第十六章：日志管理

尚硅谷云计算 Linux 课程

版本：V1.0

讲师：沈超

一、日志简介

1　日志相关服务

在 CentOS 6.x 中日志服务已经由 rsyslogd 取代了原先的 syslogd 服务。Redhat 认为 syslogd 已经不能满足在工作中的需求，rsyslogd 相比 syslogd 具有一些新的特点：

- ✧ 基于 TCP 网络协议传输日志信息；
- ✧ 更安全的网络传输方式；
- ✧ 有日志消息的及时分析框架；
- ✧ 后台数据库；
- ✧ 配置文件中可以写简单的逻辑判断；
- ✧ 与 syslog 配置文件相兼容。

2　系统中常见的日志文件

日志文件	说　　明
/var/log/cron	记录了系统定时任务相关的日志。
/var/log/cups/	记录打印信息的日志
/var/log/dmesg	记录了系统在开机时内核自检的信息。也可以使用 dmesg 命令直接查看内核自检信息。
/var/log/btmp	记录错误登录的日志。这个文件是二进制文件，不能直接 vi 查看，而要使用 lastb 命令查看，命令如下： [root@localhost log]# lastb root　　tty1　　Tue Jun　4 22:38 - 22:38　(00:00) *#有人在 6 月 4 日 22:38 使用 root 用户，在本地终端 1 登录错误*
/var/log/lastlog	记录系统中所有用户最后一次的登录时间的日志。这个文件也是二进制文件，不能直接 vi，而要使用 lastlog 命令查看。
/var/log/mailog	记录邮件信息。
/var/log/message	记录系统重要信息的日志。这个日志文件中会记录 Linux 系统的绝大多数重要信息，如果系统出现问题时，首先要检查的就应该是这个日志文件。
/var/log/secure	记录验证和授权方面的信息，只要涉及账户和密码的程序都会记录。比如说系统的登录，ssh 的登录，su 切换用户，sudo 授权，甚至添加用户和修改用户密码都会记录在这个日志文件中。
/var/log/wtmp	永久记录所有用户的登录、注销信息，同时记录系统的启动、重启、关机事件。同样这个文件也是一个二进制文件，不能直接 vi，而需要使用 last 命令来查看。

/var/run/utmp	记录当前已经登录的用户的信息。这个文件会随着用户的登录和注销而不断变化，只记录当前登录用户的信息。同样这个文件不能直接 vi，而要使用 w，who，users 等命令来查询。

除了系统默认的日志之外，采用 RPM 方式安装的系统服务也会默认把日志记录在/var/log/目录中（源码包安装的服务日志是在源码包指定目录中）。不过这些日志不是由 rsyslogd 服务来记录和管理的，而是各个服务使用自己的日志管理文档来记录自身日志。

日志文件	说　　明
/var/log/httpd/	RPM 包安装的 apache 服务的默认日志目录
/var/log/mail/	RPM 包安装的邮件服务的额外日志目录
/var/log/samba/	RPM 包安装的 samba 服务的日志目录
/var/log/sssd/	守护进程安全服务目录

二、日志服务 rsyslogd

1　日志文件格式

只要是由日志服务 rsyslogd 记录的日志文件，他们的格式是一样的。基本日志格式包含以下四列：

- ✧ 事件产生的时间；
- ✧ 发生事件的服务器的主机名；
- ✧ 产生事件的服务名或程序名；
- ✧ 事件的具体信息。

2　rsyslogd 服务的配置文件

1）、　/etc/rsyslog.conf 配置文件格式

```
authpriv.*                                  /var/log/secure
#服务名称[连接符号]日志等级                    日志记录位置
#认证相关服务.所有日志等级          记录在/var/log/secure 日志中
```

- ✧ 服务名称

那么我们首先需要确定 rsyslogd 服务可以识别哪些服务的日志，也可以理解为以下这些服务委托了 rsyslogd 服务来代为管理日志。这些服务如表 16-3 所示：

