一、Linux下软件安装

1. 本地yum源配置

任务背景

某公司内网服务器大多都是最小化安装，某些情况下需要安装一些软件包。但是由于服务器上没有光驱无法挂载光盘配置本地yum源，所以希望在公司内部搭建一个内网源。

任务要求

在跳板机上配置yum源，让内网服务器可以通过网络来下载安装软件包

任务解决方案

思路：
1. 跳板机上有一个iso镜像文件（本地yum仓库）
2. 跳板机上将镜像文件所在的目录共享给服务器（ftp/nfs/samba/http）
经过分析，选择使用ftp来共享
3. 内网服务器配置内网网络yum即可

步骤：
1. 跳板机挂载镜像文件到指定位置

```
[root@jumper ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_misshou-lv_root   18G  9.9G  6.6G  61% /
tmpfs                           491M   72K  491M   1% /dev/shm
/dev/sda1                       485M   35M  426M   8% /boot
/dev/sr0                        4.2G  4.2G     0 100% /media/CentOS_6.5_Final
```

2. 安装vsftpd软件

```
[root@jumper ~]# yum -y install vsftpd
```

3.将光盘里的内容拷贝到/var/ftp目录里

```
[root@jumper ~]# rsync -av /media/CentOS_6.5_Final/ /var/ftp/
```

4. 内网服务器配置网络yum源

```
[root@server yum.repos.d]# pwd
/etc/yum.repos.d
[root@server yum.repos.d]# cat server.repo
[network]
name=ftp share
baseurl=ftp://10.1.1.1
enabled=1
gpgcheck=0
```

5. 测试验证

2. 网络yum源配置

任务1

开发人员需要安装某个软件包（epel源中有），发现现有yum源中没有，需要运维协助配置EPEL源（两种方式搭建EPEL源）

```
方法1：
下载epel软件包安装
epel-release-6-8.noarch.rpm

方法2：
直接配置
https://mirrors.aliyun.com/epel/6/x86_64/
```

任务2

测试环境需要部署zabbix监控和puppet工具，现需要配置能够提供指定软件包的安装源

```
puppet:
http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
http://yum.puppetlabs.com/puppetlabs-release-fedora-20.noarch.rpm

Zabbix:
http://repo.zabbix.com/zabbix/2.4/rhel/6/x86_64/zabbix-release-2.4-1.el6.noarch.rpm
http://repo.zabbix.com/zabbix/2.2/rhel/6/x86_64/zabbix-release-2.2-1.el6.noarch.rpm
```

二、基础服务总结

1. SSH服务

任务背景1

为了最大程度的保护公司内网服务器的安全，公司内部有一台服务器做跳板机。运维人员在维护过程中首先要统一登录到这台服务器，然后再登录到目标设备进行维护和操作。由于开发人员有时候需要通过跳板机登录到线上生产环境查看一些业务日志，所以现在需要运维人员针对不同的人员和需求对账号密码进行统一管理，并且遵循**权限最小化**原则。

任务要求

1. 跳板机上为每个开发人员创建一个账号(code1~code3)，并且只能在指定的目录里管理自己的文件，不能删除别人的文件。（跳板机完成）
2. 线上生产服务器，禁止使用root用户远程登录。（生产服务器完成）
3. 线上生产服务器sshd服务不允许使用默认端口，防止黑客进入端口扫描。（生产服务器完成）
4. 线上生产服务器的业务用户的密码使用工具随机生成。（生产服务器完成）
5. 开发人员可以使用业务用户**pos1**登录线上环境来查看日志（/pos/logs/xxx）

解决方案

```
创建用户：code1~code3
[root@jumper-server ~]# useradd code1
[root@jumper-server ~]# useradd code2
```

```
[root@jumper-server ~]# useradd code3

[root@jumper-server ~]# id code1
uid=500(code1) gid=500(code1) groups=500(code1)
[root@jumper-server ~]# id code2
uid=501(code2) gid=501(code2) groups=501(code2)
[root@jumper-server ~]# id code3
uid=502(code3) gid=502(code3) groups=502(code3)
```

使用非交互式设置密码：

```
[root@jumper-server ~]# echo 123|passwd --stdin code2
Changing password for user code2.
passwd: all authentication tokens updated successfully.
[root@jumper-server ~]# echo 123|passwd --stdin code3
Changing password for user code3.
passwd: all authentication tokens updated successfully.
```

