

全国职业院校技能大赛改革试点赛云计算赛项

国基北盛云基础架构平台软件（私有云平台）

用户手册

V1.0

2020年10月

目 录

[简介 7](#_Toc54861058)

[1 基本环境配置 8](#_Toc54861059)

[1.1安装CentOS7说明 9](#_Toc54861060)

[1.2配置网络、主机名 9](#_Toc54861061)

[1.3配置yum源 11](#_Toc54861062)

[1.4编辑环境变量 13](#_Toc54861063)

[1.5通过脚本安装服务 15](#_Toc54861064)

[1.6安装Openstack包 15](#_Toc54861065)

[1.7配置域名解析 15](#_Toc54861066)

[1.8安装chrony服务 16](#_Toc54861067)

[1.9通过脚本安装数据库服务 16](#_Toc54861068)

[1.10安装Mysql数据库服务 17](#_Toc54861069)

[1.11安装RabbitMQ服务 18](#_Toc54861070)

[1.12安装memcahce服务 18](#_Toc54861071)

[1.13 安装etcd服务 18](#_Toc54861072)

[2 安装Keystone认证服务 19](#_Toc54861073)

[2.1 通过脚本安装keystone服务 19](#_Toc54861074)

[2.2安装keystone服务软件包 19](#_Toc54861075)

[2.3创建Keystone数据库 19](#_Toc54861076)

[2.4配置数据库连接 20](#_Toc54861077)

[2.5为keystone服务创建数据库表 20](#_Toc54861078)

[2.6创建令牌 20](#_Toc54861079)

[2.7创建签名密钥和证书 20](#_Toc54861080)

[2.8定义用户、租户和角色 23](#_Toc54861081)

[2.9 创建demo-openrc.sh 24](#_Toc54861082)

[2.10创建admin-openrc.sh 24](#_Toc54861083)

[3 安装Glance镜像服务 25](#_Toc54861084)

[3.1 通过脚本安装glance服务 25](#_Toc54861085)

[3.2 安装Glance镜像服务软件包 25](#_Toc54861086)

[3.3创建Glance数据库 25](#_Toc54861087)

[3.4配置数据库连接 25](#_Toc54861088)

[3.5为镜像服务创建数据库表 25](#_Toc54861089)

[3.6创建用户 26](#_Toc54861090)

[3.7配置镜像服务 26](#_Toc54861091)

[3.8创建Endpoint和API端点 27](#_Toc54861092)

[3.9启动服务 27](#_Toc54861093)

[3.10上传镜像 27](#_Toc54861094)

[4 安装Nova计算服务 28](#_Toc54861095)

[4.1通过脚本安装nova服务 28](#_Toc54861096)

[4.2安装Nova 计算服务软件包 28](#_Toc54861097)

[4.3创建Nova数据库 28](#_Toc54861098)

[4.4为计算服务创建数据库表 29](#_Toc54861099)

[4.5创建用户 29](#_Toc54861100)

[4.6配置计算服务 29](#_Toc54861101)

[4.7创建Endpoint和API端点 31](#_Toc54861102)

[4.8 添加配置 32](#_Toc54861103)

[4.9启动服务 32](#_Toc54861104)

[4.10验证Nova数据库是否创建成功 32](#_Toc54861105)

[4.11安装Nova计算服务软件包 33](#_Toc54861106)

[4.12配置Nova服务 33](#_Toc54861107)

[4.13检查系统处理器是否支持虚拟机的硬件加速 34](#_Toc54861108)

[4.14启动 34](#_Toc54861109)

[4.15 添加计算节点 34](#_Toc54861110)

[5 安装Neutron网络服务 35](#_Toc54861111)

[5.1通过脚本安装neutron服务 35](#_Toc54861112)

[5.2创建Neutron数据库 35](#_Toc54861113)

[5.3创建用户 35](#_Toc54861114)

[5.4创建Endpoint和API端点 35](#_Toc54861115)

[5.5安装neutron网络服务软件包 36](#_Toc54861116)

[5.6配置Neutron服务 36](#_Toc54861117)

[5.7 创建数据库 39](#_Toc54861118)

[5.8 启动服务和创建网桥 39](#_Toc54861119)

[5.9 安装软件包 39](#_Toc54861120)

[5.10 配置Neutron服务 39](#_Toc54861121)

[5.11 启动服务进而创建网桥 41](#_Toc54861122)

[6 安装Dashboard服务 41](#_Toc54861123)

[6.1通过脚本安装dashboard服务 41](#_Toc54861124)

[6.2安装Dashboard服务软件包 41](#_Toc54861125)

[6.3配置 41](#_Toc54861126)

[6.4启动服务 52](#_Toc54861127)

[6.5访问 52](#_Toc54861128)

[6.6创建云主机 52](#_Toc54861129)

[7 安装Cinder块存储服务 53](#_Toc54861130)

[7.1 通过脚本安装Cinder服务 53](#_Toc54861131)

[7.2 安装Cinder块存储服务软件包 53](#_Toc54861132)

[7.3 创建数据库 53](#_Toc54861133)

[7.4 创建用户 53](#_Toc54861134)

[7.5 创建Endpoint和API端点 54](#_Toc54861135)

[7.6 配置Cinder服务 54](#_Toc54861136)

[7.7 创建数据库 55](#_Toc54861137)

[7.8 启动服务 55](#_Toc54861138)

[7.9 安装块存储软件 56](#_Toc54861139)

[7.10 创建LVM物理和逻辑卷 56](#_Toc54861140)

[7.11 修改Cinder配置文件 56](#_Toc54861141)

[7.12 重启服务 57](#_Toc54861142)

[7.13 验证 57](#_Toc54861143)

[8 安装Swift对象存储服务 58](#_Toc54861144)

[8.1通过脚本安装Swift服务 58](#_Toc54861145)

[8.2 安装Swift对象存储服务软件包 58](#_Toc54861146)

[8.2创建用户 58](#_Toc54861147)

[8.3创建Endpoint和API端点 58](#_Toc54861148)

[8.4 编辑/etc/swift/proxy-server.conf 59](#_Toc54861149)

[8.5 创建账号、容器、对象 61](#_Toc54861150)

[8.6 编辑/etc/swift/swift.conf文件 62](#_Toc54861151)

[8.7 启动服务和赋予权限 62](#_Toc54861152)

[8.8 安装软件包 63](#_Toc54861153)

[8.9 配置rsync 63](#_Toc54861154)

[8.10 配置账号、容器和对象 65](#_Toc54861155)

[8.11 修改Swift配置文件 67](#_Toc54861156)

[8.12 重启服务和赋予权限 67](#_Toc54861157)

[9 安装Heat编配服务 68](#_Toc54861158)

[9.1通过脚本安装heat服务 68](#_Toc54861159)

[9.2安装heat编配服务软件包 68](#_Toc54861160)

[9.3创建数据库 68](#_Toc54861161)

[9.4创建用户 69](#_Toc54861162)

[9.5创建Endpoint和API端点 69](#_Toc54861163)

[9.6配置Heat服务 70](#_Toc54861164)

[9.7创建数据库 71](#_Toc54861165)

[9.8启动服务 71](#_Toc54861166)

[10 安装Zun服务 71](#_Toc54861167)

[10.1通过脚本安装Zun服务 71](#_Toc54861168)

[10.2 安装zun服务软件包 71](#_Toc54861169)

[10.3 创建数据库 71](#_Toc54861170)

[10.4 创建用户 72](#_Toc54861171)

[10.5 创建Endpoint和API端点 72](#_Toc54861172)

[10.6 配置zun服务 72](#_Toc54861173)

[10.7 创建数据库 74](#_Toc54861174)

[10.8 启动服务 74](#_Toc54861175)

[10.9 安装软件包 74](#_Toc54861176)

[10.10 配置服务 74](#_Toc54861177)

[10.11 修改内核参数 76](#_Toc54861178)

[10.12 启动服务 76](#_Toc54861179)

[10.13 上传镜像 76](#_Toc54861180)

[10.14 启动容器 77](#_Toc54861181)

[11 安装Ceilometer监控服务 77](#_Toc54861182)

[11.1通过脚本安装Ceilometer服务 77](#_Toc54861183)

[11.2 安装Ceilometer监控服务软件包 77](#_Toc54861184)

[11.3 创建数据库 78](#_Toc54861185)

[11.4 创建用户 78](#_Toc54861186)

[11.5 创建Endpoint和API端点 78](#_Toc54861187)

[11.6 配置Ceilometer 79](#_Toc54861188)

[11.7 创建监听端点 81](#_Toc54861189)

[11.8 创建数据库 81](#_Toc54861190)

[11.9 启动服务 82](#_Toc54861191)

[11.10 监控组件 82](#_Toc54861192)

[11.11 添加变量参数 83](#_Toc54861193)

[11.12 安装软件包 83](#_Toc54861194)

[11.13 配置Ceilometer 83](#_Toc54861195)

[11.14 启动服务 84](#_Toc54861196)

[12 安装Aodh监控服务 84](#_Toc54861197)

[12.1通过脚本安装Aodh服务 84](#_Toc54861198)

[12.2 创建数据库 84](#_Toc54861199)

[12.3 创建keystone用户 85](#_Toc54861200)

[12.4 创建Endpoint和API 85](#_Toc54861201)

[12.5 安装软件包 85](#_Toc54861202)

[12.6 配置aodh 85](#_Toc54861203)

[12.7 创建监听端点 86](#_Toc54861204)

[12.8 同步数据库 87](#_Toc54861205)

[12.9 启动服务 87](#_Toc54861206)

[13 添加控制节点资源到云平台 88](#_Toc54861207)

[13.1 修改openrc.sh 88](#_Toc54861208)

[13.2 运行iaas-install-nova-compute.sh 88](#_Toc54861209)

# 简介

chinaskills\_cloud\_iaas.iso镜像包含OpenStack Q版本私有云平台搭建的各项软件包、依赖包、安装脚本等，同时还提供了CentOS7.2、CentOS7.5等云主机qcow2镜像，可满足私有云平台的搭建、云平台的使用、各组件的运维操作等。

chinaskills\_cloud\_iaas.iso包含的具体内容如下：

|  |  |  |
| --- | --- | --- |
| 编号 | 软件包 | 详细信息 |
| 1 | iaas-repo | 提供安装脚本，可用安装脚本快捷部署OpenStack私有云平台 |
| 根据iaas-repo镜像源目录，可用于安装KeyStone服务，以及对keystone认证服务进行创建用户、租户、管理权限等操作 |
| 根据iaas-repo镜像源目录，可用于安装Glance服务，以及对glance服务进行上传镜像、删除镜像、创建快照等操作 |
| 根据iaas-repo镜像源目录，可用于安装Nova服务，以及对nova服务进行启动云主机、创建云主机类型、删除云主机等操作 |
| 根据iaas-repo镜像源目录，可用于安装Neutron服务，以及对neutron服务进行创建网络、删除网络、编辑网络等操作 |
| 根据iaas-repo镜像源目录，可用于安装Horzion服务，可以通过Horzion Dashboard界面对OpenStack平台进行管理 |
| 根据iaas-repo镜像源目录，可用于安装Cinder服务，以及对Cinder服务进行创建块设备、管理块设备连接、删除块设备等操作 |
| 根据iaas-repo镜像源目录，可用于安装Swift服务，以及对Swift服务进行创建容器、上传对象、删除对象等操作 |
| 根据iaas-repo镜像源目录，可用于安装Heat服务，可通过编辑模板文件，实现Heat编排操作 |
| 根据iaas-repo镜像源目录，可用于安装Ceilometer和Aodh监控服务，可通过这两个服务对私有云平台进行监控与告警 |
| 根据iaas-repo镜像源目录，可用于安装Zun服务，Zun服务可在OpenStack私有云平台中提供容器服务 |
| 2 | images | 提供CentOS7\_1804.tar（容器镜像），可用于Zun服务启动容器镜像 |
| 提供CentOS\_7.5\_x86\_64\_XD.qcow2镜像，该镜像为CentOS7.5版本的虚拟机镜像，可基于该镜像启动CentOS7.5的云主机，用于各项操作与服务搭建 |
| 提供CentOS\_7.2\_x86\_64\_XD.qcow2镜像，该镜像为CentOS7.2版本的虚拟机镜像，可基于该镜像启动CentOS7.2的云主机，用于各项操作与服务搭建 |
| 提供CentOS\_6.5\_x86\_64\_XD.qcow2镜像，该镜像为CentOS6.5版本的虚拟机镜像，可基于该镜像启动CentOS6.5的云主机，用于各项操作与服务搭建 |