服务名称	说　　明
auth（LOG_AUTH）	安全和认证相关消息（不推荐使用 authpriv 替代）
authpriv（LOG_AUTHPRIV）	安全和认证相关消息（私有的）
cron（LOG_CRON）	系统定时任务 cront 和 at 产生的日志
daemon（LOG_DAEMON）	和各个守护进程相关的日志
ftp（LOG_FTP）	ftp 守护进程产生的日志
kern（LOG_KERN）	内核产生的日志（不是用户进程产生的）
local0-local7（LOG_LOCAL0-7）	为本地使用预留的服务
lpr（LOG_LPR）	打印产生的日志
mail（LOG_MAIL）	邮件收发信息
news（LOG_NEWS）	与新闻服务器相关的日志
syslog（LOG_SYSLOG）	有 syslogd 服务产生的日志信息（虽然服务名称已经改为

	rsyslogd，但是很多配置都还是沿用了 syslogd 的，这里并没有修改服务名）。
user（LOG_USER）	用户等级类别的日志信息
uucp（LOG_UUCP）	uucp 子系统的日志信息，uucp 是早期 linux 系统进行数据传递的协议，后来也常用在新闻组服务中。

✧ 连接符号

日志服务连接日志等级的格式是：

日志服务[连接符号]日志等级　　日志记录位置

在这里连接符号可以识别为：

- “.”代表只要比后面的等级高的（包含该等级）日志都记录下来。比如：“cron.info”代表 cron 服务产生的日志，只要日志等级大于等于 info 级别，就记录
- “.=”代表只记录所需等级的日志，其他等级的都不记录。比如：“*.=emerg”代表任何日志服务产生的日志，只要等级是 emerg 等级就记录。这种用法及少见，了解就好
- “.!”代表不等于，也就是除了该等级的日志外，其他等级的日志都记录。

✧ 日志等级

✧ 等 级 名 称	说　　明
debug（LOG_DEBUG）	一般的调试信息说明
info（LOG_INFO）	基本的通知信息
notice（LOG_NOTICE）	普通信息，但是有一定的重要性
warning（LOG_WARNING）	警告信息，但是还不回影响到服务或系统的运行
err（LOG_ERR）	错误信息，一般达到 err 等级的信息以及可以影响到服务或系统的运行了。
crit（LOG_CRIT）	临界状况信息，比 err 等级还要严重
alert（LOG_ALERT）	警告状态信息，比 crit 还要严重。必须立即采取行动
emerg（LOG_EMERG）	疼痛等级信息，系统已经无法使用了
*	代表所有日志等级，比如：“authpriv.*”代表 authpriv 认证信息服务产生的日志，所有的日志等级都记录

表 16-4 日志等级

日志等级这里还可以识别“none”，如果日志等级是 none，就说明忽略这个日志服务，该服务的所有日志都不再记录。

✧ 日志记录位置

日志记录位置就是当前日志输出到哪个日志文件中保存，当然也可以把日志输出到打印机打印，或者输出到远程日志服务器上（当然日志服务器要允许接收才行）。日志的记录位置也是固定的，我们来学习下：

- 日志文件的绝对路径。这是最常见的日志保存方法，如“/var/log/secure”就是保存系统验证和授权信息日志的。
- 系统设备文件。如“/dev/lp0”代表第一台打印机，如果日志保存位置是打印机设备的话，当有日志时就会在打印机打印（不太符合可持续发展战略哦-_-!）。
- 转发给远程主机。因为可以选择使用 TCP 协议和 UDP 协议传输日志信息，所以有两种发

送格式。如使用“@192.168.0.210:514”，就会把日志内容使用 UDP 协议发送到 192.168.0.210 的 UDP 514 端口上；如果使用“@@192.168.0.210:514”就会把日志内容使用 TCP 协议发送到 192.168.0.210 的 TCP 514 端口上，其中 514 是日志服务默认端口。当然只要 192.168.0.210 同意接收此日志，就可以把日志内容保存在日志服务器上。

