[NTP时间同步服务器及客户端 1](#_Toc984929356)

[cron计划任务 2](#_Toc2072935194)

[FTP 3](#_Toc1383856877)

[NFS网络文件系统 4](#_Toc1445925741)

[配置聚合连接 5](#_Toc2105229996)

[ISCSI共享存储(网络硬盘) 6](#_Toc195427423)

[pssh远程套件工具 (类似ansible) 8](#_Toc1824786947)

[autofs的配置(自动检测并挂载) 10](#_Toc799789247)

[使用LDAP认证(客户端) 11](#_Toc1660729284)

[邮件服务 13](#_Toc400464119)

[DHCP(自动分配ip) 14](#_Toc1302743250)

[tftp服务 14](#_Toc1831058515)

[PXE装机(DHCP,TFPT) 15](#_Toc405004016)

[(cobbler)网络装机 20](#_Toc1239470531)

# NTP时间同步服务器及客户端

修改日期与时间

date -s "年-月-日 时:分:秒 "

# date 查看时间

# date -s "2018-2-5 16:09:00"

# hwclock 还原系统时间

NTP网络时间协议

• Network Time Protocol

– NTP服务器为客户机提供标准时间

– NTP客户机需要与NTP服务器保持沟通

**客户端:**

**安装包:**chrony

配置文件:/etc/chrony.conf

server classroom.example.com iburst #指定ntp时间同步服务器

服务名: chronyd

**验证：**

[root@server0 /]# date -s "2008-9-9 10:1" 改时间

[root@server0 /]# systemctl restart chronyd

[root@server0 /]# date 查看时间有没有同步

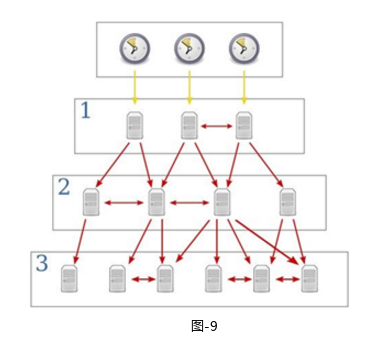
[root@server0 /]# hwclock 同步系统时间

**服务端:0**

安装包,配置文件,服务名同客户端.

Network Time Protocol（网络时间协议）采用的是分层设计，如图-9所示，Stratum层的总数限制在15以内。

服务端



1）安装软件包

# yum -y install chrony

# rpm -qc chrony                        //查看配置文件列表

/etc/chrony.conf

/etc/chrony.keys

.. ..

2)修改配置文件

# cat /etc/chrony.conf

.. ..

server 0.centos.pool.ntp.org iburst //server用户客户端指向上层NTP服务器

allow 192.168.4.0/24                        //允许那个IP或网络访问NTP

#deny 192.168.4.1                       //拒绝那个IP或网络访问NTP

local stratum 10                            //设置NTP服务器的层数量

.. ..

4)启动NTP服务

# systemctl restart chronyd

# systemctl enable chronyd

5)设置防火墙

# firewall-cmd --set-default-zone=trusted

1. 检查

chronyc sources -v

\* 同步服务器

+ 备用服务器

- 链接信号不好

# cron计划任务

• 用途:按照设置的时间间隔为用户反复执行某一项固定的系统任务

• **系统服务**:**crond**

• **日志文件**:**/var/log/crond**

编辑计划任务:**crontab** **-e**  [**-u 用户名**]

时间 执行任务

– 分 时 日 月 周 任务命令行(绝对路径)

\*: 匹配范围内任意时间

,: 分隔多个不连续的时间点

-: 指定连续时间范围

/n: 指定时间频率,每n ...

例子

[root@server0 /]# crontab -e

\*/1 \* \* \* \* date >> /opt/time.txt

每分钟记录当前的系统时间，写入/opt/time.txt

# FTP

**服务端:**

FTP: 文件传输协议(21)

**默认共享路径：/var/ftp**

**默认访问用户身份：ftp**

FTP客户端访问FTP服务端时,登陆的位置为该用户在服务器上的家目录

安装包名: vsftpd

服务名: vsftpd

/etc/vsftpd/vsftpd.conf：vsftpd 的核心配置文件

/etc/vsftpd/ftpusers：用于指定哪些用户不能访问FTP 服务器。 黑名单

/etc/vsftpd/user\_list：指定允许使用vsftpd 的用户列表文件。 白名单

[root@server0 ~]# yum -y install vsftpd

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd

客户端 : desktop0.example.com

[root@server0 ~]# firefox ftp://127.0.0.1 #访问本机服务

记得将server0和desktop0的防火墙设为**trusted**

[root@server0~]# touch /var/ftp/safa.txt

**客户端访问**

1. 图形界面下,其他位置里面输入ftp地址,就可以像普通文件一样访问
2. 通过浏览器访问
3. lftp访问

[root@desktop0 ~]# yum -y install lftp

[root@desktop0 ~]# lftp 172.25.0.11

lftp 172.25.0.11:~> ls

ftp 匿名允许上传:

1. 修改本地文件权限
2. 修改配置文件 //etc/vsftpd/vsftpd.conf

anon\_upload\_enable=YES

# NFS网络文件系统

• Network File System,网络文件系统

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

nfs只能在linux之间共享

• Network File System,网络文件系统

– 用途:为客户机提供共享使用的文件夹

– 协议:**NFS**(TCP/UDP 2049)、**RPC**(TCP/UDP 111)

**服务端:**

• 所需软件包: nfs-utils

• 系统服务: nfs-server

配置文件: /etc/exports

文件系统: nfs

**exports文件解析:**

格式: 文件夹路径 客户机地址(权限) 客户机地址(权限) .. ..

/public \*(ro) #把/public共享,所有主机都能访问,只读

• 使用exports可以重载更新过的配置.或者重启nfs-server服务

— exports -r

**客户端:**

列出有哪些NFS共享资源

— showmount -e 服务器地址

1. 手动挂载NFS共享

— mount 服务器地址:文件夹路径 挂载点

2. 开机挂载配置/etc/fstab

— mount 服务器地址:文件夹路径 挂载点 nfs \_netdev 0 0

**例子:**

[root@desktop0 ~]# mkdir /mnt/nfs

[root@desktop0 ~]# showmount -e 172.25.0.11 #查看有哪些nfs共享

Export list for 172.25.0.11:

/public \*

[root@desktop0 ~]# vim /etc/fstab #实现开机自动挂载

172.25.0.11:/public /mnt/nfs nfs defaults,\_netdev 0 0

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

配置聚合连接

（**聚合连接 = 网卡绑定 = 链路聚合**）

作用：网卡设备的冗余

热备份(activebackup)连接冗余 (活跃状态 备份状态)

eth1 team0-slave eth2 team0-slave

**team0** 虚拟网卡 master 192.168.1.1 (team0是自己设的名字）

**1.制作虚拟网卡 team0**

参考 **man teamd.conf** 全文查找**/example** 按**n**跳转匹配项

# nmcli connection add type team autoconnect yes con-name team0 ifname team0 config '{"runner": {"name": "activebackup"}}'

# nmcli connection 添加 类型为 team 的设备

每次开机自动启用 配置文件命名为 team0 网卡显示的名字为 team0

team0网卡内部成员工作模式为 '{"runner": {"name": "activebackup"}}' （热备）

**2. 为 team0添加成员**

# nmcli connection add type **team-slave** con-name team0-1 ifname eth1 **master team0**

# nmcli connection add type **team-slave** con-name team0-2 ifname eth2 **master team0**

# 添加 类型为 team-slave 的设备 配置文件命名为 team0-1 网卡为eth1

主设备为team0

**3.配置team0的ipv4地址与激活**

# nmcli connection modify team0 ipv4.method manual

ipv4.addresses 192.168.1.1/24 connection.autoconnect yes

# nmcli connection up team0 #激活team0网卡

# nmcli connection up team0-1 #激活team0-1成员

# nmcli connection up team0-2 #激活team0-2成员

**4.错误的解决方法**

1.删除

[root@server0 ~]# nmcli connection delete team0

[root@server0 ~]# nmcli connection delete team0-1

[root@server0 ~]# nmcli connection delete team0-2

2.重新配置

**5.专用于team测试查看的命令**

[root@server0 ~]# teamdctl team0 state #查看team0信息

[root@server0 ~]# ifconfig eth1 down #禁用eth1网卡

[root@server0 ~]# teamdctl team0 state #查看team0信息

[root@server0 ~]# ifconfig eth1 up #启用eth1网卡

一般关闭，开启网卡的操作

关闭eth0,#ifdown eth0

开启eth0,#ifup eth0

# ISCSI共享存储(网络硬盘)

**iSCSI磁盘的工作模式**

.Internet SCSI,网际SCSI接口

- 一种基于**C/S**架构的虚拟磁盘技术

- 服务器提供磁盘空间，客户机连接并当成本地磁盘使用

**iSCSI磁盘的构成**

做iscsi服务需要三个概念

**backstore**：**后端真正的存储设备(实物)**

**target: 共享名(虚拟)**

**lun: 绑定、关联存储设备**

iSCSI Qualified Name 名称规范

- **iqn.yyyy-mm.倒序域名：自定义标识 (yyyy-mm:年月)**

如:iqn.2018-02.com.example:data

- 用来识别target磁盘组，也用来识别客户机身份

**服务端**

在**server0**上准备一个磁盘分区(vdb1)3G,可以使用fdisk或者parted分区工具得到

安装包: targetcli

服务名: target

**.基本交互操作**

• 建立后端存储

- /backstores/block create 后端存储名 实际设备路径

• 建立磁盘组(相当于制作存放分区的箱子)

