**Ubuntu12.04 OpenStack Folsom**

**安装（VLAN模式）**

（文档版本：R1.0 Rev.7）

目录

[第一章 介绍 1](#_Toc354910952)

[第二章 控制节点 2](#_Toc354910953)

[2.1 操作系统 2](#_Toc354910954)

[2.2 源 2](#_Toc354910955)

[2.3 Hostname设置（可选） 2](#_Toc354910956)

[2.4 网络 2](#_Toc354910957)

[2.5 NTP服务器 3](#_Toc354910958)

[2.6 环境变量 4](#_Toc354910959)

[2.7 Mysql 4](#_Toc354910960)

[2.7.1 安装 4](#_Toc354910961)

[2.7.2 设置 5](#_Toc354910962)

[2.7.3 创建数据库 5](#_Toc354910963)

[2.8 RabbitMQ 6](#_Toc354910964)

[2.8.1 安装 6](#_Toc354910965)

[2.8.2 设置 6](#_Toc354910966)

[2.9 Keystone 6](#_Toc354910967)

[2.9.1 安装 6](#_Toc354910968)

[2.9.2 配置 6](#_Toc354910969)

[2.10 Glance 13](#_Toc354910970)

[2.10.1 安装 13](#_Toc354910971)

[2.10.2 配置 13](#_Toc354910972)

[2.10.3 下载Image(测试环境用) 14](#_Toc354910973)

[2.10.4 测试 14](#_Toc354910974)

[2.11 Nova 14](#_Toc354910975)

[2.11.1 安装 14](#_Toc354910976)

[2.11.2 配置 15](#_Toc354910977)

[2.12 Open-vSwitch 17](#_Toc354910978)

[2.12.1 安装 17](#_Toc354910979)

[2.12.2 配置 17](#_Toc354910980)

[2.13 Quantum 17](#_Toc354910981)

[2.13.1 安装 17](#_Toc354910982)

[2.13.2 配置 17](#_Toc354910983)

[2.14 Cinder 19](#_Toc354910984)

[2.14.1 安装 19](#_Toc354910985)

[2.14.2 配置 19](#_Toc354910986)

[2.15 Horizon 20](#_Toc354910987)

[2.15.1 安装 20](#_Toc354910988)

[2.15.2 配置 20](#_Toc354910989)

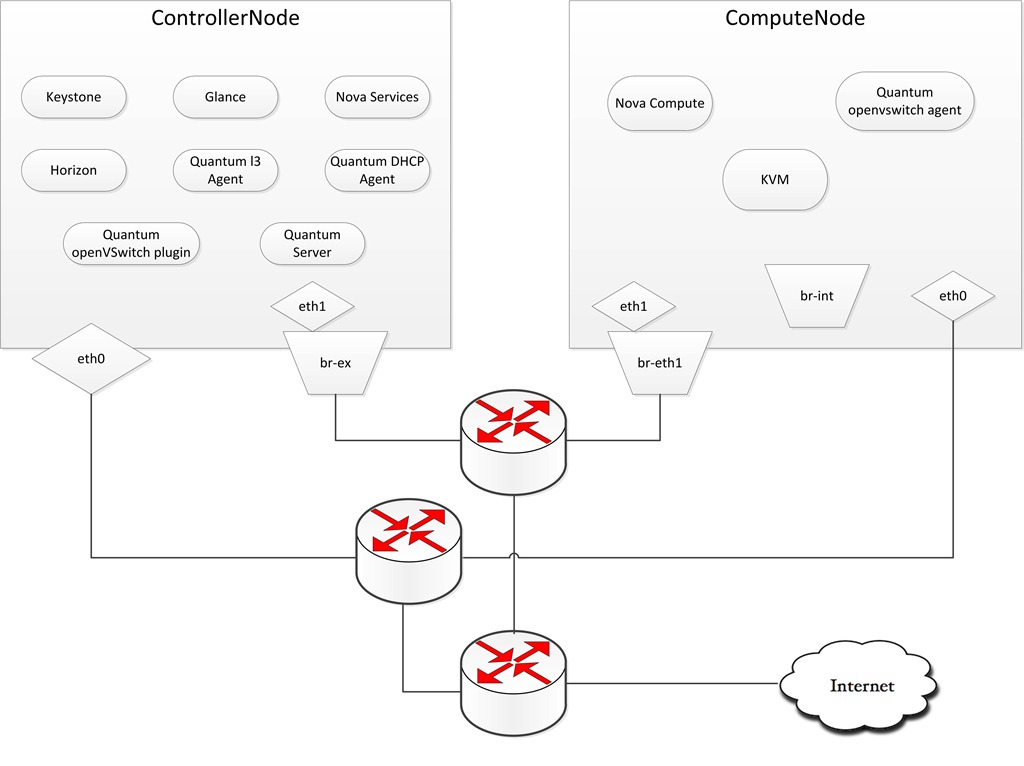
# 介绍

|  |  |  |
| --- | --- | --- |
|  | 控制节点（2块网卡） | 计算节点（2块网卡） |
| 管理网络（eth0） | 10.1.199.58/24 | 10.1.199.6/24 |
|  | 10.1.199.68/24 |  |
| hostname | controller | compute1 |
| 服务 | MySQL  RabbitMQ  Nova  Glance  Keystone  Quantum | kvm  quantum client  nova-compute |