创建相应的目录给开发人员使用：

```
[root@jumper-server ~]# mkdir /data/code -p
[root@jumper-server ~]# ll -d /data/code/
drwxr-xr-x 2 root root 4096 Aug 26 09:39 /data/code/
```

创建code组并且将code1~code3成员加入其中：

```
[root@jumper-server ~]# groupadd code
[root@jumper-server ~]# usermod -G code code1
[root@jumper-server tmp]# gpasswd -a code2 code
Adding user code2 to group code
[root@jumper-server tmp]# id code2
uid=501(code2) gid=501(code2) groups=501(code2),503(code)
[root@jumper-server tmp]# gpasswd -a code3 code
[root@jumper-server tmp]# id code2
uid=501(code2) gid=501(code2) groups=501(code2),503(code)

[root@jumper-server tmp]# tail /etc/group
...
code:x:503:code1,code2,code3
```

更改目录权限：

```
[root@jumper-server ~]# ll -d /data/code/
drwxr-xr-x 2 root root 4096 Aug 26 09:39 /data/code/
[root@jumper-server ~]# chgrp code /data/code/
[root@jumper-server ~]# ll -d /data/code/
drwxr-xr-x 2 root code 4096 Aug 26 09:39 /data/code/
[root@jumper-server ~]# chmod g+w /data/code/
[root@jumper-server ~]# ll -d /data/code/
drwxrwxr-x 2 root code 4096 Aug 26 09:39 /data/code/
```

权限最小化：

```
[root@jumper-server code]# chmod o+t /data/code/
[root@jumper-server code]# ll -d /data/code/
drwxrwxr-t 2 root code 4096 Aug 26 10:44 /data/code/
[root@jumper-server code]# su - code1
[code1@jumper-server ~]$ cd /data/code/
```

```
[code1@jumper-server code]$ ll
total 0
-rw-rw-r-- 1 code1 code1 0 Aug 26 10:43 file1
-rw-rw-r-- 1 code2 code2 0 Aug 26 10:44 file2
[code1@jumper-server code]$ rm -f file2
rm: cannot remove `file2': Operation not permitted
```

禁止root远程登录：
思路：
1. 通过修改配置文件完成
2. 通过查看man文档来找答案

步骤：

```
[root@app1-server ~]# vim /etc/ssh/sshd_config
...
#PermitRootLogin yes
PermitRootLogin no
...

[root@app1-server ~]# /etc/init.d/sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

测试验证：

```
[root@jumper-server ~]# ssh root@10.1.1.1
root@10.1.1.1's password:
Permission denied, please try again.
root@10.1.1.1's password:
Permission denied, please try again.
root@10.1.1.1's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

说明：不能使用root直接登录，但是可以使用其他用户登录成功后切换到root。

```
[root@jumper-server ~]# ssh pos1@10.1.1.1
pos1@10.1.1.1's password:
Last login: Sun Aug 26 11:50:05 2018 from 10.1.1.250
[pos1@app1-server ~]$
[pos1@app1-server ~]$ su - root
Password:
```

将默认的22号端口更改为10022
思路：
1. 查看在当前服务器中10022是否被使用

```
netstat -a|grep 10022
ss -a|grep 10022
lsof -i 10022
grep 10022 /etc/services
```

2. 修改配置文件

```
[root@app1-server ~]# vim /etc/ssh/sshd_config
#Port 22
Port 10022
```

```
[root@app1-server ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]

3. 测试验证
[root@jumper-server ~]# ssh -lpos1 10.1.1.1
ssh: connect to host 10.1.1.1 port 22: Connection refused
[root@jumper-server ~]# ssh -lpos1 10.1.1.1 -p10022
pos1@10.1.1.1's password:
Last login: Sun Aug 26 15:08:20 2018 from 10.1.1.250

说明:
如果更改了端口，那么远程连接时必须指定端口号。
```

任务背景2

经过一段时间后，开发人员和运维人员都觉得使用密码SSH登录的方式太麻烦（每次登录都需要输入密码，难记又容易泄露密码）。为了安全和便利性方面考虑，要求运维人员给所有服务器实现免密码登录。

