

一 客户端开机启动,引导项

1.本机磁盘 2.光驱设备 3.U 盘 4.网卡进行网络安装

二 rsync

2.1 rsyn : remote sync 远程同步

支持本地复制,或与其他 SSH\rsync 主机同步

官网 <http://rsync.samba.org/>

2.2 rsync 同步传输

命令:rsync [选项...] 源目录 目标目录

同步与复制的差异

复制:完全拷贝源到目标

同步:增量拷贝,只传输变化过的数据

2.3 操作选项

-n: 测试同步过程,不做实际修改

--delete: 删除目标文件夹内多余的文档

-a: 归档模式,相当于-rlptgoD

-v: 显示详细操作信息

-z: 传输过程中启用压缩/解压 [建议 1G 以上启用]

-r: 递归,包括目录/子目录及所有文件

-l: 保留符号链接文件

- p\t: 保留文件的权限\时间标记
- o\g: 保留文件的属主\属组标记
- D: 保留设备文件及其他特殊文件

2.4 操作示例

```
[root@svr7 ~]# mkdir /nsd19
```

```
[root@svr7 ~]# mkdir /todir
```

```
[root@svr7 ~]# cp /etc/passwd /nsd19
```

```
[root@svr7 ~]# touch /nsd19/1.txt
```

```
[root@svr7 ~]# rsync -av /nsd19 /todir //同步 nsd19 目录及其内容到/todir
```

sending incremental file list

nsd19/ **//同步的内容**

nsd19/1.txt

nsd19/passwd

sent 1,157 bytes received 58 bytes 2,430.00 bytes/sec

total size is 968 speedup is 0.80

```
[root@svr7 ~]# rm -rf /todir/*
```

```
[root@svr7 ~]# rsync -av /nsd19/ /todir //同步 nsd19 的内容到/todir
```

sending incremental file list

✓

1.txt

//同步的内容

passwd

sent 1,142 bytes received 57 bytes 2,398.00 bytes/sec

total size is 968 speedup is 0.81

```
[root@svr7 ~]# echo haha >> /nsd19/1.txt //修改 /nsd19 的内容
```

```
[root@svr7 ~]# cat /nsd19/1.txt
```

haha

```
[root@svr7 ~]# rsync -av /nsd19/ /todir/
```

//同步/nsd19 目录内修改的内容到/todir 目录内,两个目录后都接

"/符号,此符号可用 TAB 键自动补齐

sending incremental file list

1.txt

sent 137 bytes received 35 bytes 344.00 bytes/sec

total size is 973 speedup is 5.66

```
[root@svr7 ~]# ls /todir
```

1.txt passwd

```
[root@svr7 ~]# mkdir /todir/abcdefg // /todir 内增加内容
```

```
[root@svr7 ~]# ls /todir
```

1.txt abcdefg passwd

```
[root@svr7 ~]# rsync -av --delete /nsd19/ /todir/
```

//删除/todir 内相对于/nsd19 内多余的内容

sending incremental file list

deleting abcdefg/

✓

sent 88 bytes received 27 bytes 230.00 bytes/sec

total size is 973 speedup is 8.46

2.5 远程同步

rsync + SSH 同步

与远程的 SSH 目录保持同步

下行: rsync [...] user@host:远程目录 本地目录

上行: rsync [...] 本地目录 user@host:远程目录

```
[root@svr7 ~]# rsync -av --delete /nsd19/ root@192.168.4.207:/opt/
```

//虚拟机 A 上将/nsd19 内数据上传到虚拟机 B /opt 目录内

Warning: Permanently added '192.168.4.207' (ECDSA) to the list of known hosts.

root@192.168.4.207's password:

sending file list

✓

//上行的内容

1.txt

abc1.txt

abc2.txt

abc3.txt

abc4.txt

abc5.txt

passwd

sent 1,435 bytes received 152 bytes 352.67 bytes/sec

total size is 973 speedup is 0.61

[root@pc207 ~]# ls /opt //查看虚拟机 B 上 /opt 内容

1.txt abc1.txt abc2.txt abc3.txt abc4.txt abc5.txt passwd

三 实时远程同步

实时 rsync + SSH,其中 SSH 需要密码.

解决方案:通过公钥 私钥(两个文件的配对性),实现 SSH 无密码验证.

3.1 生成 公钥 私钥

虚拟机 A

[root@svr7 ~]# **ssh-keygen** //生成公钥 私钥命令

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): //默认

Enter passphrase (empty for no passphrase): //不输入密码

Enter same passphrase again: //确认不输入密码

Your identification has been saved in /root/.ssh/id_rsa.

Your public key has been saved in /root/.ssh/id_rsa.pub.