- 用户名。如“root”，就会把日志发送给 root 用户，当然 root 要在在线，否则就收不到日志信息了。发送日志给用户时，可以使用“*”代表发送给所有在线用户，如“mail.**”就会把 mail 服务产生的所有级别的日志发送给所有在线用户。如果需要把日志发送给多个在线用户，用户名之间用“，”分隔。

忽略或丢弃日志。如果接受日志的对象是“~”，代表这个日志不会记录，而被直接丢弃。如“local3.*~”代表忽略 local3 服务类型所有的日志都不记录。

2）、 /etc/rsyslog.conf 配置文件的内容

```
[root@localhost ~]# vi /etc/rsyslog.conf
#查看配置文件的内容
# rsyslog v5 configuration file

# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### MODULES ####
#加载模块

$ModLoad imuxsock # provides support for local system logging (e.g. via logger command)
#加载 imuxsock 模块，为本地系统登录提供支持
$ModLoad imklog   # provides kernel logging support (previously done by rklogd)
#加载 imklog 模块，为内核登录提供支持
#$ModLoad immark  # provides --MARK-- message capability
#加载 immark 模块，提供标记信息的能力

# Provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514
#加载 UPD 模块，允许使用 UDP 的 514 端口接收采用 UDP 协议转发的日志

# Provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514
#加载 TCP 模块，允许使用 TCP 的 514 端口接收采用 TCP 协议转发的日志

#### GLOBAL DIRECTIVES ####
#定义全局设置

# Use default timestamp format
```

```
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
#定义日志的时间使用默认的时间戳格式

# File syncing capability is disabled by default. This feature is usually not required,
# not useful and an extreme performance hit
#$ActionFileEnableSync on
#文件同步功能。默认没有开启，是注释的。

# Include all config files in /etc/rsyslog.d/
$IncludeConfig /etc/rsyslog.d/*.conf
#包含/etc/rsyslog.d/目录中所有的“.conf”子配置文件。也就是说这个目录中的所有
#子配置文件也同时生效。



#### RULES ####
#日志文件保存规则

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console
#kern 服务.所有日志级别                 保存在/dev/console
#这个日志默认没有开启，如果需要，则取消注释

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
#所有服务.info 以上级别的日志保存在/var/log/messages 日志中。
#mail，authpriv，cron 的日志不记录在/var/log/messages 日志文件中，因为它们都有自己的日志文件。
#所以/var/log/messages 日志是最重要的系统日志文件，需要经常查看！

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure
#用户认证服务所有级别的日志保存在/var/log/secure 日志中

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog
#mail 服务的所有级别的日志保存在/var/log/maillog 日志中。
#“-”号的含义是日志先在内存之中保存，当日志够多之后，再向文件中保存。

# Log cron stuff
cron.*                                                  /var/log/cron
#计划任务的所有日志保存在/var/log/cron 日志中
```

```
# Everybody gets emergency messages
*.emerg                                                 *
#所有日志服务的疼痛等级日志对所有在线用户广播。

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
#uucp 和 news 日志服务的 crit 以上的日志保存在/var/log/sppoler 日志文件中。

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
#loacl7 日志服务的所有日志写入/var/log/boot.log 日志中。
#会把开机时的检测信息在显示到屏幕的同时，写入/var/log/boot.log 日志中

# ### begin forwarding rule ###
#定义转发规则
# The statement between the begin ... end define a SINGLE forwarding
# rule. They belong together, do NOT split them. If you create multiple
# forwarding rules, duplicate the whole block!
# Remote Logging (we use TCP for reliable delivery)
#
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
#$WorkDirectory /var/lib/rsyslog # where to place spool files
#$ActionQueueFileName fwdRule1 # unique name prefix for spool files
#$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
#$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
#$ActionQueueType LinkedList   # run asynchronously
#$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514
# ### end of the forwarding rule ##
```

