

云计算部署与管理

NSD CLOUD

DAY01

内容

上午	09:00 ~ 09:30	Openstack概述
	09:30 ~ 10:20	
	10:30 ~ 11:20	部署安装环境
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	部署Openstack
	15:00 ~ 15:50	
	16:10 ~ 17:00	Openstack操作基础
	17:10 ~ 18:00	总结和答疑



Openstack概述

Openstack概述

云计算简介

什么是云计算

IaaS

PaaS

SaaS

Openstack简介

什么是Openstack

Openstack主要组件

Openstack版本

Openstack结构图

云计算简介



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（资源包括网络，服务器，存储，应用软件，服务）
- 这些资源能够被快速提供，只需投入很少的管理工作，或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
- 消费者不管理或控制任何云计算基础设施, 但能控制操作系统的选择、存储空间、部署的应用
- IaaS通常分为三两种用法: 公有云共的和、私有云的和混合云



PaaS

- PaaS是Platform-as-a-Service的缩写，意思是平台即服务
- 云计算时代相应的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，而且包括针对该平台的技术支持服务，甚至针对该平台而进行的应用系统开发、优化等服务
- 简单地说，PaaS平台就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是Software-as-a-Service（软件即服务）的简称
- 它是一种通过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，而改用向提供商租用基于Web的软件，来管理企业经营活动，且无需对软件进行维护，服务提供商会全权管理和维护软件，软件厂商在向客户提供互联网应用的同时，也提供软件的离线操作和本地数据存储，让用户随时随地都可以使用其定购的软件和服务



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



Openstack主要组件（续1）

- 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，如果密码认证、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



Openstack主要组件（续2）

- Neutron
 - 一种软件定义网络服务
 - 用于创建网络、子网、路由器、管理浮动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



Openstack主要组件（续3）

- 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，如果数据库文件



Openstack主要组件（续4）

- 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，能够与Keystone交互实现认证，与Glance交互实现镜像管理
 - Nova被设计成在标准硬件上能够进行水平扩展
 - 启动实例时，如果有需要则下载镜像