The key fingerprint is:

SHA256:bdW27JFW0n28LTeUBHy+5XFKN7lVtzC1VlYngsdXQFc root@svr7.tedu.cn

The key's randomart image is:

```
+---[RSA 2048]-----+
|          ooo*=E|
|          .o*oB0|
|          ...O=@|
|          ..o+@O|
|          S o .B*@|
|          .  o.*o|
|          .  |
|          |
|          |
|          |
```

+---[SHA256]-----+

[root@svr7 ~]# ls /root/.ssh/

authorized_keys id_rsa id_rsa.pub known_hosts

id_rsa: 私钥 id_rsa.pub: 公钥 known_hosts:记录连接过的远程主机

authorized_keys: 记录公钥

3.2 将公钥上传到虚拟机 B

一台主机上可以存储多个公钥

```
[root@svr7 ~]# ssh-copy-id root@192.168.4.207
```

//虚拟机 A 上将公钥传递给虚拟机 B

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
```

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any  
that are already installed
```

```
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now  
it is to install the new keys
```

root@192.168.4.207's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.4.207'"

and check to make sure that only the key(s) you wanted were added.

```
[root@svr7 ~]# rsync -av --delete /nsd19/ root@192.168.4.207:/opt/ //验证
```

sending incremental file list //验证,无需再输入密码

deleting 2.txt

✓

sent 177 bytes received 24 bytes 134.00 bytes/sec

total size is 973 speedup is 4.84

拥有私钥的主机,能无密码登录拥有公钥的机器.

3.3 监控/nsd19 目录变化

Linux 内核的 inotify 机制

提供 inotify-tools 控制工具可条用此机制实现监控

找到 inotify-tools

真机

```
[student@room9pc01 ~]$ scp /linux-soft/01/tools.tar.gz root@192.168.4.7:/root/
```

虚拟机 A

```
[root@svr7 ~]# ls /root/
```

```
Desktop  tools.tar.gz
```

```
tar -xf /root/tools.tar.gz -C /
```

```
ls /tools/
```

安装 gcc 与 make

解包

```
[root@svr7 ~]# ls /root/
```

```
Desktop  tools.tar.gz
```

```
[root@svr7 ~]# tar -xf tools.tar.gz -C /
```

```
[root@svr7 ~]# ls /
```

```
bin    dev    home  lib64  mnt    opt    root  sbin  sys  todir  usr
```

```
boot  etc  lib  media  nsd19  proc  run  srv  tmp  tools  var
```

```
[root@svr7 ~]# ls /tools
```


inotify-tools-3.13.tar.gz other

```
[root@svr7 ~]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /opt
```

```
[root@svr7 ~]# cd /opt/inotify-tools-3.13/
```

运行 configure 脚本进行配置检测

```
[root@svr7 inotify-tools-3.13]# ./configure
```

运行 make 进行编译

```
[root@svr7 inotify-tools-3.13]#make
```

运行 make install 进行安装

```
[root@svr7 inotify-tools-3.13]#make install
```

验证

```
[root@svr7 inotify-tools-3.13]# ls /usr/local/bin/inotifywait
```

/usr/local/bin/inotifywait //此路径下有 inotifywait 在表示成功

inotify 命令

-m:持续监控,捕获一个事件后不退出

-r:递归监控,包括子目录及文件

-q:减少屏幕输出信息

-e:指定监视的 modify\move\create\delete\attrib(属性) 等事件类别

-e 不写默认监控所有

3.4 编写脚本实现监视与同步的结合

```
inotifywait -rq /nsd19/
```

```
rsync -av --delete /nsd19/ root@192.168.4.207:/opt/
```

书写脚本 rsync.sh

循环:for 循环适合写有次数的循环

死循环:while 循环适合写无限次数的循环

while [条件]

do

重复执行的代码

done

while 命令

do

重复执行的代码

done

虚拟机 A

```
[root@svr7 ~]# vim rsync.sh //创建 shell 脚本并编写
```

```
#!/bin/bash
```

```
while inotifywait -rqq /nsd19/
```

do

```
rsync -a --delete /nsd19/ root@192.168.4.207:/opt/
```

done

```
[root@svr7 ~]# ll /root/rsync.sh //查看并修改 rsync.sh 的权限并修改
```

```
-rw-r--r-- 1 root root 112 7 月 22 11:36 /root/rsync.sh
```

```
[root@svr7 ~]# chmod +x /root/rsync.sh
```

```
[root@svr7 ~]# ll /root/rsync.sh
```

```
-rwxr-xr-x 1 root root 112 7 月 22 11:36 /root/rsync.sh
```

```
[root@svr7 ~]# /root/rsync.sh & //放入后台执行
```

```
[root@svr7 ~]# jobs //查看后台运行的进程
```

```
[1]+ 运行中 /root/rsync.sh &
```

四 cobbler 装机平台(看笔记能搭)

4.1 基本概念

是一款快速的网络系统部署工具

集中管理所需服务,如 DHCP\DNS\TFTP\WEB

内部集成了一个镜像版本仓库

内部集成了一个 ks 应答文件仓库

提供了 yum 源管理\web 界面管理\API 接口\电源管理

与 PXE 相比,比 PXE 简单,多版本系统部署简单,没 PXE 自动化自定义度高

4.2 手动设置一个包含图形功能的虚拟机

```
root 1 lisi 1
```

主机名 cobbler.tedu.cn

```
hostname cobbler.tedu.cn
```

```
echo cobbler.tedu.cn > /etc/hostname
```