# 1 基本环境配置

云计算平台的拓扑图如图1所示，IP地址规划如下图所示。

**云计算IaaS控制节点**



**Internet**

eth0

**核心交换机**

**中心防火墙**

Rabbit消息服务

Neutron Server网络服务

Dashboard管理界面

Mysql数据库

eth0

Nova计算控制服务

Glance镜像服务

Keystone安全认证服务

192.168.100.10/24

192.168.200.10/24

eth1

eth1

Cinder存储控制服务

Swift 代理服务

Heat编配服务

Ceilometer 监控服务

192.168.100.1/24

**云计算IaaS计算节点**

192.168.200.1/24

Nova Compute计算服务

192.168.100.20/24

Neutron节点网络服务

Cinder Volume存储服务

Swift 存储服务

192.168.200.20/24

Ceilometer监控代理

本次搭建采用双节点安装，即controller node控制节点和compute node计算节点。**enp8s0**为内部管理网络，**enp9s0**为外部网络。存储节点安装操作系统时划分两个空白分区以sda，sdb为例。作为cinder和swift存储磁盘，搭建 ftp服务器作为搭建云平台的yum源。配置文件中密码需要根据实际环境进行配置。

## 1.1安装CentOS7说明

**【CentOS7版本】**

CentOS7系统选择1804版本：CentOS-7-x86\_64-DVD-1804.iso

**【空白分区划分】**

CentOS7的安装与CentOS6.5的安装有明显的区别。在CentOS7安装过程中，设置分区都需要一个挂载点，这样一来就无法创建两个空白的磁盘分区作为cinder服务和swift服务的存储磁盘了。

所以我们应该在系统安装过程中留下足够的磁盘大小，系统安装完成后，使用命令parted划分新分区，然后使用mkfs.xfs进行文件系统格式化，完成空白分区的划分。具体命令如下：

[root@compute ~]# parted /dev/md126

(parted) mkpart swift 702G 803G //创建swift分区，从702G到803G

[root@compute ~]# mkfs.xfs /dev/md126p5

## 1.2配置网络、主机名

修改和添加/etc/sysconfig/network-scripts/ifcfg-**enp**\*（具体的网口）文件。

**（1）controller节点**

**配置网络：**

**enp8s0:** **192.168.100.10**

DEVICE=enp8s0

TYPE=Ethernet

ONBOOT=yes

NM\_CONTROLLED=no

BOOTPROTO=static

IPADDR=192.168.100.10

PREFIX=24

GATEWAY=192.168.100.1

**enp9s0:** **192.168.200.10**

DEVICE=enp9s0

TYPE=Ethernet

ONBOOT=yes

NM\_CONTROLLED=no

BOOTPROTO=static

IPADDR=192.168.200.10

PREFIX=24

**配置主机名：**

# hostnamectl set-hostname controller

按ctrl+d 退出 重新登陆

**（2）compute 节点**

**配置网络：**

**enp8s0:** **192.168.100.20**

DEVICE=enp8s0

TYPE=Ethernet

ONBOOT=yes

NM\_CONTROLLED=no

BOOTPROTO=static

IPADDR=192.168.100.20

PREFIX=24

GATEWAY=192.168.100.1

**enp9s0:** **192.168.200.20**

DEVICE=enp9s0

TYPE=Ethernet

ONBOOT=yes

NM\_CONTROLLED=no

BOOTPROTO=static

IPADDR=192.168.200.20

PREFIX=24

**配置主机名：**

# hostnamectl set-hostname compute

按ctrl+d 退出 重新登陆

## 1.3配置yum源

**#Controller和compute节点**

**（1）yum源备份**

#mv /etc/yum.repos.d/\* /opt/

**（2）创建repo文件**

**【controller】**

在/etc/yum.repos.d创建centos.repo源文件

[centos]

name=centos

baseurl=file:///opt/centos

gpgcheck=0

enabled=1

[iaas]

name=iaas

baseurl=file:///opt/iaas-repo

gpgcheck=0

enabled=1

【compute】

在/etc/yum.repos.d创建centos.repo源文件

[centos]

name=centos

baseurl=ftp://192.168.100.10/centos

gpgcheck=0

enabled=1

[iaas]

name=iaas

baseurl=ftp://192.168.100.10/iaas-repo

gpgcheck=0

enabled=1

**（3）挂载iso文件**

**【挂载CentOS-7-x86\_64-DVD-1804.iso】**

[root@controller ~]# mount -o loop CentOS-7-x86\_64-DVD-1804.iso /mnt/

[root@controller ~]# mkdir /opt/centos

[root@controller ~]# cp -rvf /mnt/\* /opt/centos/

[root@controller ~]# umount /mnt/

**【挂载XianDian-IaaS-v2.4.iso】**

[root@controller ~]# mount -o loop XianDian-IaaS-v2.4.iso /mnt/

[root@controller ~]# cp -rvf /mnt/\* /opt/

[root@controller ~]# umount /mnt/

**（4）搭建ftp服务器，开启并设置自启**

[root@controller ~]# yum install vsftpd -y

[root@controller ~]# vi /etc/vsftpd/vsftpd.conf

添加anon\_root=/opt/

保存退出

[root@controller ~]# systemctl start vsftpd

[root@controller ~]# systemctl enable vsftpd

**（5）配置防火墙和Selinux**

**【controller/compute】**

编辑selinux文件

# vi /etc/selinux/config

SELINUX=permissive

关闭防火墙并设置开机不自启

# systemctl stop firewalld.service

# systemctl disable firewalld.service

# yum remove -y NetworkManager firewalld

# yum -y install iptables-services

# systemctl enable iptables

# systemctl restart iptables

# iptables -F

# iptables -X

# iptables -Z

# service iptables save

**（6）清除缓存，验证yum源**

**【controller/compute】**

# yum clean all

# yum list

## 1.4编辑环境变量

**# controller和compute节点**

# yum install iaas-xiandian -y

编辑文件/etc/xiandian/openrc.sh,此文件是安装过程中的各项参数，根据每项参数上一行的说明及服务器实际情况进行配置。

HOST\_IP=192.168.100.10

HOST\_PASS=000000

HOST\_NAME=controller

HOST\_IP\_NODE=192.168.100.20

HOST\_PASS\_NODE=000000

HOST\_NAME\_NODE=compute

network\_segment\_IP=192.168.100.0/24

RABBIT\_USER=openstack

RABBIT\_PASS=000000

DB\_PASS=000000

DOMAIN\_NAME=demo

ADMIN\_PASS=000000

DEMO\_PASS=000000

KEYSTONE\_DBPASS=000000

GLANCE\_DBPASS=000000

GLANCE\_PASS=000000

NOVA\_DBPASS=000000

NOVA\_PASS=000000

NEUTRON\_DBPASS=000000

NEUTRON\_PASS=000000

METADATA\_SECRET=000000

INTERFACE\_IP=192.168.100.10/192.168.100.20**（controllerIP/computeIP）**

INTERFACE\_NAME=enp9s0 （外部网络网卡名称）

Physical\_NAME=provider （外部网络适配器名称）

minvlan=101 （vlan网络范围的第一个vlanID）

maxvlan=200 （vlan网络范围的最后一个vlanID）

CINDER\_DBPASS=000000

CINDER\_PASS=000000

BLOCK\_DISK=md126p4 （空白分区）

SWIFT\_PASS=000000

OBJECT\_DISK=md126p5 （空白分区）

STORAGE\_LOCAL\_NET\_IP=192.168.100.20

HEAT\_DBPASS=000000

HEAT\_PASS=000000

ZUN\_DBPASS=000000

ZUN\_PASS=000000

KURYR\_DBPASS=000000

KURYR\_PASS=000000

CEILOMETER\_DBPASS=000000

CEILOMETER\_PASS=000000

AODH\_DBPASS=000000

AODH\_PASS=000000

## 1.5通过脚本安装服务

**1.6-1.8的基础配置操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**# Controller节点和Compute节点**

**执行脚本iaas-pre-host.sh进行安装**

[root@controller ~]# iaas-pre-host.sh

**# 安装完成后同时重启**

[root@controller ~]# reboot

## 1.6安装Openstack包

**# controller和compute节点**

# yum -y install openstack-utils openstack-selinux python-openstackclient

# yum upgrade

## 1.7配置域名解析

修改/etc/hosts添加一下内容

**（1）controller 节点**

192.168.100.10 controller

192.168.100.20 compute

1. **compute 节点**

192.168.100.10 controller

192.168.100.20 compute

## 1.8安装chrony服务

**（1）controller和compute节点**

# yum install -y chrony

**（2）配置controller节点**

编辑/etc/chrony.conf文件

添加以下内容（删除默认sever规则）

server controller iburst

allow 192.168.100.0/24

local stratum 10

启动ntp服务器

# systemctl restart chronyd

# systemctl enable chronyd

**（3）配置compute节点**

编辑/etc/chrony.conf文件

添加以下内容（删除默认sever规则）

server controller iburst

启动ntp服务器

# systemctl restart chronyd

# systemctl enable chronyd

## 1.9通过脚本安装数据库服务

**1.10-1.13基础服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**# Controller节点**

**执行脚本iaas-install-mysql.sh进行安装**

## 1.10安装Mysql数据库服务

**（1）安装mysql服务**

# yum install -y mariadb mariadb-server python2-PyMySQL

**（2）修改mysql配置文件参数**

修改 /etc/my.cnf文件[mysqld]中添加

max\_connections=10000

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

**（3）启动服务**

#systemctl enable mariadb.service

#systemctl start mariadb.service

**（4）修改/usr/lib/systemd/system/mariadb.service文件参数**

[Service]

新添加两行如下参数：

**LimitNOFILE=10000**

**LimitNPROC=10000**

**（5）修改/etc/my.cnf.d/auth\_gssapi.cnf文件参数**

[mariadb]

注释一行参数

**#plugin-load-add=auth\_gssapi.so**

**（6）重新加载系统服务，并重启mariadb服务**

# systemctl daemon-reload

# service mariadb restart

**（7）配置Mysql**

# mysql\_secure\_installation

**按enter确认后设置数据库root密码**

**Remove anonymous users? [Y/n] y**

**Disallow root login remotely? [Y/n] n**

**Remove test database and access to it? [Y/n] y**

**Reload privilege tables now? [Y/n] y**

**（8）compute节点**

#yum -y install MySQL-python

## 1.11安装RabbitMQ服务

# yum install -y rabbitmq-server

# systemctl enable rabbitmq-server.service

# systemctl restart rabbitmq-server.service

# rabbitmqctl add\_user $RABBIT\_USER $RABBIT\_PASS

# rabbitmqctl set\_permissions $RABBIT\_USER ".\*" ".\*" ".\*"

## 1.12安装memcahce服务

# yum install memcached python-memcached

# systemctl enable memcached.service

# systemctl restart memcached.service

## 1.13 安装etcd服务

# yum install etcd –y

**（1）修改/etc/etcd/etcd.conf配置文件，添加以下内容：**

ETCD\_LISTEN\_PEER\_URLS="http://192.168.100.10:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.100.10:2379"