要求：

1. 控制节点和计算节点需要2块网卡，计算节点需要支持虚拟化
2. 全部的命令都是在root下运行

网络拓扑图：



# 控制节点

## 操作系统

安装ubuntu 12.04.1 Server版本，最小化安装，只需要安装SSH server就可以。Cinder 需要一个单独的分区或者硬盘。

## 源

目前Folsom进入ubuntu 12.04的官方的源，不过需要你手工添加。

|  |
| --- |
| cat <<EOF >>/etc/apt/sources.list  deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-proposed/folsom main  deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/folsom main  EOF |

运行下面命令：

|  |
| --- |
| apt-get install ubuntu-cloud-keyring  apt-get update && apt-get -y dist-upgrade |

## Hostname设置（可选）

|  |
| --- |
| # cat /etc/hostname  controller  # cat /etc/hosts  127.0.0.1 localhost  10.0.0.58 controller  10.0.0.6 compute1  # hostname  controller |

## 网络

直接设置 /etc/network/interface

|  |
| --- |
| # cat /etc/network/interfaces  # This file describes network interfaces avaiulable on your system  # and how to activate them. For more information, see interfaces(5).  # Modified by convert\_static.sh.  auto lo  iface lo inet loopback  auto eth0  iface eth0 inet static  address 10.0.0.58  netmask 255.255.255.0  network 10.0.0.0  gateway 10.0.0.1  dns-nameservers 8.8.8.8  auto eth1  iface eth1 inet static  address 10.0.0.68  netmask 255.255.255.0 |

重启服务：

|  |
| --- |
| /etc/init.d/networking restart |

设置IP转发：

|  |
| --- |
| sed -i -r 's/^\s\*#(net\.ipv4\.ip\_forward=1.\*)/\1/' /etc/sysctl.conf  echo 1 > /proc/sys/net/ipv4/ip\_forward |

检查修改结果：

|  |
| --- |
| # sysctl -p  net.ipv4.ip\_forward = 1 |

## NTP服务器

编辑 /etc/ntp.conf ，在 server ntp.ubuntu.com 下添加两行：

|  |
| --- |
| server ntp.ubuntu.com  server 127.127.1.0  fudge 127.127.1.0 stratum 10 |

或直接运行下面命令：

|  |
| --- |
| sed -i 's/server ntp.ubuntu.com/server ntp.ubuntu.com\nserver 127.127.1.0\nfudge 127.127.1.0 stratum 10/g' /etc/ntp.conf |

重启NTP服务：

|  |
| --- |
| service ntp restart |

## 环境变量

|  |
| --- |
| cat >/root/novarc <<EOF  export OS\_TENANT\_NAME=admin  export OS\_USERNAME=admin  export OS\_PASSWORD=123456  export MYSQL\_PASS=123456  export SERVICE\_PASSWORD=123456  export RABBIT\_PASSWORD=123456  export FIXED\_RANGE=10.0.0.0/24  export FLOATING\_RANGE=$(/sbin/ifconfig eth0 | awk '/inet addr/ {print $2}' | cut -f2 -d ":" | awk -F "." '{print $1"."$2"."$3}').224/27  export OS\_AUTH\_URL="http://localhost:5000/v2.0/"  export SERVICE\_ENDPOINT="http://localhost:35357/v2.0"  export SERVICE\_TOKEN=$(openssl rand -hex 10)  export MASTER="$(/sbin/ifconfig eth0 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"  export LOCAL\_IP="$(/sbin/ifconfig eth1 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"  EOF |

你可以根据你的需要调整用户的密码。

|  |
| --- |
| source novarc  echo "source novarc">>.bashrc |

## Mysql

下面是我们需要用到的数据库：

|  |  |  |
| --- | --- | --- |
| 数据库 | 用户 | 密码 |
| mysql | root | 123456 |
| nova | nova | 123456 |
| keystone | keystone | 123456 |
| glance | glance | 123456 |
| cinder | cinder | 123456 |
| quantum | quantum | 123456 |

### 安装

设置自动安装，无需输入密码：

|  |
| --- |
| cat <<MYSQL\_PRESEED | debconf-set-selections  mysql-server-5.5 mysql-server/root\_password password $MYSQL\_PASS  mysql-server-5.5 mysql-server/root\_password\_again password $MYSQL\_PASS  mysql-server-5.5 mysql-server/start\_on\_boot boolean true  MYSQL\_PRESEED |

安装mysql：

|  |
| --- |
| apt-get -y install mysql-server python-mysqldb |

### 设置

运行远程访问mysql：

|  |
| --- |
| sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf |

重启服务：

|  |
| --- |
| service mysql restart |

### 创建数据库

|  |
| --- |
| mysql -uroot –p123456  GRANT ALL PRIVILEGES ON \*.\* TO root@'%' IDENTIFIED BY '123456';  GRANT ALL PRIVILEGES ON \*.\* TO root@localhost IDENTIFIED BY '123456';  CREATE DATABASE nova;  GRANT ALL PRIVILEGES ON nova.\* TO nova@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON nova.\* TO nova@localhost IDENTIFIED BY '123456' WITH GRANT OPTION;  CREATE DATABASE glance;  GRANT ALL PRIVILEGES ON glance.\* TO glance@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON glance.\* TO glance@localhost IDENTIFIED BY '123456' WITH GRANT OPTION;  CREATE DATABASE keystone;  GRANT ALL PRIVILEGES ON keystone.\* TO keystone@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON keystone.\* TO keystone@localhost IDENTIFIED BY '123456' WITH GRANT OPTION;  CREATE DATABASE cinder;  GRANT ALL PRIVILEGES ON cinder.\* TO cinder@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON cinder.\* TO cinder@localhost IDENTIFIED BY '123456' WITH GRANT OPTION;  CREATE DATABASE quantum;  GRANT ALL PRIVILEGES ON quantum.\* TO quantum@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON quantum.\* TO quantum@localhost IDENTIFIED BY '123456' WITH GRANT OPTION;  FLUSH PRIVILEGES;  QUIT |