任务要求

所有开发人员通过远程业务用户pos登录生产服务器实现免密码登录。

解决方案

```
1. 确保线上app1服务器上有pos用户
[root@app1-server ~]# id pos
uid=504(pos) gid=504(pos) groups=504(pos)
[root@app1-server ~]# echo 123|passwd --stdin pos

2. 跳板机上的开发人员code1~code3分别生成一对秘钥
[code1@jumper-server .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/code1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/code1/.ssh/id_rsa.
Your public key has been saved in /home/code1/.ssh/id_rsa.pub.
The key fingerprint is:
21:10:21:06:0b:d0:13:e9:52:7b:89:fc:82:cb:f4:ba code1@jumper-server
The key's randomart image is:
+--[ RSA 2048]----+
|=o+o+o           |
|.o=. .           |
|.+ + .. .        |
|. = o  . .       |
| o o    S        |
|... .            |
|o...             |
|.. .             |
| Eo              |
+-----------------+
```

```
[code1@jumper-server .ssh]$ ll
total 12
-rw------- 1 code1 code1 1675 Aug 28 09:37 id_rsa     私钥
-rw-r--r-- 1 code1 code1  401 Aug 28 09:37 id_rsa.pub 公钥
-rw-r--r-- 1 code1 code1  390 Aug 26 11:27 known_hosts

3. 跳板机上的code1~code3人员将自己的公钥远程拷贝到线上app1的pos用户的加目录里（~/.ssh/xxx）
[code1@jumper-server .ssh]$ ssh-copy-id -i pos@10.1.1.1
pos@10.1.1.1's password:
Now try logging into the machine, with "ssh 'pos@10.1.1.1'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
或者
[code1@jumper-server .ssh]$ scp id_rsa.pub pos@10.1.1.1:/home/pos/.ssh/authorized_keys
pos@10.1.1.1's password:
id_rsa.pub                                                                    100%  401
    0.4KB/s   00:00
[code1@jumper-server .ssh]$

4. 测试验证
[code1@jumper-server ~]$ ssh pos@10.1.1.1
[pos@app1-server ~]$
```

2. RSYNC数据同步

任务背景

某公司为了保证开发人员线上代码的安全性，现需要对开发人员的代码进行备份。

任务要求

1. 线上MIS系统上的部分java代码需要备份到另外一台主机上。
2. 线上MIS系统服务器的java代码存放目录为：/app/java_project。
3. 每天凌晨1:03分备份机器上自动同步MIS上/app/java_project目录里的内容

解决方案

```
1. 在线上环境创建相应的目录
[root@app1-server ~]# mkdir /app/java_project -p
[root@app1-server ~]# touch /app/java_project/file{1..5}

2. 在线上环境中将rsync作为后台程序运行
1）创建配置文件
[root@app1-server ~]# cat /etc/rsyncd.conf
[app1]
path = /app/java_project
log file = /var/log/rsync.log
```

```
2）作为后台程序启动它
[root@app1-server ~]# rsync --daemon
3）查看端口是否处于监听状态
[root@app1-server ~]# ss -nltp|grep 873
LISTEN     0      5                        :::873                     :::*      users:
(("rsync",7875,5))
LISTEN     0      5                         *:873                      *:*      users:
(("rsync",7875,4))

3. 在备份机上创建备份目录
mkdir /backup

[root@jumper-server ~]# rsync -a 10.1.1.1::      查看远程主机的模块名
app1

使用命令将线上环境的文件拉取到本地
[root@jumper-server ~]# rsync -av 10.1.1.1::app1 /backup/
或者
[root@jumper-server ~]# rsync -av rsync://10.1.1.1/app1 /backup/

说明:
作为后台服务运行时，不需要密码就会直接同步。

4. 编写脚本，然后交给计划任务去执行
[root@jumper-server ~]# cat /root/1.sh
#!/bin/bash
rsync -av rsync://10.1.1.1/app1 /backup/

[root@jumper-server ~]# crontab -l
03 01 * * * /root/1.sh &>/dev/null


5. rsync服务开机自动启动
[root@app1-server ~]# cat /etc/rc.local      开机最后读取的一个文件
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
rsync --daemon             增加该行即可
```