ETCD\_NAME="controller"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.100.10:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.100.10:2379"

ETCD\_INITIAL\_CLUSTER="controller=http://192.168.100.10:2380"

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

ETCD\_INITIAL\_CLUSTER\_STATE="new"

**（2）启动服务**

# systemctl start etcd

# systemctl enable etcd

# 2 安装Keystone认证服务

#**Controller**

## 2.1 通过脚本安装keystone服务

**2.2-2.10的认证服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**# Controller节点**

**执行脚本iaas-install-keystone.sh进行安装。**

## 2.2安装keystone服务软件包

yum install -y openstack-keystone httpd mod\_wsgi

## 2.3创建Keystone数据库

# mysql –u root -p（此处数据库密码为之前安装Mysql设置的密码）mysql> CREATE DATABASE keystone;mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY '**KEYSTONE\_DBPASS**';mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY '**KEYSTONE\_DBPASS**';mysql> exit

## 2.4配置数据库连接

# crudini --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:$KEYSTONE\_DBPASS@$HOST\_NAME/keystone

## 2.5为keystone服务创建数据库表

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

## 2.6创建令牌

#ADMIN\_TOKEN=$(openssl rand -hex 10)

# crudini --set /etc/keystone/keystone.conf DEFAULT admin\_token $ADMIN\_TOKEN

# crudini --set /etc/keystone/keystone.conf token provider fernet

## 2.7创建签名密钥和证书

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

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

修改/etc/httpd/conf/httpd.conf配置文件将ServerName [www.example.com:80](http://www.example.com:80) 替换为ServerName controller

创建/etc/httpd/conf.d/wsgi-keystone.conf文件，内容如下：

Listen 5000

Listen 35357

<VirtualHost \*:5000>

WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone group=keystone display-name=%{GROUP}

WSGIProcessGroup keystone-public

WSGIScriptAlias / /usr/bin/keystone-wsgi-public

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

LimitRequestBody 114688

<IfVersion >= 2.4>

ErrorLogFormat "%{cu}t %M"

</IfVersion>

ErrorLog /var/log/httpd/keystone.log

CustomLog /var/log/httpd/keystone\_access.log combined

<Directory /usr/bin>

<IfVersion >= 2.4>

Require all granted

</IfVersion>

<IfVersion < 2.4>

Order allow,deny

Allow from all

</IfVersion>

</Directory>

</VirtualHost>

<VirtualHost \*:35357>

WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone group=keystone display-name=%{GROUP}

WSGIProcessGroup keystone-admin

WSGIScriptAlias / /usr/bin/keystone-wsgi-admin

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

LimitRequestBody 114688

<IfVersion >= 2.4>

ErrorLogFormat "%{cu}t %M"

</IfVersion>

ErrorLog /var/log/httpd/keystone.log

CustomLog /var/log/httpd/keystone\_access.log combined

<Directory /usr/bin>

<IfVersion >= 2.4>

Require all granted

</IfVersion>

<IfVersion < 2.4>

Order allow,deny

Allow from all

</IfVersion>

</Directory>

</VirtualHost>

Alias /identity /usr/bin/keystone-wsgi-public

<Location /identity>

SetHandler wsgi-script

Options +ExecCGI

WSGIProcessGroup keystone-public

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

</Location>

Alias /identity\_admin /usr/bin/keystone-wsgi-admin

<Location /identity\_admin>

SetHandler wsgi-script

Options +ExecCGI

WSGIProcessGroup keystone-admin

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

</Location>

#systemctl enable httpd.service

#systemctl start httpd.service

## 2.8定义用户、租户和角色

**（1）设置环境变量**

export OS\_TOKEN=$ADMIN\_TOKEN

export OS\_URL=http://controller:35357/v3

export OS\_IDENTITY\_API\_VERSION=3

**（2）创建keystone相关内容**

openstack service create --name keystone --description "OpenStack Identity" identity

openstack endpoint create --region RegionOne identity public http://$HOST\_NAME:5000/v3

openstack endpoint create --region RegionOne identity internal http://$HOST\_NAME:5000/v3

openstack endpoint create --region RegionOne identity admin http://$HOST\_NAME:35357/v3

openstack domain create --description "Default Domain" $DOMAIN\_NAME

openstack project create --domain $DOMAIN\_NAME --description "Admin Project" admin

openstack user create --domain $DOMAIN\_NAME --password $ADMIN\_PASS admin

openstack role create admin

openstack role add --project admin --user admin admin

openstack project create --domain $DOMAIN\_NAME --description "Service Project" service

openstack project create --domain $DOMAIN\_NAME --description "Demo Project" demo

openstack user create --domain $DOMAIN\_NAME --password $DEMO\_PASS demo

openstack role create user

openstack role add --project demo --user demo user

**（3）清除环境变量**

#unset OS\_TOKEN OS\_URL

## 2.9 创建demo-openrc.sh

创建demo环境变量demo-openrc.sh

export OS\_PROJECT\_DOMAIN\_NAME=$DOMAIN\_NAME

export OS\_USER\_DOMAIN\_NAME=$DOMAIN\_NAME

export OS\_PROJECT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=$DEMO\_PASS

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

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

## 2.10创建admin-openrc.sh

创建admin环境变量admin-openrc.sh

export OS\_PROJECT\_DOMAIN\_NAME=$DOMAIN\_NAME

export OS\_USER\_DOMAIN\_NAME=$DOMAIN\_NAME

export OS\_PROJECT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=$ADMIN\_PASS

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

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

生效环境变量

#source admin-openrc.sh

# 3 安装Glance镜像服务

#**Controller**

## 3.1 通过脚本安装glance服务

**3.2-3.9的镜像服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**# Controller 节点**

**执行脚本iaas-install-glance.sh进行安装**

## 3.2 安装Glance镜像服务软件包

# yum install -y openstack-glance

## 3.3创建Glance数据库

#mysql -u root -p

mysql> CREATE DATABASE glance;

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY '**GLANCE\_DBPASS**';

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY '**GLANCE\_DBPASS**';

## 3.4配置数据库连接

# crudini --set /etc/glance/glance-api.conf database connection mysql+pymysql://glance:$GLANCE\_DBPASS@$HOST\_NAME/glance

# crudini --set /etc/glance/glance-registry.conf database connection mysql+pymysql://glance:$GLANCE\_DBPASS@$HOST\_NAME/glance

## 3.5为镜像服务创建数据库表

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

## 3.6创建用户

# openstack user create --domain $DOMAIN\_NAME --password $GLANCE\_PASS glance

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

## 3.7配置镜像服务

# crudini --set /etc/glance/glance-api.conf database connection mysql+pymysql://glance:$GLANCE\_DBPASS@$HOST\_NAME/glance

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken auth\_type password

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken project\_name service

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken username glance

# crudini --set /etc/glance/glance-api.conf keystone\_authtoken password $GLANCE\_PASS

# crudini --set /etc/glance/glance-api.conf paste\_deploy flavor keystone

# crudini --set /etc/glance/glance-api.conf glance\_store stores file,http

# crudini --set /etc/glance/glance-api.conf glance\_store default\_store file

# crudini --set /etc/glance/glance-api.conf glance\_store filesystem\_store\_datadir /var/lib/glance/images/

# crudini --set /etc/glance/glance-registry.conf database connection mysql+pymysql://glance:$GLANCE\_DBPASS@$HOST\_NAME/glance

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_type password

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken project\_name service

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken username glance

# crudini --set /etc/glance/glance-registry.conf keystone\_authtoken password $GLANCE\_PASS

# crudini --set /etc/glance/glance-registry.conf paste\_deploy flavor keystone

## 3.8创建Endpoint和API端点

# openstack service create --name glance --description "OpenStack Image" image

# openstack endpoint create --region RegionOne image public http://$HOST\_NAME:9292

# openstack endpoint create --region RegionOne image internal http://$HOST\_NAME:9292

# openstack endpoint create --region RegionOne image admin http://$HOST\_NAME:9292

## 3.9启动服务

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

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

## 3.10上传镜像

首先下载（Wget）提供的系统镜像到本地，本次以上传CentOS\_7.5\_x86\_64镜像为例。

可以安装Wget，从Ftp服务器上下载镜像到本地。

# source admin-openrc.sh

# glance image-create --name "CentOS7.5" --disk-format qcow2 --container-format bare --progress < /opt/images/CentOS\_7.5\_x86\_64\_XD.qcow2

# 4 安装Nova计算服务

#**Controller**

## 4.1通过脚本安装nova服务

**4.2-4.15计算服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-nova-controller.sh进行安装**

**#Compute节点**

**执行脚本iaas-install-nova-compute.sh进行安装**

## 4.2安装Nova 计算服务软件包

# yum install openstack-nova-api openstack-nova-conductor openstack-nova-console openstack-nova-novncproxy openstack-nova-scheduler openstack-nova-placement-api -y

## 4.3创建Nova数据库

# mysql -u root -p

mysql> CREATE DATABASE nova;

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY '**NOVA\_DBPASS**';

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY '**NOVA\_DBPASS**';

mysql> create database IF NOT EXISTS nova\_api;

mysql> GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'localhost' IDENTIFIED BY '**NOVA\_DBPASS**' ;

mysql> GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'%' IDENTIFIED BY '**NOVA\_DBPASS**' ;

mysql> create database IF NOT EXISTS nova\_cell0;

mysql> GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'localhost' IDENTIFIED BY '**NOVA\_DBPASS**' ;

mysql> GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'%' IDENTIFIED BY '**NOVA\_DBPASS**' ;

修改数据库连接

# crudini --set /etc/nova/nova.conf database connection mysql+pymysql://nova:$NOVA\_DBPASS@$HOST\_NAME/nova

# crudini --set /etc/nova/nova.conf api\_database connection mysql+pymysql://nova:$NOVA\_DBPASS@$HOST\_NAME/nova\_api

## 4.4为计算服务创建数据库表

# 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

## 4.5创建用户

# openstack user create --domain $DOMAIN\_NAME --password $NOVA\_PASS nova

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

## 4.6配置计算服务

# crudini --set /etc/nova/nova.conf DEFAULT enabled\_apis osapi\_compute,metadata

# crudini --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:$NOVA\_DBPASS@$HOST\_NAME

# crudini --set /etc/nova/nova.conf DEFAULT my\_ip $HOST\_IP

# crudini --set /etc/nova/nova.conf DEFAULT use\_neutron True

# crudini --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver

#

# crudini --set /etc/nova/nova.conf api auth\_strategy keystone

#

# crudini --set /etc/nova/nova.conf api\_database connection mysql+pymysql://nova:$NOVA\_DBPASS@$HOST\_NAME/nova\_api

#

# crudini --set /etc/nova/nova.conf database connection mysql+pymysql://nova:$NOVA\_DBPASS@$HOST\_NAME/nova

#

# crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000/v3

# crudini --set /etc/nova/nova.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_type password

# crudini --set /etc/nova/nova.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf keystone\_authtoken project\_name service