- /iscsi/ create 磁盘组的IQN名称 (注意空格号）

• 将共享的分区放入箱子中

- /iscsi/磁盘组名/tpg1/luns create 后端存储路径

• 指定提供服务的ip地址

- /iscsi/磁盘组名/tpg1/acls create 客户机IQN标识

• 访问控制的设置

- /iscsi/磁盘组名/tpg1/portals create IP地址 端口号

**.其他辅助配置指令**

**- ls、saveconfig、exit**

# backstores/block create back\_store /dev/vdb1//把刚刚分的分区加入后端存储

# iscsi/ create iqn.2018-02.com.example:data //创建一个iscsi共享（共享名称).注意名称规范

# iscsi/iqn.2018-02.com.example:data/tpg1/luns create /backstores/block/back\_store //把共享名和后端的设备通过lun关联在一起

#iscsi/iqn.2018-02.com.example:data/tpg1/acls create iqn.2018-02.com.example:desktop0

//创建了一个访问的口令:iqn.2018-02.com.example:desktop0

//以后仅知道口令的客户端才可以访问共享

# iscsi/iqn.2018-02.com.example:data/tpg1/portals create 172.25.0.11 //以后客户端访问本机的172.25.0.11的**3260**端口旧可以访问到共享

# saveconfig 保存

# exit

**#firewall-cmd --set-default-zone=trusted**

# systemctl restart target #重起target服务

# systemctl enable target #设置为开机自启动

**客户端**

安装包:iscsi-initiator-utils

服务名:iscsi

配置文件: /etc/iscsi/initiatorname.iscsi

在desktop操作 (服务端防火墙一定设成trusted )

**1. 安装iscsi-initiator-utils**

# yum -y install iscsi-initiator-utils

//在实验环境可以跳过，生产环境需要确认

**2. 修改/etc/iscsi/initiatorname.iscsi**

客户机配置IQN名称 :/etc/iscsi/initiatorname.iscsi

# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-02.com.example:desktop0 //注意在这里给客户端配置访问口令，一定要与服务器的ACL一致

# systemctl restart iscsid //刷新IQN标识

可以通过man帮助来找到客户机配置IQN的名称

# ***man iscsiadm*** //找参考命令，在最后输入 **/EXAMPLES** 可以找到下面两个命令行.复制时注意最好将命令行缩小成一行复制，另外后面空白部分（相当于回车的作用）不要复制.

**3.连接、发现iSCSI磁盘**

.发现iSCSI磁盘

- iscsiadm -m discovery -t st -p 服务器地址：[端口]

# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover //发现对方服务器上的共享

.连接iSCSI磁盘

- iscsiadm -m node -t st -p 服务器地址：[端口]

#iscsiadm --mode node --targetname iqn.2018-02.com.example:data --portal 172.25.0.11:3260 --login