3. FTP服务

任务背景

某创业公司刚刚起步，随着业务量的增多，咨询和投诉的用户也越来越多，公司的客服部门由原来的2个增加到5个。客服部门平时需要处理大量的用户反馈，不管是邮件，还是ＱＱ，还是电话，客服人员都会针对每一次的用户反馈做详细的记录，但是由于客观原因，客服人员没有成熟稳定的客户服务系统，所以希望运维部门能够提供一个可以通过浏览器查看并下载的方式来管理这些文档，并且随时跟踪客户的反馈情况。

任务要求

1. 客服人员必须使用用户名密码(kefu/123)的方式登录服务器来下载相应文档。
2. 不允许匿名用户访问
3. 客服部门的相关文档保存在指定的目录里/data/kefu
4. 客服用户使用用户kefu/123登录后就只能在默认的/data/kefu目录里活动

解决方案

```
1. 在ftp-server端创建用户并且给密码
[root@ftp-server ~]# useradd kefu
[root@ftp-server ~]# echo 123|passwd --stdin kefu

2. 禁止匿名用户访问FTP服务
[root@ftp-server vsftpd]# vim vsftpd.conf
anonymous_enable=NO

重启服务测试验证
[root@client ~]# ftp 10.1.1.1
Connected to 10.1.1.1 (10.1.1.1).
220 (vsFTPd 2.2.2)
Name (10.1.1.1:root): ftp
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> exit
221 Goodbye.
[root@client ~]# lftp 10.1.1.1
lftp 10.1.1.1:~> ls
Interrupt
lftp 10.1.1.1:~> exit

3. 客服人员相关的文档保存在指定的目录里/data/kefu
1) 在ftp服务端创建相应的目录来保存文件
# mkdir /data/kefu -p
2) 通过修改配置文件告诉ftp服务将文件保存到指定目录里
local_root=/data/kefu
3) 重启服务测试验证
[root@client ~]# ftp 10.1.1.1
Connected to 10.1.1.1 (10.1.1.1).
220 (vsFTPd 2.2.2)
Name (10.1.1.1:root): kefu
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/data/kefu"

4. 禁锢本地用户的数据目录
修改配置文件
chroot_local_user=YES
重启服务测试验证
```

```
[root@client ~]# ftp 10.1.1.1
Connected to 10.1.1.1 (10.1.1.1).
220 (vsFTPd 2.2.2)
Name (10.1.1.1:root): kefu
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp> cd /etc
550 Failed to change directory.
ftp> cd /home
550 Failed to change directory.
```

4. NFS服务

任务背景

由于业务驱动，为了提高用户的访问效率，现需要将原有web服务器上的静态资源文件分离出来，单独保存到一台文件服务器上。

任务要求

1. 一台应用服务器web-server部署apache，静态网页资源存放在另外一台NFS服务器上
2. 对于NFS服务器上保存的静态资源实行实时备份

解决方案

```
环境准备：
web-server:10.1.1.1        安装httpd软件并且启动服务
nfs-server:10.1.1.250
backup:10.1.1.2

步骤：
1. 搭建web服务（在10.1.1.1 web-server完成）
[root@web-server ~]# service iptables stop
[root@web-server ~]# chkconfig --list|grep iptables
iptables          0:off 1:off 2:off 3:off 4:off 5:off 6:off
[root@web-server ~]# getenforce
Disabled

[root@web-server ~]# rpm -q httpd
httpd-2.2.15-29.el6.centos.x86_64

[root@web-server ~]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for web-server
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1
for ServerName
```

```
                                                           [  OK  ]
[root@web-server ~]# netstat -nltp|grep 80
tcp        0      0 :::80                       :::*                        LISTEN
 5842/httpd
```

测试静态页面:

```
[root@web-server ~]# echo this is test page > /var/www/html/index.html
IE:
http://10.1.1.1
Linux:
[root@web-server ~]# yum -y install elinks
[root@web-server ~]# elinks http://10.1.1.1
```

2. 搭建NFS服务 (nfs-server 10.1.1.250)

1) 安装相关软件

```
[root@nfs-server ~]# rpm -q rpcbind
package rpcbind is not installed
[root@nfs-server ~]# rpm -aq|grep ^nfs

[root@nfs-server ~]# yum -y install nfs-utils rpcbind

[root@nfs-server ~]# rpm -aq|grep ^nfs
nfs-utils-lib-1.1.5-6.el6.x86_64
nfs-utils-1.2.3-39.el6.x86_64
[root@nfs-server ~]# rpm -q rpcbind
rpcbind-0.2.0-11.el6.x86_64
```