3)、 定义自己的日志

```
[root@localhost ~]# vi /etc/rsyslog.conf
#写入一下一句话
*.crit                  /var/log/alert.log
#把所有服务的“临界点”以上的错误都保存在/var/log/alert.log 日志中

[root@localhost ~]# service rsyslog restart
关闭系统日志记录器:                                        [确定]
启动系统日志记录器:                                        [确定]
#重启 rsyslog 服务
[root@localhost ~]# ll /var/log/alert.log
```

```
-rw-------. 1 root root 0 6月   5 10:33 /var/log/alert.log
# alert.log日志就生成了
```

三、日志轮替

1　日志文件的命名规则

日志轮替最主要的作用就是把旧的日志文件移动并改名，同时建立新的空日志文件，当旧日志文件超出保存的范围之后，就会进行删除。那么旧的日志文件改名之后，如何命名呢？主要依靠/etc/logrotate.conf 配置文件中“dateext”参数：

- ✧ 如果配置文件中拥有“dateext”参数，那么日志会用日期来作为日志文件的后缀，例如“secure-20180605”。这样的话日志文件名不会重叠，所以也就不需要日志文件的改名，只需要保存指定的日志个数，删除多余的日志文件即可。
- ✧ 如果配置文件中没有“dateext”参数，那么日志文件就需要进行改名了。当第一次进行日志轮替时，当前的“secure”日志会自动改名为“secure.1”，然后新建“secure”日志，用来保存新的日志。当第二次进行日志轮替时，“secure.1”会自动改名为“secure.2”，当前的“secure”日志会自动改名为“secure.1”，然后也会新建“secure”日志，用来保存新的日志，以此类推。

2　logrotate 配置文件

```
[root@localhost ~]# vi /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly
#每周对日志文件进行一次轮替

# keep 4 weeks worth of backlogs
rotate 4
#保存4个日志文件，也就是说如果进行了5次日志轮替，就会删除第一个备份日志

# create new (empty) log files after rotating old ones
create
#在日志轮替时，自动创建新的日志文件

# use date as a suffix of the rotated file
dateext
#使用日期作为日志轮替文件的后缀

# uncomment this if you want your log files compressed
#compress
#日志文件是否压缩。如果取消注释，则日志会在转储的同时进行压缩

#以上日志日志配置为默认配置，如果需要轮替的日志没有设定独立的参数，那么都会遵守以上参数。
#如果轮替日志配置了独立参数，那么独立参数优先级更高。
```

```
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
#包含/etc/logrotate.d/目录中所有的子配置文件。也就是说会把这个目录中所有子配置文件读取进来，
#进行日志轮替。

# no packages own wtmp and btmp -- we'll rotate them here
#以下两个轮替日志有自己的独立参数，如果和默认的参数冲突，则独立参数生效。
/var/log/wtmp {
#以下参数仅对此目录有效
    monthly
    #每月对日志文件进行一次轮替
    create 0664 root utmp
    #建立的新日志文件，权限是 0664，所有者是 root，所属组是 utmp 组
        minsize 1M
        #日志文件最小轮替大小是 1MB。也就是日志一定要超过 1MB 才会轮替，否则就算
        #时间达到一个月，也不进行日志转储
    rotate 1
    #仅保留一个日志备份。也就是只有 wtmp 和 wtmp.1 日志保留而已
}

/var/log/btmp {
#以下参数只对/var/log/btmp 生效
    missingok
    #如果日志不存在，则忽略该日志的警告信息
    monthly
    create 0600 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
```

logrotate 配置文件的主要参数，我们通过表 16-4 来说明下：

参　数	参　数　说　明
daily	日志的轮替周期是每天
weekly	日志的轮替周期是每周
monthly	日志的轮替周期是每月
rotate 数字	保留的日志文件的个数。0 指没有备份
compress	日志轮替时，旧的日志进行压缩
create mode owner group	建立新日志，同时指定新日志的权限与所有者和所属组。如 create 0600 root utmp
mail address	当日志轮替时，输出内容通过邮件发送到指定的邮件地址。如 mail shenc@lamp.net
missingok	如果日志不存在，则忽略该日志的警告信息

