CLOUD-day02

云平台部署与管理

# Openstack概述

## 什么是云计算

•基于互联网的相关服务的增加、使用和交付模式

•这种模式提供可用的、便捷的、按需的网络访问,进入可配置的计算资源共享池

•这些资源能够被快速提供,只需投入很少的管理工作,或与服务供应商进行很少的交互

•通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源

## IaaS

• IaaS(Infrastructure as a Service),即基础设施即服务

• 提供给消费者的服务是对所有计算基础设施的利用,包括处理CPU、内存、存储、网络和其它基本的计算资源,用户能够部署和运行任意软件,包括操作系统和应用程序

• IaaS通常分为三种用法:公有云、私有云和混合云

## PaaS

• PaaS (Platform-as-a-Service),意思是平台即服务

• 以服务器平台或者开发环境作为服务进行提供就成为了PaaS

• PaaS运营商所需提供的服务,不仅仅是单纯的基础平台,还针对该平台的技术支持服务,甚至针对该平台而进行行的应用系统开发、优化等服务

• 简单地说,PaaS平台是指云环境中的应用基础设施服务,也可以说是中间件即服务

## SaaS

• SaaS(Software-as-a-Service)软件即服务,是一种通过Internet提供软件的模式,厂商将应用软件统一部署在自己的服务器上,客户可以根据自己实际需求,通过互联网向厂商定购所需的应用软件服务

• 用户不用再购买软件,而是向提供商租用基于Web的软件,来管理企业经营活劢,不用对软件进行维护,提供商会全权管理和维护软件,同时也提供软件的离线操作和本地数据存储

