# 第四阶段 CLOUD云计算

## DAY01 虚拟化 virsh管理 云平台 华为云

[root@room9pc01 ~]# yum -y install git

[root@room9pc01 ~]# git clone **git://43.254.90.134/nsd1903.git**

正克隆到 'nsd1903'...

remote: Counting objects: 4, done.

remote: Compressing objects: 100% (4/4), done.

remote: Total 4 (delta 0), reused 0 (delta 0)

接收对象中: 100% (4/4), done.

[root@room9pc01 ~]# ls

nsd1903

[root@room9pc01 ~]# cd nsd1903

[root@room9pc01 nsd1903]# ls

**nginx.service readme.txt**

**云平台上运行的都是虚拟机和容器**

### 虚拟化

**环境介绍:第四阶段专用虚拟机模板(其中没有封装多的软件,自己需要啥就安装啥)**

[root@room9pc01 ~]# vim /etc/clone.conf 修改文件

name=base

[root@room9pc01 ~]# clone-vm7**(base-vm基础模板创建虚拟机) 克隆虚拟机**

一).kvm简介

##### 1.虚拟化概念

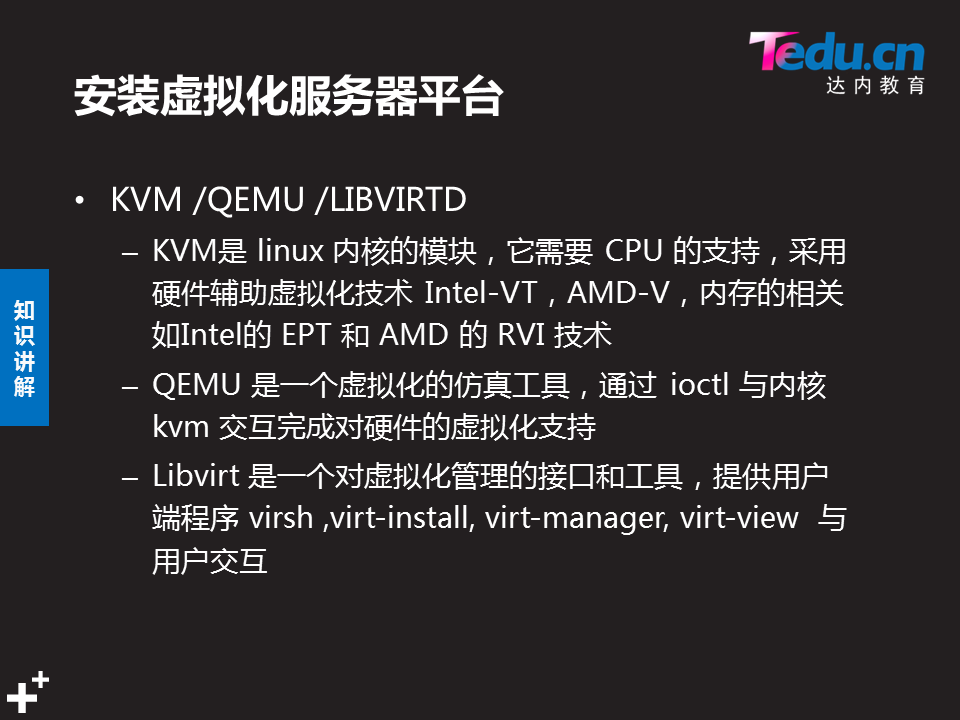


**Esxi(linux+vm)生产环境中使用的虚拟化软件**

**Vmware收费版**

**现阶段主流的为kvm vsphere**

##### **安装虚拟化服务器平台(由kvm qemu libvirt构成)**



**Kvm内核模块**

**Qemu底层仿真工具**

**Libvirt用户接口**

[root@room9pc01 ~]# ps -efww (查看进程 ww全部详细信息)

[root@room9pc01 ~]# ps -ef

1)安装虚拟化平台需要安装4个必备软件





二).管理kvm平台

##### 1.virsh管理工具(只对redhat7适用,其他系统可以该配置文件)

**提供管理各虚拟机的命令接口:**

* 支持交互式模式,查看/创建/停止/关闭
* **格式:virsh 控制指令 [虚拟机名称] [参数]**

##### Virsh虚拟机管理命令

**-list [--all] 列出虚拟机**

**-start|shutdown|reboot 虚拟机启动/停止/重启**

**-destroy 强制停止虚拟机**

**-define|undefine 根据xml文件创建/删除虚拟机**

**-console 链接虚拟机 (改ip)**

**-edit 修改虚拟机的配置**

[root@room9pc01 ]# virsh edit tedu\_node46

**-autostart 设置虚拟机自启**

**-domiflist 查看虚拟机网卡信息**

[root@room9pc01 nsd1903]# virsh domiflist tedu\_node46

接口 类型 源 型号 MAC

-------------------------------------------------------

vnet27 bridge vbr virtio 52:54:00:9d:18:14

**-domblklist查看虚拟机硬盘信息**

[root@room9pc01 nsd1903]# virsh domblklist tedu\_node46

目标 源

------------------------------------------------

vda /var/lib/libvirt/images/tedu\_node46.img

##### Virsh虚拟交换机网络管理命令

**- net-list [--all] 列出虚拟网络**

[root@room9pc01 nsd1903]# virsh net-list

名称 状态 自动开始 持久

default 活动 是 是

private1 活动 是 是

private2 活动 是 是

**- net-start 启动虚拟交换机**

**- net-destroy 强制停止虚拟交换机**

**- net-define 根据xml文件创建虚拟交换机**

**- net-undefine 删掉一个虚拟网络设备**

**- net-edit 修改虚拟交换机配置**

**- net-autostart 设置虚拟交换机自启**

1）列出当前正在运行的虚拟机

[root@room9pc01 ~]# virsh list

Id Name State

1 node1 running

步骤二：查看虚拟机的信息

[root@room9pc01 ~]# virsh **dominfo**  nova

步骤三：管理虚拟机

1）启动虚拟机

[root@room9pc01 ~]# virsh start node1

2）重启虚拟机

[root@room9pc01 ~]# virsh reboot node1

3）强制关闭虚拟机

[root@room9pc01 ~]# virsh destroy node1

4）设置虚拟机开机自动运行

[root@room9pc01 ~]# virsh autostart node1

5）查看虚拟机网卡信息

[root@room9pc01 ~]# virsh domiflist node1

接口 类型 源 型号 MAC

vnet0 bridge vbr virtio 52:54:00:5f:8b:11

vnet1 bridge private1 virtio 52:54:00:19:a4:f3

6）查看虚拟机硬盘信息

[root@room9pc01 ~]# virsh domblklist node1

目标 源

vda /var/lib/libvirt/images/node1.img

vdb /var/lib/libvirt/images/disk.img

##### 4.常用的镜像盘的类型



##### 5.qemu-img命令

* Qemu-img是虚拟机的磁盘管理命令,支持多种磁盘格式,(raw,qcow2.vdi.vmdk)
* Qemu-img命令格式
* **qemu-img 命令 参数 块文件名称 大小**

**qemu-img create -f qcow2 -b 原始盘 前端盘 前端盘大小**

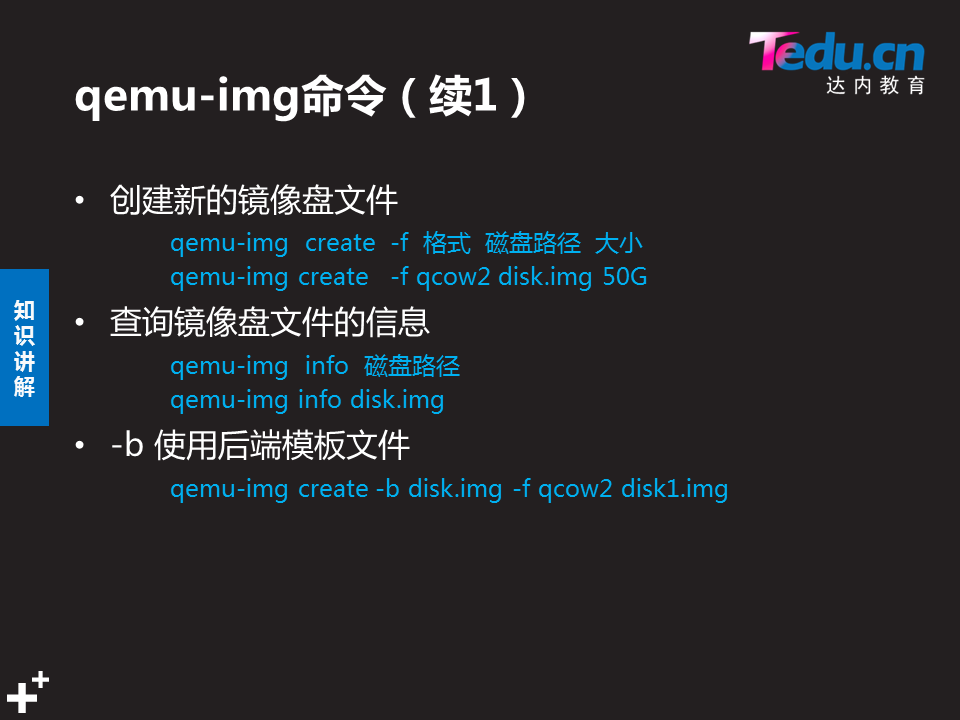
* 常用的命令

**-create 创建一个磁盘**

**-convert 转换磁盘格式**

**-info查看磁盘信息**

**-resize 扩展磁盘空间**



[root@room9pc01 ~]# **du -sh /var/lib/libvirt/images/tedu\_node46.img**

20M /var/lib/libvirt/images/tedu\_node46.img **#当前大小**

[root@ ~]# **qemu-img info /var/lib/libvirt/images/tedu\_node46.img**

**查看磁盘信息**

image: /var/lib/libvirt/images/tedu\_node46.img

file format: qcow2

virtual size: 20G (21474836480 bytes) #实际容量大小

**backing file: /var/lib/libvirt/images/.node\_base.qcow2(后端盘位置)**

###### 一：创建一个新的镜像盘文件(写时复制)

* **1).qemu-img命令格式：qemu-img 命令 参数 块文件名称 大小**

[root@room9pc01 ~]# **qemu-img create -f qcow2 disk.img 50G**

**//qcow2为创建的格式**

Formatting 'disk.img', fmt=qcow2 size=53687091200 encryption=off cluster\_size=65536 lazy\_refcounts=off

* **2）使用后端模板文件创建一个新的镜像盘文件 备注：-b使用后端模板文件**

[root@room9pc01 ~]# **qemu-img create -b disk.img -f qcow2 disk1.img**

Formatting 'disk1.img', fmt=qcow2 size=53687091200 backing\_file='disk.img' encryption=off cluster\_size=65536 lazy\_refcounts=off

* **3）使用后端模板文件创建一个16G的镜像盘文件**

[@room9pc1 ~]# **qemu-img create -b disk.img -f qcow2 disk2.img 16G**

Formatting 'disk1.img', fmt=qcow2 size=53687091200 backing\_file='disk.img' encryption=off cluster\_size=65536 lazy\_refcounts=off

##### 6.cow技术原理(写时复制)



* **案例: #虚拟机删根后在恢复虚拟机系统**

**将tedu\_node46主机的根/删掉**

**[root@B ~]# rm -rf /\***

**用45的后端模板创建新的46磁盘镜像**

**[student@room9pc01 ~]$ cd /var/lib/libvirt/images/**

[student@ images]$ **qemu-img create -f qcow2 -b .node\_base.qcow2 tedu\_node46.img**

[student@room9pc01 images]$ **virsh destroy tedu\_node4**

域 tedu\_node46 被删除

[student@room9pc01 images]$ **virsh start tedu\_node46**

域 tedu\_node46 已开始

[student@room9pc01 images]$ **virsh console tedu\_node46**

连接到域 tedu\_node46

换码符为 ^]