Openstack主要组件（续5）

- 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

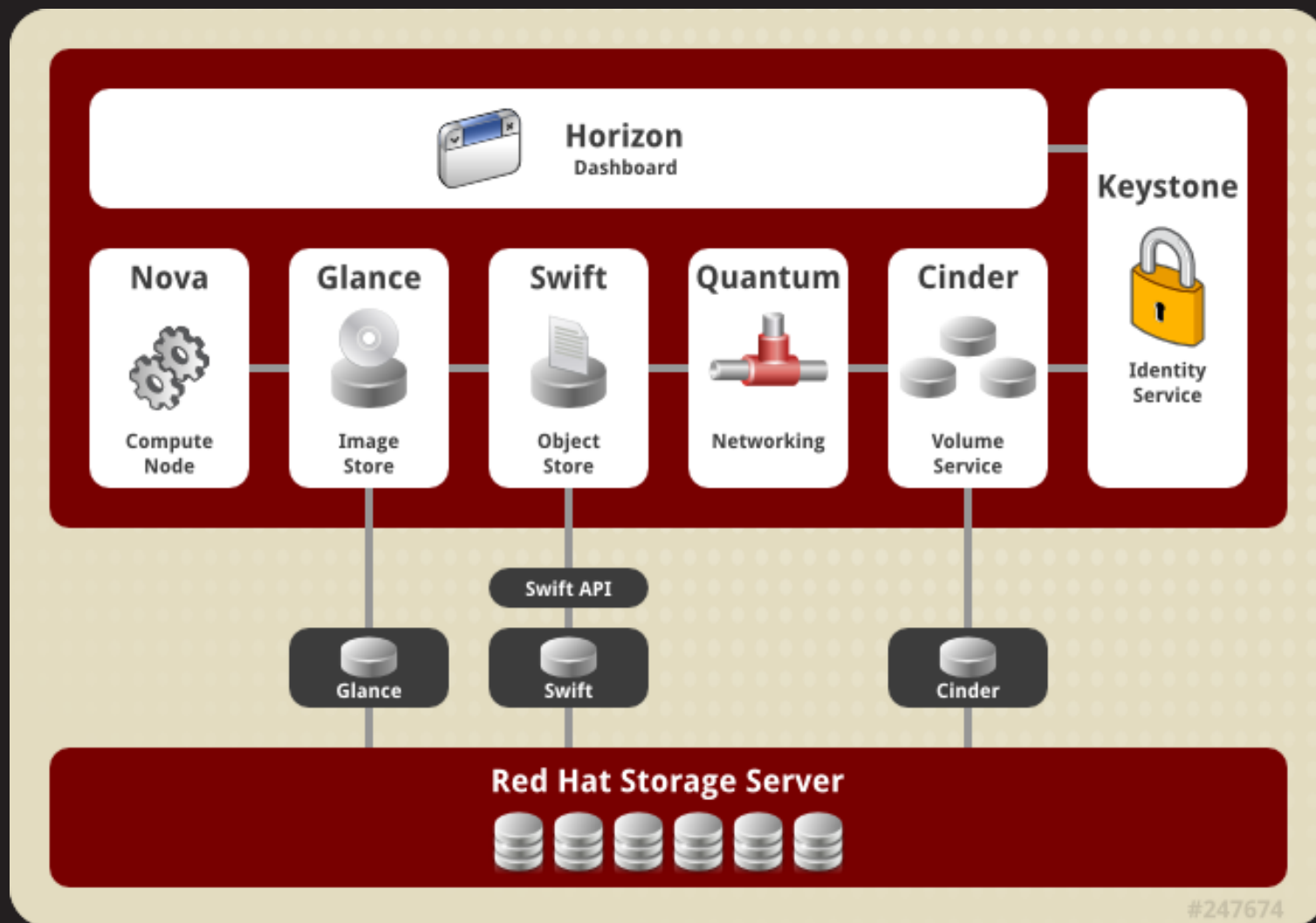


Openstack版本



Openstack架构图

知识讲解



#247674

部署安装环境

部署安装环境

Yum仓库及名称解析

配置yum仓库

设置DNS服务器

NTP服务

时区

UTC时间

配置NTP服务

启动并测试

Yum仓库及名称解析

配置yum仓库

- RHEL7光盘信息必须提供
- RHEL7OSP光盘拥有众多目录，每个目录都是一个仓库，因此，需要把这些目录都配置为yum源
- RHEL7-extras 扩展包光盘在安装的时候也是需要的



案例1：配置yum仓库

1. RHEL7光盘内容作为仓库源
2. 配置 RHEL7-extars 内容加入仓库源
3. RHEL7OSP 光盘中包含多个目录，每个目录都是仓库源（可以使用脚本生成）



设置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，并且为我们的主机提供域名解析哦
 - 将 openstack.tedu.cn 域名对应的 IP 解析到我们的安装openstac 的服务器
- 注：DNS 服务器不能与 openstack 安装在同一台主机上



案例2：配置DNS服务器

1. 允许DNS服务器为所有的客户端提供服务
2. 解析域名 openstack.tedu.cn
3. 解析域名 nova01.tedu.cn



NTP服务

时区

- 时区：由于同一个时间点上面，整个地球的时间应该都不一样，为了解决这个问题，地球就被分成了24个时区
- 由于地球被人类以『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』这个城市所在纵剖面上
- 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

- 计算时间的时候，最准确的计算应该是使用『原子震荡周期』所计算的物理时钟了(Atomic Clock, 也被称为原子钟)，这也被定义为标准时间
- 我们常常看见的 UTC 也就是 Coordinated Universal Time (协和标准时间)就是利用这种 Atomic Clock 为基准所定义出来的正确时间
- 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



配置NTP服务

- 安装

```
[root@vh01 ~]# yum install -y chrony
```

- 修改配置文件

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

```
// 设置同步网络时间服务器
```

```
pool asia.pool.ntp.org iburst prefer
```

```
# Ignore stratum in source selection.
```

```
bindacqaddress 0.0.0.0
```

```
allow 0/0
```

```
cmdallow 127.0.0.1
```



启动并测试

- 启动服务

```
[root@service ~]# systemctl start chronyd  
[root@service ~]# systemctl enable chronyd
```

- 在另一台服务器上测试校准时间

```
[root@service ~]# vim /etc/chrony.conf  
server time.tedu.cn iburst
```

```
[root@service ~]# systemctl start chronyd  
[root@service ~]# chronyc sources -v
```

