一、基础命令串讲

1. 用户组管理

重点掌握：

```
useradd
usermod
groupadd
gpasswd
chage
```

课堂实战：

1. 添加3个用户，用户harry，natasha，sarsh，要求harry，natasha用户点附加组为admin组，sarsh用户点登录shell为非交互式shell。密码均为redhat。
2. 修改harry用户的家目录为/home/itcast/redhat/harry .
3. 修改natasha，sarsh用户的主组为itcast，并且可以登录系统.
4. 新建一个公司名称itcast，3个部门 cw， rs， sc；每个部门建立2个用户，如 cw01 cw02，rs01，rs02，sc01，sc02；boss01 管理公司所有部门；所有用户的密码设置为“123456”.
5. 用户账号有效期3个月(90天)，第一次登录强制修改密码，每隔15天更新一次密码。

2. 权限管理

理解：

普通权限（rwx）的含义；

三个特殊权限的含义；

ACL访问控制策略的用途；

遮罩权限（默认权限）；

重点掌握：

```
chmod
chown
chgrp
setfacl
getfacl
umask
```

熟悉常见环境变量和shell的配置文件：

```
/etc/profile
系统和用户的环境变量信息，当用户第一次登录时,该文件被读取
/etc/bashrc
```

每个运行的bash信息（系统别名、函数及默认权限的定义），当bash shell被打开时,该文件被读取
~/.bashrc
当前用户的bash信息，当用户登录和每次打开新的shell时该文件被读取
~/.bash_profile
当前用户的环境变量，当用户登录时，该文件被读取
~/.bash_history 记录历史命令
~/.bash_logout 当用户退出bash或者终端模式下退出登录会首先执行该文件里的代码，然后再退出。

用户登录后所读取相关环境的顺序：
/etc/profile(系统和每个用户的环境变量信息)—> ~/.bash_profile(当前用户的环境变量)—> ~/.bashrc（当前用户的bash信息）—> /etc/bashrc（每个运行的bash信息）—> ~/.bash_logout （退出bash shell时执行该文件）

课堂实战：

1. 使用普通用户stu1登录系统，并在/u01/STU1目录下创建一个文件zhangsan，内容为：I am jack, I want to study hard,I can do it,come on。
2. 使用stu2用户登录系统，并修改stu1用户刚刚创建的文件zhangsan，增加内容：我要和你挑战！并在相同的目录下创建一个自己的文件lisi，内容同上 I want to challenge you 。
3. stu3用户同时可以查看stu1和stu2两个用户的文件，但是不能做任何修改。

3. 文件操作

重点掌握：

3.1 文件查找

命令查找：
which command //在PATH环境变量中查找
whereis command //查找一个命令的二进制文件，源码及手册

普通文件查找：
find 精确查找，磁盘搜索， io读写，cpu开销大
命令格式：
用法1：根据选项查找文件并输出到屏幕
find path -option 关键字
用法2：根据选项找出文件并做出相应处理
find path -option 关键字 [-exec 或 -ok command] {} \;

命令选项：
-name 按照文件名查找文件
-iname 按照文件名忽略大小写查找
-perm 按照文件权限来查找文件
-size 按照文件大小来查找
-type 按照文件类型来查找
-user 属主
-group 属组
-mtime +n 按文件更改时间来查找文件，-n指n天以内，+n指n天以前
-atime -n 按文件访问时间来查
-ctime n 按文件创建时间来查找文件，-n指n天以内，+n指n天以前

动作：

-exec：对匹配的文件执行该参数所给出的shell命令。
-ok： 和-exec的作用相同，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。
-delete：删除文件
-ls：列出文件
-print：打印

示例：
```
# find /home -user jack -group hr
# find /home -user jack -a -group hr
# find /home -user jack -o -group hr
```

扩充：管道"|"和xargs的区别

|:将上一个命令所执行的结果作为下一条命令的标准输入
xargs:将上一条命令所执行的结果作为下一条命令的参数

-n：指定单行显示的参数个数
-d：定义分割符，默认是以空格和换行符
-i|-I：指定替换字符，用于替换。{}

```
[root@node1 data]# find /data/ -type f -mtime -2|xargs -ti mv {} {}.bak
mv /data/2018-07-07.dp /data/2018-07-07.dp.bak
mv /data/2018-07-08.dp /data/2018-07-08.dp.bak
mv /data/2018-07-06.dp /data/2018-07-06.dp.bak
```