# Openstack简介

## 什么是Openstack

• OpenStack是一个由NASA(美国国家航空航天局)和Rackspace合作研发并发起的项目

• OpenStack是一套IaaS解决方案

• OpenStack是一个开源的云计算管理平台

• 以Apache许可证为授权

## Openstack主要组件

### • Horizon

– 用于管理Openstack各种服务的、基于web的管理接口

– 通过图形界面实现创建用户、管理网络、启动实例等操作

### • Keystone

– 为其他服务提供认证和授权的集中身份管理服务

– 也提供了集中的目录服务

– 支持多种身份认证模式,如密码认证、令牌认证、以及AWS(亚马逊Web服务)登陆

– 为用户和其他服务提供了SSO认证服务

### • Neutron

– 一种软件定义网络服务

– 用于创建网络、子网、路由器、管理浮劢IP地址

– 可以实现虚拟交换机、虚拟路由器

– 可用于在项目中创建VPN

### • Cinder

– 为虚拟机管理存储卷的服务

– 为运行在Nova中的实例提供永久的块存储

– 可以通过快照进行数据备份

– 经常应用在实例存储环境中,如数据库文件

### • Nova

– 在节点上用于管理虚拟机的服务

– Nova是一个分布式的服务,能够不Keystone交互实现认证,不Glance交互实现镜像管理

– Nova被设计成在标准硬件上能够迚行水平扩展

– 启动实例时,如果有则需要下载镜像

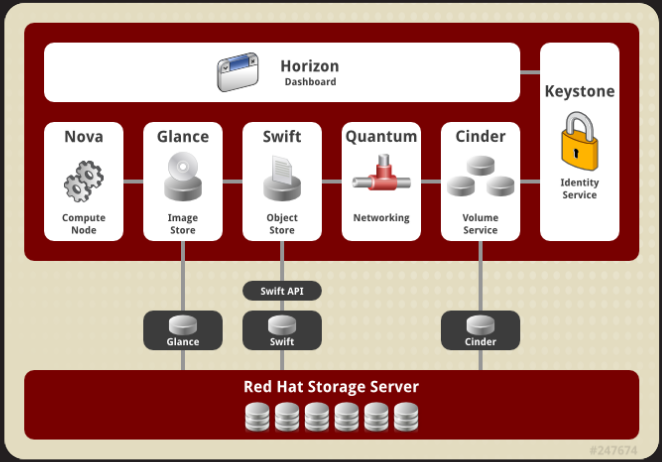
### • Glance

– 扮演虚拟机镜像注册的角色

– 允许用户为直接存储拷贝服务器镜像

– 这些镜像可以用于新建虚拟机的模板

## Openstack结构图



# 部署安装环境

## 搭建DNS服务器

## • 系统环境准备

– openstack 安装时候需要使用外部 dns 来解析域名,并且还需要外部时间服务器来保证所有节点的时间保持一致

– 我们需要创建一个 dns 服务器,并且为主机提供域名解析

– 将 openstack.tedu.cn 域名对应的 IP 解析到我们安装的openstack 的服务器

• 注:DNS 服务器不能与 openstack 安装在同一台主机上

[root@room9pc01 ～]# yum -y reinstall bind bind-chroot

[root@room9pc01 ～]# vim /etc/named.conf

options {

listen-on port 53 { 192.168.1.254; };

// listen-on-v6 port 53 { ::1; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

allow-query { any; };

forwarders {114.114.114.114; };

dnssec-enable no;

dnssec-validation no;

}

[root@room9pc01 docs]# systemctl restart named

## 搭建时间同步服务器

### 时区

• 时区:由于同一个时间点上,整个地球的时间应该都不一样,为了解决这个问题,地球被分成了24 个时区

• 由于地球被人们以『经纬度』坐标来进行定位,而经度为零的地点在英国格林威治市所在的纵剖面上

• 地球一圈是 360 度角,这 360 度角共分为 24 个时区,一个时区就是 15 度角

### UTC时间

• 计算时间最准确的方式是使用原子钟时间(原子震荡周期计算的时间),被定义为标准时间

• UTC(协和标准时间),就是利用原子钟为基准所定义的时间

• UTC 标准时间是以 GMT 这个时区为主,本地时间与UTC 时间的时差就是本地时间与GMT 时间的时差

[root@room9pc01 ~]# vim /etc/chrony.conf

server ntp.aliyun.com iburst

bindacqaddress 0.0.0.0

allow 0/0

[root@room9pc01 ~]# chronyc sources -v

### 虚拟机要求：

|  |  |
| --- | --- |
| 虚拟机1 openstack | 虚拟机2 nova |
| hostname openstack  内存 9.5G  硬盘 系统50G  硬盘 20G 空盘  网卡 2块  vbr  private1 | hostname nova01  内存 5.0G  硬盘 系统50G  网卡 2块  vbr  private1 |

## 步骤：

创建前端盘

# cd /var/lib/libvirt/images/

# qemu-img create -b node.qcow2 -f qcow2 openstack.img 50G

# qemu-img create -b node.qcow2 -f qcow2 nova01.img 50G

**添加大小为空的20G硬盘**

# qemu-img create -f qcow2 disk.img 20G

**修改模版文件中的名称为虚拟机名称**

# sed "s/node/openstack/" node.xml > /etc/libvirt/qemu/openstack.xml

# sed "s/node/nova01/" node.xml > /etc/libvirt/qemu/nova01.xml

**编辑虚拟机文件中的磁盘大小，添加网卡，添加空磁盘**

# vim /etc/libvirt/qemu/openstack.xml

# vim /etc/libvirt/qemu/nova01.xml

**根据配置文件创建虚拟机openstack、nova01**

# virsh define /etc/libvirt/qemu/openstack.xml

# virsh define /etc/libvirt/qemu/nova01.xml

# virsh start openstack //启动虚拟机openstack

# virsh start nova01 //启动虚拟机nova01

### 进入虚拟机，做初始修改

**初始配置openstack**

# virsh console openstack

# hostnamectl set-hostname openstack

# vim /etc/sysconfig/network-scripts/ifcfg-eth0

IPADDR="192.168.1.10"

NETMASK="255.255.255.0"