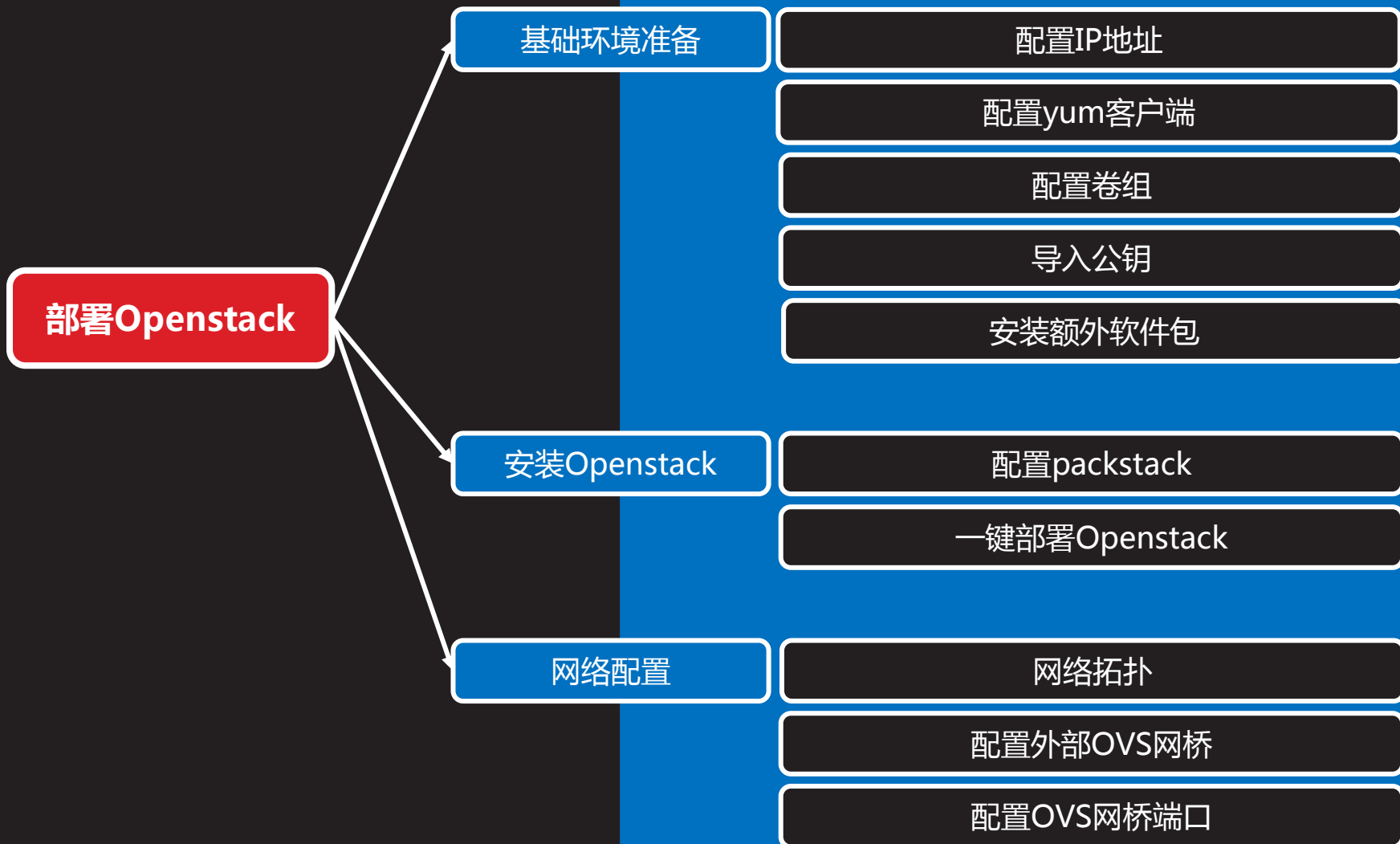


案例3：配置NTP服务器

1. 将NTP服务与DNS服务部署在同一台主机上
2. 确认NTP服务器的时区是东八区
3. 确认NTP服务器的时间准确
4. 计划安装openstack的服务器与NTP服务器进行时间校正



部署Openstack



基础环境准备



配置IP地址

- 配置eth0为公共网络，网络地址192.168.1.0/24
- 配置eth1为隧道接口，网络地址192.168.2.0/24
- 关闭NetworkManager服务
- 禁用 SELINUX
- 卸载 firewalld



配置yum客户端

- 将RHEL7光盘作为基础的yum源
- 将RHEL7-extars光盘作为扩展的yum源
- 将RHEL7OSP光盘中所有目录作为yum源

```
[root@nova ~]# yum repolist
```

源标识	源名称	状态
calamari	calamari	15
common	common	76
director	director	39
extra	extra	54
...	...	