3.2 文件打包压缩

压缩工具：
zip：兼容类unix与windows，可以压缩多个文件或目录
-r:递归压缩
用法：zip 压缩后的文件 需要压缩的文件
注意：
zip压缩默认压缩后的格式就是.zip；当然也可以加后缀.zip,一般都加上
解压：
unzip
-d:指定解压目录

gzip|bzip2|xz：压缩单个文件,用法相同
用法：gzip 需要压缩的文件

gzip：压缩速度快，压缩率低，cpu开销比较低
解压：
gunzip或者gzip -d

bzip2：压缩速度慢，压缩率高，cpu开销大
解压：
bzip2或者bzip2 -d
xz：压缩率高，解压速度快，压缩时间较长，cpu消耗相对较大
解压：
unxz 或者 xz -d

```
打包工具：
tar：打包(压缩)多个文件，不会改变文件的属性与权限
用法：tar 选项 打包压缩后的文件   需要打包压缩的文件
参数：
-c 打包
-z gzip
-j bzip2
-J xz
-v 显示详细信息
-f 指定包名
-x 解压
-r：读取追加文件
-C：指定解压路径

注意：
  1. 如果已经将文件压缩打包，那么就不能追加；如果只是打包就可以追加。
  2. 参数顺序需要注意，最好把-f参数放到所有参数后面。
  3. 当出现以下提示时，加一个大P参数解决。
        tar: Removing leading `/' from member names
```

课堂实战：

实战1：

在/home/test目录中创建10个文件，并且修改file1到file5的时间为当前系统时间的5天前，file6的时间为9月1号，file7的时间为9月2 号。

要求：

1）找出5天以前的文件并将其删掉。

2）找出昨天的文件并将其拷贝到10.1.1.2(自己另一台主机)上的/home/test目录里，redhat用户密码为123

实战2：

1. 找出根下的所有块设备文件，并且将标准输出及标准错误重定向到/tmp/find.test文件中
2. 找出/etc/下面以.conf结尾的文件，并将其复制到/home/下的backup目录中
3. 将/home/backup下的所有文件全部打包压缩到/tmp/当前系统日期.tar.gz
4. 将/tmp/当前系统日期.tar.gz解压到新建目录/tmp/test中，并打包成"当前系统日期.tar"

4. 进程控制

重点掌握：

```
静态查看进程信息ps：
ps aux
ps auxf：显示父子关系
ps -ef

a：显示现行终端机下的所有进程，包括其他用户的进程；
u：显示进程拥有者、状态、资源占用等的详细信息（注意有“-”和无“-”的区别）。
x：显示没有控制终端的进程。通常与 a 这个参数一起使用，可列出较完整信息。
-e：显示所有进程。
```

-f：完整输出显示进程之间的父子关系
-l：较长、较详细的将该 PID 的的信息列出

pidof：查看指定进程的PID
pstree：查看进程树

```
[root@MissHou ~]# ps aux|head
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  19356  1432 ?        Ss   19:41   0:03 /sbin/init
```

USER:运行进程的用户
PID:进程ID
%CPU:CPU占用率
%MEM:内存占用率
VSZ:占用虚拟内存
RSS:占用实际内存 驻留内存
TTY:进程运行的终端
STAT:进程状态 man ps (/STATE)

R 运行
S 可中断睡眠 Sleep
D 不可中断睡眠
T 停止的进程
Z 僵尸进程
X 死掉的进程

Ss s进程的领导者，父进程
S< <优先级较高的进程
SN N优先级较低的进程
R+ +表示是前台的进程组
Sl 以线程的方式运行

START: 进程的启动时间
TIME: 进程占用CPU的总时间
COMMAND: 进程文件，进程名

动态查看进程信息top：
在top的执行过程中，还可以使用以下的按键命令：
h|? 帮助
M 按内存的使用排序
P 按CPU使用排序
N 以PID的大小排序
R 对排序进行反转
f 自定义显示字段
1 显示所有CPU的负载
T：按该进程使用的CPU时间累积排序
k：给某个PID一个信号（signal），默认值是信号15
r：重新安排一个进程的优先级别
s：改变两次刷新之间的时间。默认是5秒
q：退出程序。