GATEWAY="192.168.1.254"

# cd /etc/sysconfig/network-scripts/

# cp ifcfg-eth0 ifcfg-eth1

# vim ifcfg-eth1

IPADDR="192.168.4.10"

NETMASK="255.255.255.0"

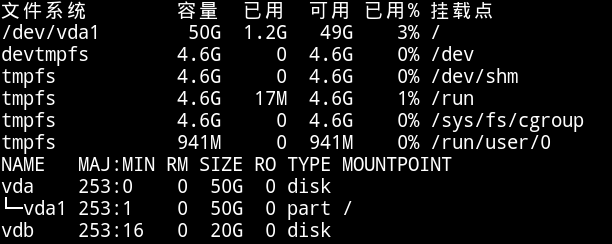
# vim /etc/hosts //追加

......

192.168.1.10 openstack

192.168.1.11 nova01

# df -h && lsblk



# reboot

### 初始配置nova01

# virsh console nova01

# hostnamectl set-hostname nova01

# vim /etc/sysconfig/network-scripts/ifcfg-eth0

IPADDR="192.168.1.11"

NETMASK="255.255.255.0"

GATEWAY="192.168.1.254"

# cd /etc/sysconfig/network-scripts/

# cp ifcfg-eth0 ifcfg-eth1

# vim ifcfg-eth1

IPADDR="192.168.4.11"

NETMASK="255.255.255.0"

# brctl show //查看网卡信息

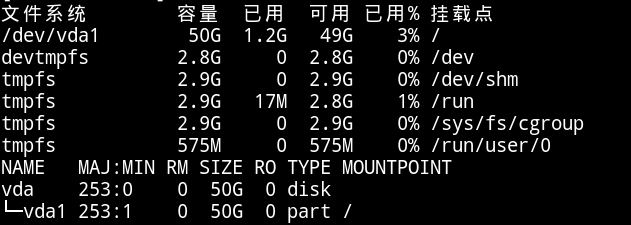
# vim /etc/hosts //追加

......