## RabbitMQ

### 安装

|  |
| --- |
| apt-get -y install rabbitmq-server |

### 设置

修改默认密码

我们把默认密码 guest，改成password

|  |
| --- |
| rabbitmqctl change\_password guest $RABBIT\_PASSWORD |

## Keystone

### 安装

|  |
| --- |
| apt-get -y install keystone python-keystone python-keystoneclient |

### 配置

编辑 /etc/keystone/keystone.conf

|  |
| --- |
| [DEFAULT]  admin\_token = d111cf2d97251a9e0422  bind\_host = 0.0.0.0  public\_port = 5000  admin\_port = 35357  compute\_port = 8774  verbose = True  debug = True  log\_file = keystone.log  log\_dir = /var/log/keystone  log\_config = /etc/keystone/logging.conf  [sql]  connection = mysql://keystone:password@10.0.0.58:3306/keystone  idle\_timeout = 200 |

或者直接运行下面脚本

|  |
| --- |
| sed -i -e " s/# admin\_token = ADMIN/admin\_token = $SERVICE\_TOKEN/g; s/# bind\_host = 0.0.0.0/bind\_host = 0.0.0.0/g; s/# public\_port = 5000/public\_port = 5000/g; s/# admin\_port = 35357/admin\_port = 35357/g; s/# compute\_port = 8774/compute\_port = 8774/g; s/# verbose = True/verbose = True/g; s/# idle\_timeout/idle\_timeout/g" /etc/keystone/keystone.conf |

使用mysql数据库

|  |
| --- |
| sed -i '/connection = .\*/{s|sqlite:///.\*|mysql://'"keystone"':'"$MYSQL\_PASS"'@'"$MASTER"'/keystone|g}' /etc/keystone/keystone.conf |

重启服务和初始化数据库

|  |
| --- |
| service keystone restart  keystone-manage db\_sync |

导入keystone数据（搭建测试环境用）：

|  |
| --- |
| #!/bin/sh  # Keystone data  # Written by Martin Gerhard Loschwitz / Hastexo  # Modified by Emilien Macchi / StackOps  #  # Support: openstack@lists.launchpad.net  # License: Apache Software License (ASL) 2.0  #  #ADMIN\_PASSWORD=${ADMIN\_PASSWORD:-password}  ADMIN\_PASSWORD=${ADMIN\_PASSWORD:-$OS\_PASSWORD}  #SERVICE\_PASSWORD=${SERVICE\_PASSWORD:-$ADMIN\_PASSWORD}  #export SERVICE\_TOKEN="password"  export SERVICE\_ENDPOINT="http://localhost:35357/v2.0"  SERVICE\_TENANT\_NAME=${SERVICE\_TENANT\_NAME:-service}  get\_id () {  echo `$@ | awk '/ id / { print $4 }'`  }  # Tenants  ADMIN\_TENANT=$(get\_id keystone tenant-create --name=admin)  SERVICE\_TENANT=$(get\_id keystone tenant-create --name=$SERVICE\_TENANT\_NAME)  DEMO\_TENANT=$(get\_id keystone tenant-create --name=demo)  INVIS\_TENANT=$(get\_id keystone tenant-create --name=invisible\_to\_admin)  # Users  ADMIN\_USER=$(get\_id keystone user-create --name=admin --pass="$ADMIN\_PASSWORD" --email=admin@domain.com)  DEMO\_USER=$(get\_id keystone user-create --name=demo --pass="$ADMIN\_PASSWORD" --email=demo@domain.com)  # Roles  ADMIN\_ROLE=$(get\_id keystone role-create --name=admin)  KEYSTONEADMIN\_ROLE=$(get\_id keystone role-create --name=KeystoneAdmin)  KEYSTONESERVICE\_ROLE=$(get\_id keystone role-create --name=KeystoneServiceAdmin)  # Add Roles to Users in Tenants  keystone user-role-add --user-id $ADMIN\_USER --role-id $ADMIN\_ROLE --tenant-id $ADMIN\_TENANT  keystone user-role-add --user-id $ADMIN\_USER --role-id $ADMIN\_ROLE --tenant-id $DEMO\_TENANT  keystone user-role-add --user-id $ADMIN\_USER --role-id $KEYSTONEADMIN\_ROLE --tenant-id $ADMIN\_TENANT  keystone user-role-add --user-id $ADMIN\_USER --role-id $KEYSTONESERVICE\_ROLE --tenant-id $ADMIN\_TENANT  # The Member role is used by Horizon and Swift  MEMBER\_ROLE=$(get\_id keystone role-create --name=Member)  keystone user-role-add --user-id $DEMO\_USER --role-id $MEMBER\_ROLE --tenant-id $DEMO\_TENANT  keystone user-role-add --user-id $DEMO\_USER --role-id $MEMBER\_ROLE --tenant-id $INVIS\_TENANT  # Configure service users/roles  NOVA\_USER=$(get\_id keystone user-create --name=nova --pass="$SERVICE\_PASSWORD" --tenant-id $SERVICE\_TENANT --email=nova@domain.com)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $NOVA\_USER --role-id $ADMIN\_ROLE  GLANCE\_USER=$(get\_id keystone user-create --name=glance --pass="$SERVICE\_PASSWORD" --tenant-id $SERVICE\_TENANT --email=glance@domain.com)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $GLANCE\_USER --role-id $ADMIN\_ROLE  SWIFT\_USER=$(get\_id keystone user-create --name=swift --pass="$SERVICE\_PASSWORD" --tenant-id $SERVICE\_TENANT --email=swift@domain.com)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $SWIFT\_USER --role-id $ADMIN\_ROLE  RESELLER\_ROLE=$(get\_id keystone role-create --name=ResellerAdmin)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $NOVA\_USER --role-id $RESELLER\_ROLE  QUANTUM\_USER=$(get\_id keystone user-create --name=quantum --pass="$SERVICE\_PASSWORD" --tenant-id $SERVICE\_TENANT --email=quantum@domain.com)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $QUANTUM\_USER --role-id $ADMIN\_ROLE  CINDER\_USER=$(get\_id keystone user-create --name=cinder --pass="$SERVICE\_PASSWORD" --tenant-id $SERVICE\_TENANT --email=cinder@domain.com)  keystone user-role-add --tenant-id $SERVICE\_TENANT --user-id $CINDER\_USER --role-id $ADMIN\_ROLE |