# crudini --set /etc/nova/nova.conf keystone\_authtoken username nova

# crudini --set /etc/nova/nova.conf keystone\_authtoken password $NOVA\_PASS

#

# crudini --set /etc/nova/nova.conf vnc enabled true

# crudini --set /etc/nova/nova.conf vnc server\_listen $HOST\_IP

# crudini --set /etc/nova/nova.conf vnc server\_proxyclient\_address $HOST\_IP

#

# crudini --set /etc/nova/nova.conf glance api\_servers http://$HOST\_NAME:9292

#

# crudini --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp

#

# crudini --set /etc/nova/nova.conf placement os\_region\_name RegionOne

# crudini --set /etc/nova/nova.conf placement project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf placement project\_name service

# crudini --set /etc/nova/nova.conf placement auth\_type password

# crudini --set /etc/nova/nova.conf placement user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf placement auth\_url http://$HOST\_NAME:5000/v3

# crudini --set /etc/nova/nova.conf placement username placement

# crudini --set /etc/nova/nova.conf placement password $NOVA\_PASS

## 4.7创建Endpoint和API端点

# openstack service create --name nova --description "OpenStack Compute" compute

# openstack endpoint create --region RegionOne compute public http://$HOST\_NAME:8774/v2.1

# openstack endpoint create --region RegionOne compute internal http://$HOST\_NAME:8774/v2.1

# openstack endpoint create --region RegionOne compute admin http://$HOST\_NAME:8774/v2.1

# openstack user create --domain $DOMAIN\_NAME --password $NOVA\_PASS 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://$HOST\_NAME:8778

# openstack endpoint create --region RegionOne placement internal http://$HOST\_NAME:8778

# openstack endpoint create --region RegionOne placement admin http://$HOST\_NAME:8778

## 4.8 添加配置

在/etc/httpd/conf.d/00-nova-placement-api.conf文件中添加如下配置

<Directory /usr/bin>

<IfVersion >= 2.4>

Require all granted

</IfVersion>

<IfVersion < 2.4>

Order allow,deny

Allow from all

</IfVersion>

</Directory>

## 4.9启动服务

# systemctl enable openstack-nova-api.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service

# systemctl start openstack-nova-api.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service

# systemctl restart httpd memcached

## 4.10验证Nova数据库是否创建成功

# nova-manage cell\_v2 list\_cells

#**Compute**

## 4.11安装Nova计算服务软件包

# yum install openstack-nova-compute -y

## 4.12配置Nova服务

# crudini --set /etc/nova/nova.conf DEFAULT enabled\_apis osapi\_compute,metadata

# crudini --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:$NOVA\_DBPASS@$HOST\_NAME

# crudini --set /etc/nova/nova.conf DEFAULT my\_ip $HOST\_IP\_NODE

# crudini --set /etc/nova/nova.conf DEFAULT use\_neutron True

# crudini --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver

# crudini --set /etc/nova/nova.conf api auth\_strategy keystone

# crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000/v3

# crudini --set /etc/nova/nova.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/nova/nova.conf keystone\_authtoken auth\_type password

# crudini --set /etc/nova/nova.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf keystone\_authtoken project\_name service

# crudini --set /etc/nova/nova.conf keystone\_authtoken username nova

# crudini --set /etc/nova/nova.conf keystone\_authtoken password $NOVA\_PASS

# crudini --set /etc/nova/nova.conf vnc enabled True

# crudini --set /etc/nova/nova.conf vnc server\_listen 0.0.0.0

# crudini --set /etc/nova/nova.conf vnc server\_proxyclient\_address $HOST\_IP\_NODE

# crudini --set /etc/nova/nova.conf vnc novncproxy\_base\_url http://$HOST\_IP:6080/vnc\_auto.html

# crudini --set /etc/nova/nova.conf glance api\_servers http://$HOST\_NAME:9292

# crudini --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp

# crudini --set /etc/nova/nova.conf placement os\_region\_name RegionOne

# crudini --set /etc/nova/nova.conf placement project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf placement project\_name service

# crudini --set /etc/nova/nova.conf placement auth\_type password

# crudini --set /etc/nova/nova.conf placement user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf placement auth\_url http://$HOST\_NAME:5000/v3

# crudini --set /etc/nova/nova.conf placement username placement

# crudini --set /etc/nova/nova.conf placement password $NOVA\_PASS

## 4.13检查系统处理器是否支持虚拟机的硬件加速

执行命令

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

（1）如果该命令返回一个1或更大的值,说明你的系统支持硬件加速，通常不需要额外的配置。

（2）如果这个指令返回一个0值，说明你的系统不支持硬件加速，你必须配置libvirt取代KVM来使用QEMU。

# crudini --set /etc/nova/nova.conf libvirt virt\_type qemu

## 4.14启动

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

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

## 4.15 添加计算节点

**#controller**

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

# 5 安装Neutron网络服务

**#Controller节点**

## 5.1通过脚本安装neutron服务

**5.2-5.11网络服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-neutron-controller.sh进行安装**

**#Compute节点**

**执行脚本iaas-install-neutron-compute.sh进行安装**

## 5.2创建Neutron数据库

#mysql -u root -p

mysql> CREATE DATABASE neutron;

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY '$NEUTRON\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY '$NEUTRON\_DBPASS';

## 5.3创建用户

# openstack user create --domain $DOMAIN\_NAME --password $NEUTRON\_PASS neutron

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

## 5.4创建Endpoint和API端点

# openstack service create --name neutron --description "OpenStack Networking" network

# openstack endpoint create --region RegionOne network public http://$HOST\_NAME:9696

# openstack endpoint create --region RegionOne network internal http://$HOST\_NAME:9696

# openstack endpoint create --region RegionOne network admin http://$HOST\_NAME:9696

## 5.5安装neutron网络服务软件包

# yum install openstack-neutron openstack-neutron-ml2 openstack-neutron-linuxbridge ebtables -y

## 5.6配置Neutron服务

# crudini --set /etc/neutron/neutron.conf DEFAULT core\_plugin ml2

# crudini --set /etc/neutron/neutron.conf DEFAULT service\_plugins router

# crudini --set /etc/neutron/neutron.conf DEFAULT allow\_overlapping\_ips true

# crudini --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:$NEUTRON\_DBPASS@$HOST\_NAME

# crudini --set /etc/neutron/neutron.conf DEFAULT auth\_strategy keystone

# crudini --set /etc/neutron/neutron.conf DEFAULT notify\_nova\_on\_port\_status\_changes true

# crudini --set /etc/neutron/neutron.conf DEFAULT notify\_nova\_on\_port\_data\_changes true

# crudini --set /etc/neutron/neutron.conf database connection mysql+pymysql://neutron:$NEUTRON\_DBPASS@$HOST\_NAME/neutron

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken username neutron

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken password $NEUTRON\_PASS

# crudini --set /etc/neutron/neutron.conf nova auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/neutron/neutron.conf nova auth\_type password

# crudini --set /etc/neutron/neutron.conf nova project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf nova user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf nova region\_name RegionOne

# crudini --set /etc/neutron/neutron.conf nova project\_name service

# crudini --set /etc/neutron/neutron.conf nova username nova

# crudini --set /etc/neutron/neutron.conf nova password $NOVA\_PASS

# crudini --set /etc/neutron/neutron.conf oslo\_concurrency lock\_path /var/lib/neutron/tmp

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 type\_drivers flat,vlan,vxlan

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 tenant\_network\_types vxlan

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers linuxbridge,l2population

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 extension\_drivers port\_security

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_flat flat\_networks $Physical\_NAME

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_vlan network\_vlan\_ranges $Physical\_NAME:$minvlan:$maxvlan

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_vxlan vni\_ranges $minvlan:$maxvlan

# crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup enable\_ipset true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings $Physical\_NAME:$INTERFACE\_NAME

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip $INTERFACE\_IP

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver

# crudini --set /etc/neutron/l3\_agent.ini DEFAULT interface\_driver linuxbridge

# crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT interface\_driver linuxbridge

# crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT dhcp\_driver neutron.agent.linux.dhcp.Dnsmasq

# crudini --set /etc/neutron/dhcp\_agent.ini DEFAULT enable\_isolated\_metadata true

# #/etc/neutron/metadata\_agent.ini

# crudini --set /etc/neutron/metadata\_agent.ini DEFAULT nova\_metadata\_host $HOST\_NAME

# crudini --set /etc/neutron/metadata\_agent.ini DEFAULT metadata\_proxy\_shared\_secret $METADATA\_SECRET

# crudini --set /etc/nova/nova.conf neutron url http://$HOST\_NAME:9696

# crudini --set /etc/nova/nova.conf neutron auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/nova/nova.conf neutron auth\_type password

# crudini --set /etc/nova/nova.conf neutron project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf neutron user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf neutron region\_name RegionOne

# crudini --set /etc/nova/nova.conf neutron project\_name service

# crudini --set /etc/nova/nova.conf neutron username neutron

# crudini --set /etc/nova/nova.conf neutron password $NEUTRON\_PASS

# crudini --set /etc/nova/nova.conf neutron service\_metadata\_proxy true

# crudini --set /etc/nova/nova.conf neutron metadata\_proxy\_shared\_secret $METADATA\_SECRET

## 5.7 创建数据库

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

# 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

## 5.8 启动服务和创建网桥

systemctl restart openstack-nova-api.service

systemctl enable neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service

systemctl restart neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service

**#Compute节点**

## 5.9 安装软件包

# yum install openstack-neutron-linuxbridge ebtables ipset net-tools -y

## 5.10 配置Neutron服务

# crudini --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:$NEUTRON\_DBPASS@$HOST\_NAME

# crudini --set /etc/neutron/neutron.conf DEFAULT auth\_strategy keystone

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken username neutron

# crudini --set /etc/neutron/neutron.conf keystone\_authtoken password $NEUTRON\_PASS

# crudini --set /etc/neutron/neutron.conf oslo\_concurrency lock\_path /var/lib/neutron/tmp

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings provider:$INTERFACE\_NAME

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip $INTERFACE\_IP

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true

# crudini --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver

# crudini --set /etc/nova/nova.conf neutron url http://$HOST\_NAME:9696

# crudini --set /etc/nova/nova.conf neutron auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/nova/nova.conf neutron auth\_type password

# crudini --set /etc/nova/nova.conf neutron project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf neutron user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/nova/nova.conf neutron region\_name RegionOne