2) 查看软件包的文件列表

```
/etc/init.d/rpcbind
/etc/init.d/nfs

/etc/exports    定义共享的文件和共享给谁
```

3) 创建一个共享目录给web-server来存放静态文件

```
[root@nfs-server ~]# mkdir /share/data -p
```

4) 将共享目录共享给web-server端

```
[root@nfs-server ~]# cat /etc/exports
/share/data 10.1.1.0/24(rw)
```

5) 启动服务

```
[root@nfs-server ~]# service rpcbind start
Starting rpcbind:                                          [  OK  ]
[root@nfs-server ~]# service nfs start
Starting NFS services:                                     [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting RPC idmapd:                                       [  OK  ]

[root@nfs-server ~]# netstat -nltp|grep 111
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
 1805/rpcbind
```

```
tcp        0      0 :::111                  :::*                    LISTEN
 1805/rpcbind


3. web-server (10.1.1.1) 端挂载使用共享目录
[root@web-server ~]# mount -t nfs 10.1.1.250:/share/data /var/www/html/

开机自动挂载:
echo "mount -t nfs 10.1.1.250:/share/data /var/www/html/" >> /etc/rc.local

4. 客户端测试验证
IE:
http://10.1.1.1
Linux:
[root@web-server ~]# elinks http://10.1.1.1

5. 实时备份nfs-server上的静态资源文件，以下内容在nfs-server完成
1) 在nfs-server上安装inotify-tools工具
解压:
[root@nfs-server soft]# tar -xf inotify-tools-3.13.tar.gz
进入解压目录安装:
[root@nfs-server soft]# cd inotify-tools-3.13
[root@nfs-server inotify-tools-3.13]# ./configure
[root@nfs-server inotify-tools-3.13]# make
[root@nfs-server inotify-tools-3.13]# make install

2) 编写脚本实时监控
[root@nfs-server ~]# cat 1.sh
#!/bin/bash
/usr/local/bin/inotifywait -mrq -e modify,delete,create,attrib,move /share/data |while read
events
        do
                rsync -a --delete /share/data/ 10.1.1.2:/web_backup
                echo "`date +%F\ %T`出现事件$events" >> /var/log/rsync.log 2>&1
        done

后台执行脚本:
./1.sh &

3) 测试验证
```

5. SAMBA服务

任务背景

某公司为了方便部门资料的共享，需要针对不同部门不同成员来共享一些资料，现需要运维人员来根据具体需求协助完成文件共享服务的搭建。

任务要求

1. 财务部门,资料目录为/smb/itcast_cw ， 财务总监(cw01)有可读可写权限，财务部门员工可读，boss01对其有管理权限。

2. 市场部门,资料目录为/smb/itcast_sc，市场部门员工可读可写，**公司员工**可以查询资料，boss02对其有管理权限(可读写)，vip用户可以查询。
3. 人事部门,资料目录为/smb/itcast_rs ，人事总监(rs01)可读写，人事部门员工可以对市场部查询
4. 公共区，公共目录为/smb/itcast_pub，要求只能自己管理自己的文件，不能删除别人的文件,只允许公司内部员工使用

解决方案

```
1. 创建相应的目录用户组
[root@smb-server ~]# mkdir /smb/{cw,rs,sc,pub} -p
[root@smb-server ~]# groupadd itcast
[root@smb-server ~]# groupadd cw
[root@smb-server ~]# groupadd rs
[root@smb-server ~]# groupadd sc
[root@smb-server ~]# useradd cw01 -g cw -G itcast
[root@smb-server ~]# useradd cw02 -g cw -G itcast
[root@smb-server ~]# useradd rs01 -g rs -G itcast
[root@smb-server ~]# useradd rs02 -g rs -G itcast
[root@smb-server ~]# useradd sc01 -g sc -G itcast
[root@smb-server ~]# useradd sc02 -g sc -G itcast
[root@smb-server ~]# useradd boss01 -g itcast
[root@smb-server ~]# useradd boss02 -g itcast
[root@smb-server ~]# useradd vip

2. 统一更改权限：原则权限最小化
//权限最小化
[root@smb-server ~]# chmod 750 -R /smb
[root@smb-server ~]# chgrp itcast /smb
[root@smb-server ~]# chgrp cw /smb/cw
[root@smb-server ~]# chgrp rs /smb/rs
[root@smb-server ~]# chgrp sc /smb/sc
[root@smb-server ~]# chgrp itcast /smb/pub

3. 搭建samba服务，根据需求完成任务
[root@smb-server ~]# vim /etc/samba/smb.conf
追加如下内容：
[cw]
   path=/smb/cw
   valid users = boss01,@cw
    write list = cw01,boss01


 [sc]
   path=/smb/sc
   valid users = @itcast,vip
   write list = @sc,boss02

[rs]
   path=/smb/rs
   valid users = @rs
   write list = rs01

[pub]
```

```
   path=/smb/pub
   valid users = @itcast
   writable = yes
```

