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

任务分析

1. 明白不同服务的日志如何定义
    - 服务本身自己管理日志
    - 交给rsyslog程序去管理日志（重点）
2. rsyslog程序如何管理其他服务的日志（掌握）
3. 日志如何轮转？使用logrotate工具+cron服务（了解）

涉及知识点

- 日志的级别及作用
- rsyslog服务日志管理
    - 本地日志管理（重点）
    - 远程日志管理
- logrotate工具对日志进行切割轮转

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。

二、系统日志文件介绍

日志文件	介绍说明	备注
/var/log/messages	系统的日志文件	
/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/dmesg	核心的启动日志（硬件相关）	
/var/log/boot.log	系统引导日志，记录开机启动信息	
/var/log/xferlog	ftp服务的日志(上传和下载文件)	
/var/log/maillog	邮件服务的日志	
/var/log/wtmp	记录所有的登入和登出	last -f 查看
/var/log/btmp	记录失败的登入尝试	

三、日志级别

`man syslog`

说明：日志信息分为以下级别，从上到下级别依次降低

日志级别	描述	备注
none	表示不记录服务的所有信息	不算是一个等级
0 emerg	系统不可用	
1 alert	特别留意的报警信息，需要马上处理	
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	

rsyslog程序——>服务——>功能：管理维护系统和服务的日志——>配置文件(/etc/rsyslog.conf)

四、日志配置

1. 日志定义相关符号

符号	描述	举例
.	用来分隔服务和日志级别	mail(服务).info(日志级别)
*	任何服务，或者任何日志级别	*.info 和 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 和 mail.info
;	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
!	取反，排除操作，前面有相同服务的表达式，这个操作才有意义	
-	用于指定目标文件时，代表异步写入	-/var/log/maillog

举例说明：

```
cron.err                cron服务大于等于err级别日志
cron.=info              cron服务等于info级别日志
mail,cron.err           cron和mail服务大于等于err级别日志
cron.info;cron.!=err    012456
cron.info;cron.!err     456
```

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
local0-7
```

实战演练

rsyslog程序管理本地和远程日志思路

- 安装软件
- 根据需求修改配置文件
- 启动服务
- 测试验证

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 /etc/rsyslog.conf
#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@server ~]# tail -f /var/log/test_cron.log
Jan  3 17:24:33 server test cronlog[44746]: 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. 确定FTP服务日志服务自己是否可以管理（了解man文档）
    - FTP服务本身可以管理日志
    - 两种日志格式:
        - xferlog_enable=YES
        - FTP服务自己日志记录格式；vsftpd_log_file=/var/log/vsftpd.log
        - xferlog格式: xferlog_std_format=YES;xferlog_file=/var/log/xferlog
    - FTP服务日志也可以让rsyslog程序管理
        - syslog_enable=YES ftp日志记录到ftp设备载体上
3. 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）
    - 修改配置文件syslog_enable=YES打开即可
4. 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    //FTP服务日志默认会记录到ftp的设备载体上
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1小结:

任务目标：将FTP服务的日志让rsyslog程序管理

任务解决方案：

1. 修改ftp服务的配置文件，目的指定日志的管理方式：syslog_enable=YES,默认日志载体ftp
2. 修改rsyslog服务的配置文件，目的管理来自ftp载体上的日志（实现了管理ftp服务的日志）
3. 测试验证（记录的ftp上传和下载的日志）

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(sshd_config)——>local0）
2. rsyslog程序将来自于local0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 local0


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
local0.*                                                        /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

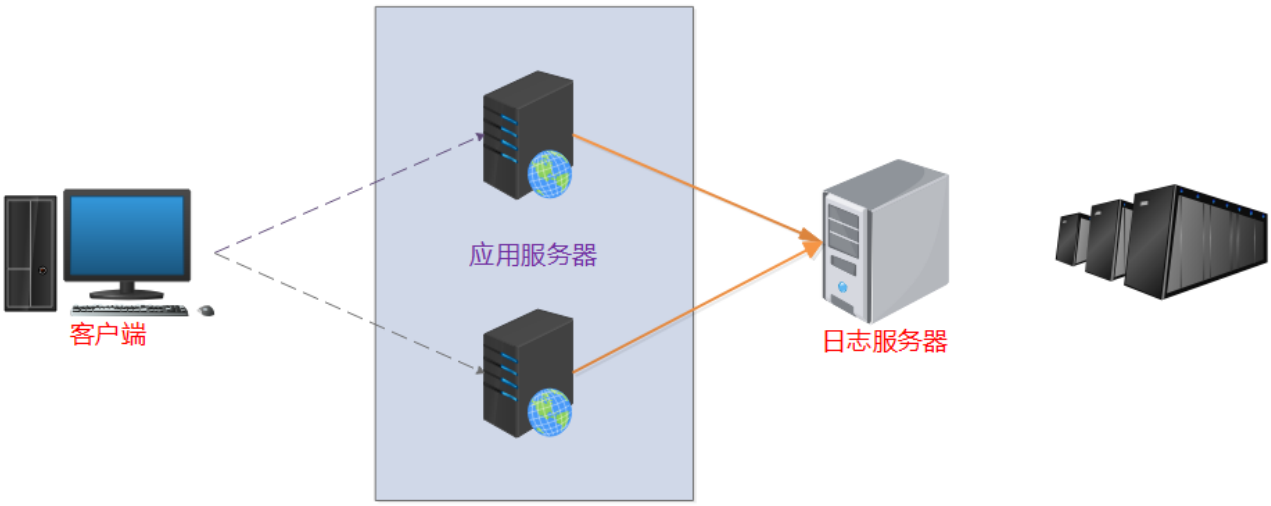
服务日志：

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:10.1.1.2

ssh-server(jumper)：10.1.1.1

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 -nltup|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                      :::*
 7801/rsyslogd

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local0
2）删除原来的本地保存
#local0.*                                                  /var/log/sshd.log
local0.*          @@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
client端：10.1.1.2
[root@MissHou ~]# ssh 10.1.1.1
root@10.1.1.1'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 10.1.1.3 port 41978 ssh2
Sep  9 15:18:04 jumper sshd[1844]: Received disconnect from 10.1.1.3: 11: disconnected by
user
```

总结：

日志远程管理

ssh日志————>日志服务器

1. 本地服务的日志(日志服务器的客户端)需要远程管理——>rsyslog程序
    - 服务日志设备载体定义（服务本身完成）
    - 在rsyslog.conf文件里指定该服务的日志远程发送： 日志设备载体.日志级别 `@@log-server:514`

2. 日志服务端需要打开514端口——>rsyslog程序
3. 如果发送不成功请检查你的网络（防火墙和selinux）

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
    上面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：

1. 将ssh服务的日志单独记录 /var/log/ssh.log
2. 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：
# 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/ssh.log {
missingok
daily
rotate 5
size 5M
nodateext
}

测试验证：
# logrotate -f /etc/logrotate.conf
```

任务三总结：

logrotate 工具轮转 + crond服务

对ssh服务日志轮转，方法：

1. 修改/etc/logrotate.conf主配置文件

```
/var/log/ssh {
   轮转需求

}
```

2. 在/etc/logrotate.d/目录里创建一个子配置文件

```
vim /etc/logrotate.d/ssh
/var/log/ssh {
   轮转需求

}
```

- **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
  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

参考官档：访问日志内容定制

格式字符串	描述
%%	百分号。
%a	请求的客户端IP地址（请参阅 mod_remoteip模块）。
%{c}a	连接的基础对等IP地址（请参阅 mod_remoteip模块）。
%A	本地IP地址。
%B	响应大小（以字节为单位），不包括HTTP头。
%b	响应大小（以字节为单位），不包括HTTP头。在CLF格式中，即-没有发送字节时 的' '而不是0。
%{*VARNAME*}C	发送到服务器的请求中的cookie *VARNAME*的内容。仅完全支持版本0 cookie。
%D	服务请求所需的时间，以微秒为单位。
%{*VARNAME*}e	环境变量*VARNAME*的内容。
%f	文件名。
%h	远程主机名。如果HostnameLookups设置为 Off，将记录IP地址，这是默认值。如果它仅记录少数主机的主机名，则可能具有按名称提及它们的访问控制指令，请参阅Require主机文档。
%H	请求协议。
%{*VARNAME*}i	*VARNAME*:发送到服务器的请求中标题行的内容。其他模块（例如mod_headers）所做的更改会影响这一点。如果您对大多数模块修改它之前请求标头的内容感兴趣，请使用mod_setenvif 将标头复制到内部环境变量中并使用上述描述该值。 %{*VARNAME*}e
%k	此连接上处理的keepalive请求数。有趣的KeepAlive是，如果 正在使用，那么，例如，'1'表示在初始一个之后的第一个keepalive请求，'2'表示第二个，等等; 否则这始终为0（表示初始请求）。
%l	远程日志名称（来自identd，如果提供）。除非mod_ident存在并且IdentityCheck已设置， 否则这将返回破折号On。
%L	来自错误日志的请求日志ID（如果没有将任何内容记录到此请求的错误日志中，则为“-”）。查找匹配的错误日志行以查看哪些请求导致了什么错误。
%m	请求方法。
%{*VARNAME*}n	来自另一个模块的note *VARNAME*的内容。
%{*VARNAME*}o	*VARNAME*:回复中标题行的内容。
%p	服务请求的服务器的规范端口。
%{*format*}p	服务请求的服务器的规范端口，服务器的实际端口或客户端的实际端口。有效的格式canonical，local或remote。
%P	为请求提供服务的子进程ID。
%{*format*}P	为请求提供服务的子进程ID或线程ID。有效的格式是pid，tid和hextid。 hextid要求APR 1.2.0或更高。
%q	查询字符串（?如果查询字符串存在则前缀，否则为空字符串）。
%r	第一行请求。

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的请求。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rotatelogs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/rotatelogs -l /usr/local/apache2/logs/error_log-%Y%m%d
86400"
CustomLog "| /usr/local/apache2/bin/rotatelogs -l /usr/local/apache2/logs/access_log-%Y%m%d
86400" common
```

说明：
1. rotatelogs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-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>
```

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 中找得到。
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

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