# crudini --set /etc/nova/nova.conf neutron project\_name service

# crudini --set /etc/nova/nova.conf neutron username neutron

# crudini --set /etc/nova/nova.conf neutron password $NEUTRON\_PASS

## 5.11 启动服务进而创建网桥

# systemctl restart openstack-nova-compute.service

# systemctl start neutron-linuxbridge-agent.service

# systemctl enable neutron-linuxbridge-agent.service

# 6 安装Dashboard服务

## 6.1通过脚本安装dashboard服务

**6.2-6.4dashboard的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller**

**执行脚本iaas-install-dashboard.sh进行安装**

## 6.2安装Dashboard服务软件包

# yum install openstack-dashboard –y

## 6.3配置

修改/etc/openstack-dashboard/local\_settings内容如下

修改

|  |
| --- |
| import os  from django.utils.translation import ugettext\_lazy as \_  from openstack\_dashboard.settings import HORIZON\_CONFIG  DEBUG = False  WEBROOT = '/dashboard/'  ALLOWED\_HOSTS = ['\*', 'two.example.com']  OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True  OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "Default"  LOCAL\_PATH = '/tmp'  SECRET\_KEY='31880d3983dd796f54c8'  CACHES = {  'default': {  'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',  },  }  EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'  OPENSTACK\_HOST = "controller"  OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST  OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"  OPENSTACK\_KEYSTONE\_BACKEND = {  'name': 'native',  'can\_edit\_user': True,  'can\_edit\_group': True,  'can\_edit\_project': True,  'can\_edit\_domain': True,  'can\_edit\_role': True,  }  OPENSTACK\_HYPERVISOR\_FEATURES = {  'can\_set\_mount\_point': False,  'can\_set\_password': False,  'requires\_keypair': False,  'enable\_quotas': True  }  OPENSTACK\_CINDER\_FEATURES = {  'enable\_backup': False,  }  OPENSTACK\_NEUTRON\_NETWORK = {  'enable\_router': True,  'enable\_quotas': True,  'enable\_ipv6': True,  'enable\_distributed\_router': False,  'enable\_ha\_router': False,  'enable\_fip\_topology\_check': True,  'supported\_vnic\_types': ['\*'],  'physical\_networks': [],  }  OPENSTACK\_HEAT\_STACK = {  'enable\_user\_pass': True,  }  IMAGE\_CUSTOM\_PROPERTY\_TITLES = {  "architecture": \_("Architecture"),  "kernel\_id": \_("Kernel ID"),  "ramdisk\_id": \_("Ramdisk ID"),  "image\_state": \_("Euca2ools state"),  "project\_id": \_("Project ID"),  "image\_type": \_("Image Type"),  }  IMAGE\_RESERVED\_CUSTOM\_PROPERTIES = []  API\_RESULT\_LIMIT = 1000  API\_RESULT\_PAGE\_SIZE = 20  SWIFT\_FILE\_TRANSFER\_CHUNK\_SIZE = 512 \* 1024  INSTANCE\_LOG\_LENGTH = 35  DROPDOWN\_MAX\_ITEMS = 30  TIME\_ZONE = "UTC"  POLICY\_FILES\_PATH = '/etc/openstack-dashboard'  LOGGING = {  'version': 1,  'disable\_existing\_loggers': False,  'formatters': {  'console': {  'format': '%(levelname)s %(name)s %(message)s'  },  'operation': {  'format': '%(message)s'  },  },  'handlers': {  'null': {  'level': 'DEBUG',  'class': 'logging.NullHandler',  },  'console': {  'level': 'INFO',  'class': 'logging.StreamHandler',  'formatter': 'console',  },  'operation': {  'level': 'INFO',  'class': 'logging.StreamHandler',  'formatter': 'operation',  },  },  'loggers': {  'horizon': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'horizon.operation\_log': {  'handlers': ['operation'],  'level': 'INFO',  'propagate': False,  },  'openstack\_dashboard': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'novaclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'cinderclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'keystoneauth': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'keystoneclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'glanceclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'neutronclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'swiftclient': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'oslo\_policy': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'openstack\_auth': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'nose.plugins.manager': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'django': {  'handlers': ['console'],  'level': 'DEBUG',  'propagate': False,  },  'django.db.backends': {  'handlers': ['null'],  'propagate': False,  },  'requests': {  'handlers': ['null'],  'propagate': False,  },  'urllib3': {  'handlers': ['null'],  'propagate': False,  },  'chardet.charsetprober': {  'handlers': ['null'],  'propagate': False,  },  'iso8601': {  'handlers': ['null'],  'propagate': False,  },  'scss': {  'handlers': ['null'],  'propagate': False,  },  },  }  SECURITY\_GROUP\_RULES = {  'all\_tcp': {  'name': \_('All TCP'),  'ip\_protocol': 'tcp',  'from\_port': '1',  'to\_port': '65535',  },  'all\_udp': {  'name': \_('All UDP'),  'ip\_protocol': 'udp',  'from\_port': '1',  'to\_port': '65535',  },  'all\_icmp': {  'name': \_('All ICMP'),  'ip\_protocol': 'icmp',  'from\_port': '-1',  'to\_port': '-1',  },  'ssh': {  'name': 'SSH',  'ip\_protocol': 'tcp',  'from\_port': '22',  'to\_port': '22',  },  'smtp': {  'name': 'SMTP',  'ip\_protocol': 'tcp',  'from\_port': '25',  'to\_port': '25',  },  'dns': {  'name': 'DNS',  'ip\_protocol': 'tcp',  'from\_port': '53',  'to\_port': '53',  },  'http': {  'name': 'HTTP',  'ip\_protocol': 'tcp',  'from\_port': '80',  'to\_port': '80',  },  'pop3': {  'name': 'POP3',  'ip\_protocol': 'tcp',  'from\_port': '110',  'to\_port': '110',  },  'imap': {  'name': 'IMAP',  'ip\_protocol': 'tcp',  'from\_port': '143',  'to\_port': '143',  },  'ldap': {  'name': 'LDAP',  'ip\_protocol': 'tcp',  'from\_port': '389',  'to\_port': '389',  },  'https': {  'name': 'HTTPS',  'ip\_protocol': 'tcp',  'from\_port': '443',  'to\_port': '443',  },  'smtps': {  'name': 'SMTPS',  'ip\_protocol': 'tcp',  'from\_port': '465',  'to\_port': '465',  },  'imaps': {  'name': 'IMAPS',  'ip\_protocol': 'tcp',  'from\_port': '993',  'to\_port': '993',  },  'pop3s': {  'name': 'POP3S',  'ip\_protocol': 'tcp',  'from\_port': '995',  'to\_port': '995',  },  'ms\_sql': {  'name': 'MS SQL',  'ip\_protocol': 'tcp',  'from\_port': '1433',  'to\_port': '1433',  },  'mysql': {  'name': 'MYSQL',  'ip\_protocol': 'tcp',  'from\_port': '3306',  'to\_port': '3306',  },  'rdp': {  'name': 'RDP',  'ip\_protocol': 'tcp',  'from\_port': '3389',  'to\_port': '3389',  },  }  REST\_API\_REQUIRED\_SETTINGS = ['OPENSTACK\_HYPERVISOR\_FEATURES',  'LAUNCH\_INSTANCE\_DEFAULTS',  'OPENSTACK\_IMAGE\_FORMATS',  'OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN',  'CREATE\_IMAGE\_DEFAULTS',  'ENFORCE\_PASSWORD\_CHECK']  ALLOWED\_PRIVATE\_SUBNET\_CIDR = {'ipv4': [], 'ipv6': []}  SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'  CACHES = {  'default': {  'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',  'LOCATION': 'controller:11211',  }  }  OPENSTACK\_API\_VERSIONS = {  "identity": 3,  "image": 2,  "volume": 2,  } |

## 6.4启动服务

# systemctl restart httpd.service memcached.service

## 6.5访问

打开浏览器访问Dashboard

[http://controller](http://controller/)（或本机内网ip）/dashboard

**注：检查防火墙规则，确保允许http服务相关端口通行，或者关闭防火墙。**

## 6.6创建云主机

（1）管理员->资源管理->云主机类型->创建云主机类型

（2）管理员->网络->网络->创建网络

1. 项目->网络->安全组->管理规则->添加规则（ICMP、TCP、UDP）

（4）项目->资源管理->云主机->创建云主机

# 7 安装Cinder块存储服务

## 7.1 通过脚本安装Cinder服务

**7.2-7.12块存储服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller**

**执行脚本iaas-install-cinder-controller.sh进行安装**

**#Compute节点**

**执行脚本****iaas-install-cinder-compute.sh进行安装**

## 7.2 安装Cinder块存储服务软件包

# yum install openstack-cinder

## 7.3 创建数据库

# mysql -u root -p

mysql> CREATE DATABASE cinder;

mysql> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY '$CINDER\_DBPASS';

mysql> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY '$CINDER\_DBPASS';

## 7.4 创建用户

# openstack user create --domain $DOMAIN\_NAME --password $CINDER\_PASS cinder

# openstack role add --project service --user cinder admin

## 7.5 创建Endpoint和API端点

# openstack service create --name cinder --description "OpenStack Block Store" volume

# openstack service create --name cinderv2 --description "OpenStack Block Store" volumev2

# openstack service create --name cinderv3 --description "OpenStack Block Store" volumev3

# openstack endpoint create --region RegionOne volume public http://$HOST\_NAME:8776/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volume internal http://$HOST\_NAME:8776/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volume admin http://$HOST\_NAME:8776/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volumev2 public http://$HOST\_NAME:8776/v2/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volumev2 internal http://$HOST\_NAME:8776/v2/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volumev2 admin http://$HOST\_NAME:8776/v2/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volumev3 public http://$HOST\_NAME:8776/v3/%\(tenant\_id\)s

#openstack endpoint create --region RegionOne volumev3 internal http://$HOST\_NAME:8776/v3/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne volumev3 admin http://$HOST\_NAME:8776/v3/%\(tenant\_id\)s

## 7.6 配置Cinder服务

# crudini --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder:$CINDER\_DBPASS@$HOST\_NAME/cinder

# crudini --set /etc/cinder/cinder.conf DEFAULT rpc\_backend rabbit

# crudini --set /etc/cinder/cinder.conf oslo\_messaging\_rabbit rabbit\_host $HOST\_NAME

# crudini --set /etc/cinder/cinder.conf oslo\_messaging\_rabbit rabbit\_userid $RABBIT\_USER

# crudini --set /etc/cinder/cinder.conf oslo\_messaging\_rabbit rabbit\_password $RABBIT\_PASS

# crudini --set /etc/cinder/cinder.conf DEFAULT auth\_strategy keystone

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_type password

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_name service

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken username cinder

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken password $CINDER\_PASS

# crudini --set /etc/cinder/cinder.conf DEFAULT my\_ip $HOST\_IP

# crudini --set /etc/cinder/cinder.conf oslo\_concurrency lock\_path /var/lib/cinder/tmp

# crudini --set /etc/nova/nova.conf cinder os\_region\_name RegionOne

## 7.7 创建数据库

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

## 7.8 启动服务

# systemctl restart openstack-nova-api.service

# systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service

# systemctl restart openstack-cinder-api.service openstack-cinder-scheduler.service

## 7.9 安装块存储软件

**#compute**

# yum install lvm2 device-mapper-persistent-data openstack-cinder targetcli python-keystone -y

# systemctl enable lvm2-lvmetad.service

# systemctl restart lvm2-lvmetad.service

## 7.10 创建LVM物理和逻辑卷

以磁盘**/dev/sda**为例

# pvcreate –f **/dev/sda**

# vgcreate cinder-volumes **/dev/sda**

## 7.11 修改Cinder配置文件

# crudini --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder:$CINDER\_DBPASS@$HOST\_NAME/cinder

# crudini --set /etc/cinder/cinder.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/cinder/cinder.conf DEFAULT auth\_strategy keystone