导入endpoint（搭建测试环境用）：

|  |
| --- |
| #!/bin/sh  # Keystone Endpoints  # Written by Martin Gerhard Loschwitz / Hastexo  # Modified by Emilien Macchi / StackOps  #  # Support: openstack@lists.launchpad.net  # License: Apache Software License (ASL) 2.0  #  # MySQL definitions  MYSQL\_USER=keystone  MYSQL\_DATABASE=keystone  MYSQL\_HOST=$MASTER  MYSQL\_PASSWORD=$MYSQL\_PASS  # Keystone definitions  KEYSTONE\_REGION=RegionOne  #SERVICE\_TOKEN=password  SERVICE\_ENDPOINT="http://localhost:35357/v2.0"  # other definitions  #MASTER="192.168.0.1"  while getopts "u:D:p:m:K:R:E:S:T:vh" opt; do  case $opt in  u)  MYSQL\_USER=$OPTARG  ;;  D)  MYSQL\_DATABASE=$OPTARG  ;;  p)  MYSQL\_PASSWORD=$OPTARG  ;;  m)  MYSQL\_HOST=$OPTARG  ;;  K)  MASTER=$OPTARG  ;;  R)  KEYSTONE\_REGION=$OPTARG  ;;  E)  export SERVICE\_ENDPOINT=$OPTARG  ;;  S)  SWIFT\_MASTER=$OPTARG  ;;  T)  export SERVICE\_TOKEN=$OPTARG  ;;  v)  set -x  ;;  h)  cat <<EOF  Usage: $0 [-m mysql\_hostname] [-u mysql\_username] [-D mysql\_database] [-p mysql\_password]  [-K keystone\_master ] [ -R keystone\_region ] [ -E keystone\_endpoint\_url ]  [ -S swift\_master ] [ -T keystone\_token ]    Add -v for verbose mode, -h to display this message.  EOF  exit 0  ;;  \?)  echo "Unknown option -$OPTARG" >&2  exit 1  ;;  :)  echo "Option -$OPTARG requires an argument" >&2  exit 1  ;;  esac  done  if [ -z "$KEYSTONE\_REGION" ]; then  echo "Keystone region not set. Please set with -R option or set KEYSTONE\_REGION variable." >&2  missing\_args="true"  fi  if [ -z "$SERVICE\_TOKEN" ]; then  echo "Keystone service token not set. Please set with -T option or set SERVICE\_TOKEN variable." >&2  missing\_args="true"  fi  if [ -z "$SERVICE\_ENDPOINT" ]; then  echo "Keystone service endpoint not set. Please set with -E option or set SERVICE\_ENDPOINT variable." >&2  missing\_args="true"  fi  if [ -z "$MYSQL\_PASSWORD" ]; then  echo "MySQL password not set. Please set with -p option or set MYSQL\_PASSWORD variable." >&2  missing\_args="true"  fi  if [ -n "$missing\_args" ]; then  exit 1  fi    keystone service-create --name nova --type compute --description 'OpenStack Compute Service'  keystone service-create --name cinder --type volume --description 'OpenStack Volume Service'  keystone service-create --name glance --type image --description 'OpenStack Image Service'  keystone service-create --name swift --type object-store --description 'OpenStack Storage Service'  keystone service-create --name keystone --type identity --description 'OpenStack Identity'  keystone service-create --name ec2 --type ec2 --description 'OpenStack EC2 service'  keystone service-create --name quantum --type network --description 'OpenStack Networking service'  create\_endpoint () {  case $1 in  compute)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':8774/v2/$(tenant\_id)s' --adminurl 'http://'"$MASTER"':8774/v2/$(tenant\_id)s' --internalurl 'http://'"$MASTER"':8774/v2/$(tenant\_id)s'  ;;  volume)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':8776/v1/$(tenant\_id)s' --adminurl 'http://'"$MASTER"':8776/v1/$(tenant\_id)s' --internalurl 'http://'"$MASTER"':8776/v1/$(tenant\_id)s'  ;;  image)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':9292/v2' --adminurl 'http://'"$MASTER"':9292/v2' --internalurl 'http://'"$MASTER"':9292/v2'  ;;  object-store)  if [ $SWIFT\_MASTER ]; then  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$SWIFT\_MASTER"':8080/v1/AUTH\_$(tenant\_id)s' --adminurl 'http://'"$SWIFT\_MASTER"':8080/v1' --internalurl 'http://'"$SWIFT\_MASTER"':8080/v1/AUTH\_$(tenant\_id)s'  else  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':8080/v1/AUTH\_$(tenant\_id)s' --adminurl 'http://'"$MASTER"':8080/v1' --internalurl 'http://'"$MASTER"':8080/v1/AUTH\_$(tenant\_id)s'  fi  ;;  identity)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':5000/v2.0' --adminurl 'http://'"$MASTER"':35357/v2.0' --internalurl 'http://'"$MASTER"':5000/v2.0'  ;;  ec2)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':8773/services/Cloud' --adminurl 'http://'"$MASTER"':8773/services/Admin' --internalurl 'http://'"$MASTER"':8773/services/Cloud'  ;;  network)  keystone endpoint-create --region $KEYSTONE\_REGION --service-id $2 --publicurl 'http://'"$MASTER"':9696/' --adminurl 'http://'"$MASTER"':9696/' --internalurl 'http://'"$MASTER"':9696/'  ;;  esac  }  for i in compute volume image object-store identity ec2 network; do  id=`mysql -h"$MYSQL\_HOST" -u"$MYSQL\_USER" -p"$MYSQL\_PASSWORD" "$MYSQL\_DATABASE" -ss -e "SELECT id FROM service WHERE type='"$i"';"` || exit 1  create\_endpoint $i $id  done |