top命令常用的选项：
-d：后面可以接秒数，指定每两次屏幕信息刷新之间的时间间隔；
-p：指定某个进程来进行监控；
-b -n：以批处理方式执行top命令。通常使用数据流重定向，将处理结果输出为文件；

```
[root@MissHou ~]# top
[root@MissHou ~]# top -d 1
[root@MissHou ~]# top -d 1 -p 10126                  查看指定进程的动态信息
[root@MissHou ~]# top -d 1 -u apache                 查看指定用户的进程
[root@MissHou ~]# top -d 1 -b -n 2 > top.txt         将2次top信息写入到文件


进程控制：
kill,killall,pkill

给进程发送信号
# kill -l          列出所有支持的信号
编号 信号名
1) SIGHUP       重新加载配置
2) SIGINT       键盘中断^C
3) SIGQUIT      键盘退出
9) SIGKILL      强制终止
15) SIGTERM     终止（正常结束），缺省信号
18) SIGCONT     继续
19) SIGSTOP     停止
20)SIGTSTP      暂停^Z


# fg  把最后放在后台运行的进程再次放到终端前台
# fg %1  将作业1调回到前台
# bg %2 把后台编号为2的进程恢复运行状态
```

二、基础服务串讲

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。

7. DNS服务

任务背景

随着公司内部服务器的不断增多，现阶段公司所有服务器虽然都配置了主机名和IP的hosts解析，考虑到hosts文件的更新滞后，为了长远考虑，我们需要搭建一台DNS服务器，所有服务器都统一使用DNS服务来解析。

任务要求

搭建一台DNS服务器，配置正反向解析

```
环境如下：
以下主机属于web.cluster域
dns-server  10.1.1.250     提供主机名的正向和反向解析

web1  10.1.1.10   web1.web.cluster
web2  10.1.1.20   web2.web.cluster
web3  10.1.1.30   web3.web.cluster
app1  10.1.1.1    app1.web.cluster
app2  10.1.1.2    app2.web.cluster
app3  10.1.1.3    app3.web.cluster

搭建DNS服务，统一解析以上主机的域名(实现正反向解析)
```

解决方案

```
1. 关闭防火墙和selinux
2. 配置yum源
3. 软件三步曲
1) 安装软件   bind
[root@dns-server ~]# yum -y install bind
2) 确定是否成功安装
[root@dns-server ~]# rpm -q bind
bind-9.8.2-0.17.rc1.el6_4.6.x86_64
3) 查看软件的文件列表
[root@dns-server ~]# rpm -ql bind
/etc/logrotate.d/named        日志轮转文件
/etc/named                    配置文件的主目录
/etc/named.conf               主配置文件
/etc/named.rfc1912.zones       zone文件，定义域
/etc/rc.d/init.d/named         启动脚本

/usr/sbin/named               二进制命令
/usr/sbin/named-checkconf     检查配置文件的命令   named.conf   named.rfc1912.zones
/usr/sbin/named-checkzone     检查区域文件的命令

/var/log/named.log            日志文件
```

```
/var/named                    数据文件的主目录
/var/named/data
/var/named/named.ca           根域服务器
/var/named/named.empty
/var/named/named.localhost    正向解析区域文件的模板
/var/named/named.loopback     反向解析区域文件的模板
/var/named/slaves          从dns服务器下载文件的默认路径
/var/run/named             进程文件
```

4. 根据需求通过修改配置文件来完成服务的搭建

1) 修改主配置文件

```
[root@dns-server ~]# vim /etc/named.conf
options {
   listen-on port 53 { 127.0.0.1;any; };     定义监听方式，any代表全网监听
   listen-on-v6 port 53 { ::1; };
   directory   "/var/named";
   dump-file   "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
   allow-query     { localhost;any; };       允许任何人来查询
   recursion yes;
.....
};
....
```

2) 修改子配置文件

```
[root@dns-server ~]# vim /etc/named.rfc1912.zones
```

在该文件的后面追加以下内容：

```
zone "web.cluster" IN {
        type master;
        file "web.cluster.zone";
        allow-update { none; };
};
zone "1.1.10.in-addr.arpa" IN {
        type master;
        file "10.1.1.zone";
        allow-update { none; };
};
```

3) 需要在指定的数据目录里创建相应的zone文件

```
[root@dns-server ~]# cd /var/named/
[root@dns-server named]# cp -p named.localhost web.cluster.zone
[root@dns-server named]# cp -p named.loopback 10.1.1.zone
```