设置 IP 192.168.4.123/24

4.1 基本概念

```
nmcli connection modify eth0 ipv4.method manual ipv4.addresses '192.16
```

```
8.4.123/24' connection.autoconnect yes //设置 IP 和子掩
```

```
nmcli connection up eth0
```

防火墙设置为 trusted firewall-cmd --set-default-zone=trusted

关闭 selinux

```
setenforce 0
```

```
getenforce
```

```
vim /etc/selinux/config
```

```
SELINUX=permissive
```

搭建 yum

```
[root@cobbler ~]#cd /etc/yum.repos.d
```

```
[root@cobbler ~]#mkdir repo
```

```
[root@cobbler ~]#mv *repo repo
```

```
[root@cobbler ~]#vim /etc/yum.repos.d/local_repo.repo
```

//修改 yum 配置文件

```
[local_repo]
```

```
name=centos1804
```

```
baseurl="ftp://192.168.4.254/centos-1804"
```

enabled=1

gpgcheck=0

[root@cobbler ~]#yum repolist //验证

[root@cobbler ~]#yum -y install xeyes //验证

4.3 真机传输 cobbler 到虚拟机

[student@room9pc01 ~]\$ scp /home/student/桌面/cobbler.zip root@192.168.4.123:/root/

[root@cobbler ~]# ls

4.4 解压 cobbler.zip

[root@cobbler ~]# unzip cobbler.zip -d /

[root@cobbler ~]# ls /cobbler/

cobbler_boot.tar.gz cobbler_rpm.zip cobbler_web.png

[root@cobbler cobbler]# unzip /cobbler/cobbler_rpm.zip -d /opt/

[root@cobbler cobbler]# ls /opt/cobbler/

4.5 安装 cobbler 主程序\工具包等

[root@cobbler cobbler]# yum -y install dhcp httpd mod_ssl

[root@cobbler cobbler]# yum -y install /opt/cobbler/*.rpm //安装所有 rpm 包

[root@cobbler cobbler]# rpm -q cobbler //验证 cobbler 是否安装

cobbler-2.8.2-1.el7.x86_64

cobbler cobbler 主程序

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
httpd	http 服务
mod_ssl	为 apache web 提供密码保护

4.6 设置 cobbler 配置文件

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

:set nu

272 next_server: 192.168.4.123    //设置下一个服务器为本机

384 server: 192.168.4.123    //设置本机为 cobbler 服务器

242 manage_dhcp: 1    //设置 cobbler 管理 dhcp 服务

292 pxe_just_once: 1    //防止客户端重复安装操作系统
```

4.7 配置 cobbler 的 DHCP

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

末行模式:%s/旧/新/g    //全文替换旧为新

末行模式:%s /192.168.1/192.168.4/g    //全文替换 192.168.1 为 192.168.4
```

4.8 绝对路径解压 cobbler_boot.tar.gz //众多的引导文件

```
[root@cobbler cobbler]# tar -tf /cobbler/cobbler_boot.tar.gz    //查看包内容

[root@cobbler cobbler]# tar -xPf /cobbler/cobbler_boot.tar.gz    //绝对路径释放文件

[root@cobbler cobbler]# ls /var/lib/cobbler/loaders/    //查看绝对路径下的文件
```

4.9 启动相关服务并设置开机自启

```
[root@cobbler ~]# systemctl restart cobblerd  
[root@cobbler ~]# systemctl enable cobblerd  
[root@cobbler ~]# systemctl restart httpd  
[root@cobbler ~]# systemctl enable httpd  
[root@cobbler ~]# systemctl restart tftp  
[root@cobbler ~]# systemctl enable tftp  
[root@cobbler ~]# systemctl restart rsyncd //rsyncd 为同步服务  
[root@cobbler ~]# systemctl enable rsyncd
```

4.10 同步刷新 cobbler 配置

```
[root@cobbler ~]# cobbler sync //检测所有配置是否正确
```

4.11 cobbler 应用

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

```
[root@cobbler ~]# mkdir dvd
```

```
[root@cobbler ~]# mount /dev/cdrom /dvd/
```

```
[root@cobbler ~]# ls /dvd
```

```
[root@cobbler ~]# cobbler import --path=/dvd --name=centos7
```

cobbler 导入的镜像放在:/var/www/cobbler/ks_mirror

4.12 KVM 新机安装系统

步骤同 PXE: root 密码:cobbler

五 环境变量

都为大写

PATH:存放执行命令程序的搜寻路径

```
[root@cobbler /]# vim /opt/hello.sh
```

```
[root@cobbler /]# chmod +x /opt/hello.sh
```

```
[root@cobbler /]# /opt/hello.sh
```

hello

```
[root@cobbler /]# echo $PATH
```

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin //路径任选一个

```
[root@cobbler /]# cp /opt/hello.sh /usr/bin/ //拷贝后即刻生效
```

```
[root@cobbler /]# hello.sh //测试
```

hello

六 权限的掩码 umask 值: 与目录默认权限有关

```
[root@cobbler ~]# umask
```

0022

影响所有新建目录的权限值

新建目录的默认权限值=777-022=755

新建文件的默认权限 644(目录基础上减去了执行权限)

修改 umask 命令:umask 值