## Glance

### 安装

|  |
| --- |
| apt-get -y install glance glance-api python-glanceclient glance-common |

### 配置

编辑/etc/glance/glance-api.conf 和 /etc/glance/glance-registry.conf ,两个文件，都是修改4个地方

|  |
| --- |
| sql\_connection = mysql://glance:password@10.0.0.58/glance  admin\_tenant\_name = service  admin\_user = glance  admin\_password = password |

或者直接运行下面脚本实现

|  |
| --- |
| sed -i -e " s/%SERVICE\_TENANT\_NAME%/service/g; s/%SERVICE\_USER%/glance/g; s/%SERVICE\_PASSWORD%/$SERVICE\_PASSWORD/g; " /etc/glance/glance-api.conf /etc/glance/glance-registry.conf  sed -i '/sql\_connection = .\*/{s|sqlite:///.\*|mysql://'"glance"':'"$MYSQL\_PASS"'@'"$MASTER"'/glance|g}' /etc/glance/glance-registry.conf /etc/glance/glance-api.conf |

编辑 /etc/glance/glance-api.conf

|  |
| --- |
| #notifier\_strategy = noop  notifier\_strategy = rabbit  #rabbit\_password = guest  rabbit\_password = password |

运行下面命令进行修改

|  |
| --- |
| sed -i " s/notifier\_strategy = noop/notifier\_strategy = rabbit/g;s/rabbit\_password = guest/rabbit\_password = $RABBIT\_PASSWORD/g;" /etc/glance/glance-api.conf |

运行下面命令

|  |
| --- |
| cat <<EOF >>/etc/glance/glance-api.conf  flavor = keystone+cachemanagement  EOF  cat <<EOF >>/etc/glance/glance-registry.conf  flavor = keystone  EOF |

重启服务

|  |
| --- |
| service glance-api restart && service glance-registry restart |

同步数据库

|  |
| --- |
| glance-manage db\_sync |

### 下载Image(测试环境用)

下载CirrOS的image作为测试使用，只有10M。如果是ubuntu官方的image，220M，并且ubuntu官方的image，都是需要使用密钥登陆。

**CirrOS：**

下载image：

|  |
| --- |
| wget https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86\_64-disk.img |

上传image：

|  |
| --- |
| glance image-create --name=cirros-0.3.0-x86\_64 --public --container-format=bare \  --disk-format=qcow2 < /root/cirros-0.3.0-x86\_64-disk.img |

Cirros，是可以使用用户名和密码登陆，也可以使用密钥登陆

user：cirros

password：cubswin:)

**Ubuntu官方image：**

下载image：

|  |
| --- |
| wget http://cloud-images.ubuntu.com/precise/current/precise-server-cloudimg-amd64-disk1.img |

上传image：

|  |
| --- |
| glance image-create --name="Ubuntu 12.04 cloudimg amd64" --public \  --container-format=ovf --disk-format=qcow2 < /root/precise-server-cloudimg-amd64-disk1.img |

user：ubuntu

只能使用密钥登陆。

### 测试

查看image

glance image-list查看image详细信息

glance image-show <image的UUID>

## Nova

### 安装

|  |
| --- |
| apt-get -y install nova-api nova-cert nova-common \  nova-scheduler python-nova python-novaclient nova-consoleauth novnc nova-novncproxy |

### 配置

编辑 /etc/nova/api-paste.ini

|  |
| --- |
| [filter:authtoken]  paste.filter\_factory = keystone.middleware.auth\_token:filter\_factory  auth\_host = 10.1.199.58  auth\_port = 35357  auth\_protocol = http  admin\_tenant\_name = service  admin\_user = nova  admin\_password = password  signing\_dirname = /tmp/keystone-signing-nova |