4. 将用户加入到smb数据库里

```
[root@smb-server ~]# smbpasswd -a cw01
New SMB password:
Retype new SMB password:
Added user cw01.
[root@smb-server ~]# smbpasswd -a cw02
New SMB password:
Retype new SMB password:
Added user cw02.
[root@smb-server ~]# smbpasswd -a rs01
New SMB password:
Retype new SMB password:
Added user rs01.
[root@smb-server ~]# smbpasswd -a rs02
New SMB password:
Retype new SMB password:
Added user rs02.
```

。。。。

5. 启动服务

```
[root@smb-server ~]# service nmb start
Starting NMB services:                                     [  OK  ]
[root@smb-server ~]# service smb start
Starting SMB services:                                     [  OK  ]
[root@smb-server ~]#
```

6. 测试验证

```
[root@client ~]# smbclient //10.1.1.250/cw -U cw01
Enter cw01's password:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.6.23-51.el6]
smb: \> ls      测试是否可以查看
NT_STATUS_ACCESS_DENIED listing \*           无法查看cw部门的资料
smb: \>
```

解决:

```
[root@smb-server ~]# ll -d /smb
drwxr-x--- 6 root root 4096 Sep  4 17:05 /smb
[root@smb-server ~]# setfacl -m g:itcast:rx /smb/

smb: \> mkdir bbb    测试是否可以写
NT_STATUS_ACCESS_DENIED making remote directory \bbb
```

解决:

```
[root@smb-server ~]# chmod g+w /smb/cw
```

补充:

```
[root@smb-server ~]# setfacl -x g:itcast /smb/        //删除一个文件上指定的acl
[root@smb-server ~]# getfacl /smb/
getfacl: Removing leading '/' from absolute path names
# file: smb/
# owner: root
# group: root
user::rwx
group::r-x
mask::r-x
other::---

[root@smb-server ~]# setfacl -b /smb/           //删除文件上所有的acl策略
[root@smb-server ~]# getfacl /smb/
getfacl: Removing leading '/' from absolute path names
# file: smb/
# owner: root
# group: root
user::rwx
group::r-x
other::---
```

6. DHCP服务

任务背景

运维负责的工作不仅只有服务器， 公司内部的相关I T 支持也需要懂得（很多时候 不见得运维要亲自去做，但是必须懂）。目前公司所有员工加一起共30多人，每个人都有自己的办公电脑，再加上手机等设备基本上总设备在80-100左右。设备的大量增多需要公司的内网划分出明确的内网网段， 之后 IP地址的分配肯定不能手动设定，这就要用到DHCP 来动态分配IP地址。于是乎运维需要架设DHCP服务器（DHCP大多多直接使用网络设备分发，现在也有使用Linux服务架设DHCP服务，然后分网段分发再结合交换机 ）

任务要求

搭建一台DHCP服务器，给办公人员提供自动分配**172.16.0.0/24**网段的IP地址

解决方案

```
# vim /etc/dhcp/dhcpd.conf
option domain-name "itcast.cc";
option domain-name-servers 172.16.0.254, 8.8.8.8;
default-lease-time 3600;
max-lease-time 7200;
log-facility local7;

subnet 10.1.1.0 netmask 255.255.255.0 {
  range 10.1.1.100 10.1.1.200;
  option routers 10.1.1.254;
  option broadcast-address 10.1.1.255;
}
```

注意：DHCP服务器必须要有10.1.1.0/24网段的IP地址存在。