# crudini --set /etc/cinder/cinder.conf DEFAULT enabled\_backends lvm

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken auth\_type password

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken project\_name service

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken username cinder

# crudini --set /etc/cinder/cinder.conf keystone\_authtoken password $CINDER\_PASS

# crudini --set /etc/cinder/cinder.conf DEFAULT my\_ip $HOST\_IP\_NODE

# crudini --set /etc/cinder/cinder.conf lvm volume\_driver cinder.volume.drivers.lvm.LVMVolumeDriver

# crudini --set /etc/cinder/cinder.conf lvm volume\_group cinder-volumes

# crudini --set /etc/cinder/cinder.conf lvm iscsi\_protocol iscsi

# crudini --set /etc/cinder/cinder.conf lvm iscsi\_helper lioadm

# crudini --set /etc/cinder/cinder.conf DEFAULT glance\_api\_servers http://$HOST\_NAME:9292

# crudini --set /etc/cinder/cinder.conf oslo\_concurrency lock\_path /var/lib/cinder/tmp

## 7.12 重启服务

# systemctl enable openstack-cinder-volume.service target.service

# systemctl restart openstack-cinder-volume.service target.service

## 7.13 验证

**#Controller**

使用cinder create 创建一个新的卷

# cinder create --display-name myVolume 1

通过cinder list 命令查看是否正确创建

# cinder list

# 8 安装Swift对象存储服务

## 8.1通过脚本安装Swift服务

**8.2-8.12对象存储服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller**

**执行脚本iaas-install-swift-controller.sh进行安装**

**#Compute节点**

**执行脚本iaas-install-swift-compute.sh进行安装**

## 8.2 安装Swift对象存储服务软件包

# yum install openstack-swift-proxy python-swiftclient python-keystoneclient python-keystonemiddleware memcached -y

## 8.2创建用户

# openstack user create --domain $DOMAIN\_NAME --password $SWIFT\_PASS swift

# openstack role add --project service --user swift admin

## 8.3创建Endpoint和API端点

# openstack service create --name swift --description "OpenStack Object Storage" object-store

# openstack endpoint create --region RegionOne object-store public http://$HOST\_NAME:8080/v1/AUTH\_%\(tenant\_id\)s

# openstack endpoint create --region RegionOne object-store internal http://$HOST\_NAME:8080/v1/AUTH\_%\(tenant\_id\)s

# openstack endpoint create --region RegionOne object-store admin http://$HOST\_NAME:8080/v1

## 8.4 编辑/etc/swift/proxy-server.conf

编辑配置文件如下

[DEFAULT]

bind\_port = 8080

swift\_dir = /etc/swift

user = swift

[pipeline:main]

pipeline = catch\_errors gatekeeper healthcheck proxy-logging cache container\_sync bulk ratelimit authtoken keystoneauth container-quotas account-quotas slo dlo versioned\_writes proxy-logging proxy-server

[app:proxy-server]

use = egg:swift#proxy

account\_autocreate = True

[filter:tempauth]

use = egg:swift#tempauth

user\_admin\_admin = admin .admin .reseller\_admin

user\_test\_tester = testing .admin

user\_test2\_tester2 = testing2 .admin

user\_test\_tester3 = testing3

user\_test5\_tester5 = testing5 service

[filter:authtoken]

paste.filter\_factory = keystonemiddleware.auth\_token:filter\_factory

auth\_uri = http://$HOST\_NAME:5000

auth\_url = http://$HOST\_NAME:35357

memcached\_servers = $HOST\_NAME:11211

auth\_type = password

project\_domain\_name = $DOMAIN\_NAME

user\_domain\_name = $DOMAIN\_NAME

project\_name = service

username = swift

password = $SWIFT\_PASS

delay\_auth\_decision = True

[filter:keystoneauth]

use = egg:swift#keystoneauth

operator\_roles = admin,user

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:cache]

memcache\_servers = $HOST\_NAME:11211

use = egg:swift#memcache

[filter:ratelimit]

use = egg:swift#ratelimit

[filter:domain\_remap]

use = egg:swift#domain\_remap

[filter:catch\_errors]

use = egg:swift#catch\_errors

[filter:cname\_lookup]

use = egg:swift#cname\_lookup

[filter:staticweb]

use = egg:swift#staticweb

[filter:tempurl]

use = egg:swift#tempurl

[filter:formpost]

use = egg:swift#formpost

[filter:name\_check]

use = egg:swift#name\_check

[filter:list-endpoints]

use = egg:swift#list\_endpoints

[filter:proxy-logging]

use = egg:swift#proxy\_logging

[filter:bulk]

use = egg:swift#bulk

[filter:slo]

use = egg:swift#slo

[filter:dlo]

use = egg:swift#dlo

[filter:container-quotas]

use = egg:swift#container\_quotas

[filter:account-quotas]

use = egg:swift#account\_quotas

[filter:gatekeeper]

use = egg:swift#gatekeeper

[filter:container\_sync]

use = egg:swift#container\_sync

[filter:xprofile]

use = egg:swift#xprofile

[filter:versioned\_writes]

use = egg:swift#versioned\_writes

## 8.5 创建账号、容器、对象

存储节点存储磁盘名称以sdb为例

swift-ring-builder account.builder create 18 1 1

swift-ring-builder account.builder add --region 1 --zone 1 --ip $STORAGE\_LOCAL\_NET\_IP --port 6002 --device $OBJECT\_DISK --weight 100

swift-ring-builder account.builder

swift-ring-builder account.builder rebalance

swift-ring-builder container.builder create 10 1 1

swift-ring-builder container.builder add --region 1 --zone 1 --ip $STORAGE\_LOCAL\_NET\_IP --port 6001 --device $OBJECT\_DISK --weight 100

swift-ring-builder container.builder

swift-ring-builder container.builder rebalance

swift-ring-builder object.builder create 10 1 1

swift-ring-builder object.builder add --region 1 --zone 1 --ip $STORAGE\_LOCAL\_NET\_IP --port 6000 --device $OBJECT\_DISK --weight 100

swift-ring-builder object.builder

swift-ring-builder object.builder rebalance

## 8.6 编辑/etc/swift/swift.conf文件

编辑如下

[swift-hash]

swift\_hash\_path\_suffix = changeme

swift\_hash\_path\_prefix = changeme

[storage-policy:0]

name = Policy-0

default = yes

aliases = yellow, orange

[swift-constraints]

## 8.7 启动服务和赋予权限

chown -R root:swift /etc/swift

systemctl enable openstack-swift-proxy.service memcached.service

systemctl restart openstack-swift-proxy.service memcached.service

## 8.8 安装软件包

**#Compute节点**

存储节点存储磁盘名称以sdb为例

# yum install xfsprogs rsync openstack-swift-account openstack-swift-container openstack-swift-object -y

# mkfs.xfs -i size=1024 -f /dev/sdb

# echo "/dev/sdb /swift/node xfs loop,noatime,nodiratime,nobarrier,logbufs=8 0 0" >> /etc/fstab

# mkdir -p /swift/node/sdb

# mount /dev/sdb /swift/node/sdb

# scp controller:/etc/swift/\*.ring.gz /etc/swift/

## 8.9 配置rsync

**（1）编辑/etc/rsyncd.conf文件如下**

pid file = /var/run/rsyncd.pid

log file = /var/log/rsyncd.log

uid = swift

gid = swift

address = 127.0.0.1

[account]

path = /swift/node

read only = false

write only = no

list = yes

incoming chmod = 0644

outgoing chmod = 0644

max connections = 25

lock file = /var/lock/account.lock

[container]

path = /swift/node

read only = false

write only = no

list = yes

incoming chmod = 0644

outgoing chmod = 0644

max connections = 25

lock file = /var/lock/container.lock

[object]

path = /swift/node

read only = false

write only = no

list = yes

incoming chmod = 0644

outgoing chmod = 0644

max connections = 25

lock file = /var/lock/object.lock

[swift\_server]

path = /etc/swift

read only = true

write only = no

list = yes

incoming chmod = 0644

outgoing chmod = 0644

max connections = 5

lock file = /var/lock/swift\_server.lock

**（2）启动服务**

# systemctl enable rsyncd.service

#systemctl restart rsyncd.service

## 8.10 配置账号、容器和对象

**（1）修改/etc/swift/account-server.conf配置文件**

[DEFAULT]

bind\_port = 6002

user = swift

swift\_dir = /etc/swift

devices = /swift/node

mount\_check = false

[pipeline:main]

pipeline = healthcheck recon account-server

[app:account-server]

use = egg:swift#account

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

[account-replicator]

[account-auditor]

[account-reaper]

[filter:xprofile]

use = egg:swift#xprofile

**（2）修改/etc/swift/container-server.conf配置文件**

[DEFAULT]

bind\_port = 6001

user = swift

swift\_dir = /etc/swift

devices = /swift/node

mount\_check = false

[pipeline:main]

pipeline = healthcheck recon container-server

[app:container-server]

use = egg:swift#container

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

[container-replicator]

[container-updater]

[container-auditor]

[container-sync]

[filter:xprofile]

use = egg:swift#xprofile

**（3）修改/etc/swift/object-server.conf配置文件**

[DEFAULT]

bind\_port = 6000

user = swift

swift\_dir = /etc/swift

devices = /swift/node

mount\_check = false

[pipeline:main]

pipeline = healthcheck recon object-server

[app:object-server]

use = egg:swift#object

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

recon\_lock\_path = /var/lock

[object-replicator]

[object-reconstructor]

[object-updater]

[object-auditor]

[filter:xprofile]

use = egg:swift#xprofile

## 8.11 修改Swift配置文件

修改/etc/swift/swift.conf

[swift-hash]

swift\_hash\_path\_suffix = changeme

swift\_hash\_path\_prefix = changeme

[storage-policy:0]

name = Policy-0

default = yes

aliases = yellow, orange

[swift-constraints]

## 8.12 重启服务和赋予权限

# chown -R swift:swift /swift/node

# mkdir -p /var/cache/swift

# chown -R root:swift /var/cache/swift

# chmod -R 775 /var/cache/swift

# chown -R root:swift /etc/swift

# systemctl enable openstack-swift-account.service openstack-swift-account-auditor.service openstack-swift-account-reaper.service openstack-swift-account-replicator.service

# systemctl restart openstack-swift-account.service openstack-swift-account-auditor.service openstack-swift-account-reaper.service openstack-swift-account-replicator.service

# systemctl enable openstack-swift-container.service openstack-swift-container-auditor.service openstack-swift-container-replicator.service openstack-swift-container-updater.service

# systemctl restart openstack-swift-container.service openstack-swift-container-auditor.service openstack-swift-container-replicator.service openstack-swift-container-updater.service

# systemctl enable openstack-swift-object.service openstack-swift-object-auditor.service openstack-swift-object-replicator.service openstack-swift-object-updater.service

# systemctl restart openstack-swift-object.service openstack-swift-object-auditor.service openstack-swift-object-replicator.service openstack-swift-object-updater.service

# 9 安装Heat编配服务

**# Controller节点**

## 9.1通过脚本安装heat服务

**9.2-9.8编配服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-heat.sh进行安装**

## 9.2安装heat编配服务软件包

# yum install openstack-heat-api openstack-heat-api-cfn openstack-heat-engine openstack-heat-ui -y

## 9.3创建数据库

# mysql -u root -p

mysql> CREATE DATABASE heat;

mysql> GRANT ALL PRIVILEGES ON heat.\* TO 'heat'@'localhost' IDENTIFIED BY '$HEAT\_DBPASS'; mysql> GRANT ALL PRIVILEGES ON heat.\* TO 'heat'@'%' IDENTIFIED BY '$HEAT\_DBPASS';

## 9.4创建用户

# openstack user create --domain $DOMAIN\_NAME --password $HEAT\_PASS heat

# openstack role add --project service --user heat admin

# openstack domain create --description "Stack projects and users" heat

# openstack user create --domain heat --password $HEAT\_PASS heat\_domain\_admin

# openstack role add --domain heat --user-domain heat --user heat\_domain\_admin admin

# openstack role create heat\_stack\_owner

# openstack role add --project demo --user demo heat\_stack\_owner

# openstack role create heat\_stack\_user

## 9.5创建Endpoint和API端点

# openstack service create --name heat --description "Orchestration" orchestration

# openstack service create --name heat-cfn --description "Orchestration" cloudformation

# openstack endpoint create --region RegionOne orchestration public http://$HOST\_NAME:8004/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne orchestration internal http://$HOST\_NAME:8004/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne orchestration admin http://$HOST\_NAME:8004/v1/%\(tenant\_id\)s