### virsh管理

#### .xml管理

##### 修改xml文件

对虚拟机的配置进行调整

编辑格式: **virsh edit 虚拟机名**

###### 1).修改CPU/memory

**修改前**

[root@a ~]# lscpu

Byte Order: Little Endian

CPU(s): 2

[root@a ~]# free -h(易读单位)

total used free

Mem: 1488282

[student@room9pc01 ~]$ **virsh edit A云**

<domain type='kvm'>

<name>A云</name>

<uuid>2a9c55fd-9134-42d3-a8c7-32bdea4361b1</uuid>

**<memory unit='KiB'>1188282</memory> #修改内存**

<currentMemory unit='KiB'>1188282</currentMemory>

**<vcpu placement='static'>6</vcpu> #修改cpu个数为6个**

**重置:**

[student@room9pc01 ~]$ virsh destroy A云

域 A云 被删除

[student@room9pc01 ~]$ virsh start A云

域 A云 已开始

**修改后:**

[root@a ~]# lscpu

Byte Order: Little Endian

CPU(s): 6

[root@a ~]# free

total used free

Mem: 1188282

###### 2).修改磁盘/网络

[student@room9pc01 ~]$ **virsh edit A云**

<disk type='file' device='disk'>

<driver name='qemu' type='qcow2'/>

<source file='/**var/lib/libvirt/images/tedu\_node45.img'**/> #磁盘

<interface type='bridge'>

<mac address='52:54:00:be:75:d1'/>

<source **bridge='vbr'/**> #网络

<model type='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>

student@room9pc01 ~]$ virsh destroy A云

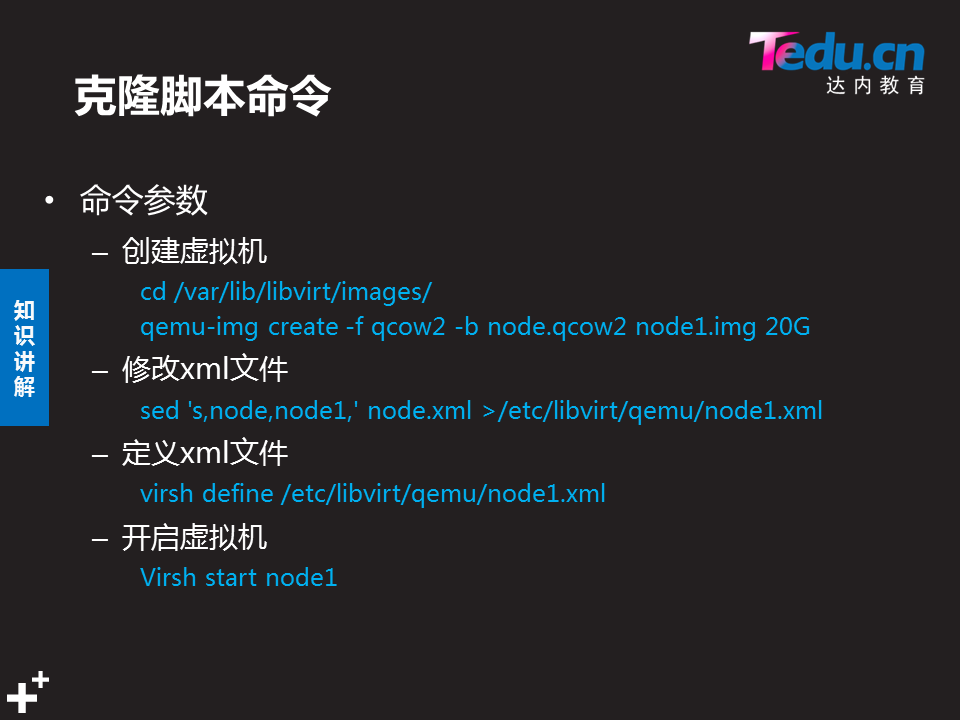
域 A云 被删除

[student@room9pc01 ~]$ virsh start A云

域 A云 已开始

#### .批量创建虚拟机

##### 克隆脚本命令



[student@room9pc01 images]$ ls -a .\*

.node\_base.xml **(xml模板文件)**

**创建**

[student@room9pc01 images]$ cd /var/lib/libvirt/images/

[s@room9pc01 images]$ **qemu-img create -f qcow2 -b .node \_base.qcow2 nsd1903.img 30G #创建nsd1903磁盘文件**

[@mages]]$ **cp .node\_base.xml /etc/libvirt/qemu/nsd1903.xml**

**#创建nsd1903xml文件**

**定义nsd1903.xml文件**

[student@room9pc01 qemu]$ vim nsd1903.xml

<name>**nsd1903**</name> **#修改name**

<source file='/var/lib/libvirt/images**/nsd1903.img**'/> **#修改磁盘文件**

**定义xml文件**

[student@room9pc01 qemu]$ virsh define nsd1903.xml

定义域 nsd1903（从 nsd1903.xml）

[student@room9pc01 qemu]$ virsh start nsd1903

域 nsd1903 已开始

**登录:**

[student@room9pc01 qemu]$ virsh console nsd1903

连接到域 nsd1903

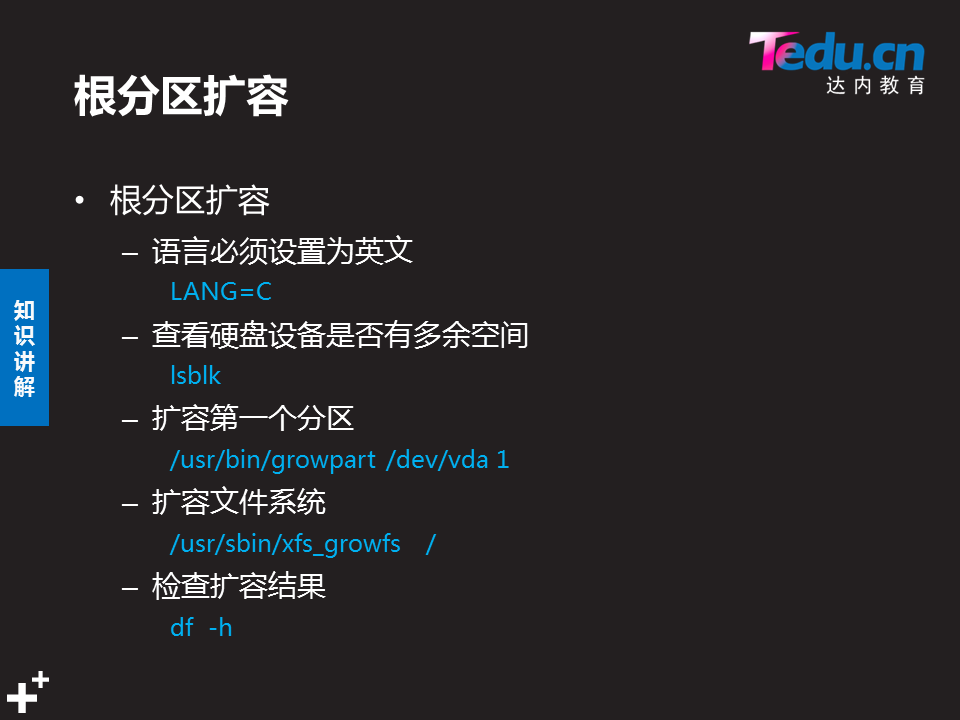
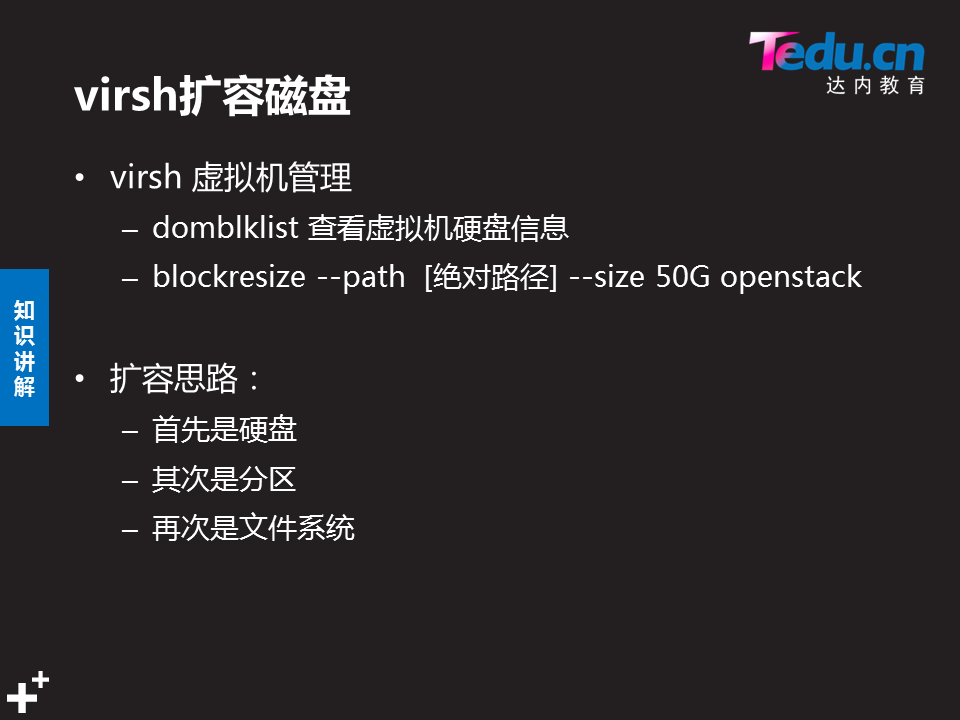
换码符为 ^]

##### 2.网卡及**配置文件**



**NetworkManager 工作在rhel7 network工作在rhel6**

##### Virsh动态扩容磁盘



* Virsh虚拟机管理

--domblklist 查看虚拟机硬盘信息

**格式: virsh blockresize --path [绝对路径] --size 大小G openstack(名字)**

**扩容思路:**

* **首先扩容硬盘**
* **在扩展分区**
* **文件系统扩容**

###### 1).扩容硬盘

**动态扩容前**

[root@a ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 20G 0 disk

└─vda1 253:1 0 2G 0 part /

**[student@room9pc01 ~]$ virsh domblklist A云**

目标 源

------------------------------------------------

vda /var/lib/libvirt/images/tedu\_node45.img

**[student@room9pc01 ~]$ virsh blockresize --path /var/lib/libvirt/images/tedu\_node45.img --size 50G A云(待扩展主机名)**

重新定义 '/var/lib/libvirt/images/tedu\_node45.img' 块设备大小

###### **2).扩容分区**

**[root@a ~]# lsblk**

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 50G 0 disk

└─vda1 253:1 0 **2G** 0 part /

[root@a ~]# **LANG=en growpart /dev/vda 1**

CHANGED: partition=1 start=2048 old: size=4192256 end=4194304

**[root@a ~]# lsblk**

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 50G 0 disk

└─vda1 253:1  **0 50G**  0 part /

###### 3).扩容文件系统

**[root@a ~]# df -h**

Filesystem Size Used Avail Use% Mounted on

/dev/vda1  **2.0G** 950M 1.1G 47% / **#文件系统未扩容**

**[root@a ~]# xfs\_growfs /dev/vda1**

[root@a ~]# **df -h**

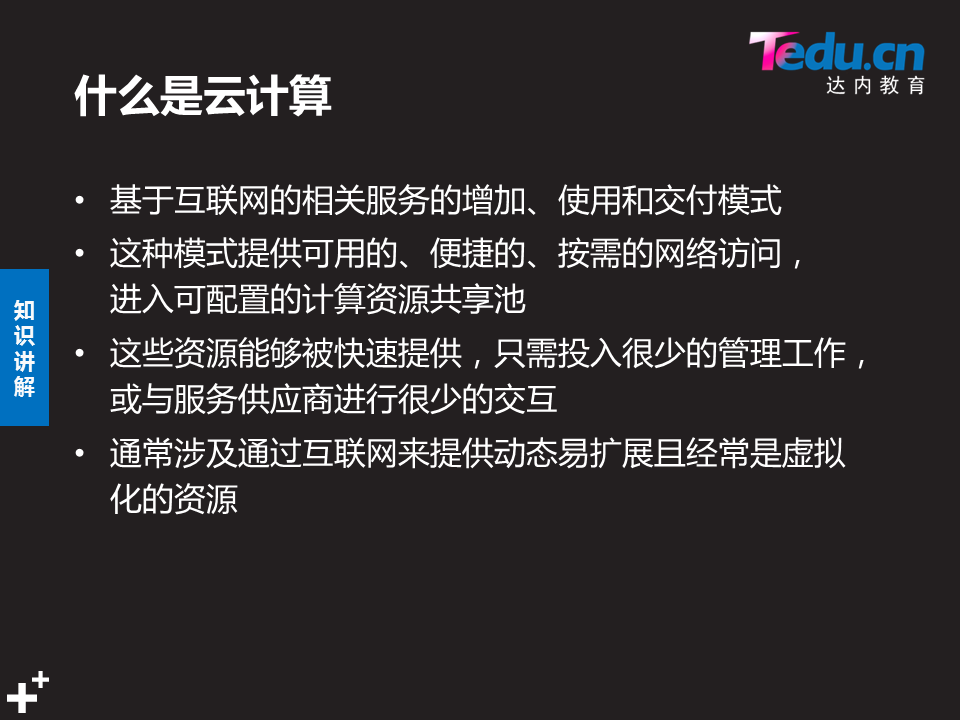
Filesystem Size Used Avail Use% Mounted on

/dev/vda1 50G 953M **50G**  2% / **文件系统扩容**

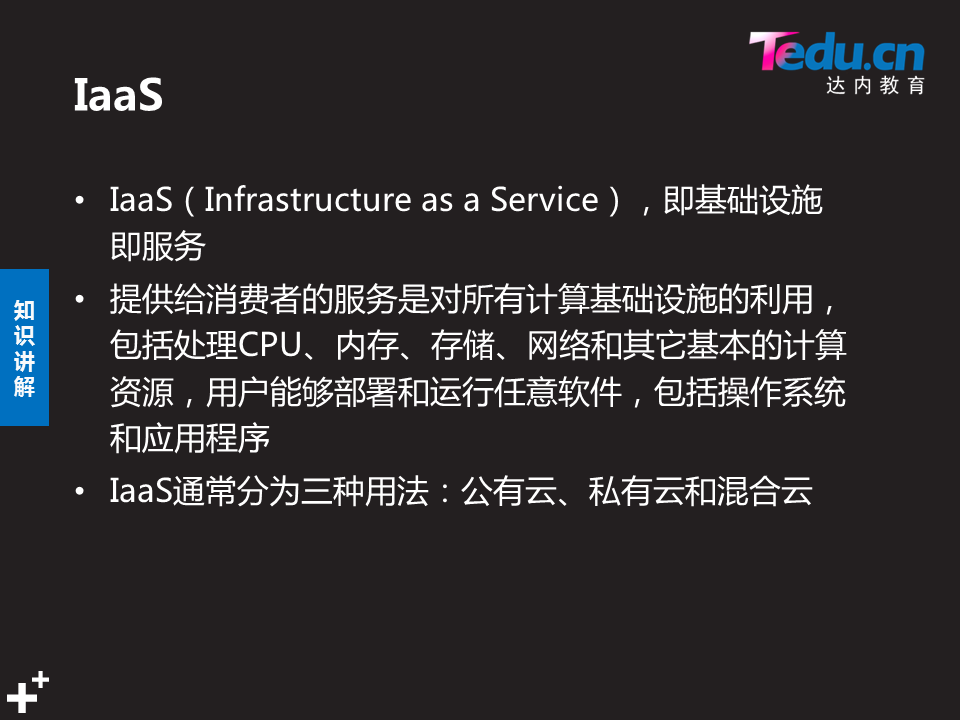
### 云平台

#### .云计算基础

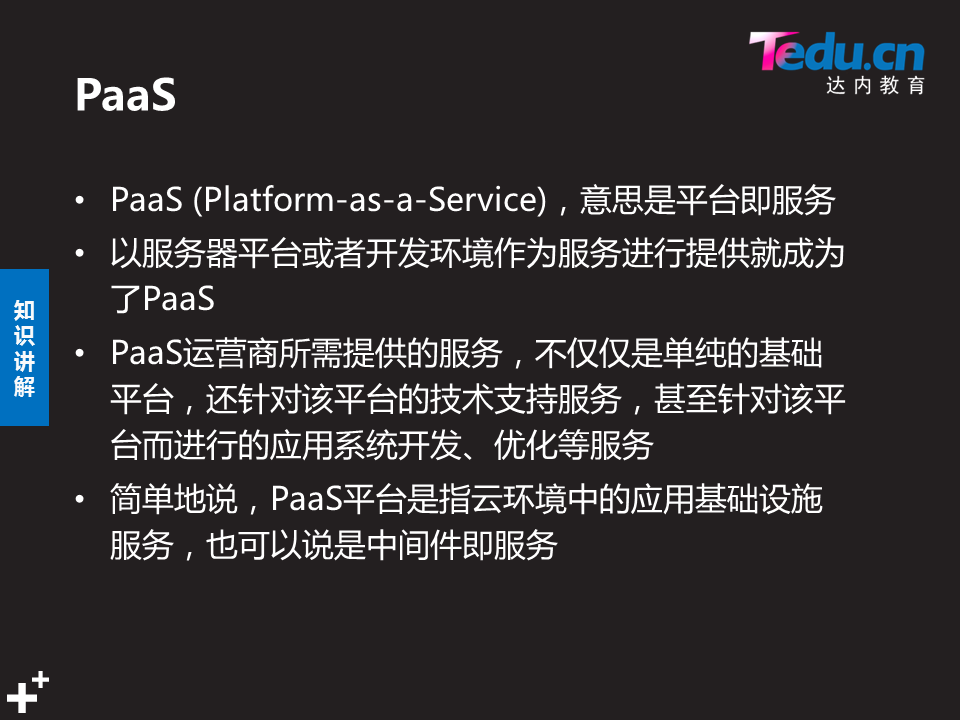
##### 什么是云计算



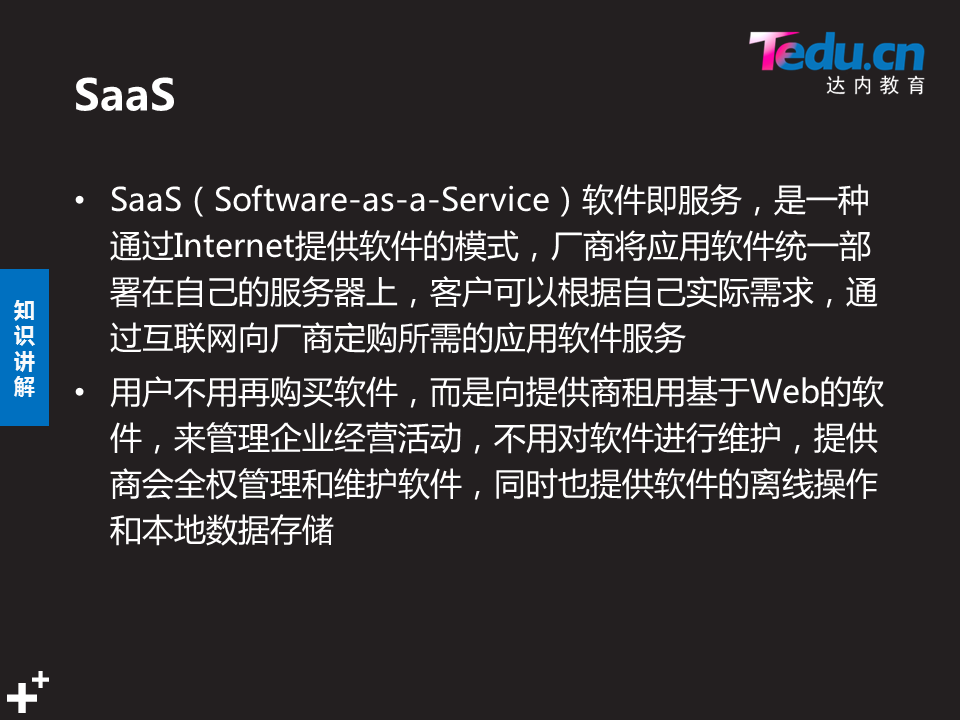
##### 2.Iaas(百度.阿里等)/基础设施云



##### 3.Paas(类似与淘宝平台)/平台云



##### 4.SaaS/软件云

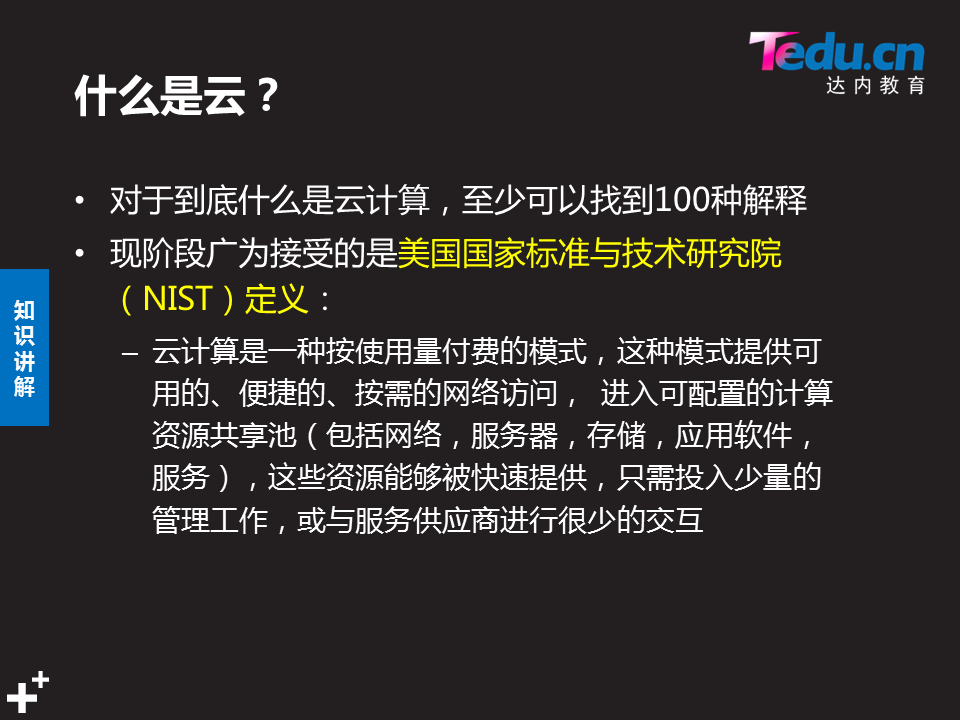


#### .openstack简介

##### 什么是Openstack



##### 什么是云



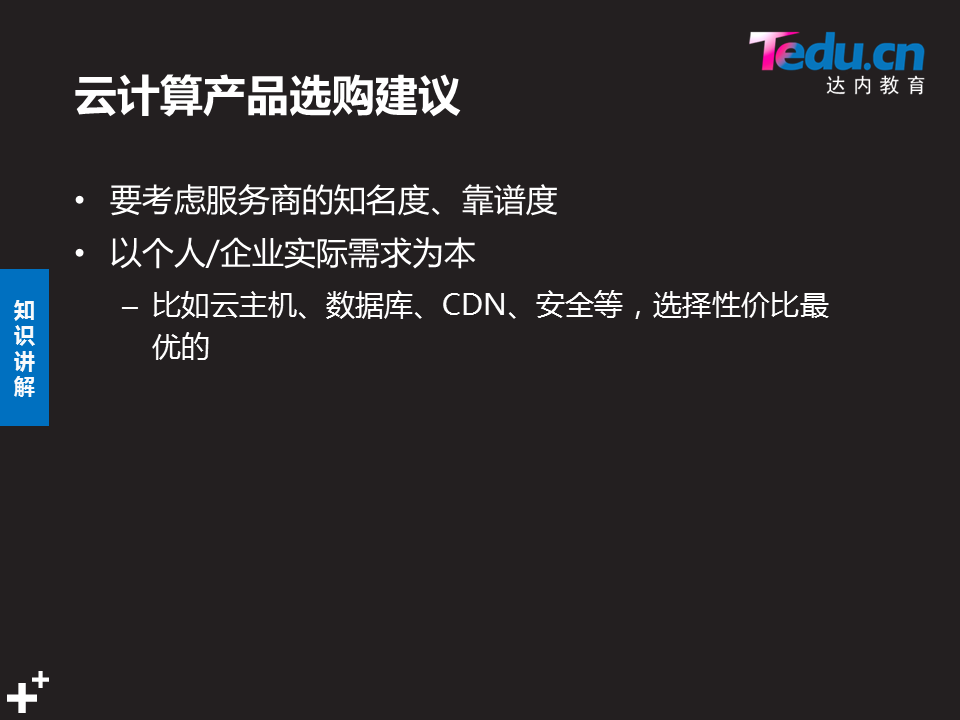
##### 3.云计算三大服务模式



##### 4.知名云服务商



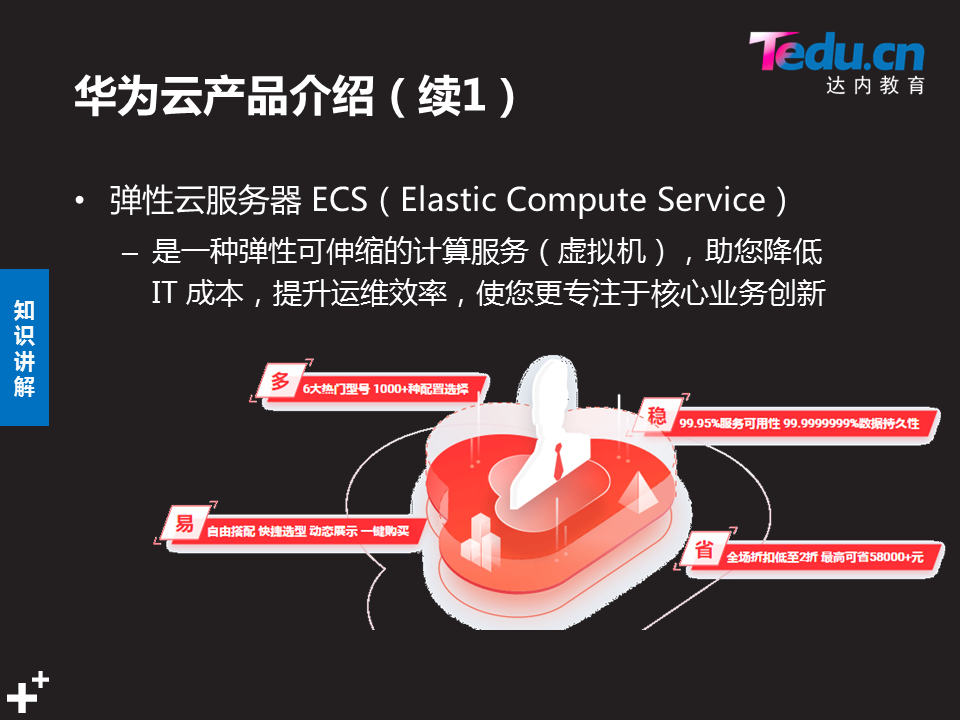
##### 云产品选购建议



### 华为云(ecs云服务器管理)

* **华为云产品介绍**





#### .华为云控制台

**华为云在openstack基础上二次开发(openstack云的代码贡献率除了openstack自己,华为第一)**

* **注册华为云用户**

本案例要求：

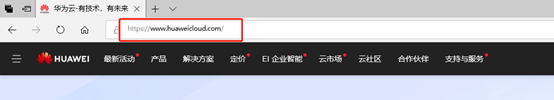
访问官网 https://huaweicloud.com/

注册华为云用户（需手机号验证）

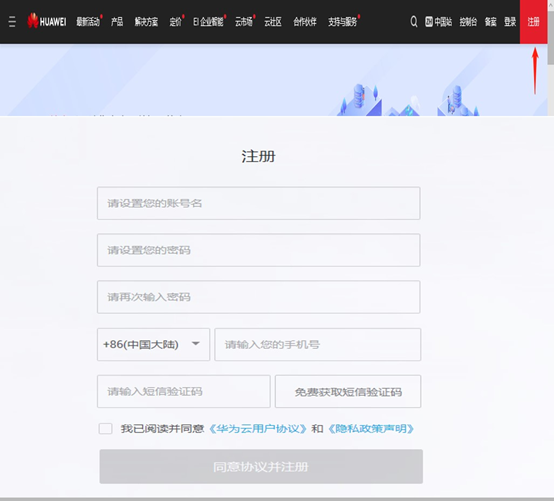
登录并完成实名认证

为账号充值不少于100元（不用时可提现）

##### 1）访问官网，如图-1所示：

****

##### 2）注册用户，如图-2所示：

****

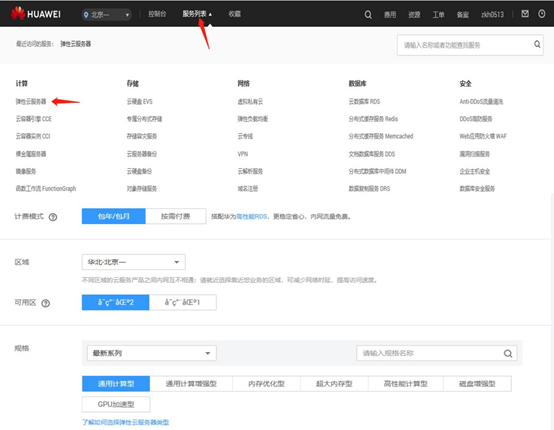
##### 3）登陆并完成实名认证，如图-3所示：

****

#### 二).ECS选购及操作

* 选购一台ECS云服务器
* 按需付费、通用计算型1vCPUs/1GB、硬盘40GB
* 独享带宽按流量、镜像选CentOS 7.4 x64
* 用户 root，密码 tedu.cn1234
* 通过“远程登录”进入此ECS云服务器的系统

##### 1）选购一台云服务器，如图-4所示：



##### 2）结果如图-5所示



##### 3）ECS基本操作，如图-6所示：



### 五.云服务器web建站

* 在ECS云服务器上启用 httpd 服务
* 测试网页

#### 1）在ECS云服务器Web建站上启用 httpd 服务

[root@svr1 ~]# yum -y install httpd //安装httpd

.. ..

[root@svr1 ~]# systemctl restart httpd //启动服务

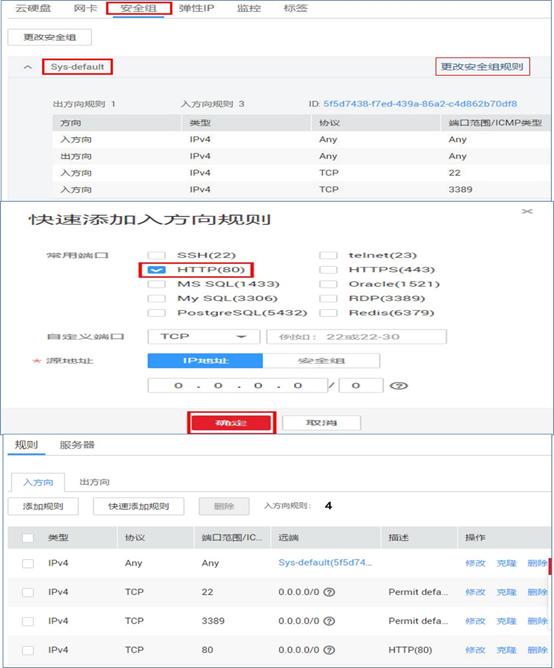
[root@svr1 ~]# systemctl enable httpd //设开机自运行

[root@svr1 ~]# systemctl status httpd //检查服务状态

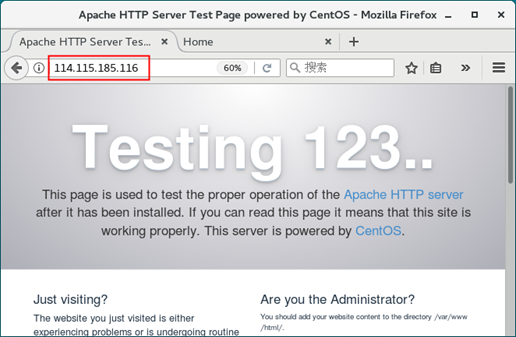
.. .. Active: active (running) .. .. //正在运行中

#### 2）ECS实例需开放Web服务端口

* 备注：22端口：Linux服务器远程控制
* 80端口：普通网站服务
* 443端口：加密网站服务



#### 3）测试页面，如图-8所示：



## DAY02 OPEnstack概述 部署安装环境 部署openstack openstack操作基础

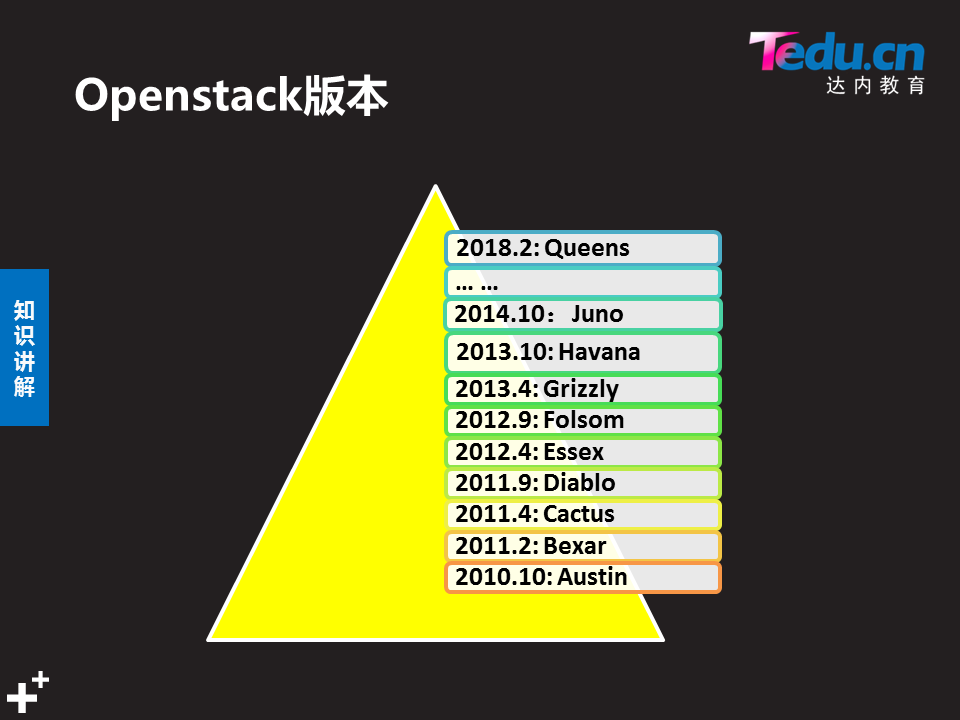
### 一.openstack概述

#### 1.什么是Openstack

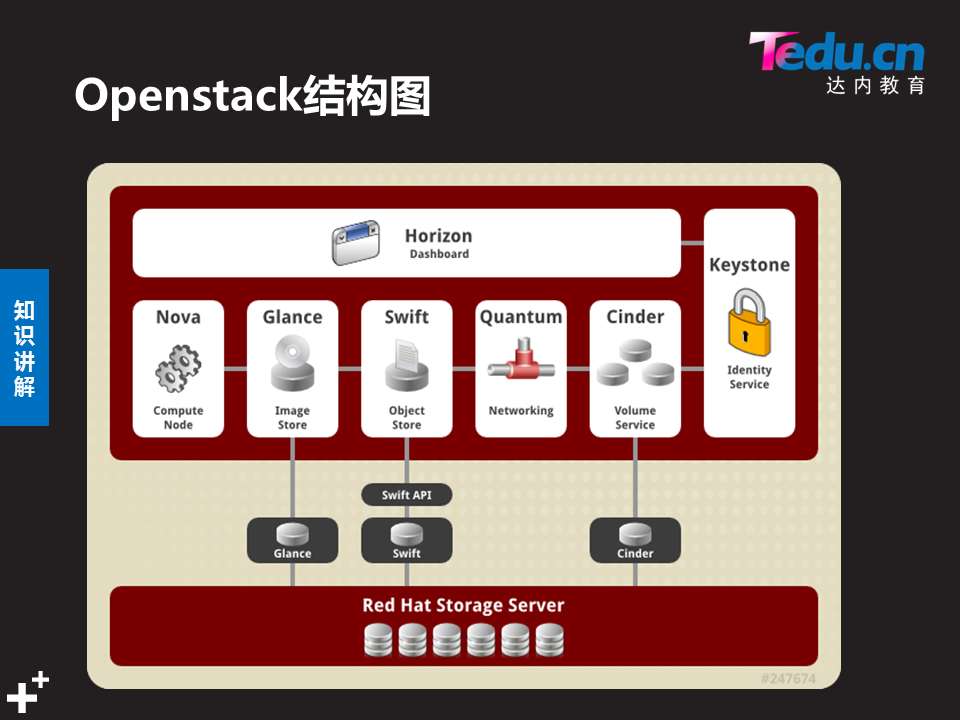


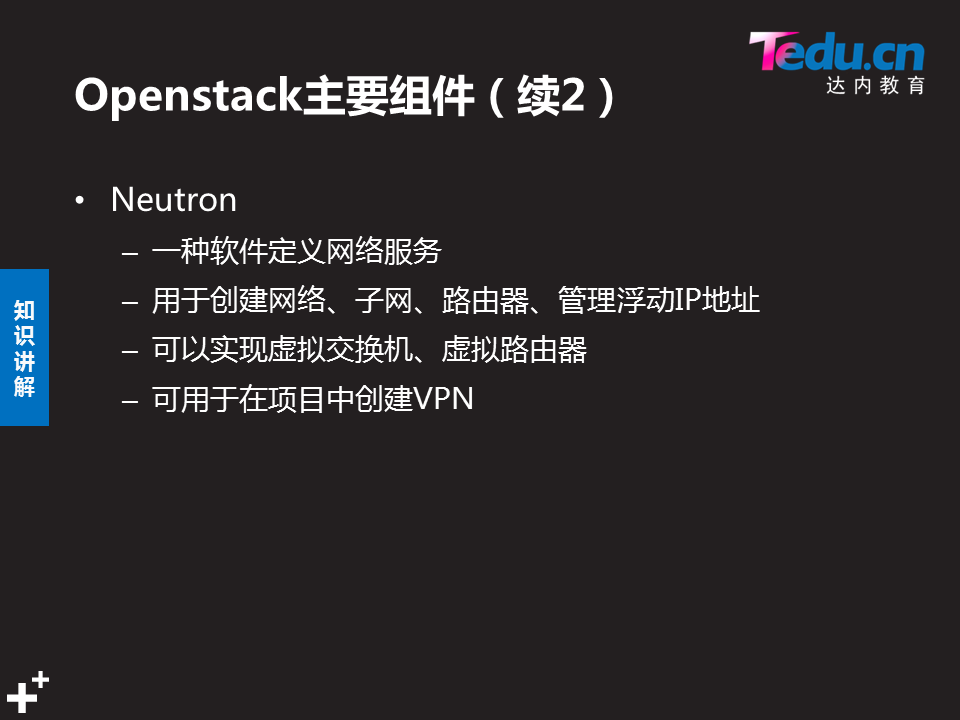
由Python开发

#### openstack版本

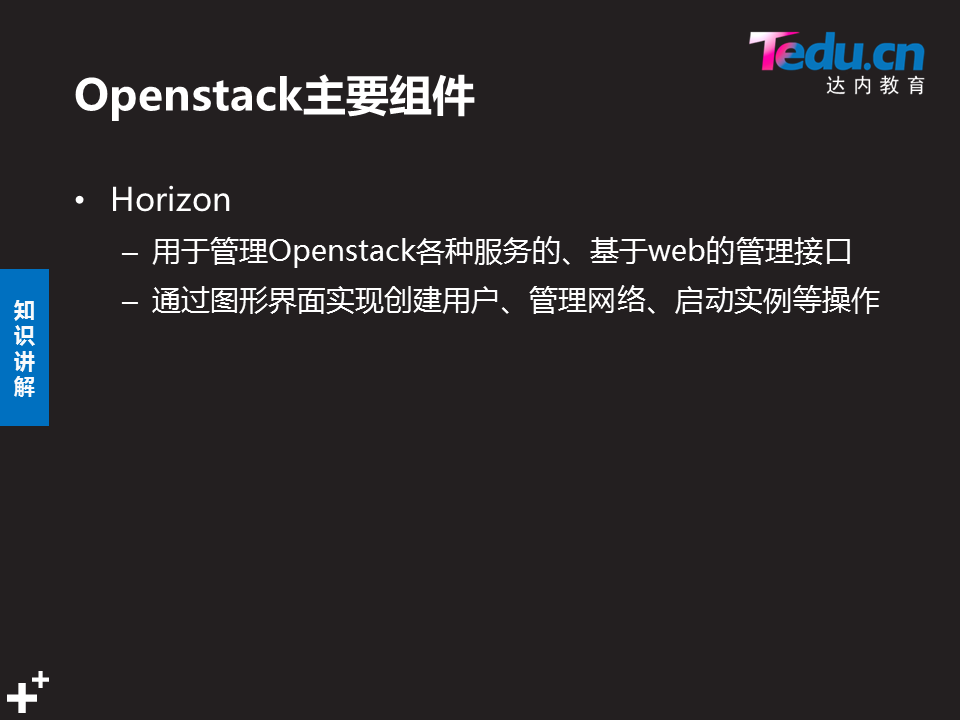


#### openstack结构

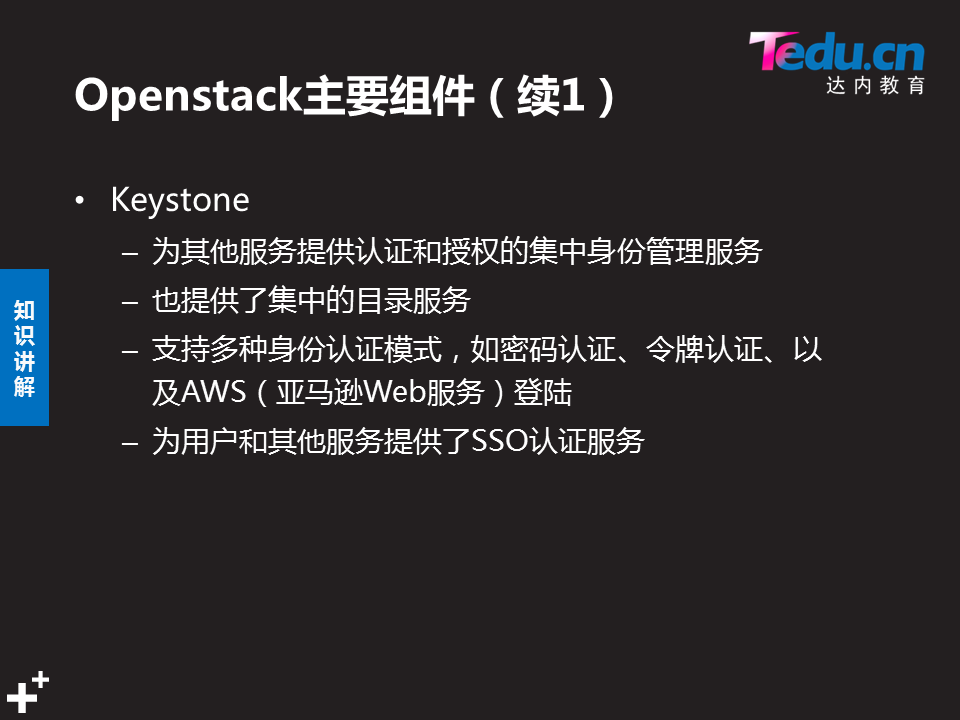




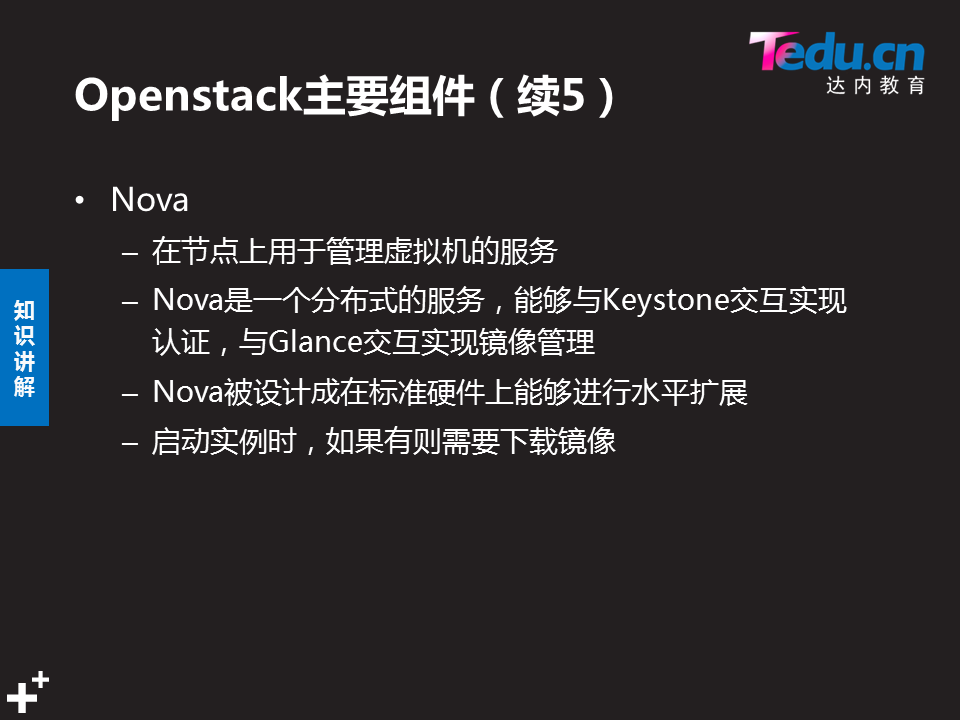
* **Horizon:提供和web页面基于Apache(管理节点)**



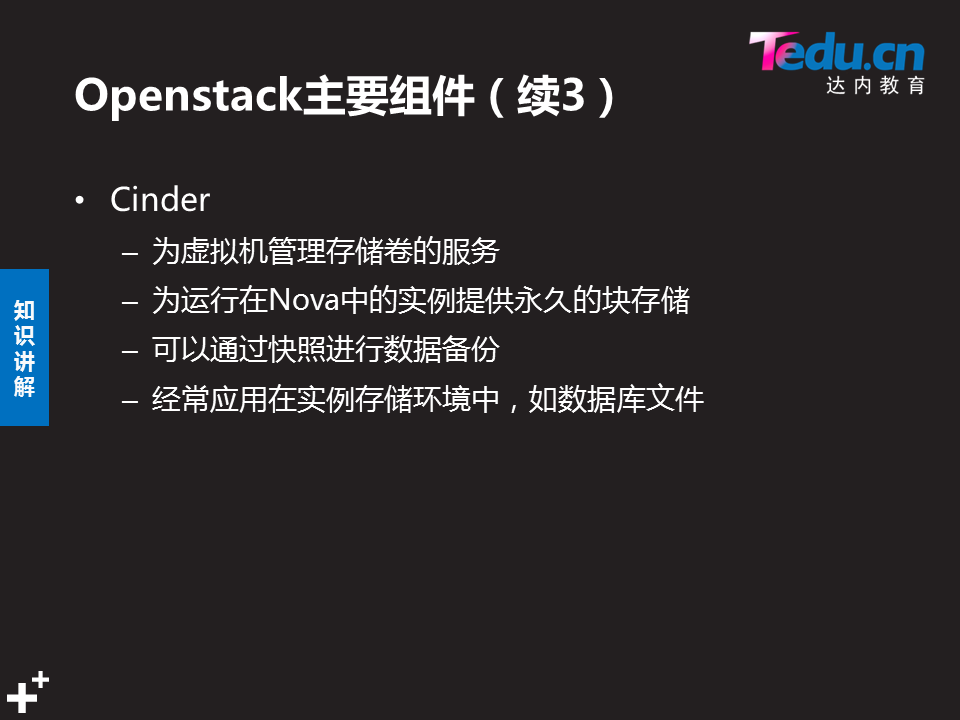
* **Keystone:实现所有机器和组件之间的统一认证**

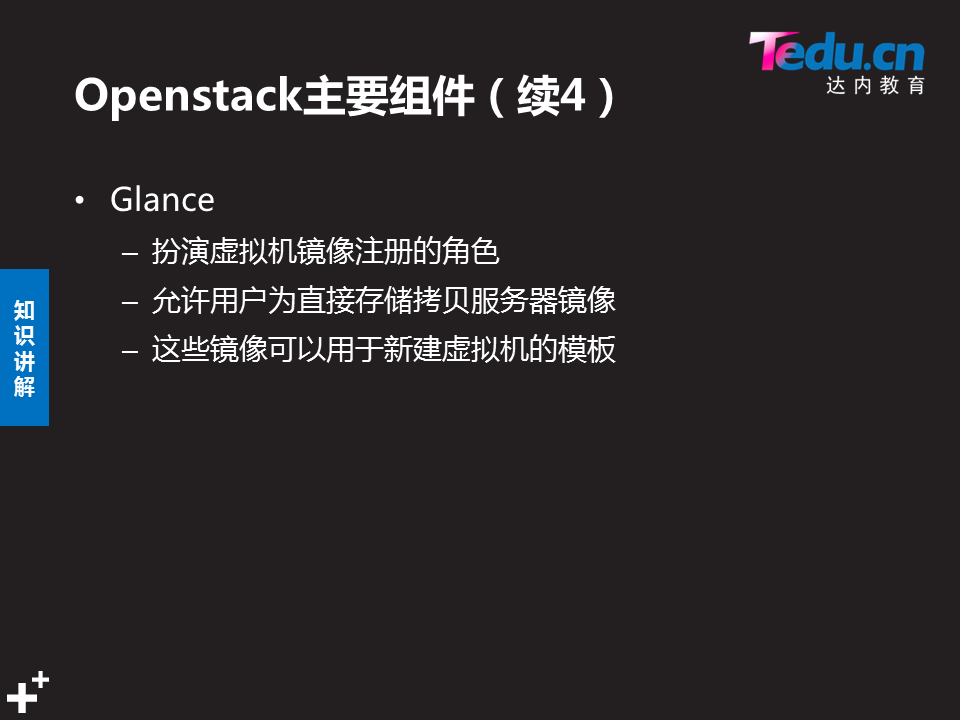


* **Nova:为web的master提供虚拟机的节点平台,有多少台虚拟机就得有多少个Nova组件**



* **Glance:磁盘镜像组件,类似于yum的功能,只要将后端盘放入glance,所有的主机都可以用其应用软件了**
* **cinder卷存储**





* **Swift:对象存储**
* **Quantum:网络组件**

### 部署openstack安装环境

#### 1.准备虚拟机

**要求配置基本环境：**

**配置三台虚拟机**

* Openstack管理主机

openstack : 192.168.1.10

2CPU，8G 内存，50G 硬盘

* nova01 和 nova02作为额外节点

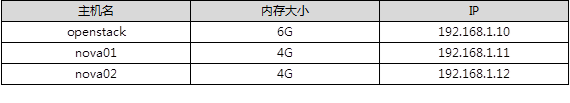
配置静态IP ifcfg-eth0

2CPU，4.5G 内存，100G 硬盘

nova: 192.168.1.11, 192.168.1.12

配置主机名 /etc/hosts，能够相互 ping 通

配置 dns 服务器 /etc/resolv.conf



##### 1.配置openstack

[root@localhost ~]# eip 10

[root@localhost ~]# hostnamectl set-hostname openstack

[root@openstack ~]# reboot

[root@localhost ~]# **cd /etc/sysconfig/network-scripts/ ifcfg-eth0**

# Generated by dracut initrd

DEVICE="eth0"

ONBOOT="yes"

NM\_CONTROLLED="no"

TYPE="Ethernet"

BOOTPROTO="static"

PERSISTENT\_DHCLIENT="yes"

IPADDR=192.168.1.10

NEMASK=255.255.255.0

GATEWAY=192.168.1.254

[student@room9pc01]$ cd /var/lib/libvirt/images/

[student@room9pc01]$ virsh blockresize --path tedu\_node47.img --size 100G tedu\_node47

[root@openstack ~]# LANG=en growpart /dev/vda 1

[root@openstack ~]# xfs\_growfs /dev/vda1

[student@room9pc01 images]$ **virsh edit tedu\_node47**

<memory unit='KiB'>9000000</memory>

<currentMemory unit='KiB'>90000000</currentMemory>

<vcpu placement='static'>2</vcpu>

[student@room9pc01 images]$ lscpu

CPU(s): 4

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/vda1 100G 917M 100G 1% /

##### 2.配置node01 node02

[root@localhost ~]# eip 11

[root@localhost ~]# hostnamectl set-hostname node01

[root@localhost ~]#reboot

[student@room9pc01]$ virsh blockresize --path /var/lib/libvirt/images/tedu\_node48.img --size 100G tedu\_node48

[root@node01 ~]# LANG=en growpart /dev/vda 1

[root@node01 ~]# xfs\_growfs /dev/vda1

[student@room9pc01 images]$ virsh edit tedu\_node48

<memory unit='KiB'>5000000</memory>

<vcpu placement='static'>2</vcpu>

[student@room9pc01 images]$ virsh destroy tedu\_node48

域 tedu\_node48 被删除

[student@room9pc01 images]$ virsh start tedu\_node48

域 tedu\_node48 已开始

* Node02

[root@localhost network-scripts]# cd /etc/sysconfig/network-scripts/

[root@localhost network-scripts]# vim ifcfg-eth0 #配置文件修改ip

# Generated by dracut initrd

DEVICE="eth0"

ONBOOT="yes"

NM\_CONTROLLED="no"

TYPE="Ethernet"

BOOTPROTO=**"static" #修改为static静态**

PERSISTENT\_DHCLiIENT="yes"

IPADDR=192.168.1.12

NEMASK=255.255.255.0

GATEWAY=192.168.1.254

[root@localhost ]# hostnamectl set-hostname node02

[student@room9pc01 images]$ virsh edit tedu\_node49

编辑了域 tedu\_node49 XML 配置。

<memory unit='KiB'>5000000</memory>

<currentMemory unit='KiB'>5000000</currentMemory>

<vcpu placement='static'>2</vcpu>

[student@room9pc01 images]$ virsh destroy tedu\_node49

域 tedu\_node49 被删除

[student@room9pc01 images]$ virsh start tedu\_node49

域 tedu\_node49 已开始

* **时间服务器(一台为例)**

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

...

server 192.168.1.254 iburst

[root@openstack ~]# systemctl restart chronyd

[root@openstack ~]# chronyc sources -v

210 Number of sources = 1

.-- Source mode '^' = server, '=' = peer, '#' = local clock.

/ .- Source state '\*' = current synced, '+' = combined , '-' = not combined,

#### 2.配置yum仓库

* 三台主机配置yum源（以一台主机为例，共10670个软件包）挂载三个镜像

**/var/lib/libvirt/images/iso/CentOS7-1804.iso(系统软件)**

**/linux-soft/04/openstack/RHEL7-extras.iso (提供Python依赖包)**

**/linux-soft/04/openstack/RHEL7OSP-10.iso (openstack软件仓)**

[student@room9pc01 images]$ cd /var/ftp

[student@room9pc01 ftp]$ mkdir extras

[student@room9pc01 ftp]$ mkdir openstack

[student@room9pc01 ftp]$ mkdir centos-1804

[student@room9pc01 openstack]$ vim /etc/fstab

**/var/lib/libvirt/images/iso/CentOS7-1804.iso(系统软件)** /var/ftp/centos-1804 iso9660 defaults 0 0

[student@room9pc01 ~]$ cd /linux-soft/04/openstack/

RHEL7-extras.iso RHEL7OSP-10.iso

[student@room9pc01 openstack]$ ls

[@room9pc01 openstack]$ **mount RHEL7-extras.iso /var/ftp/extras/**

mount: /dev/loop1 写保护，将以只读方式挂载

[t@room9pc01 openstack]$ **mount RHEL7OSP-10.iso /var/ftp/openstack/**

mount: /dev/loop2 写保护，将以只读方式挂载

* **定义yum源仓库**

[root@ openstack]# **vim /etc/yum.repos.d/CentOS.repo**

[local\_repo]

name=CentOS-$releasever - Base

baseurl="ftp://192.168.1.254/centos-1804"

enabled=1

gpgcheck=1

[extras]

name=extras

baseurl=ftp://192.168.1.254/extras

enabled=1

gpgcheck=0

[package]

name=package

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-rpms

enabled=1

gpgcheck=0

[devtools]

name=devtools

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-devtools-rpms

enabled=1

gpgcheck=0

[root@openstack ~]# yum repolist

源标识 源名称 状态

devtools devtools 3

extras extras 76

local\_repo CentOS-7 - Base 9,911

package package 68

repolist: 10,670

* **远程同步到Nova01 和nova02下(远程同步不改变属性和权限,增量)**

[root@openstack ~]# rsync -avXSH /etc/yum.repos.d nova02:/etc/ (**X保留扩展权限,S保留软连接,H保留硬链接)**

[root@openstack ~]# rsync -av /etc/yum.repos.d/ nova01:/etc/yum.repos.d/

#### 3.设置三台DNS(一个为例)

[@openstack~]# vim /etc/hosts

#将openstack.tedu.cn域名对应的ip解析到我们安装的openstack服务器

192.168.1.10 openstack

192.168.1.11 nova01

192.168.1.12 nova02

[root@openstack ~]# vim /etc/resolv.conf

#openstack安装时需要使用外部dns来解析域名

; generated by /usr/sbin/dhclient-script

#search localhost //注释search开头的行

nameserver 192.168.1.254

### 部署openstack

#### 一).基础环境准备

##### 1.安装额外软件包（三台机器操作，这里以一台为例）

[root@openstack ~]# for i in 11 12 10

> do

> ssh -X root@192.168.1.$i "yum -y install qemu-kvm libvirt-client libvirt-daemon libvirt-daemon-driver-qemu python-setuptools"

> done

##### 检测是否卸载firewalld 和 NetworkManager

[root@openstack ~]# rpm -qa | grep NetworkManager\* #无如果则卸载

[root@openstack ~]# rpm -qa | grep firewalld\*

##### 3.检查配置主机网络参数

[root@openstack ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

# Generated by dracut initrd

DEVICE="eth0"

ONBOOT="yes"

NM\_CONTROLLED="no"

TYPE="Ethernet"

BOOTPROTO="static"

PERSISTENT\_DHCLIENT="yes"

IPADDR=192.168.1.10

NEMASK=255.255.255.0

GATEWAY=192.168.1.254

##### 验证主机名是否互通

[root@openstack ~]# ping openstack

...

64 bytes from openstack (192.168.1.10): icmp\_seq=1 ttl=255 time=0.023 ms

64 bytes from openstack (192.168.1.10): icmp\_seq=2 ttl=255 time=0.027 ms

...