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@nova ~]# vgs
```

```
VG                #PV #LV #SN Attr   Vsize   VFree
cinder-volumes    1   0   0  wz--n- 1020.00m 1020.00m
```



导入公钥

- 安装openstack期间，系统会要求密钥验证
- 我们手工导入系统密钥
- 密钥文件在光盘中提供

```
[root@nova ~]# rpm --import RPM-GPG-KEY-CentOS-7
```



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

```
[root@nova ~]# yum install -y qemu-kvm libvirt-client libvirt-daemon  
libvirt-daemon-driver-qemu python-setuptools
```



案例4：基础环境准备

1. 禁用 selinux
2. 卸载 firewalld
3. 卸载 NetworkManager
4. 配置主机IP地址
5. 配置主机yum客户端
6. 创建cinder-volumes的卷组
7. 导入公钥
8. 安装依赖的软件包



安装openstack



配置packstack

- 安装packstack

```
[root@vh02 ~]# yum install -y openstack-packstack
```

```
[root@vh02 ~]# packstack --gen-answer-file answer.txt
```

- 修改应答文件

```
[root@nova ~]# vim answer.txt
```

```
CONFIG_DEFAULT_PASSWORD=Taren1
```

```
CONFIG_SWIFT_INSTALL=n
```

```
CONFIG_NTP_SERVERS=192.168.1.253
```

```
CONFIG_CINDER_VOLUMES_CREATE=n
```

```
CONFIG_NEUTRON_ML2_TYPE_DRIVERS=flat,vxlan
```

```
CONFIG_NEUTRON_ML2_VXLAN_GROUP=239.1.1.5
```

```
CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=physnet1:br-ex
```

```
CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:eth0
```

```
CONFIG_NEUTRON_OVS_TUNNEL_IF=eth1
```

```
CONFIG_PROVISION_DEMO=n
```



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
- 根据主机配置不同，安装过程需要20分钟左右或更久
- 如果出现错误，根据屏幕上给出的日志文件进行排错

```
[root@vh02 ~]# packstack --answer-file answer.txt
```



案例5：部署Openstack

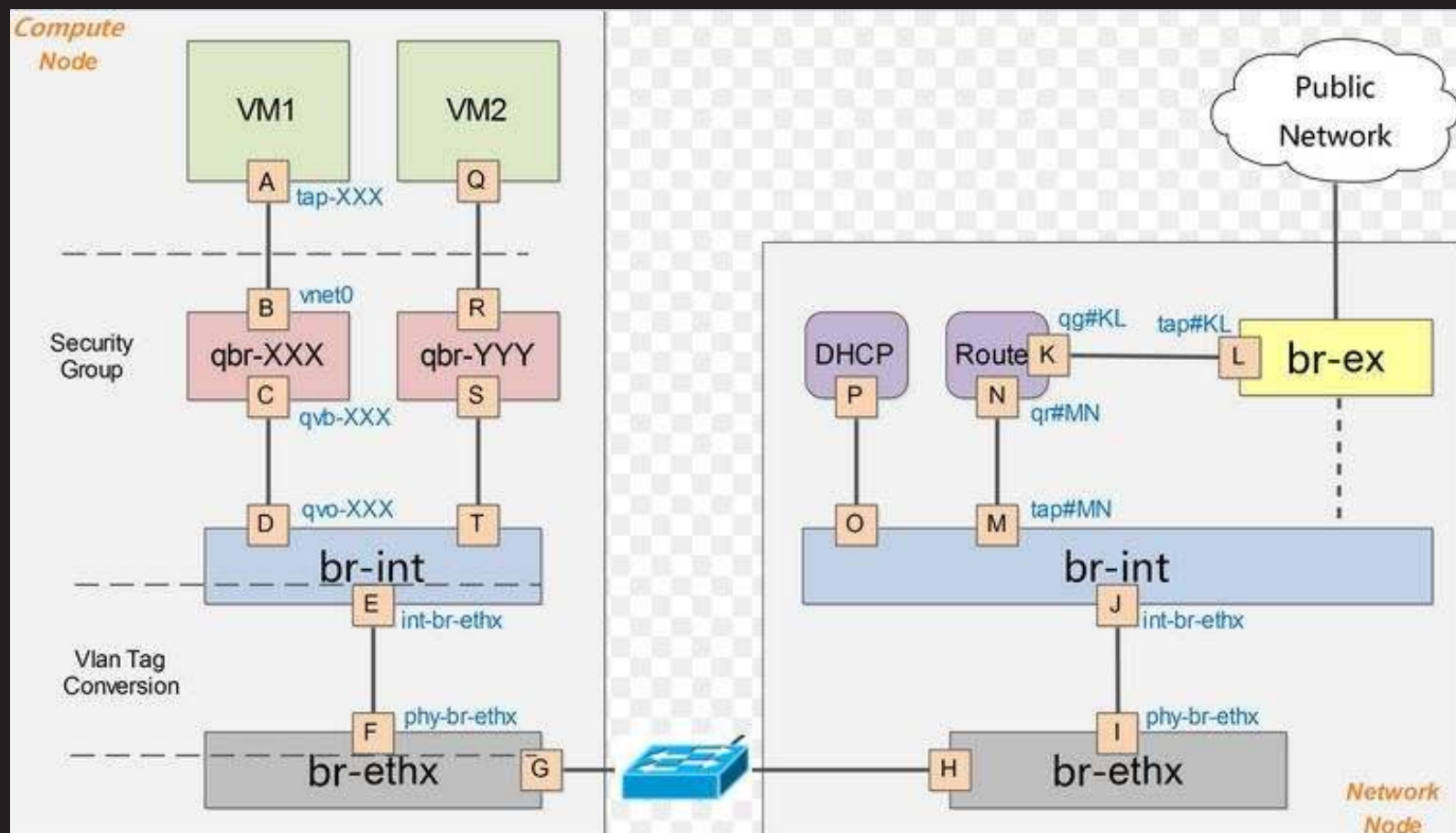
1. 通过packstack部署Openstack
2. 根据相关日志文件进行排错



网络配置

网络拓扑

- 多计算节点的拓扑如下所示



配置外部OVS网桥

- 配置br-ex为外部OVS网桥