//挂载172.25.0.11服务器上的共享iqn.2018-02.com.example:data

//如果挂载提示authentication，口令不对

# systemctl restart iscsi 重启iscsi服务加载共享存储

# lsblk 可以找到共享的硬盘,可以对他格式化,挂载等操作

# pssh远程套件工具 (类似ansible)

安装包: pssh

安装pssh后会提供如下命令：

/usr/bin/pnuke 提供远程杀死进程的功能

/usr/bin/prsync

/usr/bin/pscp.pssh 提供并发拷贝文件功能

/usr/bin/pslurp 提供远程下载功能

/usr/bin/pssh 提供并发远程连接功能

语法格式

# man pssh              //通过man帮助查看工具选项的作用

pssh提供并发远程连接功能

-A                使用密码远程其他主机（默认使用密钥）

-i                将输出显示在屏幕

-H                设置需要连接的主机

-h                设置主机列表文件

-p                设置并发数量

-t                设置超时时间

-o dir            设置标准输出信息保存的目录

-e dir            设置错误输出信息保存的目录

-x                传递参数给ssh

# man pscp.pssh                    //通过man帮助查看工具选项的作用

pscp.pssh提供并发拷贝文件功能

-r    递归拷贝目录

其他选项基本与pssh一致

**准备工作**

1）安装软件包

# rpm -ivh pssh-2.3.1-5.el7.noarch.rpm

2)修改/etc/hosts本地解析文件

cat /etc/hosts

… …

192.168.2.100 host1

192.168.2.200 host2

192.168.4.100 host3

… …

3)创建主机列表文件

# cat /root/host.txt            //每行一个用户名、IP或域名

… …

root@host1

host2

host3

… …

**使用密码批量远程**

2)使用密码远程多台主机执行命令，屏幕显示标准和错误输出信息

pssh -i -A -H 'host1 host2 host3' -x '-o StrictHostKeyChecking=no' echo hello

3)使用密码远程多台主机执行命令，不显示标准和错误输出信息，通过读取host.txt读取主机信息

pssh -A -h /root/host.txt -x '-o StrictHostKeyChecking=no' echo hello

**使用密钥批量、多并发远程其他主机**

1）生成密钥并发送密钥到其他主机

# ssh-keygen -N '' -f /root/.ssh/id\_rsa //非交互生成密钥文件

# ssh-copy-id -i /root/.ssh/id\_rsa.pub host1

# ssh-copy-id -i /root/.ssh/id\_rsa.pub host2

# ssh-copy-id -i /root/.ssh/id\_rsa.pub host3

2)使用密钥远程其他主机

# pssh -h host.txt -x '-o StrictH批量、多并发拷贝数据到其他主机

ostKeyChecking=no' echo hello

3)使用密钥远程其他主机，将标准输出信息写入到/tmp目录

# pssh -h host.txt -o /tmp/ -x '-o StrictHostKeyChecking=no' echo hello

**批量、多并发拷贝数据到其他主机**

2)将本地的etc.tar.gz拷贝到远程主机的根目录下

# pscp.pssh -h host.txt -x '-o StrictHostKeyChecking=no' etc.tar.gz /

3)递归将本地的/etc目录拷贝到远程主机的/tmp目录下

#pscp.pssh -r -h host.txt -x '-o StrictHostKeyChecking=no' /etc /tmp

步骤五：批量、多并发从其他主机下载数据到本机

2)将远程主机的/etc/目录，拷贝到当前目录下，存放在对应IP下的tmp目录

# pslurp -r -h host.txt -x '-o StrictHostKeyChecking=no' /etc /tmp

3)将远程主机的/etc/目录，拷贝到media下，存放在对应IP下的tmp目录

# pslurp -r -h host.txt -x '-o StrictHostKeyChecking=no' -L /media /etc /tmp

**批量、多并发杀死其他主机的进程**

2)将远程主机上的sleep进程杀死

# pnuke -h host.txt -x '-o StrictHostKeyChecking=no' sleep

3)将远程主机上的test相关脚本都杀死（如：test1,testtt,test2等等）

# pnuke -h host.txt -x '-o StrictHostKeyChecking=no' test

4)将远程主机上的test.sh脚本杀死

# pnuke -h host.txt -x '-o StrictHostKeyChecking=no' test.sh

# autofs的配置(自动检测并挂载)

安装包 autofs

服务名 autofs

配置文件:

/etc/auto.master （指定监控点）

/etc/auto.guests (配置挂载策略)

结合LDAP,讲

mkdir /home/guests

vim /etc/auto.master （指定监控点）

/home/guests /etc/auto.guests

vim /etc/auto.guests (配置挂载策略)

**\*** -rw classroom.example.com:/home/guests/&

systemctl restart autofs

systemctl enable autofs

[root@server0 ~]# ls /home/guests/ (底下不会显示信息）

[root@server0 ~]# su - ldapuser0

[ldapuser0@server0 ~]$ exit

[root@server0 ~]# ls /home/guests/

ldapuser0

[root@server0 ~]# su - ldapuser1

[ldapuser1@server0 ~]$ exit

[root@server0 ~]# ls /home/guests/

ldapuser0 ldapuser1

**（只有切换进去，触发后才能在/home/guests中显示）**

# 使用LDAP认证(客户端)

• 轻量级目录访问协议(LDAP)

