' -e '/^$/d'  /etc/nova/nova.conf

# 2、修改模块如下, vim /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 = neutron_pass
```

6. 重新启动Nova Compute服务

```
systemctl restart openstack-nova-compute.service
```

7. 启动Linux网桥代理并开机自启动

```
systemctl enable neutron-linuxbridge-agent.service
systemctl start neutron-linuxbridge-agent.service
```

8. 验证(在控制节点上操作)

```
openstack extension list --network
openstack network agent list  # 注意：一共4个，其中两个是Linux bridge agent说明成功
```

十二、创建一台虚拟机（控制节点）

注意： 以下步骤均在控制节点上操作

12.1、创建网络

1. 获取keystone管理员凭证

```
cd ~
. admin-openrc
```

2. 创建网络

```
openstack network create  --share --external \
  --provider-physical-network provider \
  --provider-network-type flat provider

openstack network list  # 查看
```

3. 创建子网

```
openstack subnet create --network provider \
  --allocation-pool start=10.0.0.100,end=10.0.0.200 \
  --dns-nameserver 10.0.0.2 --gateway 10.0.0.2 \
  --subnet-range 10.0.0.0/24 provider-sub

openstack subnet list
```

12.2、创建主机规格

1. 获取keystone管理员凭证

```
cd ~
. admin-openrc
```

2. 创建主机规格

```
openstack flavor create --id 0 --vcpus 1 --ram 64 --disk 1 m1.nano
# openstack flavor create 创建主机
# --id 主机ID
# --vcpus cpu数量
# --ram 64 (默认是MB，可以写成G)
# --disk 磁盘 (默认单位是G)
```

12.3、创建一个实例

1. 获取demo用户权限凭证

```
cd ~
. demo-openrc
```

2. 生成秘钥对

```
ssh-keygen -q -N ""
```

3. 将密钥放在openstack上

```
openstack keypair create --public-key ~/.ssh/id_rsa.pub mykey
```

4. 验证密码是否创建成功

```
nova keypair-list
```

5. 添加安全组规则

```
# 允许ICMP (ping
openstack security group rule create --proto icmp default

# 允许安全shell (SSH) 访问
openstack security group rule create --proto tcp --dst-port 22 default
```

6. 查看创建实例需要的相关信息

```
openstack flavor list
openstack image list
openstack network list
openstack security group list
openstack keypair list
```

7. 创建并启动实例

```
openstack server create --flavor m1.nano --image cirros \
  --nic net-id=9e07c3d5-9a9e-496c-90b6-ba294f8b0699  \
  --security-group default \
  --key-name mykey hello-instance


# —flavor: 类型名称
# —image: 镜像名称
# --nic: 指定网络ID，根据刚刚openstack network list查到的网络ID填写，不是子网哦
# —security-group：安全组名
```

8. 查看实例状态

```
[root@controller ~]# openstack server list
+--------------------------------------+----------------+--------+---------------------+--------+---------+
| ID                                   | Name           | Status | Networks            | Image  | Flavor  |
+--------------------------------------+----------------+--------+---------------------+--------+---------+
| 0d94ce6d-ae08-4ace-a183-3ecd44ccba56 | hello-instance | ACTIVE | provider=10.0.0.138 | cirros | m1.nano |
+--------------------------------------+----------------+--------+---------------------+--------+---------+
```

12.4、登录实例

12.4.1、通过SSH登录

```
ping 10.0.0.138

ssh cirros@10.0.0.138
```

12.4.2、 通过WEB页面登录

1. 获取Keystone demo用户权限凭证

```
cd ~
. demo-openrc
```

2. 启用并查看实例web登录的url

```
[root@controller ~]# openstack console url show hello-instance
+-------+-------------------------------------------------------------------------------------------+
| Field | Value                                                                                     |
+-------+-------------------------------------------------------------------------------------------+
| type  | novnc                                                                                     |
| url   | http://controller:6080/vnc_auto.html?path=%3Ftoken%3D56c1d801-c6ce-40d3-a998-9343c9af925e |
+-------+-------------------------------------------------------------------------------------------+
```

3. 修改win10的host文件
如果你用的也是Windows 10，那么hosts文件在C:\Windows\System32\drivers\etc路径下，将 `10.0.0.11 controller` 加入到hosts文件。

4. 在浏览器上登录
复制上面url到浏览器地址栏登录