```
[root@nova ~]# cat /etc/sysconfig/network-scripts/ifcfg-br-ex
ONBOOT="yes"
NM_CONTROLLED="no"
IPADDR="192.168.1.66"
PREFIX="24"
GATEWAY="192.168.1.254"
DEVICE=br-ex
NAME=br-ex
DEVICETYPE=ovs
OVSBOOTPROTO="static"
TYPE=OVSBridge
```



配置外部OVS网桥端口

- 配置eth0为外部OVS网桥的端口

```
[root@nova ~]# cat /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

```
[root@nova ~]# ovs-vsctl show
Bridge br-ex
    Controller "tcp:127.0.0.1:6633"
        is_connected: true
    fail_mode: secure
    ... ..
Port br-ex
    Interface br-ex
        type: internal
Port "eth0"
    Interface "eth0"
```



Openstack操作基础

Openstack操作基础

Horizon配置

概述

功能与特点

基本配置

项目管理

基本概念

在Horizon中管理项目

命令行接口基础

通过命令行管理项目

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、创建密钥对，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
- 访问与安全管理：创建安全群组，管理密匙对，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

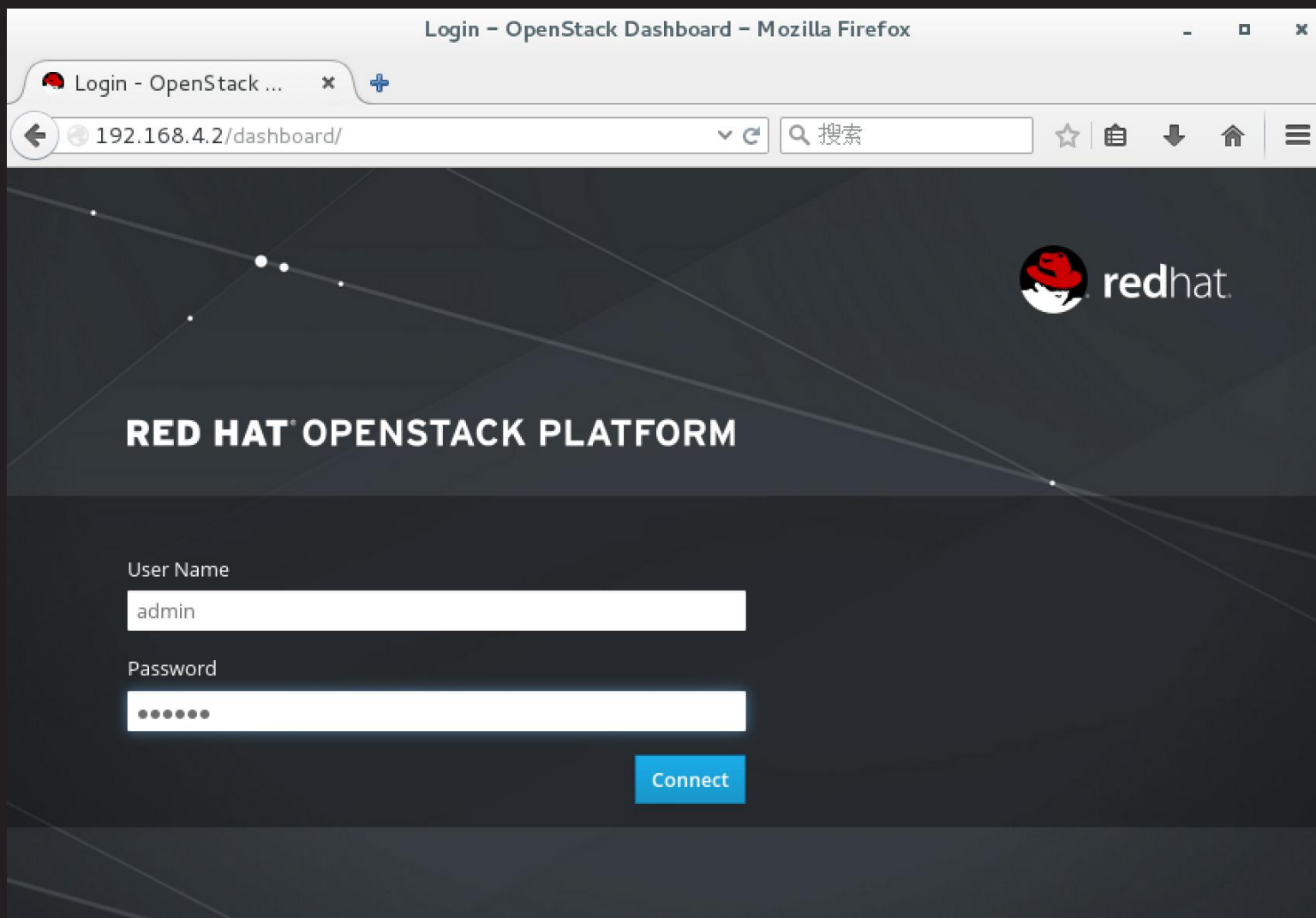


Horizon

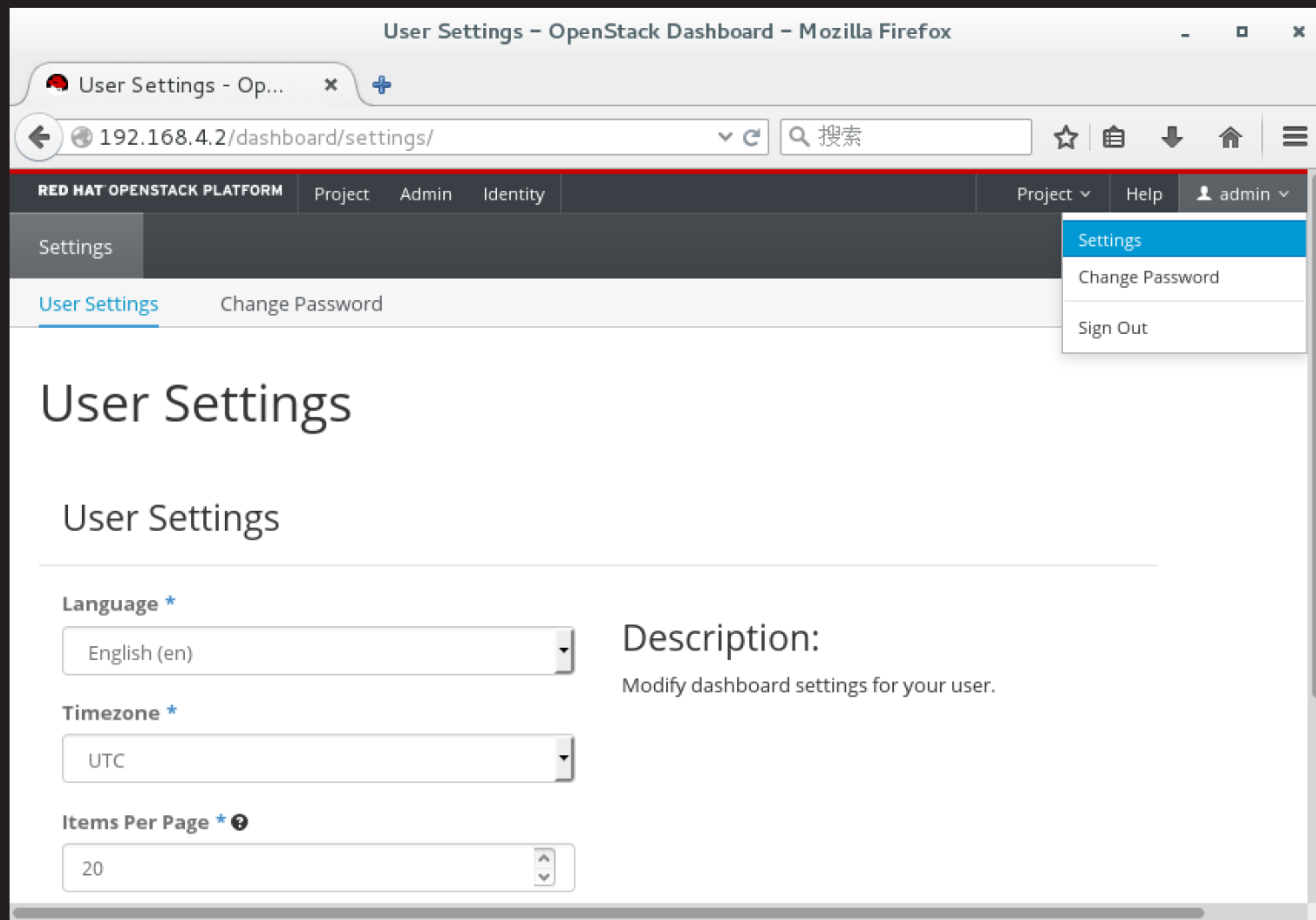
- Horizon BUG 处理
- 安装虽然没有报错，但默认无法打开 Horizon，这是一个软件的配置 BUG
 - /etc/httpd/conf.d/15-horizon_vhost.conf
 - WSGIApplicationGroup %{GLOBAL}
 - ServerAlias localhost
 - WSGIDaemonProcess apache group=apache processes=3 threads=10 user=apache
 - WSGIProcessGroup apache
 - WSGIApplicationGroup %{GLOBAL} <--- 这里添加
- 重新载入配置文件 apache graceful



基本配置



基本配置（续1）



项目管理



基本概念

- 项目：一组隔离的资源 and 对象。由一组关联的用户进行管理
- 在老的版本里，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、管理虚资资源
- 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



基本概念（续1）

- 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：与安装的各个服务相关联



在Horizon中管理项目