192.168.1.10 openstack

192.168.1.11 nova01

# df -h && lsblk



# reboot

### 真机准备yum源

[root@room9pc01 ~]# mkdir /var/ftp/CentOS-1807/

[root@room9pc01 ~]# mkdir /var/ftp/RHE-ext/

[root@room9pc01 ~]# mkdir /var/ftp/RHE-OSP/

[root@room9pc01 ~]# vim /etc/fstab

/root/桌面/CLOUD/ISO/CentOS7-1708.iso /var/ftp/CentOS-1807/ iso9660 defaults 0 0

/root/桌面/CLOUD/ISO/RHEL7-extras.iso /var/ftp/RHE-ext/ iso9660 defaults 0 0

/root/桌面/CLOUD/ISO/RHEL7OSP-10.iso /var/ftp/RHE-OSP/ iso9660 defaults 0 0

[root@room9pc01 ~]# mount -a

**虚拟机书写yum源（内容详见文件CentOS-1807.repo）**

<https://github.com/>asdzhubo/CLOUD/blob/master/day02/CentOS-1807.repo

**配置时间同步服务(两台虚拟机做相同操作)**

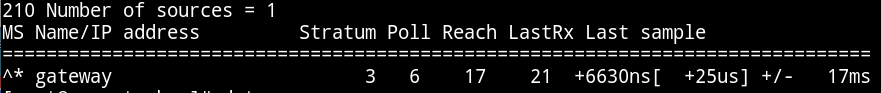
# vim /etc/chrony.conf

server gateway iburst

# systemctl restart chronyd

# systemctl enable chronyd

# chronyc sources



# timedatectl



## 配置卷组

• Openstack为虚拟机提供的云硬盘,本质上是本地的逻辑卷

• 逻辑卷创建于名为cinder-volumes的卷组

• 没有物理卷可以使用 loop 文件替代

– losetup loop0 /var/disk.img

– pvcreate /dev/loop0

– vgcreate cinder-volumes /dev/loop0

[root@openstack ~]# yum -y install lvm2 //安装软件提供lv命令

[root@openstack ~]# pvcreate /dev/vdb //创建物理卷

[root@openstack ~]# vgcreate cinder-volumes /dev/vdb //创建卷组

[root@openstack ~]# vgs //查看

## 导入公钥

• 安装openstack期间,系统会要求密钥验证

• 我仧手工导入系统密钥

• 密钥文件在光盘中提供

## 安装额外软件包

• 安装openstack期间,有些软件包所依赖的软件包,并没有在安装过程中安装

• 这些软件包需提前安装

• 本地RPM包也可以通过yum进行安装

### 在openstack与nova01上安装额外软件

# yum -y install qemu-kvm libvirt-client libvirt-daemon libvirt-daemon-driver-qemu python-setuptools

## 检查openstack环境部署

• 是否禁用selinux

• 是否卸载firewalld和NetworkManager

• 检查配置主机网络参数(静态IP)

• 检查配置主机yum源(12个)