– Lightweight Directory Access Protocol

– 由服务器来集中存储并向客户端提供的信息,存储方式类似于DNS分层结构

– 提供的信息包括:用户名、密码

• 为一组客户机集中提供可登录的用户账号

– 本地用户账户密码存放在：/etc/passwd /etc/shadow

– 网络用户账户密码存放在: 用户名、密码信息存储在 LDAP 服务端

**如何加入LDAP域**

服务端：classroom.example.com

客户端：server0.example.com 指定 LDAP 服务端位置

1.安装一个客户端sssd软件,与LDAP 服务端沟通

[root@server0 /]# yum -y install sssd

2.安装authconfig-gtk 图形的工具 配置sssd软件

[root@server0 /]# yum -y install authconfig-gtk

[root@server0 /]# rpm -q authconfig-gtk

3.运行authconfig-gtk 图形的工具

[root@server0 /]# authconfig-gtk

选择LDAP

dc=example,dc=com #指定服务端域名

classroom.example.com #指定服务端主机名

勾选TLS加密

使用证书加密: http://classroom.example.com/pub/example-ca.crt

选择LDAP密码

4.重起客户端服务sssd服务，设置开机自启动

[root@server0 /]# systemctl restart sssd #重起sssd服务

[root@server0 /]# systemctl enable sssd #设置开机自启动

5.验证

[root@server0 ~]# grep 'ldapuser0' /etc/passwd

[root@server0 ~]# id ldapuser0 //此时并没有

**验证**

**一. 家目录漫游 (在LDAP认证完即可操作）**

1.访问NFS共享

• Network File System,网络文件系统

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

NFS服务器 ：classroom.example.com

客户端： server0.example.com

查看共享classroom.example.com

[root@server0 ~]# showmount -e classroom.example.com

Export list for classroom.example.com:

/home/guests 172.25.0.0/255.255.0.0 **共享路径**

手动挂载LDAP家目录

访问共享内容,将服务端的共享文件夹数据，挂载到本地/mnt以本地的/mnt作为访问点

# mount classroom.example.com:/home/guests /mnt

# ls /mnt/

# mkdir /home/guests

# mount classroom.example.com:/home/guests/ /home/guests/

# ls /home/guests/

[root@server0 ~]# su - ldapuser0

**2. Autofs服务 (自动检测远端并挂载)**

yum -y install autofs

mkdir /home/guests

修改配置文件

vim /etc/auto.master （指定监控点）

/home/guests /etc/auto.guests

vim /etc/auto.guests (配置挂载策略)

\* -rw classroom.example.com:/home/guests/&

启动服务

systemctl restart autofs

systemctl enable autofs

验证

[root@server0 ~]# ls /home/guests/ (底下不会显示信息）

[root@server0 ~]# su - ldapuser0

[ldapuser0@server0 ~]$ exit

[root@server0 ~]# ls /home/guests/

ldapuser0

[root@server0 ~]# su - ldapuser1

[ldapuser1@server0 ~]$ exit

[root@server0 ~]# ls /home/guests/

ldapuser0 ldapuser1

**（只有切换进去，触发后才能在/home/guests中显示）**

# 邮件服务

邮件系统(**postfix**)可以做类似于163的邮件系统 postfix默认安装

DNS服务器：classroom.example.com

**1.快速部署posrfix邮件服务器**

#vim /etc/postfix/main.cf

99:myorigin = $mydomain

//默认补全域名后缀

116:inet\_interfaces = all

//允许所有人给自己发邮件，默认仅接收本机发的邮件

164：mydestination = server0.example.com

//什么邮件我才接收

#systemctl restart **postfix**

**2.使用mail命令发信/收信**

.mail发信操作

#mail -s 标题 收件人

内容

内容

**.** //独立的点回车表示邮件内容结束

#mail -s hello root

hello the world

.