或者直接运行命令

|  |
| --- |
| sed -i -e "s/127.0.0.1/$MASTER/g; s/%SERVICE\_TENANT\_NAME%/service/g; s/%SERVICE\_USER%/nova/g; s/%SERVICE\_PASSWORD%/$SERVICE\_PASSWORD/g; " /etc/nova/api-paste.ini |

创建 /etc/nova/nova.conf 文件，直接在终端输入下面的命令，运行就可以。

|  |
| --- |
| cat >/etc/nova/nova.conf <<EOF  [DEFAULT]  logdir=/var/log/nova  state\_path=/var/lib/nova  lock\_path=/run/lock/nova  verbose=False  api\_paste\_config=/etc/nova/api-paste.ini  scheduler\_driver=nova.scheduler.simple.SimpleScheduler  s3\_host=$MASTER  ec2\_host=$MASTER  ec2\_dmz\_host=$MASTER  rabbit\_host=$MASTER  rabbit\_password=$RABBIT\_PASSWORD  cc\_host=$MASTER  nova\_url=http://$MASTER:8774/v1.1/  sql\_connection=mysql://nova:$MYSQL\_PASS@$MASTER/nova  ec2\_url=http://$MASTER:8773/services/Cloud  root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf  # Auth  use\_deprecated\_auth=false  auth\_strategy=keystone  keystone\_ec2\_url=http://$MASTER:5000/v2.0/ec2tokens  # Imaging service  glance\_api\_servers=$MASTER:9292  image\_service=nova.image.glance.GlanceImageService  # Vnc configuration  novnc\_enabled=true  novncproxy\_base\_url=http://$MASTER:6080/vnc\_auto.html  novncproxy\_port=6080  vncserver\_proxyclient\_address=127.0.0.1  vncserver\_listen=0.0.0.0  # Network settings  network\_api\_class=nova.network.quantumv2.api.API  quantum\_url=http://$MASTER:9696  quantum\_auth\_strategy=keystone  quantum\_admin\_tenant\_name=service  quantum\_admin\_username=quantum  quantum\_admin\_password=$SERVICE\_PASSWORD  quantum\_admin\_auth\_url=http://$MASTER:35357/v2.0  libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver  linuxnet\_interface\_driver=nova.network.linux\_net.LinuxOVSInterfaceDriver  firewall\_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver  # Compute #  compute\_driver=libvirt.LibvirtDriver  # Cinder #  volume\_api\_class=nova.volume.cinder.API  osapi\_volume\_listen\_port=5900  EOF |

同步数据库

|  |
| --- |
| nova-manage db sync |

重启服务

|  |
| --- |
| service nova-api restart  service nova-cert restart  service nova-consoleauth restart  service nova-scheduler restart  service nova-novncproxy restart |

或者

|  |
| --- |
| cd /etc/init.d/; for i in $( ls nova-\* ); do sudo service $i restart; done |

检查服务

|  |
| --- |
| nova-manage service list |

## Open-vSwitch

### 安装

|  |
| --- |
| apt-get install -y openvswitch-switch |

### 配置

设置网络

|  |
| --- |
| ovs-vsctl add-br br-ex  ovs-vsctl add-port br-ex eth1 |

列出已创建的桥

|  |
| --- |
| ovs-vsctl list-br  ovs-vsctl show |

## Quantum

### 安装

|  |
| --- |
| apt-get -y install quantum-server python-cliff \  quantum-plugin-openvswitch-agent \  quantum-l3-agent quantum-dhcp-agent python-pyparsing |

### 配置

编辑 /etc/quantum/quantum.conf

|  |
| --- |
| auth\_strategy = keystone  fake\_rabbit = False  rabbit\_host = 10.0.0.58  rabbit\_password = password |

或者运行下面命令

|  |
| --- |
| sed -i -e " s/# auth\_strategy/auth\_strategy/g; s/# fake\_rabbit/fake\_rabbit/g; s/# rabbit\_host = localhost/rabbit\_host = $MASTER/g; s/# rabbit\_password = guest/rabbit\_password = $RABBIT\_PASSWORD/g" /etc/quantum/quantum.conf |

编辑 /etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini

|  |
| --- |
| #Under the database section  [DATABASE]  sql\_connection = mysql://quantum:password@10.0.0.58/quantum  #Under the OVS section  [OVS]  tenant\_network\_type=vlan  network\_vlan\_ranges = physnet1:1:4094 |

或者运行下面命令

|  |
| --- |
| sed -i -e " s/# Example: tenant\_network\_type = gre/tenant\_network\_type = vlan/g; s/# Example: network\_vlan\_ranges = physnet1:1000:2999/network\_vlan\_ranges = physnet1:1:4094/g" /etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini  sed -i '/sql\_connection = .\*/{s|sqlite:///.\*|mysql://'"quantum"':'"password"'@'"$MASTER"'/quantum|g}' /etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini |

编辑 /etc/quantum/l3\_agent.ini 和 /etc/quantum/api-paste.ini

|  |
| --- |
| [DEFAULT]  admin\_tenant\_name = service  admin\_user = quantum  admin\_password = password |

或者运行下面命令

|  |
| --- |
| sed -i -e " s/%SERVICE\_TENANT\_NAME%/service/g; s/%SERVICE\_USER%/quantum/g; s/%SERVICE\_PASSWORD%/$SERVICE\_PASSWORD/g; " /etc/quantum/l3\_agent.ini /etc/quantum/api-paste.ini |