- 通过Horizon可以进行项目的创建和删除

RED HAT OPENSTACK PLATFORM

项目 管理员 身份管理

项目 ▾ 帮助 admin ▾

身份管理

项目 用户

项目

筛选

+

 创建项目

删除项目

<input type="checkbox"/>	名称	描述	项目ID	激活	动作
<input type="checkbox"/>	admin	admin tenant	46214a2e0afb4d61a7e5e691571f8bcc	True	管理成员 ▾
<input type="checkbox"/>	services	Tenant for the openstack services	79595f43f85e4ba2b741ed73995317fd	True	管理成员 ▾

正在显示 2 项



命令行接口基础

- 初始化环境变量

```
[root@vh02 ~]# source ~/keystonerc_admin  
[root@vh02 ~(keystone_admin)]# env | grep OS  
HOSTNAME=vh02.tedu.cn  
OS_REGION_NAME=RegionOne  
OS_PASSWORD=redhat  
OS_AUTH_URL=http://192.168.4.2:5000/v2.0  
OS_USERNAME=admin  
OS_TENANT_NAME=admin
```

- 使用帮助

```
[root@vh02 ~(keystone_admin)]# openstack help
```



通过命令行管理项目

- 创建名为myproject项目

```
[root@vh02 ~(keystone_admin)]# openstack project create myproject
```