notifempty	如果日志为空文件，则不进行日志轮替
minsize 大小	日志轮替的最小值。也就是日志一定要达到这个最小值才会轮替，否则就算时间达到也不轮替
size 大小	日志只有大于指定大小才进行日志轮替，而不是按照时间轮替。如 size 100k
dateext	使用日期作为日志轮替文件的后缀。如 secure-20180605
sharedscripts	在此关键字之后的脚本只执行一次
prerotate/endscript	在日志轮替之前执行脚本命令。endscript 标示 prerotate 脚本结束。
postrotate/endscript	在日志轮替之后执行脚本命令。endscript 标示 postrotate 脚本结束

这些参数中较为不好理解的应该就是 prerotate/endscript 和 postrotate/endscript 参数了，我们利用“man logrotate”中的列子来解释下这两个参数。例如：

```
"/var/log/httpd/access.log" /var/log/httpd/error.log {
 #日志轮替的是/var/log/httpd/中 RPM 包安装的 apache 正确访问日志和错误日志
    rotate 5
     #轮替 5 次
    mail www@my.org
     #信息发送到指定邮箱
    size 100k
     #日志大于 100KB 时才进行日志轮替，不再按照时间轮替
    sharedscripts
     #以下脚本只执行一次
    postrotate
     #在日志轮替结束之后，执行以下脚本
        /usr/bin/killall -HUP httpd
         #重启 apache 服务
    endscript
     #脚本结束
}
```

prerotate 和 postrotate 主要用于在日志轮替的同时，执行指定的脚本，一般用于日志轮替之后重启服务。这里强调，如果你的日志是写入 rsyslog 服务的配置文件的，那么把新日志加入 logrotate 后，一定要重启 rsyslog 服务，否则你会发现虽然新日志建立了，但是数据还是写入了旧的日志当中。那是因为虽然 logrotate 知道日志轮替了，但是 rsyslog 服务却并不知道。同理，如果你的日志不是被 rsyslog 管理，如源码包安装的 Apache、Nginx 等服务，则需要重启 Apache 或 Nginx 服务，否则日志也不能正常轮替。

3　把自己的日志加入日志轮替

这里有两个方法：第一种方法是直接在/etc/logrotate.conf 配置文件中写入该日志的轮替策略，从而把日志加入轮替；第二种方法是在/etc/logrotate.d/目录中新建立该日志的轮替文件，在该轮替文件中写入正确的轮替策略，因为该目录中的文件都会被“include”到主配置文件中，所以也可以把日志加入轮替。我们这里推荐第二种方法，因为系统中需要轮替的日志非常多，如果全都直接写入/etc/logrotate.conf 配置文件，那么这个文件的可管理性就会非常差，不利于此文件的维护。

说起来很复杂，我们举个例子吧，还记得我们自己生产的/var/log/alert.log 日志吗？这个日志不是系统默认日志，而是我们通过/etc/rsyslog.conf 配置文件自己生成的日志，所以默认这个日志是不会轮替的。那么我们需要把这个日志加入日志轮替的策略，该怎么实现呢？我们采用第二种方法，也就是在/etc/logrotate.d/目录中建立此日志的轮替文件。具体步骤如下：

```
[root@localhost ~]# chattr +a /var/log/alert.log
#先给日志文件赋予 chattr 的 a 属性，保证日志的安全
[root@localhost ~]# vi /etc/logrotate.d/alter
#创建 alter 轮替文件，把/var/log/alert.log 加入轮替
/var/log/alert.log {
        weekly                                  ←每周轮替一次
        rotate 6                                ←保留 6 个轮替日志
        sharedscripts                           ←以下命令只执行一次
        prerotate                               ←在日志轮替之前执行
                /usr/bin/chattr -a /var/log/alert.log
                #在日志轮替之前取消 a 属性，以便让日志可以轮替
        endscript                               ←脚本结束

        sharedscripts
        postrotate                              ←在日志轮替之后执行
                /usr/bin/chattr +a /var/log/alert.log
                #日志轮替之后，重新加入 a 属性
        endscript

        sharedscripts
        postrotate
        /bin/kill -HUP $(/bin/cat /var/run/syslogd.pid 2>/dev/null) &>/dev/null
        endscript
        #重启 rsyslog 服务，保证日志轮替正常
}
```

再举个例子，我们如果需要把 Nginx 服务的日志加入日志轮替，则也需要重启 Nginx 服务，例如：

```
/date/logs/nginx/access/access.log /date/logs/nginx/access/default.log {
#假设 Nginx 的日志放在/date 目录下
    daily
    rotate 15
    sharedscripts
    postrotate
        /bin/kill -HUP $(/bin/cat /var/run/syslogd.pid) &>/dev/null
        #重启 rsyslog 服务
        /bin/kill -HUP $(/bin/cat /usr/local/nginx/logs/nginx.pid) &>/dev/null
        #重启 Nginx 服务
    endscript
}
```

4　logrotate 命令

我们日志轮替之所以可以在指定的时间备份日志，其实也要依赖系统定时任务。如果大家还记得/etc/cron.daily/目录，就会发现这个目录中是有 logrotate 文件，logrotate 通过这个文件依赖定时任务执行的。

不过 logrotate 命令的格式是什么样的呢？我们来学习下：

```
[root@localhost ~]# logrotate [选项] 配置文件名
选项：
    如果此命令没有选项，则会按照配置文件中的条件进行日志轮替
    -v:     显示日志轮替过程。加了-v 选项，会显示日志的轮替的过程
    -f:     强制进行日志轮替。不管日志轮替的条件是否已经符合，强制配置文件中所有
            的日志进行轮替
```

我们执行 logrotate 命令，并查看下执行过程：

```
[root@localhost ~]# logrotate -v /etc/logrotate.conf
#查看日志轮替的流程
…省略部分输出…
rotating pattern: /var/log/alert.log  weekly (6 rotations)
#这就是我们自己加入轮替的 alert.log 日志。
empty log files are rotated, old logs are removed
considering log /var/log/alert.log
  log does not need rotating                    ←时间不够一周，所以不进行日志轮替
…省略部分输出…
```

我们发现/var/log/alert.log 加入了日志轮替，已经被 logrotate 识别，并调用了。只是时间没有达到轮替的标准，所以没有进行轮替。那我们强制进行一次日志轮替，看看有什么结果：

```
[root@localhost ~]# logrotate -vf /etc/logrotate.conf
#强制进行日志轮替，不管是否符合轮替条件
…省略部分输出…
rotating pattern: /var/log/alert.log  forced from command line (6 rotations)
empty log files are rotated, old logs are removed
considering log /var/log/alert.log
  log needs rotating                            ←日志需要轮替
rotating log /var/log/alert.log, log->rotateCount is 6
dateext suffix '-20180607'                      ←提取日期参数
glob pattern '-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
glob finding old rotated logs failed
running prerotate script
fscreate context set to unconfined_u:object_r:var_log_t:s0
renaming /var/log/alert.log to /var/log/alert.log-20180607
#旧的日志被重命名
creating new /var/log/alert.log mode = 0600 uid = 0 gid = 0
#创建新日志文件，同时制定权限、所有者和属组
running postrotate script
…省略部分输出…
```

我们发现 alert.log 日志已经完成了日志轮替。我们查看下新产生的日志和旧日志：

```
[root@localhost ~]# ll /var/log/alert.log*
-rw-------. 1 root root   0 6月   7 10:07 /var/log/alert.log
-rw-------. 1 root root 237 6月   7 09:58 /var/log/alert.log-20180607
#旧日志文件已经转储
[root@localhost ~]# lsattr /var/log/alert.log
-----a-------e- /var/log/alert.log
#新的日志文件被自动加入了 chattr 的 a 属性。
```

logrotate 命令使用“-f”选项之后，就不管日志是否已经符合了日志轮替条件，而强制把所有的日志都进行了轮替。