4) 修改相应的区域文件（正向和反向）

```
[root@dns-server named]# cat web.cluster.zone
$TTL 1D
@  IN SOA   web.cluster. rname.invalid. (
               0  ; serial
               1D ; refresh
               1H ; retry
               1W ; expire
```

```
                3H )   ; minimum
@   NS abc.web.cluster.
abc    A   10.1.1.250
web1   A   10.1.1.10
web2   A   10.1.1.20
web3   A   10.1.1.30
app1   A   10.1.1.1
app2   A   10.1.1.2
app3   A   10.1.1.3
```

注意：NS代表nameserver，前两行必须指定当前DNS服务的IP，abc.web.cluster.

```
[root@dns-server named]# cat 10.1.1.zone
$TTL 1D
@   IN SOA    web.cluster. rname.invalid. (
                0   ; serial
                1D ; refresh
                1H ; retry
                1W ; expire
                3H )   ; minimum
@   NS abc.web.cluster.
10 PTR    web1.web.cluster.
20 PTR    web2.web.cluster.
30 PTR    web3.web.cluster.
1   PTR    app1.web.cluster.
2   PTR    app2.web.cluster.
3   PTR    app3.web.cluster.
```

注意：在反向记录文件中前面的A记录可以省略

5. 语法检测

```
[root@dns-server named]# named-checkconf /etc/named.conf
[root@dns-server named]# named-checkconf /etc/named.rfc1912.zones
[root@dns-server named]# named-checkzone web.cluster.zone web.cluster.zone
zone web.cluster.zone/IN: loaded serial 0
OK
[root@dns-server named]# named-checkzone 10.1.1.zone 10.1.1.zone
zone 10.1.1.zone/IN: loaded serial 0
OK
```

6. 启动服务，测试验证

```
[root@dns-server named]# service named start
Generating /etc/rndc.key:                                  [  OK  ]
Starting named:                                            [  OK  ]
[root@dns-server named]# netstat -nulp|grep named
udp        0      0 10.1.1.250:53               0.0.0.0:*
1550/named
udp        0      0 192.168.159.152:53          0.0.0.0:*
1550/named
udp        0      0 127.0.0.1:53                0.0.0.0:*
1550/named
udp        0      0 ::1:53                      :::*
 1550/named
```

```
[root@client ~]# echo nameserver 10.1.1.250 > /etc/resolv.conf
[root@client ~]# cat /etc/resolv.conf
nameserver 10.1.1.250
[root@client ~]# nslookup web1.web.cluster
Server:     10.1.1.250
Address: 10.1.1.250#53

Name: web1.web.cluster
Address: 10.1.1.10
[root@client ~]# nslookup 10.1.1.20
Server:     10.1.1.250
Address: 10.1.1.250#53

20.1.1.10.in-addr.arpa  name = web2.web.cluster.

[root@client ~]# nslookup app1.web.cluster
Server:     10.1.1.250
Address: 10.1.1.250#53

Name: app1.web.cluster
Address: 10.1.1.1

[root@client ~]# host app2.web.cluster
app2.web.cluster has address 10.1.1.2
```

8. 源码构建LAMP环境

任务背景

随着业务不断增长，用户基数越来越大，为更好满足用户体验，开发人员提一个工单过来，需要运维人员给开发人员部署一套预发布测试环境（和生产环境保持一致），能够保证开发人员高效的进行预发布测试等工作。

任务要求

1. 源码部署LAMP环境，和生产保持一致
2. 发布2个网站，一个WordPress博客，一个mysql的web管理网站phpMyAdmin

解决方案

1. 环境准备

```
1. 清空环境、安装相应的软件包
# yum groupinstall "Development tools" -y
# yum groupinstall "Desktop Platform Development" -y   桌面开发工具包（图形化相关包）
# yum -y install cmake
# yum -y install ncurses-devel
# yum -y install pcre-devel
# yum -y install libcurl-devel
2. 根据自己的环境，如果有安装其他版本的mysql和httpd请卸载它
# yum -y remove httpd mysql-libs
3. 将自己的主机名和ip地址绑定写在/etc/hosts文件中
```

2. 编译安装MySQL

```
1. 官方网站下载相应的软件包
mysql-5.6.25.tar.gz
2. 解压软件包
# tar -xf mysql-5.6.25.tar.gz
3. 安装
0) 创建mysql用户
# useradd mysql -r -s /sbin/nologin
1)  配置
根据需求配置：
[root@server mysql-5.6.25]# vim cmake.sh
cmake . \
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql/ \
-DMYSQL_DATADIR=/usr/local/mysql/data \
-DENABLED_LOCAL_INFILE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DSYSCONFDIR=/usr/local/mysql/etc \
-DMYSQL_UNIX_ADDR=/tmp/mysql.sock \
-DMYSQL_TCP_PORT=3306 \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DWITH_EXTRA_CHARSETS=all \
-DMYSQL_USER=mysql

[root@server mysql-5.6.25]# chmod +x cmake.sh
[root@server mysql-5.6.25]# ./cmake.sh
2) 编译安装
[root@server mysql-5.6.25]# make && make install

4. 初始化数据库
[root@server mysql-5.6.25]# cd /usr/local/mysql/
[root@server mysql]# scripts/mysql_install_db --user=mysql

5. 拷贝启动脚本到/etc/init.d/目录
[root@server mysql]# cp ./support-files/mysql.server /etc/init.d/mysql

6. 启动mysql数据库
[root@server mysql]# service mysql start
```

```
7. 修改环境变量
[root@server mysql]# echo "export PATH=/usr/local/mysql/bin:$PATH" >> /etc/profile
[root@server mysql]# source /etc/profile
```

3. 编译安装Apache

```
安装apr软件:
# tar xf apr-1.5.2.tar.bz2 -C /usr/src/
# cd /usr/src/apr-1.5.2
# ./configure
# make
# make install
安装apr-util软件:
# tar xf apr-util-1.5.4.tar.bz2 -C /usr/src/
# cd /usr/src/apr-util-1.5.4/
# ./configure --with-apr=/usr/local/apr/bin/apr-1-config    指定软件apr的路径
# make
# make install

安装httpd软件:
1. 官网下载
httpd-2.4.12.tar.bz2
2. 解压
tar -xf httpd-2.4.12.tar.bz2
3. 安装
1) 配置
[root@server httpd-2.4.12]# vim apache.sh
./configure \
--enable-modules=all \
--enable-mods-shared=all \
--enable-so \
--enable-rewrite \
--enable-ssl \
--with-mpm=prefork \
--with-apr=/usr/local/apr/bin/apr-1-config \
--with-apr-util=/usr/local/apr/bin/apu-1-config

[root@server httpd-2.4.12]# chmod +x apache.sh
[root@server httpd-2.4.12]# ./apache.sh
2) 编译安装
[root@server httpd-2.4.12]# make && make install
```

4. 编译安装php

```
1. 官网下载
php-5.6.11.tar.xz
2. 解压
tar -xf php-5.6.11.tar.xz
3. 安装
1) 配置
[root@server php-5.6.11]# vim php.sh
```

```
./configure \
--with-apxs2=/usr/local/apache2/bin/apxs \
--with-mysql=/usr/local/mysql/ \
--with-mysqli=/usr/local/mysql/bin/mysql_config \
--with-pdo-mysql=/usr/local/mysql/ \
--with-zlib \
--with-zlib-dir=/usr/local/mysql/zlib \
--with-curl \
--enable-zip \
--with-gd \
--with-freetype-dir \
--with-jpeg-dir \
--with-png-dir \
--enable-sockets \
--with-xmlrpc \
--enable-soap \
--enable-opcache \
--enable-mbstring \
--enable-mbregex \
--enable-pcntl \
--enable-shmop \
--enable-sysvmsg \
--enable-sysvsem \
--enable-sysvshm \
--enable-calendar \
--enable-bcmath

2）编译安装
[root@server php-5.6.11]# make && make install

3）确认php成功安装：
[root@server php-5.6.11]# ls /usr/local/apache2/modules/libphp5.so
注意：确认有这个libphp5.so模块文件，就表示编译php成功
```

5. 后续配置

```
1. 修改apache主配置文件
# vim /usr/local/apache2/conf/httpd.conf

//配置语言支持
LoadModule negotiation_module modules/mod_negotiation.so 此模块打开注释
Include conf/extra/httpd-languages.conf          打开此选项，扩展配置文件就生效了
...

//打开对虚拟主机的支持
Include conf/extra/httpd-vhosts.conf

//加载php模块解析php页面,添加两行，告诉httpd把.php文件交给模块去编译
LoadModule php5_module    modules/libphp5.so 找到这一句，在这句下面加上两句
//添加以下两行意思是以.php结尾的文件都认为是php程序文件，注意两句话的.php前面都是有一个空格的
AddHandler php5-script   .php
AddType text/html  .php
```

```
//默认主页加上index.php,并放在index.html前,支持php的首页文件
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>

2. 修改apache的子配置文件，优先支持中文
# vim /usr/local/apache2/conf/extra/httpd-languages.conf
DefaultLanguage zh-CN      打开注释，默认语言集改为中文
LanguagePriority zh-CN en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru
sv tr zh-TW    语言集优先集，把zh-CN 写到前面

3. 修改php.ini文件连接数据库
进入到php软件的解压目录里
[root@server php-5.6.11]# cp php.ini-production /usr/local/lib/php.ini
[root@server php-5.6.11]# vim /usr/local/lib/php.ini
.....
[MySQL]
mysql.default_port = 3306     改成对应的mysql的端口
mysql.default_socket = /tmp/mysql.sock 对应的socket文件地址

[MySQLi]
mysqli.default_port = 3306
mysqli.default_socket = /tmp/mysql.sock
```

6. 发布网站

```
1. 创建相应的数据目录
[root@server LAMP]# mkdir /www/{admin,myblog} -p

2. 将2个网站相关的程序文件解压拷贝到指定的目录里
[root@server LAMP]# unzip phpMyAdmin-4.4.11-all-languages.zip
[root@server LAMP]# cp -a phpMyAdmin-4.4.11-all-languages/* /www/admin/
[root@server LAMP]# tar xf wordpress-4.7.3-zh_CN.tar.gz
[root@server LAMP]# cp -a wordpress/* /www/myblog/

3. 修改网站数据目录的属主和属组
[root@server LAMP]# chown -R daemon. /www/

4. 配置虚拟主机:
注意:
确保在主配置文件httpd.conf已打开以下内容:
# Virtual hosts
Include conf/extra/httpd-vhosts.conf

[root@server LAMP]# cd /usr/local/apache2/conf/extra/
[root@server extra]# vim httpd-vhosts.conf
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/www/admin"
    ServerName www.admin.cc
    ServerAlias www.admin.cc
```

```
    ErrorLog "logs/admin-error_log"
    CustomLog "logs/admin-access_log" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot "/www/myblog"
    ServerName www.myblog.net
    ErrorLog "logs/myblog-error_log"
    CustomLog "logs/myblog-access_log" common
</VirtualHost>
```

5. 修改每个网站各自连接mysql数据库的配置文件

1) mysql的web管理网站phpMyAdmin相关文件

```
[root@server admin]#  cd /www/admin/
[root@server admin]# cp config.sample.inc.php config.inc.php
[root@server admin]# vim config.inc.php
....
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
....
```

2) myblog网站wordpress

进入到mysql数据库创建一个myblog库

```
[root@server extra]# mysql -uroot -p123
mysql> create database myblog;
Query OK, 1 row affected (0.00 sec)

[root@server admin]#  cd /www/myblog/
[root@server myblog]# cp wp-config-sample.php wp-config.php
[root@server myblog]# vim wp-config.php
...
// ** MySQL 设置 - 具体信息来自您正在使用的主机 ** //
/** WordPress数据库的名称 */
define('DB_NAME', 'myblog');

/** MySQL数据库用户名 */
define('DB_USER', 'root');

/** MySQL数据库密码 */
define('DB_PASSWORD', '123');

/** MySQL主机 */
define('DB_HOST', '127.0.0.1');

/** 创建数据表时默认的文字编码 */
define('DB_CHARSET', 'utf8');
```

```
/** 数据库整理类型。如不确定请勿更改 */
define('DB_COLLATE', '');
...

6. 创建myblog数据库
[root@server myblog]# mysql -p123
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.25 Source distribution

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create database myblog;         //创建myblog库来存放wordpress的数据
Query OK, 1 row affected (0.01 sec)

mysql>

7. 启动web服务
为了方便启动，做成脚本启动脚本：
# cp /usr/local/apache2/bin/apachectl /etc/init.d/apache
# service apache start
# lsof -i :80

8. 测试验证
基于域名的虚拟主机，需要客户端指定DNS或者修改hosts文件完成测试。
```

9. rsyslog日志服务

任务背景

为了提高用户体验，对于大部分的互联网企业来说，7*24小时对外提供服务经常被挂在嘴边。而真正要保证企业业务的高效、稳定运行，运维人员起着决定性的作用。运维人员常常是冲在一线的战士，而日志是指引运维人员快速判断、定位、分析问题的入口。所以对于一些重要的、常用的服务做好**日志记录和管理**是非常有必要的，并且这件事情也依然是运维人员的事情。

任务要求

1. 对文件共享服务器上ftp服务的日志单独保存到本地（文件路径自己定义）
2. 对跳板机上的ssh服务的日志单独保存到本地（文件路径自己定义）
3. 根据具体情况，对相应日志进行轮转，保证当前服务器上能够查看5天以内的日志
4. 对于web服务器的访问日志每天保存一个文件，并按照当前日期命名

解决方案

任务1

```
1. 搭建FTP服务
略
2. 修改vsftpd配置文件，目的是日志让rsyslog程序管理
1）在vsftpd.conf追加以下内容：
syslog_enable=YES
2）重启服务
service vsftpd restart

3.修改rsyslog服务的配置文件来管理本地日志
vim /etc/rsyslog.conf
ftp.*                                                  /var/log/ftp.log

4. 重启服务
[root@lamp log]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
[root@lamp log]# ls ftp.log
ftp.log

5. 测试验证
[root@lamp log]# tail -f ftp.log
Sep  9 11:41:29 lamp vsftpd[1500]: [ftp] OK LOGIN: Client "10.1.1.250", anon password "lftp@"
Sep  9 11:41:43 lamp vsftpd[1502]: [ftp] OK UPLOAD: Client "10.1.1.250", "/pub/1.sh", 288 bytes,
697.89Kbyte/sec

说明：
1. 默认情况下FTP服务本身也可以管理自己的上传下载日志，默认保存位置：/var/log/xferlog;
2. 如果不想让它自己管理修改配置文件:将xferlog_enable=YES修改为xferlog_enable=NO
```

任务2

```
1. 修改sshd的配置文件
vim /etc/ssh/sshd_config
...
SyslogFacility local1

2.修改rsyslog程序的配置文件
vim /etc/rsyslog.conf
修改一下内容：
*.info;mail.none;authpriv.none;cron.none;local1.none               /var/log/messages
authpriv.*;local1.none                                             /var/log/secure
local1.*                                                  /var/log/sshd.log

3.重启服务测试验证
```

任务3

```
方法1：修改主配置文件
vim /etc/logrotate.conf
...
/var/log/sshd.log {
```

```
missingok
daily
rotate 5
size 5M
}
```

测试验证：

```
# logrotate -f /etc/logrotate.conf
```

方法2：创建子配置文件

```
[root@log-server logrotate.d]# pwd
/etc/logrotate.d
[root@log-server logrotate.d]#  cat /etc/logrotate.d/sshd
/var/log/sshd.log {
missingok
daily
rotate 5
size 5M
nodateext
}
```

测试验证：

```
[root@jumper log]# logrotate -f /etc/logrotate.conf
```

任务4

1. 修改web服务的虚拟主机配置文件对日志进行管理：

```
# vim /usr/local/apache2/conf/extra/httpd-vhosts.conf
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/www/admin"
    ServerName www.admin.cc
    ServerAlias www.admin.cc
   ErrorLog "| /usr/local/apache2/bin/rotatelogs -l /usr/local/apache2/logs/admin-error_log-
%Y%m%d 86400"
   CustomLog "| /usr/local/apache2/bin/rotatelogs -l /usr/local/apache2/logs/admin-access_log-
%Y%m%d 86400" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot "/www/myblog"
    ServerName www.myblog.net
    ErrorLog "| /usr/local/apache2/bin/rotatelogs -l /usr/local/apache2/logs/myblog-error_log-
%Y%m%d 86400"
   CustomLog "| /usr/local/apache2/bin/rotatelogs -l /usr/local/apache2/logs/myblog-access_log-
%Y%m%d 86400" common
</VirtualHost>
```

2. 重启apache测试验证

10. iptables防火墙配置

任务背景

俗话说"人机状态好，事故不会找；手中有粮，心中不慌"。运维人员往往是服务器的最后一道防线，保障服务器安全稳定有效的运行是我们的工作职责。所以每台服务器上不管运行什么服务，开启防火墙功能是必须的。由于服务器上运行着很多程序来对外提供服务，端口也五花八门，所以需要在开启防火墙的情况下，不影响服务的正常使用。故要求运维人员根据业务情况开启防火墙并保证服务的正常运行。