.mail 收信操作

#mail [-u 用户名] //收取邮件，q退出

# DHCP(自动分配ip)

自动分配网络参数服务

• Dynamic Host Configuration Protocol

– 动态主机配置协议,由 IETF(Internet 网络工程师任务小组)组织制定,用来简化主机地址分配管理

• 主要分配以下入网参数

– IP地址/子网掩码/广播地址

– 默认网关地址、DNS服务器地址

• DHCP工作过程以广播的方式进行，原理先到先得

• DHCP地址分配的四次会话

– DISCOVERY --> OFFER --> REQUEST -->ACK

• 服务端基本概念

– 租期:允许客户机租用IP地址的时间期限,单位为秒

– 作用域:分配给客户机的IP地址所在的网段

– 地址池:用来动态分配的IP地址的范围

一、搭建基本DHCP服务

1.安装dhcp包

[root@svr7 ~]# yum -y install dhcp

[root@svr7 ~]# rpm -q dhcp

2.修改配置文件 /etc/dhcp/dhcpd.conf

可以参考/usr/share/doc/dhcp\*/dhcpd.conf.example

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 { #指定分配的网段

range 192.168.4.10 192.168.4.100; #指定的分配具体的ip地址范围

option domain-name-servers 8.8.8.8; #指定dns地址

option routers 192.168.4.254; #指定网关地址

}

3.重起dhcpd服务

systemctl restart dhcpd

# tftp服务

tftp:简单文件传输协议 端口:69

默认共享的路径：/var/lib/tftpboot

1.安装tftp-server

[root@svr7 ~]# yum -y install tftp-server

2.启动tftp服务

[root@svr7 ~]# systemctl restart tftp

[root@svr7 ~]# systemctl enable tftp

# PXE装机(DHCP,TFPT)

网络装机的优势

• 规模化:同时装配多台主机

• 自动化:装系统、配置各种服务

• 远程实现:不需要光盘、U盘等物理安装介质

什么是PXE网络

• PXE,Pre-boot eXecution Environment

– 预启动执行环境,在操作系统之前运行

– 可用于远程安装

• 工作模式

– PXE client 集成在网卡的启动芯片中

– 当计算机引导时,从网卡芯片中把PXE client调入内存执行,获取PXE server配置、显示菜单,根据用户选择将远程引导程序下载到本机运行

• 需要哪些服务组件?

– DHCP服务,分配IP地址、定位引导程序

– TFTP服务,提供引导程序下载

– HTTP服务(或FTP/NFS),提供yum安装源(系统镜像)

• 客户机应具备的条件

– 网卡芯片必须支持PXE协议

– 主板支持从网卡启动

#############################################################

搭建PXE网络装机服务器

**一、配置dhcp服务pxe的设置**

1.修改配置文件/etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.10 192.168.4.100;

option domain-name-servers 8.8.8.8;

option routers 192.168.4.254;

next-server 192.168.4.7; #指定下一个服务器地址

filename "pxelinux.0"; #指定到下一个服务器下载的文件名字

}

2.重起dhcpd服务

[root@svr7 ~]# systemctl restart dhcpd

##################################################################

pxelinux.0：网卡引导文件（安装说明书）二进制的文件，安装一个软件可以生成

**二、搭建tftp服务**

tftp:简单文件传输协议 端口:69

默认共享的路径：/var/lib/tftpboot

1.安装tftp-server

[root@svr7 ~]# yum -y install tftp-server

2.启动tftp服务

[root@svr7 ~]# systemctl restart tftp

[root@svr7 ~]# systemctl enable tftp

3.部署 pxelinux.0 引导文件

[root@svr7 ~]# yum provides \*/pxelinux.0 #查询什么软件包产生pxelinux.0

[root@svr7 ~]# yum -y install syslinux

[root@svr7 ~]# rpm -ql syslinux | grep pxelinux.0 #查询软件包安装清单

[root@svr7 ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

[root@svr7 ~]# ls /var/lib/tftpboot/

pxelinux.0 ------》 /var/lib/tftpboot/pxelinux.cfg/default

default------》vesamenu.c32、splash.png、vmlinuz、initrd.img

4.部署默认菜单文件

# mkdir /var/lib/tftpboot/pxelinux.cfg

# cp /dvd/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

# ls -l /var/lib/tftpboot/pxelinux.cfg/default

# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default

# ls -l /var/lib/tftpboot/pxelinux.cfg/default

5.部署 图形的模块 与 背景图片

vesamenu.c32 :图形的模块

splash.png :背景图片

# cp /dvd/isolinux/vesamenu.c32 /dvd/isolinux/splash.png /var/lib/tftpboot/

# ls /var/lib/tftpboot/

6.部署启动内核与驱动程序

vmlinuz :启动内核

initrd.img :驱动程序

# cp /dvd/isolinux/vmlinuz /dvd/isolinux/initrd.img /var/lib/tftpboot/

# ls /var/lib/tftpboot/

initrd.img pxelinux.cfg vesamenu.c32

pxelinux.0 splash.png vmlinuz

7.修改菜单文件

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

.........

default vesamenu.c32 #默认加载图形的模块

timeout 600 #读秒时间为60秒

........

menu background splash.png #指定背景图片

menu title NSD1801 PXE Server ! #标题信息

......

label linux

menu label Install RHEL7 #显示菜单的内容

kernel vmlinuz #加载内核

append initrd=initrd.img #加载驱动

##########################################################

初步测试：

新建一台全新的虚拟机，选择PXE网络引导安装，注意网络类型要选择 private1

**如何更改虚拟机的菜单背景**。

真机上传目录到虚拟机

[root@room9pc01 ~]# scp -r /root/桌面/tupian.zip root@192.168.4.7:/root/

root@192.168.4.7's password:

[root@sur7 ~]# ls

anaconda-ks.cfg initial-setup-ks.cfg tupian.zip 模板 图片 下载 桌面

bin ks.cfg 公共 视频 文档 音乐

[root@sur7 ~]# unzip tupian.zip

[root@sur7 ~]# ls tupian

1.jpg 2.jpg 3.jpg 4.jpg 5.jpg 6.jpg 7.jpg

[root@sur7 ~]# rm -rf /var/lib/tftpboot/splash.png

[root@sur7 ~]# cp tupian/1.jpg /var/lib/tftpboot

[root@sur7 ~]# ls /var/lib/tftpboot

[root@sur7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

menu background 1.jpg

重新安装一个虚拟机（PXE）验证，如果想再更换菜单背景的话，需要再安装一个虚拟机进行验证，原来的就没有用了。

################################################################

**三、搭建httpd服务**

1.安装httpd软件包

[root@svr7 ~]# yum -y install httpd

2.重起httpd服务，设置开机自启动

[root@svr7 ~]# systemctl restart httpd

[root@svr7 ~]# systemctl enable httpd

3.利用httpd服务共享光盘所有内容

[root@svr7 ~]# mkdir /var/www/html/rhel7

[root@svr7 ~]# mount /dev/cdrom /var/www/html/rhel7

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 ~]# ls /var/www/html/rhel7

[root@svr7 ~]# firefox http://192.168.4.7/rhel7

##############################################################

**四、配置无人值守安装，应答文件(ks)的生成**

1.安装图形的工具，system-config-kickstart

[root@svr7 ~]# yum -y install system-config-kickstart

2.运行图形的工具

[root@svr7 ~]# system-config-kickstart

第一步 查看 软件包选择 是否可用

第二步 yum仓库的标识 必须为 [development]

[root@svr7 ~]# vim /etc/yum.repos.d/dvd.repo

[root@svr7 ~]# system-config-kickstart #再一次运行，查看软件包的选择

进入Kickstart配置程序中操作

(a)基本配置 ：默认语言设置为中文简体（不是固定的）

时区为Asia/Shanghai （不是固定的）

Root密码和确认密码为123456 （必须设，密码可以随意设）

勾上安装后重启和给root密码加密

（b)安装方法 ：执行全新安装

HTTP 服务器：192.168.4.7（A虚拟机的ip）

目录：rhel7

(c)安装类型：点击安装新引导装载程序

(d)分区信息：主引导记录中点击清除主引导记录

分区中点击删除所有现存分区

磁盘标签点击初始化磁盘标签

然后在里面点击添加，挂载点为/ ,大小选项为使用磁盘上全部未用空间，格式化分区，

(e)网络配置： 添加网络设备 ，输入eth0

(f)防火墙配置：SELinux和安全级别都使用禁用

(g)软件包选择：勾选基本

(h)勾选使用解释程序：/bin/bash

在脚本中输入useradd lisi

echo 123456 | passwd --stdin lisi

然后保存在root里面即可。

[root@svr7 ~]# ls /root/ks.cfg

/root/ks.cfg

[root@svr7 ~]# vim /root/ks.cfg

###############################################################

五、利用httpd服务共享ks.cfg应答文件

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/

[root@svr7 ~]# ls /var/www/html/

###############################################################

六、修改默认菜单文件,指定ks.cfg应答文件

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label Install RHEL7

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg

重新安装一个新的虚拟机进行测试

生成新虚拟机 ： 勾上网络引导（PXE) -前进-前进-前进-选择网络一定要用虚拟网络 'private1'

##############################################################

七、总结

1.dhcp------》ip地址、next-server、filename

2.tftp------》pxelinux.0、pxelinux.cfg/default、vesamenu.c32、splash.png、vmlinuz、initrd.img、ks=http://192.168.4.7/ks.cfg