[root@openstack ~]# ping nova01

PING nova01 (192.168.1.11) 56(84) bytes of data.

64 bytes from nova01 (192.168.1.11): icmp\_seq=1 ttl=255 time=0.139 ms

...

[root@openstack ~]# ping nova02

PING nova02 (192.168.1.12) 56(84) bytes of data.

64 bytes from nova02 (192.168.1.12): icmp\_seq=1 ttl=255 time=0.251 ms

##### 检测配置主机yum源

[root@openstack ~]# yum repolist

源标识 源名称 状态

RHEL7-extras RHEL7-extras 76

RHEL7OSP-devtools RHEL7OSP-devtools 3

RHEL7OSP-package RHEL7OSP-package 680

local\_repo CentOS-7 - Base 9,911

repolist: 10,670

##### 6.检查时间同步是否可用

[root@openstack ~]# **chronyc sources -v**

210 Number of sources = 1

|| | | \

MS Name/IP address Stratum Poll Reach LastRx Last sample

==============

^\* gateway 3 7 377 28 +31us[ +89us] +/- 25ms

##### 7.检查/etc/resolv.conf 不能有 search 开头的行

[root@openstack ~]# cat /etc/resolv.conf

; generated by /usr/sbin/dhclient-script

nameserver 192.168.1.254

#### .安装openstack

* **通过packstack部署Openstack**
* **根据相关日志文件进行排错**

##### 1.安装openstack需要的openstack-packstack

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

##### 2.使用openstack创建配置应答文件

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

**//answer.ini与answer.txt是一样的，只是用vim打开answer.ini文件有颜色**

Packstack changed given value to required value /root/.ssh/id\_rsa.pub

[root@openstack ~]# vim answer.ini

42 CONFIG\_SWIFT\_INSTALL=**n**   **#是否使用对象存储**

45 CONFIG\_CEILOMETER\_INSTALL=**n**  **//计费相关模块**

49 CONFIG\_AODH\_INSTALL=**n** **//计费相关模块**

53 CONFIG\_GNOCCHI\_INSTALL=**n**   **//计费相关模块**

75 CONFIG\_NTP\_SERVERS=**192.168.1.254**  **//时间服务器的地址**

98 CONFIG\_COMPUTE\_HOSTS=**192.168.1.11** **#在哪个机器上安装neutron服务**

102 CONFIG\_NETWORK\_HOSTS=**192.168.1.10,192.168.1.11**

**#在哪个机器上配置neutron网络**

333 CONFIG\_KEYSTONE\_ADMIN\_PW=**a**   **//修改管理员的密码**

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

**//驱动类型(设置neutron需要什么样的网络协议)**

#876 CONFIG\_NEUTRON\_ML2\_VXLAN\_GROUP=239.1.1.5

**//设置组播地址,最后一个随意不能为0和255,其他固定**

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

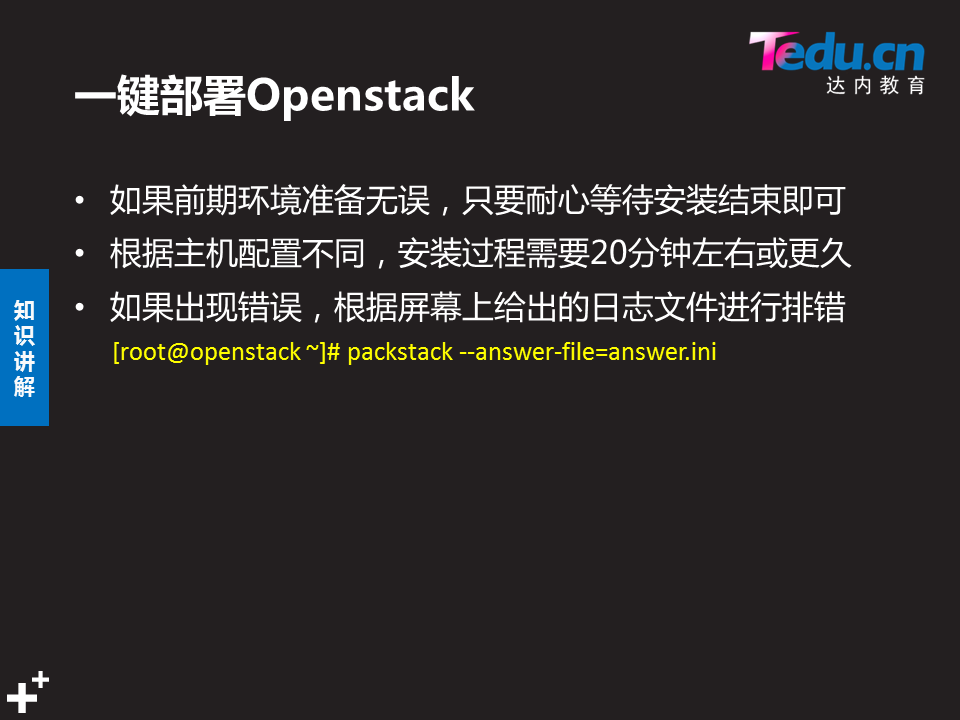
**//物理网桥的名称(配置虚拟交换机)**

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

**//br-ex桥的名称与eth0连接，管理eth0,网桥与哪个物理网卡连接**

1179 CONFIG\_PROVISION\_DEMO=**n**  **//DEMO是否测试(不使用daemon进行测试)**

##### 一键部署openstack



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

.........

Applying 192.168.1.10\_controller.pp

192.168.1.10\_controller.pp: [ DONE ]

Applying 192.168.1.11\_network.pp

Applying 192.168.1.10\_network.pp

192.168.1.10\_network.pp: [ DONE ]

192.168.1.11\_network.pp: [ DONE ]

Applying 192.168.1.11\_compute.pp

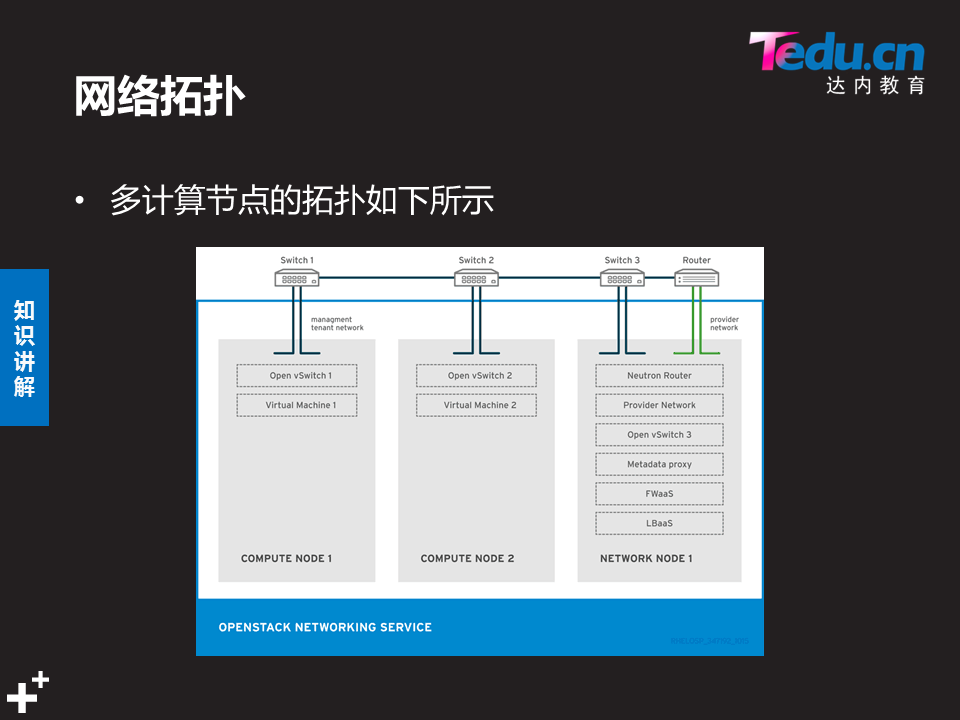
192.168.1.11\_compute.pp: [ DONE ]

Applying Puppet manifests [ DONE ]

Finalizing [ DONE ]

\*\*\*\* Installation completed successfully \*\*\*\*\*\*

#### .网络配置



* **查看外部OVS网桥**

##### 查看br-ex网桥配置（br-ex为OVS网桥设备:交换机）

[root@openstack ~]# **ifconfig**

br-ex: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet **192.168.1.10** netmask 255.255.255.0 broadcast 192.168.1.255

[root@openstack ~]# **cat /etc/sysconfig/network-scripts/ifcfg-br-ex**

ONBOOT="yes"

NM\_CONTROLLED="no"

IPADDR=**"192.168.1.10"**

**PREFIX=24**

**GATEWAY=192.168.1.254**

DEVICE=br-ex

NAME=br-ex

DEVICETYPE=ovs

OVSBOOTPROTO="static"

TYPE=OVSBridge

##### 2）查看eth0网卡配置（该网卡为OVS网桥的接口）

[root@openstack ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

NAME=eth0

DEVICETYPE=ovs

TYPE=OVSPort

OVS\_BRIDGE=br-ex

ONBOOT=yes

BOOTPROTO=none

##### 3）验证OVS配置

[root@openstack ~]# **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 phy-br-ex

Interface phy-br-ex

type: patch

options: {peer=int-br-ex}

Port "eth0"

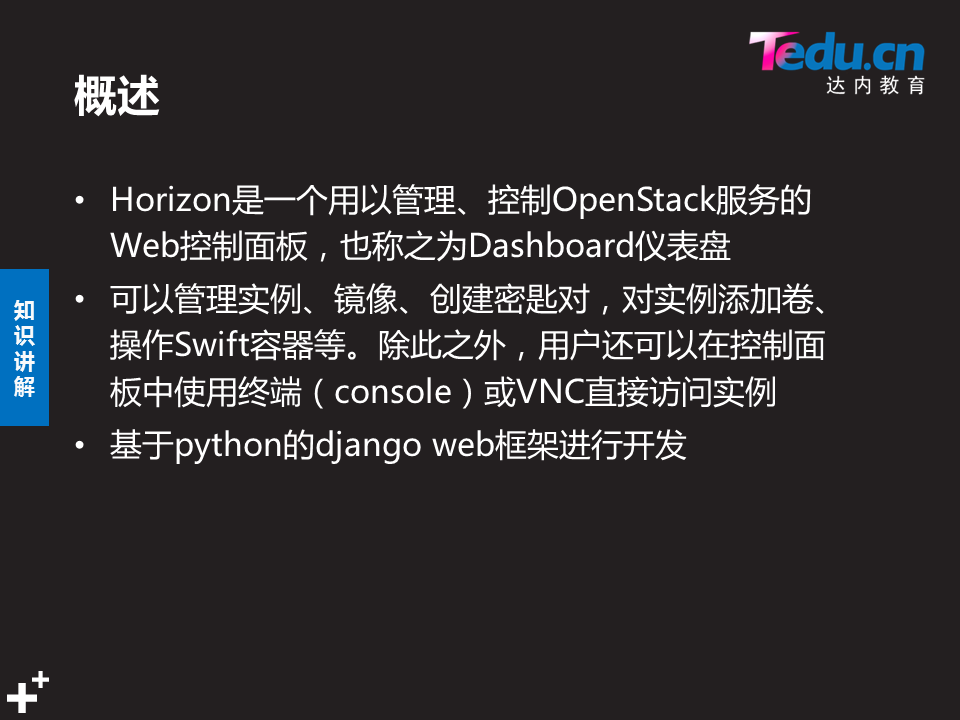
Interface "eth0"

ovs\_version: "2.5.0"