• 检查cinder-volumes卷组是否已经创建

• 检查公钥是否导入

• 查看相关软件包是否安装

• 检查NTP和DNS服务器是否可用

# 安装Openstack

## 配置packstack

### • 在openstack机安装packstack

[root@openstack ~]# yum -y install openstack-packstack

### 一键部署Openstack

• 如果前期环境准备无误,只要耐心等待安装结束即可

• 根据主机配置丌同,安装过程需要20分钟左右戒更久

• 如果出现错误,根据屏幕上给出的日志文件迚行排错

## 在root下执行

### 创建answer.ini配置文件

[root@openstack ~]# packstack --gen-answer-file answer.ini

### 修改auswer.ini配置文件

[root@openstack ~]# vim answer.ini

11 CONFIG\_DEFAULT\_PASSWORD=redhat //设置默认密码

42 CONFIG\_SWIFT\_INSTALL=n //对象存储

75 CONFIG\_NTP\_SERVERS=192.168.1.254 //自己配置的NTP时间服务器

554 CONFIG\_CINDER\_VOLUMES\_CREATE=n //cinder\_volumes卷组创建

840 CONFIG\_NEUTRON\_ML2\_TYPE\_DRIVERS=flat,vxlan

//openstack支持的网络种类；flat（扁平网络）,vxlan（隔离网络）

876 CONFIG\_NEUTRON\_ML2\_VXLAN\_GROUP=239.1.1.3 //组播

910 CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-ex

//声明一个虚拟交换机（带有物理接口），作用打通次元网络

921 CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=br-ex:eth0

//将eth0网卡转换为虚拟交换机的物理接口

936 CONFIG\_NEUTRON\_OVS\_TUNNEL\_IF=eth1 //设置隧道网络使用的网卡

1179 CONFIG\_PROVISION\_DEMO=n //测试配置,修改为不配置（相当于守护进程）

### 执行（安装openstack）

[root@openstack ~]# packstack --answer-file=answer.ini

**出现** \*\*\*\* Installation completed successfully \*\*\*\*\*\* **表示安装成功**

# Openstack操作基础

## Horizon配置

## 概述

• Horizon是一个用以管理、控制OpenStack服务的Web控制面板,也称之为Dashboard仪表盘

• 可以管理实例、镜像、创建密匙对,对实例添加卷、操作Swift容器等。除此之外,用户还可以在控制面板中使用终端(console)或VNC直接访问实例

• 基于python的django web框架进行开发

## 功能与特点

• 实例管理:创建、终止实例,查看终端日志,VNC连接,添加卷等

• 访问不安全管理:创建安全群组,管理密钥对,设置浮动IP等

• 偏好设定:对虚拟硬件模板可以进行不同偏好设定

• 镜像管理:编辑或删除镜像

• 用户管理:创建用户等

• 卷管理:创建卷和快照

• 对象存储处理:创建、删除容器和对象

## • Horizon BUG 处理

**• 安装虽然没有报错,但默认无法打开 Horizon,这是一个软件的配置 BUG**

### 解决办法

[root@openstack ~]# cd /etc/httpd/conf.d/

[root@openstack conf.d]# vim 10-aodh\_wsgi.conf

25 WSGIApplicationGroup %{GLOBAL}

//将此行复制到15-horizon\_vhost.conf文件中

[root@openstack conf.d]# vim 15-horizon\_vhost.conf

35 WSGIProcessGroup apache

36 WSGIApplicationGroup %{GLOBAL}

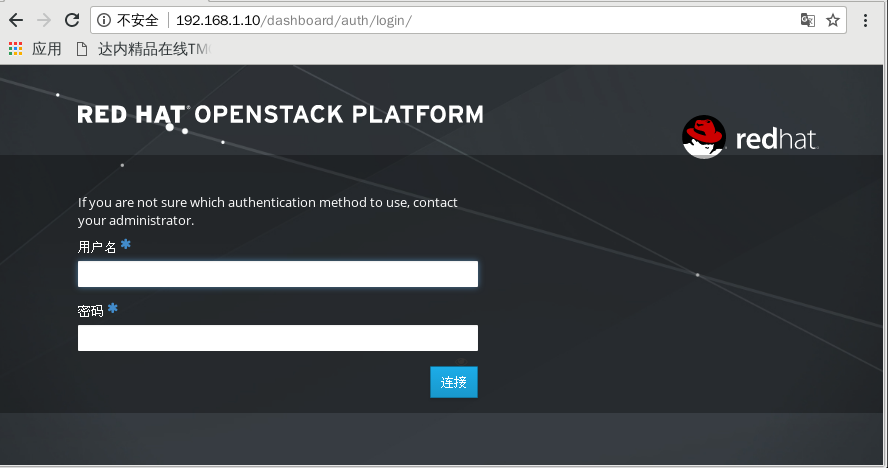
### 重新载入配置文件

[root@openstack conf.d]# systemctl restart httpd

**或者**

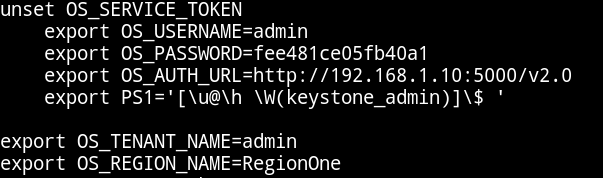
[root@openstack conf.d]# apachectl graceful