- 列出所有项目

```
[root@vh02 ~(keystone_admin)]# openstack project list
```

- 查看myproject详细信息

```
[root@vh02 ~(keystone_admin)]# openstack project show myproject
```

- 禁用与激活项目

```
[root@vh02 ~(keystone_admin)]# openstack project set --disable  
myproject
```

```
[root@vh02 ~(keystone_admin)]# openstack project set --enable  
myproject
```



通过命令行管理项目（续1）

- 查看项目配额

```
[root@vh02 ~(keystone_admin)]# nova quota-show --tenant myproject
```

- 更新可用vcpu数目为30

```
[root@vh02 ~(keystone_admin)]# nova quota-update --cores 30  
myproject
```

- 删除myproject

```
[root@vh02 ~(keystone_admin)]# openstack project delete  
myproject
```

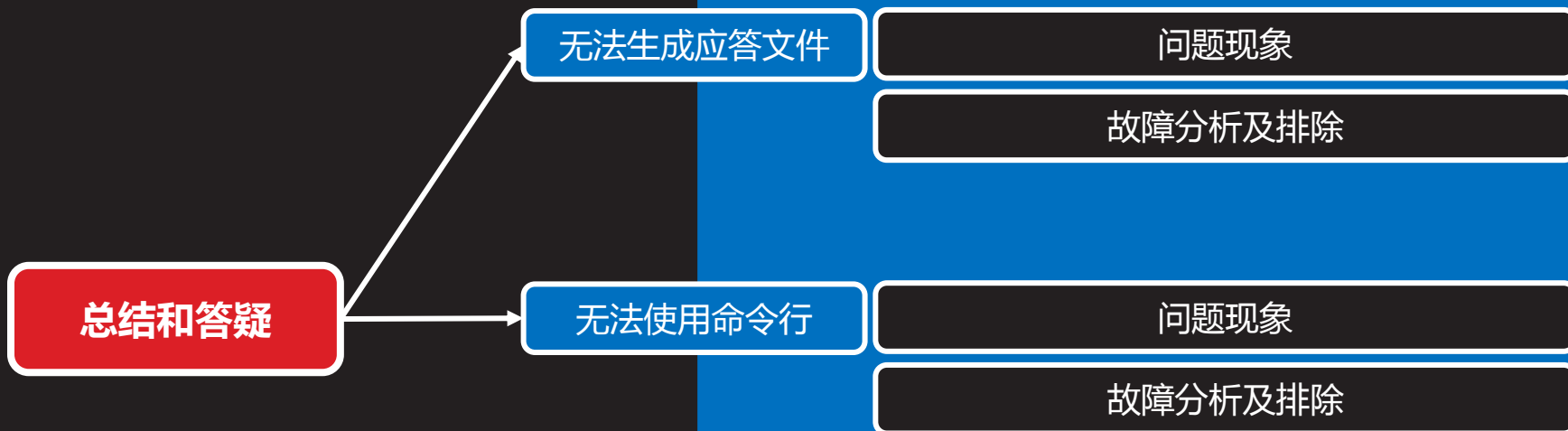


案例7：管理项目

- 通过Horizon完成以下操作
 1. 创建名为myproject的项目
 2. 查看项目信息
 3. 更新vcpu配额为30
 4. 删除myproject



总结和答疑



无法生成应答文件

问题现象

- 通过packstack生成应答文件时，命令无法成功执行，提示name resolution error



故障分析及排除

- 原因分析
 - Name resolution error提示的是名称解析错误
- 解决办法
 - 验证待安装openstack的服务器是否可以正常做域名解析，将域名解析故障排除即可



无法使用命令行

问题现象

- 通过命令管理项目时，出现以下错误

```
[root@vh02 ~]# openstack project show
```

```
Missing parameter(s):
```

```
Set a username with --os-username, OS_USERNAME, or auth.username
```

```
Set an authentication URL, with --os-auth-url, OS_AUTH_URL or  
auth.auth_url
```

```
Set a scope, such as a project or domain, set a project scope with --os-  
project-name, OS_PROJECT_NAME or auth.project_name, set a domain  
scope with --os-domain-name, OS_DOMAIN_NAME or  
auth.domain_name
```



故障分析及排除

- 原因分析
 - 执行命令时，必须具有相关的权限
 - 权限通过环境变量设置
- 解决办法
 - 初始化管理员目录下的keystonerc_admin文件即可