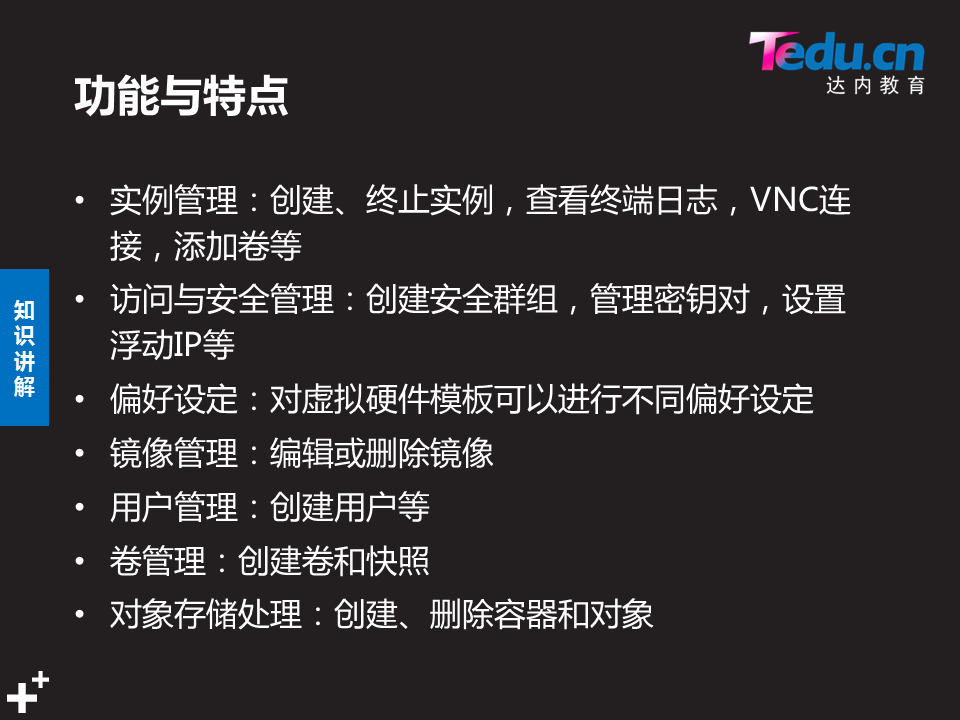
### openstack操作基础

#### 一).Horizon配置

##### 1.概述



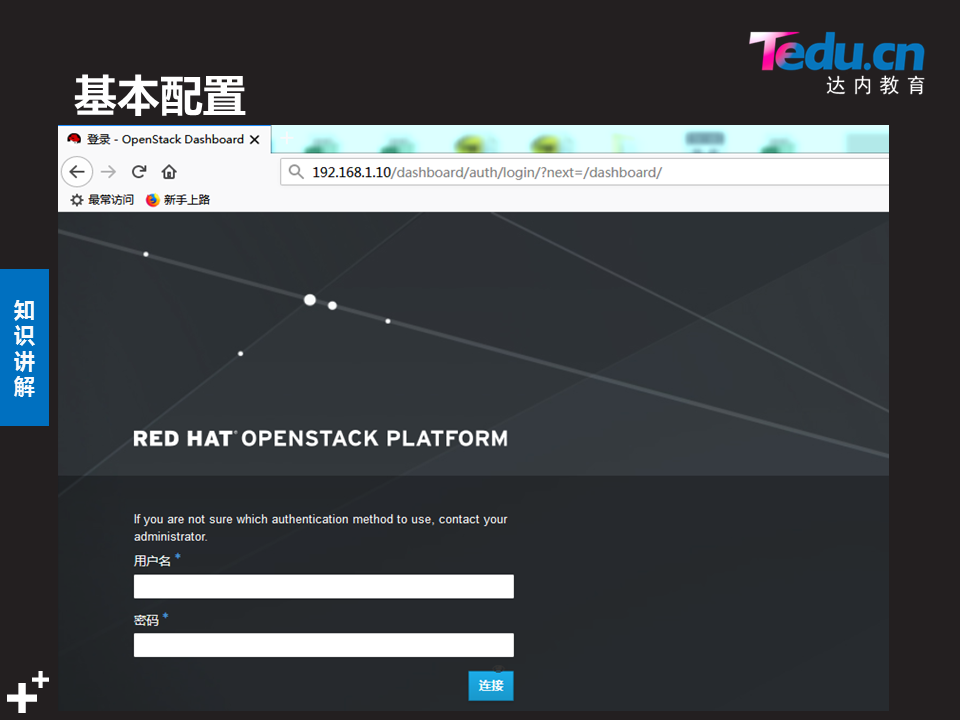
##### 功能与特点



##### Horizon



##### 基本配置



[root@openstack ~]# ls

answer.ini **keystonerc\_admin(记录云平台安装时后的用户名和密码)**

[root@openstack ~]# cat keystonerc\_admin

unset OS\_SERVICE\_TOKEN

export OS\_USERNAME=**admin //用户名**

export OS\_PASSWORD=**a //密码**

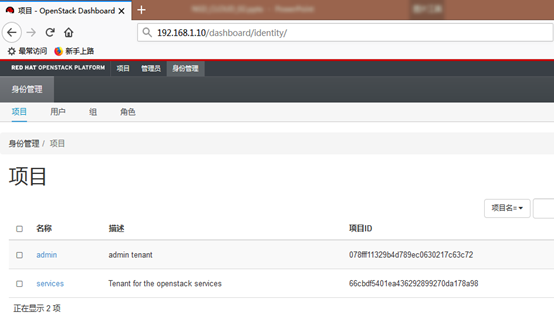
export OS\_AUTH\_URL=**http://192.168.1.10:5000/v2.0**

export PS1='[\u@\h \W(keystone\_admin)]\$ '

export OS\_TENANT\_NAME=admin

export OS\_REGION\_NAME=RegionOne

* **在火狐浏览器中输入用户名和密码，登录后页面**



#### .项目管理

##### 命令行接口基础初始化

[root@openstack ~]# source ~/keystonerc\_admin //初始化环境变量

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

##### 2.通过命令行管理项目

###### 1),创建名为myproject的项目

[roo@openstack ~(keystone\_admin)]# **openstack project create myproject**

| Field | Value |

| description | None |

| enabled | True |

| id |f73949e2701346328faf5c1272414cd0|

| name | myproject |

###### 2).查看项目信息

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

| ID | Name |

| 3fd5bffd734543d39cf742492273a47f | services |

| 7f0b43274a9f4e149357fc8393ff2c68 | admin |

| f73949e2701346328faf5c1272414cd0 | myproject |

###### 3).更新vcpu配额为30

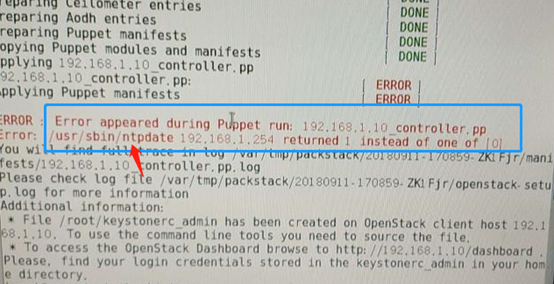
[root@openstack ~(keystone\_admin)]# **nova quota-update --cores 30 myproject**

###### 4).删除myproject

[root@openstack ~(keystone\_admin)]# **openstack project delete myproject**

### 五.安装openstack可能会出现的错误以及排错方法

#### 1.ntp时间不同步：



**解决办法：查看ntp时间服务器，是否出现\*号，若没有查看配置文件，配置ntp服务器**

[root@room9pc01 ~]# **chronyc sources -v** //出现\*号代表NTP时间可用

**^\*** 120.25.115.20 2 6 17 62 -753us[-7003us] +/- 24ms

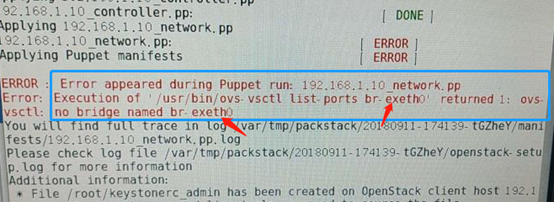
[root@openstack ~]# chronyc sources -v

^\* 192.168.1.254 3 9 377 504 +50us[-20us] +/- 24ms

[root@nova ~]# chronyc sources -v

**^\*** 192.168.1.254 3 9 377 159 -202us[-226us] +/- 24ms

#### 2.网桥名称写错：



**解决办法：检查配置文件**

[root@openstack ~]# **vim answer.ini**

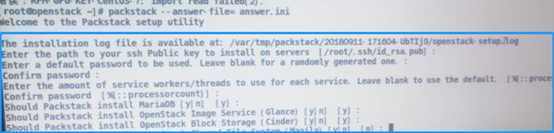
...

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

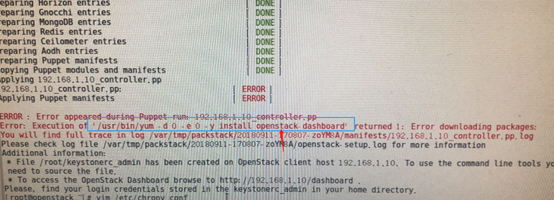
//br-ex桥的名称与eth0连接，管理eth0,网桥与哪个物理网卡连接

...

#### 3.若/root/.ssh/id\_rsa.pub，提示password，同样是配置文件没有写对

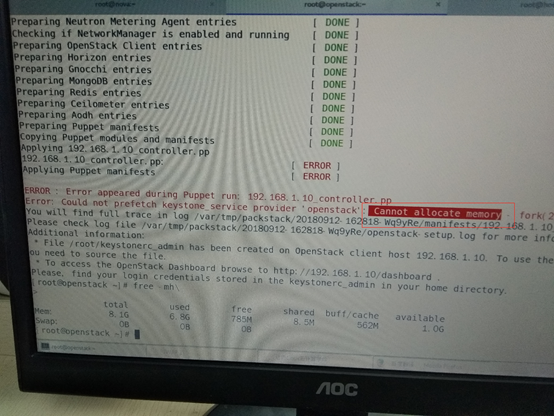


#### 4.yum源没有配置正确，如图-5所示：



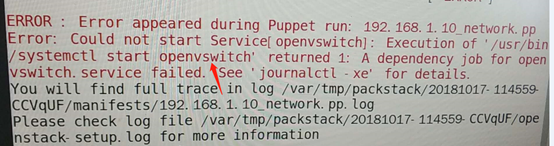
**解决办法：检查yum是否为10670个软件包，查看是否是yum源没有配置正确，之后安装oprnstack-dashboard**

#### 5.出现Cannot allocate memory,如图-6所示：



**解决办法：内存不足，重新启动主机**

#### 6.出现/usr/bin/systemctl start openvswith ... falied，说明是ssse3指令集的错误：



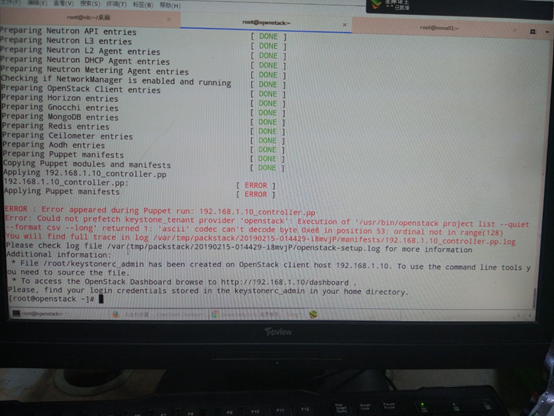
**解决办法：编辑openstack主机的xml文件，在里面添加**

[root@openstack ~]#virsh edit openstack

**<cpu mode='host-passthrough'>**

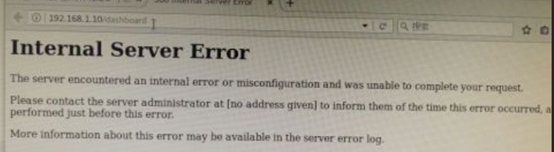
**</cpu>**

#### 7.若出现 Could not prefetch... ‘openstack’。 如图-8所示：



**配置文件里面有中文符号**

#### 8.访问openstack出错



**没有修改Apache配置文件**