# openstack endpoint create --region RegionOne cloudformation public http://$HOST\_NAME:8000/v1

# openstack endpoint create --region RegionOne cloudformation internal http://$HOST\_NAME:8000/v1

# openstack endpoint create --region RegionOne cloudformation admin http://$HOST\_NAME:8000/v1

## 9.6配置Heat服务

# crudini --set /etc/heat/heat.conf database connection mysql+pymysql://heat:$HEAT\_DBPASS@$HOST\_NAME/heat

# crudini --set /etc/heat/heat.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/heat/heat.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/heat/heat.conf keystone\_authtoken auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/heat/heat.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/heat/heat.conf keystone\_authtoken auth\_type password

# crudini --set /etc/heat/heat.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/heat/heat.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/heat/heat.conf keystone\_authtoken project\_name service

# crudini --set /etc/heat/heat.conf keystone\_authtoken username heat

# crudini --set /etc/heat/heat.conf keystone\_authtoken password $HEAT\_PASS

# crudini --set /etc/heat/heat.conf trustee auth\_plugin password

# crudini --set /etc/heat/heat.conf trustee auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/heat/heat.conf trustee username heat

# crudini --set /etc/heat/heat.conf trustee password $HEAT\_PASS

# crudini --set /etc/heat/heat.conf trustee user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/heat/heat.conf clients\_keystone auth\_uri http://$HOST\_NAME:35357

# crudini --set /etc/heat/heat.conf DEFAULT heat\_metadata\_server\_url http://$HOST\_NAME:8000

# crudini --set /etc/heat/heat.conf DEFAULT heat\_waitcondition\_server\_url http://$HOST\_NAME:8000/v1/waitcondition

# crudini --set /etc/heat/heat.conf DEFAULT stack\_domain\_admin heat\_domain\_admin

# crudini --set /etc/heat/heat.conf DEFAULT stack\_domain\_admin\_password $HEAT\_PASS

# crudini --set /etc/heat/heat.conf DEFAULT stack\_user\_domain\_name heat

## 9.7创建数据库

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

## 9.8启动服务

# systemctl enable openstack-heat-api.service openstack-heat-api-cfn.service openstack-heat-engine.service

# systemctl restart openstack-heat-api.service openstack-heat-api-cfn.service openstack-heat-engine.service

# 10 安装Zun服务

## 10.1通过脚本安装Zun服务

10**.2-**10**.12zun服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-zun-controller.sh进行安装**

**#Compute节点**

**执行脚本iaas-install-zun-compute.sh进行安装**

## 10.2 安装zun服务软件包

**#Controller节点**

# yum install python-pip git openstack-zun openstack-zun-ui –y

## 10.3 创建数据库

# mysql -u root -p

mysql> CREATE DATABASE zun;

mysql> GRANT ALL PRIVILEGES ON zun.\* TO zun@'localhost' IDENTIFIED BY **'$ZUN\_DBPASS'**; mysql> GRANT ALL PRIVILEGES ON zun.\* TO zun@'%' IDENTIFIED BY **'$ZUN\_DBPASS'**;

## 10.4 创建用户

# openstack user create --domain $DOMAIN\_NAME --password $ZUN\_PASS zun

# openstack role add --project service --user zun admin

# openstack user create --domain $DOMAIN\_NAME --password $KURYR\_PASS kuryr

# openstack role add --project service --user kuryr admin

## 10.5 创建Endpoint和API端点

# openstack service create --name zun --description "Container Service" container

# openstack endpoint create --region RegionOne container public http://$HOST\_NAME:9517/v1

# openstack endpoint create --region RegionOne container internal http://$HOST\_NAME:9517/v1

# openstack endpoint create --region RegionOne container admin http://$HOST\_NAME:9517/v1

## 10.6 配置zun服务

# crudini --set /etc/zun/zun.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/zun/zun.conf DEFAULT log\_file /var/log/zun

# crudini --set /etc/zun/zun.conf api host\_ip $HOST\_IP

# crudini --set /etc/zun/zun.conf api port 9517

# crudini --set /etc/zun/zun.conf database connection mysql+pymysql://zun:$ZUN\_DBPASS@$HOST\_NAME/zun

# crudini --set /etc/zun/zun.conf keystone\_auth memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_auth project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_auth project\_name service

# crudini --set /etc/zun/zun.conf keystone\_auth user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_auth password $ZUN\_PASS

# crudini --set /etc/zun/zun.conf keystone\_auth username zun

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_type password

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_version v3

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_protocol http

# crudini --set /etc/zun/zun.conf keystone\_auth service\_token\_roles\_required True

# crudini --set /etc/zun/zun.conf keystone\_auth endpoint\_type internalURL

# crudini --set /etc/zun/zun.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_authtoken project\_name service

# crudini --set /etc/zun/zun.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_authtoken password $ZUN\_PASS

# crudini --set /etc/zun/zun.conf keystone\_authtoken username zun

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_type password

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_version v3

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_protocol http

# crudini --set /etc/zun/zun.conf keystone\_authtoken service\_token\_roles\_required True

# crudini --set /etc/zun/zun.conf keystone\_authtoken endpoint\_type internalURL

# crudini --set /etc/zun/zun.conf oslo\_concurrency lock\_path /var/lib/zun/tmp

# crudini --set /etc/zun/zun.conf oslo\_messaging\_notifications driver messaging

# crudini --set /etc/zun/zun.conf websocket\_proxy wsproxy\_host $HOST\_IP

# crudini --set /etc/zun/zun.conf websocket\_proxy wsproxy\_port 6784

## 10.7 创建数据库

# su -s /bin/sh -c "zun-db-manage upgrade" zun

## 10.8 启动服务

# systemctl enable zun-api zun-wsproxy

# systemctl restart zun-api zun-wsproxy

# systemctl restart httpd memcached

## 10.9 安装软件包

**#compute节点**

# yum install -y yum-utils device-mapper-persistent-data lvm2

# yum install docker-ce python-pip git kuryr-libnetwork openstack-zun-compute –y

## 10.10 配置服务

# crudini --set /etc/kuryr/kuryr.conf DEFAULT bindir /usr/libexec/kuryr

# crudini --set /etc/kuryr/kuryr.conf neutron auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/kuryr/kuryr.conf neutron auth\_url http://$HOST\_NAME:35357

# crudini --set /etc/kuryr/kuryr.conf neutron username kuryr

# crudini --set /etc/kuryr/kuryr.conf neutron user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/kuryr/kuryr.conf neutron password $KURYR\_PASS

# crudini --set /etc/kuryr/kuryr.conf neutron project\_name service

# crudini --set /etc/kuryr/kuryr.conf neutron project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/kuryr/kuryr.conf neutron auth\_type password

# crudini --set /etc/zun/zun.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/zun/zun.conf DEFAULT state\_path /var/lib/zun

# crudini --set /etc/zun/zun.conf DEFAULT log\_file /var/log/zun

# crudini --set /etc/zun/zun.conf database connection mysql+pymysql://zun:$ZUN\_DBPASS@$HOST\_NAME/zun

# crudini --set /etc/zun/zun.conf keystone\_auth memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_auth project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_auth project\_name service

# crudini --set /etc/zun/zun.conf keystone\_auth user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_auth password $ZUN\_PASS

# crudini --set /etc/zun/zun.conf keystone\_auth username zun

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_type password

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_version v3

# crudini --set /etc/zun/zun.conf keystone\_auth auth\_protocol http

# crudini --set /etc/zun/zun.conf keystone\_auth service\_token\_roles\_required True

# crudini --set /etc/zun/zun.conf keystone\_auth endpoint\_type internalURL

# crudini --set /etc/zun/zun.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_uri http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_authtoken project\_name service

# crudini --set /etc/zun/zun.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/zun/zun.conf keystone\_authtoken password $ZUN\_PASS

# crudini --set /etc/zun/zun.conf keystone\_authtoken username zun

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/zun/zun.conf keystone\_authtoken auth\_type password

# crudini --set /etc/zun/zun.conf websocket\_proxy base\_url ws://$HOST\_NAME:6784/

# crudini --set /etc/zun/zun.conf oslo\_concurrency lock\_path /var/lib/zun/tmp

# crudini --set /etc/kuryr/kuryr.conf DEFAULT capability\_scope global

## 10.11 修改内核参数

修改/etc/sysctl.conf文件，添加以下内容：

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

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

net.ipv4.ip\_forward = 1

生效配置

# sysctl –p

## 10.12 启动服务

# mkdir -p /etc/systemd/system/docker.service.d

修改mkdir -p /etc/systemd/system/docker.service.d文件，添加以下内容：

[Service]

ExecStart=

ExecStart=/usr/bin/dockerd --group zun -H tcp://$HOST\_NAME\_NODE:2375 -H unix:///var/run/docker.sock --cluster-store etcd://$HOST\_NAME:2379

# systemctl daemon-reload

# systemctl restart docker

# systemctl enable docker

# systemctl enable kuryr-libnetwork

# systemctl restart kuryr-libnetwork

# systemctl enable zun-compute

# systemctl restart zun-compute

## 10.13 上传镜像

以CentOS7\_1804.tar镜像为例，CentOS7\_1804.tar镜像包存放在XianDian-IaaS-v2.4.iso镜像包中。将docker镜像上传到glance中，通过openstack使用镜像启动容器。

# source /etc/keystone/admin-openrc.sh

# openstack image create centos7.5 --public --container-format docker --disk-format raw < CentOS7\_1804.tar

## 10.14 启动容器

通过glance存储镜像启动容器

# zun run --image-driver glance centos7.5

# zun list

+--------------------------------------+--------------------+-----------+---------+------------+--------------+-------+ | uuid | name | image | status | task\_state | addresses | ports | +--------------------------------------+--------------------+-----------+---------+------------+--------------+-------+ | c01d89b6-b927-4a5e-9889-356f572e184d | psi-9-container | centos7.5 | Running | None | 172.30.15.9 | [22] | | +--------------------------------------+--------------------+-----------+---------+------------+--------------+-------+

# 11 安装Ceilometer监控服务

## 11.1通过脚本安装Ceilometer服务

11**.2-**11**.14ceilometer监控服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-ceilometer-controller.sh进行安装**

**#Compute节点**

**执行脚本iaas-install-ceilometer-compute.sh进行安装**

## 11.2 安装Ceilometer监控服务软件包

**#Controller节点**

# yum install openstack-gnocchi-api openstack-gnocchi-metricd python2-gnocchiclient openstack-ceilometer-notification openstack-ceilometer-central python2-ceilometerclient python-ceilometermiddleware -y

## 11.3 创建数据库

# mysql -u root -p

mysql> CREATE DATABASE gnocchi;

mysql> GRANT ALL PRIVILEGES ON gnocchi.\* TO gnocchi@'localhost' IDENTIFIED BY **'$CEILOMETER\_DBPASS'**; mysql> GRANT ALL PRIVILEGES ON gnocchi.\* TO gnocchi@'%' IDENTIFIED BY **'$CEILOMETER\_DBPASS'**;

## 11.4 创建用户

# openstack user create --domain $DOMAIN\_NAME --password $CEILOMETER\_PASS ceilometer

# openstack role add --project service --user ceilometer admin

# openstack user create --domain $DOMAIN\_NAME --password $CEILOMETER\_PASS gnocchi

# openstack role add --project service --user gnocchi admin

# openstack role create ResellerAdmin

# openstack role add --project service --user ceilometer ResellerAdmin

## 11.5 创建Endpoint和API端点

# openstack service create --name ceilometer --description "OpenStack Telemetry Service" metering

# openstack service create --name gnocchi --description "Metric Service" metric

# openstack endpoint create --region RegionOne metric public http://$HOST\_NAME:8041

# openstack endpoint create --region RegionOne metric internal http://$HOST\_NAME:8041

# openstack endpoint create --region RegionOne metric admin http://$HOST\_NAME:8041

## 11.6 配置Ceilometer

# crudini --set /etc/gnocchi/gnocchi.conf DEFAULT log\_dir /var/log/gnocchi

# crudini --set /etc/gnocchi/gnocchi.conf api auth\_mode keystone

# crudini --set /etc/gnocchi/gnocchi.conf database backend sqlalchemy

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken auth\_type password

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken www\_authenticate\_uri http://$HOST\_NAME:5000