任务要求

1. 所有的服务器开启防火墙，并且开机自动启动
2. iptables允许**web服务器**相关的端口通过
3. iptables允许远程管理**ssh服务**的端口通过
4. iptables允许文件共享服务**ftp、samba**的端口通过

解决方案

```
1. 开启防火墙并开机自启动
service iptables start
chkconfig iptables on

2. iptables允许web服务器相关的端口通过
iptables -t filter -P INPUT DROP
iptables -t filter -I INPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -I INPUT -p tcp --dport 443 -j ACCEPT
或者
iptables -t filter -I INPUT -p tcp -m multiport --dports 80,443 -j ACCEPT

3. iptables允许远程管理ssh服务的端口通过
iptables -t filter -I INPUT -p tcp --dport 22 -j ACCEPT

4.iptables允许文件共享服务ftp、samba的端口通过
iptables -t filter -I INPUT -p tcp -m multiport --dports 20:21,139,445,2000:2500 -s 10.1.1.0/24
-j ACCEPT

ftp配置文件中，需要固定被动模式下的端口范围：
vim /etc/vsftpd/vsftpd.conf
增加如下内容：
pasv_min_port=2000
pasv_max_port=2500
```

11. 磁盘管理之LVM

任务背景

某天接到短信报警提示，显示某主机的根分区空间使用率超过85%，该主机用于影评(mysql)和报表数据库(oracle)。经查看发现其中MySQL数据库的数据文件存放在/usr/local/mysql/中，占用根文件系统空间导致。由于前期规划不合理，没有将业务数据和系统数据分开。经研究决定，要将影评的数据库单独放到另一块磁盘上，并且实现逻辑卷管理。

任务要求

1. 保证数据库完整的情况下将影评数据库迁移到另外一块新添加的磁盘上
2. 考虑到数据增长情况，新磁盘使用lvm逻辑卷管理，方便日后动态扩容

解决方案

```
1. 添加一块物理硬盘并重启
[root@server ~]# lsblk
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
...
sdc                         8:32   0   10G  0 disk

2. 创建大小为10G的逻辑卷
1) 创建物理卷
[root@server ~]# pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created
2) 创建卷组vg02
[root@server ~]# vgcreate vg02 /dev/sdc
  Volume group "vg01" successfully created
3) 创建逻辑卷lv-mysql
[root@server ~]# lvcreate -n lv-mysql -L 9G vg02
或者
[root@server ~]# lvcreate -n lv-mysql -l 100%free vg02

4) 格式化为ext4文件系统
[root@server ~]# mkfs.ext4 /dev/vg02/lv-mysql

5) 挂载使用
a. 创建一个空的挂载点/u04
[root@server ~]# mkdir /u04
b. 挂载逻辑卷lv-mysql到/u04目录
[root@server ~]# mount /dev/vg02/lv-mysql /u04

3. 停止前端web服务
[root@server ~]# service apache stop

4. 停止mysql数据库
[root@server ~]# service mysql25 stop

5. 备份mysql数据库到另外一台备份机
备份机：10.1.1.1   备份目录:/backup

[root@server ~]# rsync -av /usr/local/mysql 10.1.1.1:/backup/

6. 将/usr/local/mysql/目录里的所有数据文件同步到逻辑卷上，即/u04目录
[root@server ~]# rsync -av /usr/local/mysql/ /u04
查看是否同步完成：
[root@server ~]# ls /u04
[root@server ~]# du -sh /u04     查看大小是否和原来mysql数据库大小一致
7. 卸载逻辑卷
[root@server ~]# umount /u04

8. 删除/usr/local/mysql/目录里原来的数据文件
注意：删之前一定要确定成功备份了！！！
```

```
[root@server ~]# rm -rf /usr/local/mysql/*

9. 挂载逻辑卷lv-mysql到mysql的安装目录/usr/local/mysql
[root@server ~]# mount /dev/vg02/lv-mysql /usr/local/mysql
开机自动挂载:
vim /etc/rc.local
...
mount /dev/vg02/lv-mysql /usr/local/mysql

10. 启动数据库
[root@server ~]# service mysql25 start

11. 启动web服务
[root@server ~]# service apache start

12.测试验证
访问之前的网站看是否可以正常访问
```