3.httpd------》http://192.168.4.7/rhel7

# (cobbler)网络装机

一、解压cobbler.zip包，cobbler主程序、工具包等

[root@room9pc01 /]# ls /root/桌面/Cobbler/

cobbler\_boot.tar.gz cobbler\_web.png cobbler.zip

[root@room9pc01 /]# unzip /root/桌面/Cobbler/cobbler.zip -d /

[root@room9pc01 /]# ls /cobbler

二、为真机搭建永久的本地的Yum仓库

1.挂载/iso/CentOS-7-x86\_64-DVD-1708.iso

[root@room9pc01 ~]# mount /iso/CentOS-7-x86\_64-DVD-1708.iso /dvd/

[root@room9pc01 ~]# ls /dvd/

2.书写Yum客户端文件

[root@room9pc01 ~]# vim /etc/yum.repos.d/nsd.repo

[development]

name=dvd

baseurl=file:///dvd

enabled=1

gpgcheck=0

[root@room9pc01 ~]# yum repolist

3.为真机永久挂载/iso/CentOS-7-x86\_64-DVD-1708.iso

默认的iso文件系统 : iso9660

默认的/dev/cdrom文件系统 : iso9660

/etc/rc.d/rc.local : 每次开机运行配置文件

[root@room9pc01 /]# ls -l /etc/rc.d/rc.local

[root@room9pc01 /]# vim /etc/rc.d/rc.local

mount /var/lib/libvirt/images/iso/CentOS-7-x86\_64-DVD-1708.iso /dvd

[root@room9pc01 /]# reboot #重起系统验证

[root@room9pc01 /]# yum repolist

##################################################################

三、安装cobbler主程序、工具包等

[root@room9pc01 /]# yum -y install /cobbler/\*.rpm

#################################################################

cobbler网络装机部署

1.安装软件 cobbler cobbler-web dhcp tftp-server pykickstart httpd

cobbler #cobbler程序包

cobbler-web #cobbler的web服务包

pykickstart #cobbler检查kickstart语法错误

httpd #Apache web服务

dhcp #Dhcp服务

tftp-server #tftp服务

2.配置cobbler

[root@svr7 /]# vim /etc/cobbler/settings

修改为 next\_server: 192.168.4.254 #设置下一个服务器还为本机

将 server: 127.0.0.1 修改为 server: 192.168.4.254 #设置本机为cobbler服务器

将 manage\_dhcp: 0 修改为 manage\_dhcp: 1 #设置cobbler管理dhcp服务

将 pxe\_just\_once: 0 修改为 pxe\_just\_once: 1 #防止客户端重复安装操作系统

3.配置cobbler的dhcp

[root@svr7 /]# vim /etc/cobbler/dhcp.template

:%s /192.168.1/192.168.4/g 在末行模式下输入

[root@svr7 /]# vim /etc/sysconfig/dhcpd

DHCPDARGS=private1 表示dhcp将只在private1 网络接口上提供DHCP服务

4.绝对路径解压cobbler\_boot.tar.gz #众多的引导文件

[root@room9pc01 /]# tar -tf /root/桌面/Cobbler/cobbler\_boot.tar.gz

[root@room9pc01 /]# tar -xPf /root/桌面/Cobbler/cobbler\_boot.tar.gz

[root@room9pc01 /]# ls /var/lib/cobbler/loaders/

5.启动相关服务

[root@svr7 /]# systemctl restart cobblerd

[root@svr7 /]# systemctl enable cobblerd

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# systemctl enable httpd

[root@svr7 /]# systemctl restart tftp

[root@svr7 /]# systemctl enable tftp

[root@svr7 /]# systemctl restart rsyncd

[root@svr7 /]# systemctl enable rsyncd

6.同步刷新cobbler配置

[root@svr7 /]# cobbler sync

[root@svr7 /]# firefox https://192.168.4.254/cobbler\_web

##########################################################################################################

cobbler import --path=挂载点 --name=导入系统命名（随意起）

导入安装镜像数据

[root@room9pc01 ~]# mount /iso/rhel-server-7.4-x86\_64-dvd.iso /dvd

mount: /dev/loop2 写保护，将以只读方式挂载

[root@room9pc01 /]# cobbler list #查看有哪些系统

[root@room9pc01 /]# cobbler import --path=/dvd --name=rhel7

[root@room9pc01 /]# mkdir /rhel6

[root@room9pc01 /]# mount /iso/rhel-server-6.7-x86\_64-dvd.iso /rhel6/

[root@room9pc01 /]# cobbler import --path=/rhel6 --name=RedHat6

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

[root@room9pc01 /]# cobbler profile report #查看cobbler导入信息

[root@room9pc01 /]# killall -9 dnsmasq #虚拟化服务会干扰DHCP服务

############################################################################################################

默认kickstart文件存放位置：/var/lib/cobbler/kickstarts/

[root@cobbler ~]# cobbler list

修改kickstart文件：

[root@cobbler ~]# cobbler profile edit --name=CentOS7.4-A --kickstart=/var/lib/cobbler/kickstarts/CentOS-7.3-x86\_64.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync