

## 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斗，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

## 涉及知识点

- rsyslog服务管理日志
- logrotate工具对日志进行切割轮转

## 课程目标

- 了解日志的级别及作用
- 能够使用rsyslog程序对服务日志进行**本地管理**（==重点）==
- 能够使用rsyslog程序对服务日志进行远程管理
- ==能够使用logrotate工具对服务日志进行轮转==

## 理论储备

### 一、rsyslog介绍

- 它提供高性能、强大的安全特性和模块化设计。虽然它最初是一个普通的口号，但rsyslog已经发展成为一种瑞士军刀式的日志记录，能够接受来自各种来源的输入，转换它们，并将结果输出到不同的目的地。性能通常被认为是“令人震惊的”。
- 在RHEL6中，syslogd已经被rsyslog取代。它可以将日志写入数据库，并可以利用模块和插件控制输入输出。
- rsyslog程序管理本地和远程日志
  - 安装软件
  - 根据需求修改配置文件
  - 启动服务
  - 测试验证

### 二、系统日志文件介绍

日志格式：文本日志/二进制日志/数据库日志

常见日志文件：

/var/log/boot.log	系统引导日志，记录开机启动信息
/var/log/dmesg	核心的启动日志（硬件相关）
/var/log/messages	系统的日志文件
/var/log/maillog	邮件服务的日志
/var/log/xferlog	ftp服务的日志
/var/log/secure	网络连接及系统登录的安全信息
/var/log/cron	定时任务的日志
/var/log/wtmp	记录所有的登入和登出
/var/log/btmp	记录失败的登入尝试

last -f 查看

### 三、日志级别

# man syslog

日志信息分为以下级别，从上到下级别依次降低

none	<-- none	不算是一个等级，它表示不记录服务的所有信息
0 emerg	<--	系统不可用
1 alert	<--	特别留意的报警信息
2 crit	<--	非常严重的状况
3 err	<--	错误信息
4 warning	<--	警告信息
5 notice	<--	稍微需要注意的信息
6 info	<--	正常信息
7 debug	<--	调试信息，开发人员使用

### 四、日志配置

#### 1. 日志定义相关符号

配置文件中常见的表示符号

.	<--	用来分隔服务和日志级别	mail.info
*	<--	任何服务，或者任何级别	*.info mail.*
=	<--	有等号表示等于某一日志级别，没有等号表示大于或者等于某一级别	mail.=info mail.info
!	<--	排除操作，前面有相同服务的表达式，这个操作才有意义	
		代表从前面表达式所包含的内容中排除某些内容	
;	<--	用于分隔不同的 服务.日志级别 组合	cron.=info;mail.info
,	<--	用于分隔不同的服务	cron,mail.=info
-	<--	用于指定目标文件时，代表异步写入	-/var/log/maillog

服务.级别日志

mail.=info	/var/log/maillog	邮件服务的等于info级别日志记录到/var/log/maillog
cron,mail.info	0 1 2 3 4 5 6	
*.info		所有服务的大于等于info级别
mail.*		邮件服务的所有级别日志 0-7
cron.info;mail.=info		
ftp.err;mail.=warning		
cron.err;ftp,mail.=err		

举例说明：

```

cron.err          0123
cron.=info  6
cron.!=info 0123457
cron.!=info  7

mail,cron,ftp.=err

cron.info;cron.!=err    456
0123456
4567
mail.info;mail.!=err    012456
0123456
0124567

```

## 2. 了解配置文件

```

*.info;mail.none;authpriv.none;cron.none    /var/log/messages
所有服务产生的日志，除了mail/验证/任务计划相关日志都记录/var/log/message
authpriv.*                                  /var/log/secure
记录所有跟验证有关日志
mail.*                                       -/var/log/maillog
记录所有跟邮件有关的日志信息
cron.*                                       /var/log/cron
记录跟任务计划查关的日志
*.emerg                                     *
把所有级别为emerg的信息发送给所有登录到系统上的用户
uucp,news.crit                              /var/log/spooler
local7.*                                    /var/log/boot.log
记录所有跟启动相关的日志信息

```

### 特别说明：

说明：

# man rsyslog.conf

The facility is one of the following keywords: auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security (same as auth), syslog, user, uucp and local0 through local7.

log facility 设备 设施：用来记录一种类型日志的日志设备

```

daemon
auth
authpriv
user
mail
lpr
news
uucp
ftp

```

## 实战演练

### 一、本地日志管理

#### 小试牛刀：

将计划任务服务的日志记录到/var/log/test\_cron.log里

#### 思路：

1. 通过修改配置文件完成
2. 重启rsyslog服务
3. 测试验证

#### 步骤：

##### 1. 修改配置文件

```
vim rsyslogconf
```

```
#cron.*                                /var/log/cron
cron.info                               /var/log/test_cron.log
```

##### 2. 启动服务

```
# service rsyslog restart
```

```
Shutting down system logger:          [ OK ]
```

```
Starting system logger:               [ OK ]
```

```
# ls /var/log/test_cron.log
```

```
/var/log/test_cron.log
```

##### 3. 测试验证

1) 编写一个计划任务，测试验证

2) 发个log消息测试

logger用于往系统中写入日志，他提供一个shell命令接口到rsyslog系统模块

```
# logger -it "test cronlog" -p cron.info "Testing cron log info"
```

-t 指定标记记录

-p 指定输入消息的优先级，优先级可以是数字或者指定为 " facility.level" 的格式。

-i 逐行记录每一次logger的进程ID

```
[root@lamp ~]# tail -f /var/log/test_cron.log
```

```
Sep  9 09:06:33 lamp test cronlog[1378]: Testing cron log info
```

#### 任务1：

搭建FTP服务，并且将ftp服务的上传下载日志单独记录到/var/log/ftp.log里

#### 思路：

1. 搭建FTP服务
2. 修改ftp服务的配置文件指定让rsyslog程序管理日志（定义设备载体）
3. 修改rsyslog服务的配置文件来指定ftp服务日志保存到哪个文件

## 步骤:

### 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
```

注意:

ftp服务的默认日志载体是ftp。

### 4. 重启服务

```
[root@lamp log]# ls ftp.log
```

```
ls: cannot access ftp.log: No such file or directory
```

```
[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
```

```
Nov 16 11:33:48 lamp vsftpd[17076]: CONNECT: Client "10.1.1.1"
```

```
Nov 16 11:33:48 lamp vsftpd[17075]: [ftp] OK LOGIN: Client "10.1.1.1", anon password "lftp@"
```

```
Nov 16 11:33:58 lamp vsftpd[17077]: [ftp] OK DOWNLOAD: Client "10.1.1.1", "/hosts", 249 bytes, 379.94Kbyte/sec
```

说明:

1. 默认情况下FTP服务本身也可以管理自己的日志，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:

把ssh服务的日志指定记录到/var/log/ssh里

## 思路:

1. 单独指定ssh服务的日志载体 (ssh服务需要配置——>local1)
2. rsyslog程序将来自于local1设备载体上的日志单独记录 (通过rsyslog.conf配置)
3. 重启相关的服务
4. 测试验证

## 步骤:

### 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

### 总结：

本地日志管理：服务的日志和系统的日志

服务日志：

1. 服务本身可以管理自己的日志，比如ftp服务。——>服务的配置文件，vsftpd.conf
2. 使用rsyslog程序管理服务的日志。rsyslog.conf——>日志设备载体.info /var/log/ftp.log

## 二、远程日志管理

**目的：**把多台服务器的日志远程记录到其中一台日志服务器集中化管理，方便对其统一分析和管理

**需求：**

将ssh服务的日志远程记录到日志管理服务器上保存。

**环境：**

log-server:10.1.1.1

ssh-server: 10.1.1.2

client:10.1.1.3 用于测试

**思路：**

1. 在ssh-server上将ssh服务的日志单独记录
2. 在ssh-server上通过修改==rsyslog.conf==文件来将本地ssh服务日志远程传送到log-server服务器上
3. 在==log-server上打开514端口==等待客户来传送日志
4. 测试验证

**步骤：**

说明：接着上面的任务2实验继续。

### 1. 在日志服务器端完成以下事情

#### 1) 安装rsyslog软件

```
[root@log-server ~]# rpm -q rsyslog
rsyslog-5.8.10-8.el6.x86_64
```

#### 2) 打开514端口

vim /etc/rsyslog.conf

```
# Provides UDP syslog reception
```

```

$ModLoad imudp
$UDPServerRun 514

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

3) 启动服务
[root@log-server ~]# service rsyslog restart
Shutting down system logger:          [ OK ]
Starting system logger:                [ OK ]

[root@log-server ~]# netstat -nltp|grep 514
tcp        0      0 0.0.0.0:514          0.0.0.0:*            LISTEN
       7801/rsyslogd
tcp        0      0 :::514              :::*                  LISTEN
       7801/rsyslogd
udp        0      0 0.0.0.0:514          0.0.0.0:*
       7801/rsyslogd
udp        0      0 :::514              :::*

2. 在ssh-server上将ssh服务的日志单独记录并远程发送
1) ssh服务的日志载体定义为local1
2) 删除原来的本地保存
#local1.*                                /var/log/sshd.log
local1.*      @@10.1.1.1:514             将local1载体上的所有日志远程日志发送到log-server
@代表UDP协议传送; @@代表tcp协议传输

3)重启rsyslog服务
service rsyslog restart

4)测试验证
客户端: 10.1.1.2
[root@MissHou ~]# ssh 10.1.1.2
root@10.1.1.2's password:
Last login: Sun Sep  9 15:16:23 2018 from lamp.itcast.cc

log-server端查看日志:
[root@log-server ~]# tail -f /var/log/message
Sep  9 15:17:45 jumper sshd[1844]: Accepted password for root from 10.1.1.2 port 41978 ssh2
Sep  9 15:18:04 jumper sshd[1844]: Received disconnect from 10.1.1.2: 11: disconnected by user

```

## 思考:

如果日志服务器管理多台服务器，那么如何区分不同的客户端？

1. 这件事情需要再日志管理服务器端完成
2. 将接收过来的日志单独管理保存称为本地日志管理
3. 可以通过定义==模板==的方式完成

解决办法:

可以通过定义模板来保存不同的日志文件:

```
[root@log-server log]# vim /etc/rsyslog.conf
```

在该文件的最后面加入以下内容:

//定义一个模板DynFile, 将日志保存在/var/log里, 文件名为system-客户端的主机名.log

```
$template DynFile, "/var/log/system-%HOSTNAME%.log"
```

//动态加载调用上面的模板

```
local1.* ?DynFile
```

## 课堂练习:

将FTP服务的日志远程保存到log-server日志管理服务器上。

步骤:

### 1. 准备环境

每台服务器标记清楚主机名

### 2. FTP服务的管理方式

1) 方法1: ftp服务自己去管理 xferlog\_enable=YES 默认/var/log/xferlog

2) 方法2: ftp服务交给rsyslog程序去管理 syslog\_enable=YES

```
[root@ftp-server ~]# vim /etc/vsftpd/vsftpd.conf
```

增加以下内容:

```
syslog_enable=YES
```

修改以下内容:

```
xferlog_enable=YES 修改为xferlog_enable=NO
```

重启ftp服务:

### 3. 修改rsyslog配置文件进行ftp服务的日志远程管理

```
[root@ftp-server ~]# vim /etc/rsyslog.conf
```

增加以下内容:

```
ftp.* @@10.1.1.1:514
```

### 4. 在log-server上完成接收过来的日志保存

```
[root@log-server log]# vim /etc/rsyslog.conf
```

改变以下内容:

```
# Provides UDP syslog reception
```

```
$ModLoad imudp
```

```
$UDPServerRun 514
```

```
# Provides TCP syslog reception
```

```
$ModLoad imtcp
```

```
$InputTCPServerRun 514
```

```
$template DynFile, "/var/log/system-%HOSTNAME%.log"
```

```
local1.* ?DynFile
```

```
ftp.* ?DynFile
```

将接收过来的ftp设备载体上的所有日志保存到/var/log/system-



xxx.log

## 5. 客户端测试验证

注意：客户端访问ftp服务进行上传下载文件

```
[root@client ~]# lftp 10.1.1.2
lftp 10.1.1.2:~> get file2
```

```
[root@log-server ~]# ls /var/log/system-ftp-server.log
/var/log/system-ftp-server.log
[root@log-server ~]# tail -f /var/log/system-ftp-server.log
Nov 16 16:39:14 ftp-server vsftpd[18270]: CONNECT: Client "10.1.1.3"
Nov 16 16:39:14 ftp-server vsftpd[18269]: [ftp] OK LOGIN: Client "10.1.1.3", anon password
"lftp@"
Nov 16 16:39:21 ftp-server vsftpd[18271]: [ftp] OK DOWNLOAD: Client "10.1.1.3", "/file1",
0.00Kbyte/sec
Nov 16 16:40:18 ftp-server vsftpd[18276]: CONNECT: Client "10.1.1.3"
Nov 16 16:40:18 ftp-server vsftpd[18275]: [ftp] OK LOGIN: Client "10.1.1.3", anon password
"lftp@"
Nov 16 16:40:18 ftp-server vsftpd[18277]: [ftp] OK DOWNLOAD: Client "10.1.1.3", "/file2",
0.00Kbyte/sec
```

## 三、日志轮转

### 1. 日志轮转介绍

日志轮循（轮转）：日志轮转，切割，备份，归档

- 为什么要日志轮转？

- 1、避免日志过大占满/var/log的文件系统
- 2、方便日志查看
- 3、将丢弃系统中最旧的日志文件，以节省空间
- 4、日志轮转的程序是logrotate
- 5、logrotate本身不是系统守护进程，它是通过计划任务crond每天执行

- 如何进行日志轮转？
  - 了解相关配置文件

```
[root@MissHou ~]# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly
#以7天为一个周期(每周轮转)
rotate 4
#每4周备份日志文件（保留4份日志文件）
create
#当老的转储文件被归档后,创建一个新的空的转储文件重新记录,权限和原来的转储文件权限一样。

dateext
```

```

#用日期来做轮转之后的文件的后缀名
#compress
#指定不压缩转储文件,如需压缩去掉注释就可以了.通过gzip压缩
include /etc/logrotate.d #加载外部目录
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly      表示此文件是每月轮转,而不会用到上面的每周轮转
    create 0664 root utmp  轮转之后创建新文件,权限是0664,属于root用户和utmp组
    minsize 1M      文件大于1M,而且周期到了,才会轮转
    #size 1M        文件大小大于1M立马轮转,不管有没有到周期
    rotate 1        保留1分日志文件,每1个月备份一次日志文件
}

/var/log/btmp {
    missingok      如果日志文件不存在,不报错
    monthly
    create 0600 root utmp
    rotate 1
}

[root@MissHou ~]# cat /etc/logrotate.d/syslog
//这个子配置文件,没有指定的参数都会以默认方式轮转

/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    sharedscripts  不管有多少个文件待轮转,prerotate 和 postrotate 代码只执行一次
    postrotate     轮转完后执行postrotate 和 endscript 之间的shell代码
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true 这一句话表示
轮转后对rsyslog的pid进行刷新(但pid其实不变)
    endscript
}

思考:
为什么轮转后需要对rsyslog的pid进行刷新呢?

```

## 2. 任务3

### • 对ssh服务日志进行轮转

要求:

1. 每天进行轮转,保留5天的日志文件
2. 日志文件大小大于5M进行轮转,不管是否到轮转周期

思路:

修改logrotate程序的主配置文件或者在/etc/logrotate.d/目录创建一个文件

步骤:

#### 方法1: 修改主配置文件

```
vim /etc/logrotate.conf
...
/var/log/ssh.log {
missingok
daily
rotate 5
size 5M
}
```

#### 测试验证:

```
[root@jumper log]# 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/ssh
/var/log/ssh.log {
missingok
daily
rotate 5
size 5M
nodateext
}
```

#### 测试验证:

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

### • 对FTP服务日志进行轮转

#### 要求:

1. 每个月轮转1次, 保留1个月的日志
2. 日志文件大小超过500M不管有没有到轮转周期必须轮转
3. 如果日志文件不存在也不报错

#### 思路:

1. 首先得在当前系统中有ftp服务的日志
  - ftp服务自己本身管理日志(/var/log/xferlog)
  - 交给rsyslog程序管理(vsftpd.conf—>syslog\_enable=YES;设备载体默认是FTP); rsyslog.conf—>ftp.\* /var/log/ftp.log
2. 根据需求去让logrotate程序轮转

```
/var/log/ftp.log {
monthly
rotate 1
size 500M
missingok
}
```

## 扩展补充

### 一、Apache日志管理

#### 1. 日志内容定制

参考官当：[访问日志内容定制](#)

自定义日志内容格式

日志文件定义：

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/www/admin"
    ServerName www.admin.cc
    ServerAlias www.admin.cc
    ErrorLog "logs/admin-error_log"           //错误日志，日志等级warning
    CustomLog "logs/admin-access_log" common  //访问日志，通用格式
</VirtualHost>
```

说明：

##### 1. 主配置文件中定义访问日志记录内容

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
CustomLog logs/access_log common
```

%h 远程主机名

%l 远程日志名称

%u 通过身份验证的远程用户。如果返回状态（%s）为401（未经授权），则可能是伪造的。

%t 收到请求的时间，格式为[18/Sep/2011:19:18:28 -0400]。最后一个数字表示与GMT的时区偏移量

#### 2. 日志轮转

方法1：优雅重启

通过使用优雅的重启，服务器可以被指示打开新的日志文件，而不会丢失任何来自客户端的现有或未决的连接。但是，为了完成这个任务，服务器必须继续在旧的日志文件中写入旧请求。因此，在重新启动之后，需要等待一段时间，然后才能对日志文件进行任何处理。

一个典型的场景是：简单地旋转日志并压缩旧日志以节省空间：

```
mv access_log access_log.old
mv error_log error_log.old
apachectl graceful
sleep 600
gzip access_log.old error_log.old
```

方法2：管道日志

管道日志的一个重要用途是允许在不重启服务器的情况下进行日志旋转。Apache HTTP服务器包括一个名为rotatelog的简单程序。例如，每24小时(每天)旋转日志，可以使用：

```
ErrorLog "| /usr/local/apache2/bin/rotatelog -l /usr/local/apache2/logs/error_log-%Y%m%d 86400"
CustomLog "| /usr/local/apache2/bin/rotatelog -l /usr/local/apache2/logs/access_log-%Y%m%d 86400" common
```

说明：

1. rotatelog程序是apache自带的一个日志切割工具。 -l参数表示使用本地系统时间为标准切割，而不是GMT时区时间。
2. /usr/local/apache2/logs/access\_log-%Y%m%d 86400  
用来指定日志文件的位置和名称，其中86400用来指定分割时间默认单位为s,也就是24小时；

### 3. 任务4

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

## 二、课后小实验

讨论为什么轮转后需要对rsyslog的pid进行刷新呢？

```
sharedscripts
    prerotate
        轮转前执行脚本
    endscript
```

```
sharedscripts
    postrotate
        轮转后执行脚本
```

```
endscript
```

为什么轮转后要做上面的kill -HUP? HUP是一个信号, 这个信号的默认操作为终止进程

小实验:

准备环境:

不以时间作为后缀去轮转

```
# logger -t "哈哈" "你好"
# tail -3 /var/log/messages      --默认这条信息在当前的messages里
```

注释掉以日期为后缀测试:

修改子配置文件 /etc/logrotate.d/syslog

```
{
    sharedscripts
    postrotate
        logger -t "呵呵" "再见!"
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
    endscript
}
```

强制轮转

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

请问这条信息在轮转后在哪个文件里?

messages.1

结论: 先写日志再刷新PID; 刷新之前写的日志文件是老的日志文件。

再次编辑子配置文件

```
{
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
        logger -t "呵呵" "我又回来啦!"
    endscript
}
```

强制轮转

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

请问这条信息在轮转后在哪个文件里?

messages

结论: 先刷新PID再写日志; 日志就会写到新的日志文件里。

### 三、日志轮转常见参数

常用的指令解释, 这些指令都可以在man logrotate 中找到。

daily                      指定转储周期为每天

monthly                    指定转储周期为每月

weekly                    <-- 每周轮转一次(monthly)

rotate 4                   <-- 同一个文件最多轮转4次, 4次之后就删除该文件

create 0664 root utmp      <-- 轮转之后创建新文件, 权限是0664, 属于root用户和utmp组

```
dateext          <-- 用日期来做轮转之后的文件的后缀名
compress         <-- 用gzip对轮转后的日志进行压缩
minsize 30K      <-- 文件大于30K，而且周期到了，才会轮转
size 30k         <-- 文件必须大于30K才会轮转，而且文件只要大于30K就会轮转
                  不管周期是否已到
missingok        <-- 如果日志文件不存在，不报错
notifempty       <-- 如果日志文件是空的，不轮转
delaycompress    <-- 下一次轮转的时候才压缩
sharedscripts    <-- 不管有多少个文件待轮转，prerotate 和 postrotate 代码只执行一次
prerotate        <-- 如果符合轮转的条件
                  则在轮转之前执行prerotate和endscript 之间的shell代码
postrotate       <-- 轮转完后执行postrotate 和 endscript 之间的shell代码
```

## 课后实战

- 日志管理服务器将远程接收过来的ftp的日志按照以下要求进行轮转
  1. 每周轮转1次，保留1个月的日志文件
  2. 日志大小超过100M并且到了轮转周期再进行轮转
  3. 以时间作为后缀轮转老的文件