### 网页地址栏输入：192.168.1.10

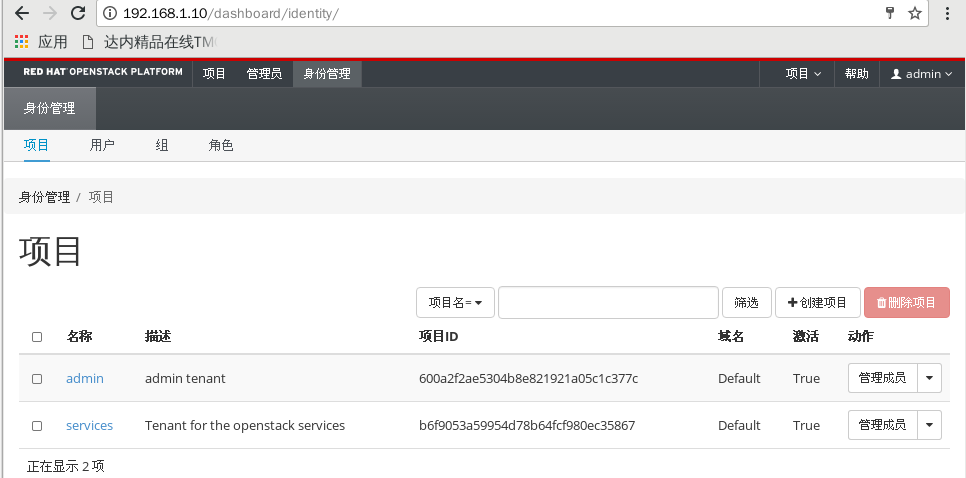


### **登录所使用的账户名和初始密码通过查看文件/root/keystonerc\_admin使用**

[root@openstack ~]# cat /root/keystonerc\_admin

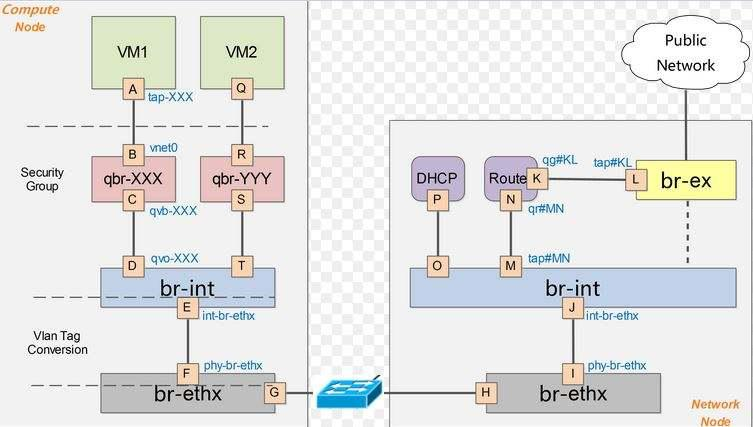


### 登入之后即可进新管理



# 网络配置

## 拓扑

****

**br-ethx：虚拟网桥整合之后显示**

## 查看外部OVS网桥

**• 配置br-ex为外部OVS网桥**/etc/sysconfig/network-scripts/ifcfg-br-ex

DEVICE=br-ex

NAME=br-ex

DEVICETYPE=ovs **//OVS模式的交换机**

OVSBOOTPROTO="static"

TYPE=OVSBridge

## 查看外部OVS网桥端口

**• 配置eth0为外部OVS网桥的端口**/etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

NAME=eth0

DEVICETYPE=ovs

TYPE=OVSPort

OVS\_BRIDGE=br-ex

ONBOOT=yes

BOOTPROTO=none

## 验证OVS配置

**1. ovs-vsctl show**

**2. ip -o addr show (-o是在一行显示)**

Port "eth0"

Interface "eth0"  **//子接口**

# 项目管理

## 基本概念

### • 项目:一组隔离的资源和对象。由一组关联的用户进行管理

### • 在旧版本里,也用租户(tenant)来表示

### • 根据配置的需求,项目对应一个组织、一个公司或是一个使用客户等

### • 项目中可以有多个用户,项目中的用户可以在该项目创建、管理虚拟资源

### • 具有admin角色的用户可以创建项目

### • 项目相关信息保存到MariaDB中

### • 缺省情况下,packstack安装的openstack中有两个

## 独立的项目

### • 缺省情况下,packstack安装的openstack中有两个独立的项目

– admin:为admin账户创建的项目

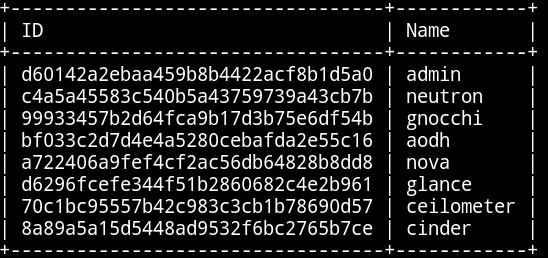
– services:不安装的各个服务相关联

## 命令行接口基础

### 初始化环境变量

[root@openstack ~]# source /root/keystonerc\_admin

[root@openstack ~(keystone\_admin)]# openstack user list



### 使用帮助(每个版本的命令有所不同)

[root@openstack ~(keystone\_admin)]# openstack help