重启服务

|  |
| --- |
| service quantum-server restart  service quantum-plugin-openvswitch-agent restart  service quantum-dhcp-agent restart  service quantum-l3-agent restart |

## Cinder

### 安装

|  |
| --- |
| apt-get install -y cinder-api cinder-scheduler cinder-volume iscsitarget \  open-iscsi iscsitarget-dkms python-cinderclient |

### 配置

分区，我的硬盘专门一个分区给volume使用

|  |
| --- |
| umount /dev/sda5  pvcreate /dev/sda5  vgcreate cinder-volumes /dev/sda5去掉开机挂载  sed -i '/nova-volume/s/^/#/' /etc/fstabiscsi  sed -i 's/false/true/g' /etc/default/iscsitarget  service iscsitarget restart  service open-iscsi restart |

编辑 /etc/cinder/cinder.conf ,直接运行下面命令就可以.

|  |
| --- |
| cat >/etc/cinder/cinder.conf <<EOF  [DEFAULT]  rootwrap\_config = /etc/cinder/rootwrap.conf  sql\_connection = mysql://cinder:$MYSQL\_PASS@$MASTER:3306/cinder  api\_paste\_confg = /etc/cinder/api-paste.ini  iscsi\_helper = ietadm  volume\_group = cinder-volumes  volume\_name\_template = volume-%s  rabbit\_password = $RABBIT\_PASSWORD  logdir = /var/log/cinder  verbose = False  auth\_strategy = keystone  EOF |

编辑 /etc/cinder/api-paste.ini

|  |
| --- |
| admin\_tenant\_name = service  admin\_user = cinder  admin\_password = password |

或者用下面命令

|  |
| --- |
| sed -i -e " s/%SERVICE\_TENANT\_NAME%/service/g; s/%SERVICE\_USER%/cinder/g; s/%SERVICE\_PASSWORD%/$SERVICE\_PASSWORD/g; " /etc/cinder/api-paste.ini |

同步数据库

|  |
| --- |
| cinder-manage db sync |

重启服务

|  |
| --- |
| service cinder-api restart  service cinder-scheduler restart  service cinder-volume restart |

## Horizon

### 安装

|  |
| --- |
| apt-get -y install apache2 libapache2-mod-wsgi openstack-dashboard memcached python-memcache |

### 配置

编辑 /etc/openstack-dashboard/local\_settings.py，删除ubuntu主题，默认的主题有问题，注释掉下面内容

|  |
| --- |
| #Comment these lines  #Enable the Ubuntu theme if it is present.  #try:  # from ubuntu\_theme import \*  #except ImportError:  # pass |

重启服务

|  |
| --- |
| service apache2 restart; service memcached restart |

访问http://10.0.0.0/horizon

user：admin

pass：123456

由于在控制节点没有安装计算服务，所以你是无法创建虚拟机。

# 计算节点

## 操作系统

操作系统最小化安装，ssh server就可以。

## 源

添加Folsom源

|  |
| --- |
| cat <<EOF >>/etc/apt/sources.list  deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-proposed/folsom main  deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/folsom main  EOF |

运行下面命令

|  |
| --- |
| apt-get install ubuntu-cloud-keyring  apt-get update && apt-get -y dist-upgrade |

## 网络

|  |
| --- |
| # cat /etc/network/interfaces  # This file describes network interfaces avaiulable on your system  # and how to activate them. For more information, see interfaces(5).  # Modified by convert\_static.sh.  auto lo  iface lo inet loopback  auto eth0  iface eth0 inet static  address 10.0.0.6  hwaddress ether 00:25:90:2d:7a:42  netmask 255.255.255.0  network 10.0.0.0  gateway 10.0.0.1  dns-search chenshake.com  dns-nameservers 8.8.8.8  # VMs Networks with OVS in tunnel mode  auto eth1  iface eth1 inet static  address 10.0.0.4  netmask 255.255.255.0 |

重启网络

|  |
| --- |
| /etc/init.d/networking restart |

IP转发

|  |
| --- |
| sed -i -r 's/^\s\*#(net\.ipv4\.ip\_forward=1.\*)/\1/' /etc/sysctl.conf  echo 1 > /proc/sys/net/ipv4/ip\_forward |

环境变量

|  |
| --- |
| cat >/root/novarc <<EOF  export CONTROLLER\_IP=10.0.0.58  export MASTER="$(/sbin/ifconfig eth0 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"  export LOCAL\_IP="$(/sbin/ifconfig eth1 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"  EOF  source novarc  echo "source novarc">>.bashrc |

## NTP

### 安装

|  |
| --- |
| apt-get -y install ntp |

### 设置

编辑 /etc/ntp.conf, 指向控制节点

|  |
| --- |
| server 10.0.0.58 |

或者运行命令

|  |
| --- |
| sed -i -e " s/server ntp.ubuntu.com/server $CONTROLLER\_IP/g" /etc/ntp.conf |

重启服务

|  |
| --- |
| service ntp restart |

## Hypervisor

### 安装

|  |
| --- |
| apt-get install -y kvm libvirt-bin pm-utils |

### 配置

编辑 /etc/libvirt/qemu.conf ，添加下面内容

|  |
| --- |
| cgroup\_device\_acl = [  "/dev/null", "/dev/full", "/dev/zero",  "/dev/random", "/dev/urandom",  "/dev/ptmx", "/dev/kvm", "/dev/kqemu",  "/dev/rtc", "/dev/hpet","/dev/net/tun",  ] |