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken project\_name service

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken username gnocchi

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken password $CEILOMETER\_PASS

# crudini --set /etc/gnocchi/gnocchi.conf keystone\_authtoken service\_token\_roles\_required true

# crudini --set /etc/gnocchi/gnocchi.conf indexer url mysql+pymysql://gnocchi:$CEILOMETER\_DBPASS@$HOST\_NAME/gnocchi

# crudini --set /etc/gnocchi/gnocchi.conf storage file\_basepath /var/lib/gnocchi

# crudini --set /etc/gnocchi/gnocchi.conf storage driver file

# crudini --set /etc/ceilometer/ceilometer.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf api auth\_mode keystone

# crudini --set /etc/ceilometer/ceilometer.conf dispatcher\_gnocchi filter\_service\_activity False

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken www\_authenticate\_uri = http://$HOST\_NAME:5000

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken auth\_url = http://$HOST\_NAME:5000

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken memcached\_servers = $HOST\_NAME:11211

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken auth\_type = password

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken project\_domain\_name = $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken user\_domain\_name = $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken project\_name = service

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken username = gnocchi

# crudini --set /etc/ceilometer/ceilometer.conf keystone\_authtoken password = $CEILOMETER\_PASS

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials auth\_type password

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials project\_name service

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials username ceilometer

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials password $CEILOMETER\_PASS

## 11.7 创建监听端点

创建/etc/httpd/conf.d/10-gnocchi\_wsgi.conf文件，添加以下内容：

Listen 8041

<VirtualHost \*:8041>

DocumentRoot /var/www/cgi-bin/gnocchi

<Directory /var/www/cgi-bin/gnocchi>

AllowOverride None

Require all granted

</Directory>

CustomLog /var/log/httpd/gnocchi\_wsgi\_access.log combined

ErrorLog /var/log/httpd/gnocchi\_wsgi\_error.log

SetEnvIf X-Forwarded-Proto https HTTPS=1

WSGIApplicationGroup %{GLOBAL}

WSGIDaemonProcess gnocchi display-name=gnocchi\_wsgi user=gnocchi group=gnocchi processes=6 threads=6

WSGIProcessGroup gnocchi

WSGIScriptAlias / /var/www/cgi-bin/gnocchi/app

</VirtualHost>

## 11.8 创建数据库

# mkdir /var/www/cgi-bin/gnocchi

# cp /usr/lib/python2.7/site-packages/gnocchi/rest/gnocchi-api /var/www/cgi-bin/gnocchi/app

# chown -R gnocchi. /var/www/cgi-bin/gnocchi

# su -s /bin/bash gnocchi -c "gnocchi-upgrade"

# su -s /bin/bash ceilometer -c "ceilometer-upgrade --skip-metering-database"

## 11.9 启动服务

# systemctl enable openstack-gnocchi-api.service openstack-gnocchi-metricd.service

# systemctl restart openstack-gnocchi-api.service openstack-gnocchi-metricd.service

#systemctl restart httpd memcached

# systemctl enable openstack-ceilometer-notification.service openstack-ceilometer-central.service

# systemctl restart openstack-ceilometer-notification.service openstack-ceilometer-central.service

## 11.10 监控组件

# crudini --set /etc/glance/glance-api.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/glance/glance-api.conf oslo\_messaging\_notifications driver messagingv2

# crudini --set /etc/glance/glance-registry.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/glance/glance-registry.conf oslo\_messaging\_notifications driver messagingv2

# systemctl restart openstack-glance-api openstack-glance-registry

# crudini --set /etc/cinder/cinder.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/cinder/cinder.conf oslo\_messaging\_notifications driver messagingv2

# systemctl restart openstack-cinder-api openstack-cinder-scheduler

# crudini --set /etc/heat/heat.conf oslo\_messaging\_notifications driver messagingv2

# crudini --set /etc/heat/heat.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# systemctl restart openstack-heat-api.service openstack-heat-api-cfn.service openstack-heat-engine.service

# crudini --set /etc/neutron/neutron.conf oslo\_messaging\_notifications driver messagingv2

# systemctl restart neutron-server.service

# crudini --set /etc/swift/proxy-server.conf filter:keystoneauth operator\_roles "admin, user, ResellerAdmin"

# crudini --set /etc/swift/proxy-server.conf pipeline:main pipeline "catch\_errors gatekeeper healthcheck proxy-logging cache container\_sync bulk ratelimit authtoken keystoneauth container-quotas account-quotas slo dlo versioned\_writes proxy-logging ceilometer proxy-server"

# crudini --set /etc/swift/proxy-server.conf filter:ceilometer paste.filter\_factory ceilometermiddleware.swift:filter\_factory

# crudini --set /etc/swift/proxy-server.conf filter:ceilometer url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME:5672/

# crudini --set /etc/swift/proxy-server.conf filter:ceilometer driver messagingv2

# crudini --set /etc/swift/proxy-server.conf filter:ceilometer topic notifications

# crudini --set /etc/swift/proxy-server.conf filter:ceilometer log\_level WARN

# systemctl restart openstack-swift-proxy.service

## 11.11 添加变量参数

# echo "export OS\_AUTH\_TYPE=password" >> /etc/keystone/admin-openrc.sh

## 11.12 安装软件包

**# compute 节点**

# yum install openstack-ceilometer-compute -y

## 11.13 配置Ceilometer

# crudini --set /etc/ceilometer/ceilometer.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials memcached\_servers = $HOST\_NAME:11211

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials project\_name service

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials auth\_type password

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials username ceilometer

# crudini --set /etc/ceilometer/ceilometer.conf service\_credentials password $CEILOMETER\_PASS

# crudini --set /etc/nova/nova.conf DEFAULT instance\_usage\_audit True

# crudini --set /etc/nova/nova.conf DEFAULT instance\_usage\_audit\_period hour

# crudini --set /etc/nova/nova.conf DEFAULT notify\_on\_state\_change vm\_and\_task\_state

# crudini --set /etc/nova/nova.conf oslo\_messaging\_notifications driver messagingv2

## 11.14 启动服务

# systemctl enable openstack-ceilometer-compute.service

# systemctl restart openstack-ceilometer-compute.service

# systemctl restart openstack-nova-compute

# 12 安装Aodh监控服务

## 12.1通过脚本安装Aodh服务

**12.2-12.9 Alarm监控服务的操作命令已经编写成shell脚本，通过脚本进行一键安装。如下：**

**#Controller节点**

**执行脚本iaas-install-aodh.sh进行安装**

## 12.2 创建数据库

# mysql -u root -p

mysql> CREATE DATABASE aodh;

mysql> GRANT ALL PRIVILEGES ON aodh.\* TO aodh@'localhost' IDENTIFIED BY '$AODH\_DBPASS';mysql> GRANT ALL PRIVILEGES ON aodh.\* TO aodh@'%' IDENTIFIED BY '$AODH\_DBPASS';

## 12.3 创建keystone用户

# openstack user create --domain $DOMAIN\_NAME --password $AODH\_PASS aodh

# openstack role add --project service --user aodh admin

## 12.4 创建Endpoint和API

# openstack service create --name aodh --description "Telemetry Alarming" alarming

# openstack endpoint create --region RegionOne alarming public http://$HOST\_NAME:8042

# openstack endpoint create --region RegionOne alarming internal http://$HOST\_NAME:8042

# openstack endpoint create --region RegionOne alarming admin http://$HOST\_NAME:8042

## 12.5 安装软件包

# yum -y install openstack-aodh-api openstack-aodh-evaluator openstack-aodh-notifier openstack-aodh-listener openstack-aodh-expirer python2-aodhclient

## 12.6 配置aodh

# crudini --set /etc/aodh/aodh.conf DEFAULT log\_dir /var/log/aodh

# crudini --set /etc/aodh/aodh.conf DEFAULT transport\_url rabbit://$RABBIT\_USER:$RABBIT\_PASS@$HOST\_NAME

# crudini --set /etc/aodh/aodh.conf api auth\_mode keystone

# crudini --set /etc/aodh/aodh.conf api gnocchi\_external\_project\_owner service

# crudini --set /etc/aodh/aodh.conf database connection mysql+pymysql://aodh:$AODH\_DBPASS@$HOST\_NAME/aodh

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken www\_authenticate\_uri http://$HOST\_NAME:5000

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken auth\_url http://$HOST\_NAME:5000

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken memcached\_servers $HOST\_NAME:11211

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken auth\_type password

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken project\_name service

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken username aodh

# crudini --set /etc/aodh/aodh.conf keystone\_authtoken password $AODH\_PASS

# crudini --set /etc/aodh/aodh.conf service\_credentials auth\_url http://$HOST\_NAME:5000/v3

# crudini --set /etc/aodh/aodh.conf service\_credentials auth\_type password

# crudini --set /etc/aodh/aodh.conf service\_credentials project\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/aodh/aodh.conf service\_credentials user\_domain\_name $DOMAIN\_NAME

# crudini --set /etc/aodh/aodh.conf service\_credentials project\_name service

# crudini --set /etc/aodh/aodh.conf service\_credentials username aodh

# crudini --set /etc/aodh/aodh.conf service\_credentials password $AODH\_PASS

# crudini --set /etc/aodh/aodh.conf service\_credentials interface internalURL

## 12.7 创建监听端点

修改/etc/httpd/conf.d/20-aodh\_wsgi.conf文件，添加以下内容：

Listen 8042

<VirtualHost \*:8042>

DocumentRoot "/var/www/cgi-bin/aodh"

<Directory "/var/www/cgi-bin/aodh">

AllowOverride None

Require all granted

</Directory>

CustomLog "/var/log/httpd/aodh\_wsgi\_access.log" combined

ErrorLog "/var/log/httpd/aodh\_wsgi\_error.log"

SetEnvIf X-Forwarded-Proto https HTTPS=1

WSGIApplicationGroup %{GLOBAL}

WSGIDaemonProcess aodh display-name=aodh\_wsgi user=aodh group=aodh processes=6 threads=3

WSGIProcessGroup aodh

WSGIScriptAlias / "/var/www/cgi-bin/aodh/app"

</VirtualHost>

## 12.8 同步数据库

# mkdir /var/www/cgi-bin/aodh

# cp /usr/lib/python2.7/site-packages/aodh/api/app.wsgi /var/www/cgi-bin/aodh/app

# chown -R aodh. /var/www/cgi-bin/aodh

# su -s /bin/bash aodh -c "aodh-dbsync"

## 12.9 启动服务

# systemctl restart openstack-aodh-evaluator openstack-aodh-notifier openstack-aodh-listener

#systemctl enable openstack-aodh-evaluator openstack-aodh-notifier openstack-aodh-listener

# systemctl restart httpd memcached

# 13 添加控制节点资源到云平台

## 13.1 修改openrc.sh

**把compute节点的IP和主机名改为controller节点的IP和主机名**

## 13.2 运行iaas-install-nova-compute.sh

**在控制节点运行iaas-install-nova-compute.sh**

**执行过程中需要确认登录controller节点和输入controller节点root用户密码。**