或者运行命令：这个地方用命令修改有点复杂，还没找到太好的办法。

|  |
| --- |
| cat <<EOF>>/etc/libvirt/qemu.conf  cgroup\_device\_acl = [  "/dev/null", "/dev/full", "/dev/zero",  "/dev/random", "/dev/urandom",  "/dev/ptmx", "/dev/kvm", "/dev/kqemu",  "/dev/rtc", "/dev/hpet","/dev/net/tun",  ]  EOF |

删除默认 virtual bridge

|  |
| --- |
| virsh net-destroy default  virsh net-undefine default |

允许迁移

编辑 /etc/libvirt/libvirtd.conf, 去掉这三行的注释

|  |
| --- |
| listen\_tls = 0  listen\_tcp = 1  auth\_tcp = "none" |

或者运行下面命令

|  |
| --- |
| sed -i '/#listen\_tls/s/#listen\_tls/listen\_tls/; /#listen\_tcp/s/#listen\_tcp/listen\_tcp/; /#auth\_tcp/s/#auth\_tcp/auth\_tcp/; /auth\_tcp/s/sasl/none/' /etc/libvirt/libvirtd.conf |

编辑 /etc/init/libvirt-bin.conf

|  |
| --- |
| env libvirtd\_opts="-d -l" |

或者使用命令

|  |
| --- |
| sed -i '/env libvirtd\_opts/s/-d/-d –l/' /etc/init/libvirt-bin.conf |

编辑 /etc/default/libvirt-bin

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

或者使用命令

|  |
| --- |
| sed -i '/libvirtd\_opts/s/-d/-d -l/' /etc/default/libvirt-bin |

重启服务

|  |
| --- |
| service libvirt-bin restartOpen-vSwitch  apt-get install -y openvswitch-switch |

创建bridge

|  |
| --- |
| ovs-vsctl add-br br-int  ovs-vsctl add-br br-eth1  ovs-vsctl add-port br-eth1 eth1 |

## Quantum

### 安装

|  |
| --- |
| apt-get -y install quantum-plugin-openvswitch-agent |

### 配置

编辑 /etc/quantum/quantum.conf , 修改和控制节点一样，直接从控制直接复制过来

|  |
| --- |
| scp root@$CONTROLLER\_IP:/etc/quantum/quantum.conf /etc/quantum/quantum.conf |

编辑 /etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini，可以从控制节点copy过来

|  |
| --- |
| scp root@$CONTROLLER\_IP:/etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini /etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini |

启动agent

|  |
| --- |
| service quantum-plugin-openvswitch-agent restart |

## Nova

### 安装

|  |
| --- |
| apt-get -y install nova-api-metadata nova-compute-kvm novnc nova-novncproxy |

### 配置

编辑 /etc/nova/api-paste.ini

|  |
| --- |
| [filter:authtoken]  paste.filter\_factory = keystone.middleware.auth\_token:filter\_factory  auth\_host = 10.0.0.58  auth\_port = 35357  auth\_protocol = http  admin\_tenant\_name = service  admin\_user = nova  admin\_password = password  signing\_dirname = /tmp/keystone-signing-nova |

或者运行下面命令,直接从控制节点复制过来就可以。

|  |
| --- |
| scp root@$CONTROLLER\_IP:/etc/nova/api-paste.ini /etc/nova/ |

编辑 /etc/nova/nova-compute.conf

|  |
| --- |
| [DEFAULT]  libvirt\_type=kvm  libvirt\_ovs\_bridge=br-int  libvirt\_vif\_type=ethernet  libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver  libvirt\_use\_virtio\_for\_bridges=True |

或者运行下面命令

|  |
| --- |
| cat > /etc/nova/nova-compute.conf <<EOF  [DEFAULT]  libvirt\_type=kvm  libvirt\_ovs\_bridge=br-int  libvirt\_vif\_type=ethernet  libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver  libvirt\_use\_virtio\_for\_bridges=True  EOF |

编辑 /etc/nova/nova.conf。我们可以从控制节点copy过来修改

|  |
| --- |
| scp root@$CONTROLLER\_IP:/etc/nova/nova.conf /etc/nova/nova.conf |

修改vnc

|  |
| --- |
| # Vnc configuration  novnc\_enabled=true  novncproxy\_base\_url=http://10.0.0.58:6080/vnc\_auto.html  novncproxy\_port=6080  vncserver\_proxyclient\_address=10.0.0.6  vncserver\_listen=10.0.0.6 |

或者使用下面命令修改

|  |
| --- |
| sed -i 's/^vncserver\_proxyclient\_address.\*$/vncserver\_proxyclient\_address='"$(/sbin/ifconfig eth0 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"'/g' /etc/nova/nova.conf  sed -i 's/^vncserver\_listen.\*$/vncserver\_listen='"$(/sbin/ifconfig eth0 | awk '/inet addr/ {print $2}' | cut -f2 -d ":")"'/g' /etc/nova/nova.conf |

重启服务

|  |
| --- |
| service nova-api-metadata restart  service nova-novncproxy restart  service nova-compute restart |

或者

|  |
| --- |
| cd /etc/init.d/; for i in $( ls nova-\* ); do sudo service $i restart; done |

查看服务

|  |
| --- |
| nova-manage service list |