Linux 下 logrotate 日志轮询操作梳理

2017-06-08 Linux爱好者

（点击上方公众号，可快速关注）

> 作者：散尽浮华
> www.cnblogs.com/kevingrace/p/6307298.html
> 如有好文章投稿，请点击 → 这里了解详情

对于 Linux 系统安全来说，日志文件是极其重要的工具。不知为何，我发现很多运维同学的服务器上都运行着一些诸如每天切分 Nginx日志之类的 CRON 脚本，大家似乎遗忘了 Logrotate，争相发明自己的轮子，这真是让人沮丧啊！就好比明明身边躺着现成的性感美女，大家却忙着自娱自乐，罪过！

logrotate 程序是一个日志文件管理工具。用于分割日志文件，删除旧的日志文件，并创建新的日志文件，起到“转储”作用。可以节省磁盘空间。下面就对 logrotate 日志轮转操作做一梳理记录。

1）配置文件介绍

Linux系统默认安装logrotate工具，它默认的配置文件在：

```
/etc/logrotate.conf
/etc/logrotate.d/
```

logrotate.conf 才主要的配置文件，logrotate.d 是一个目录，该目录里的所有文件都会被主动的读入/etc/logrotate.conf中执行。

另外，如果 /etc/logrotate.d/ 里面的文件中没有设定一些细节，则会以/etc/logrotate.conf这个文件的设定来作为默认值。

Logrotate是基于CRON来运行的，其脚本是/etc/cron.daily/logrotate，日志轮转是系统自动完成的。实际运行时，Logrotate会调用配置文件/etc/logrotate.conf。可以在/etc/logrotate.d目录里放置自定义好的配置文件，用来覆盖Logrotate的缺省值。

```
[root@huanqiu_web1 ~]# cat /etc/cron.daily/logrotate
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
```

```
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

如果等不及cron自动执行日志轮转，想手动强制切割日志，需要加-f参数；不过正式执行前最好通过Debug选项来验证一下（-d参数），这对调试也很重要：

```
# /usr/sbin/logrotate -f /etc/logrotate.d/nginx
# /usr/sbin/logrotate -d -f /etc/logrotate.d/nginx
```

logrotate 命令格式：

```
logrotate [OPTION...] <configfile>

-d, –debug ：debug模式，测试配置文件是否有错误。
-f, –force ：强制转储文件。
-m, –mail=command ：压缩日志后，发送日志到指定邮箱。
-s, –state=statefile ：使用指定的状态文件。
-v, –verbose ：显示转储过程。
```

根据日志切割设置进行操作，并显示详细信息：

```
[root@huanqiu_web1 ~]# /usr/sbin/logrotate -v /etc/logrotate.conf
[root@huanqiu_web1 ~]# /usr/sbin/logrotate -v /etc/logrotate.d/php
```

根据日志切割设置进行执行，并显示详细信息,但是不进行具体操作，debug模式

```
[root@huanqiu_web1 ~]# /usr/sbin/logrotate -d /etc/logrotate.conf
[root@huanqiu_web1 ~]# /usr/sbin/logrotate -d /etc/logrotate.d/nginx
```

查看各log文件的具体执行情况

```
[root@fangfull_web1 ~]# cat /var/lib/logrotate.status
```

2）切割介绍

比如以系统日志/var/log/message做切割来简单说明下：

- 第一次执行完rotate(轮转)之后，原本的messages会变成messages.1，而且会制造一个空的messages给系统来储存日志；

- 第二次执行之后，messages.1会变成messages.2，而messages会变成messages.1，又造成一个空的messages来储存日志！

如果仅设定保留三个日志（即轮转3次）的话，那么执行第三次时，则 messages.3这个档案就会被删除，并由后面的较新的保存日志所取代！也就是会保存最新的几个日志。

日志究竟轮换几次，这个是根据配置文件中的dateext 参数来判定的。
看下logrotate.conf配置：

```
# cat /etc/logrotate.conf
# 底下的设定是 "logrotate 的默认值" ，如果别的文件设定了其他的值，
# 就会以其它文件的设定为主
weekly        //默认每一周执行一次rotate轮转工作
rotate 4      //保留多少个日志文件(轮转几次).默认保留四个.就是指定日志文件删除之前轮转的次数，0 指没有备份
create        //自动创建新的日志文件，新的日志文件具有和原来的文件相同的权限；因为日志被改名,因此要创建一个新的来继续存储之前的日志
dateext       //这个参数很重要！就是切割后的日志文件以当前日期为格式结尾，如xxx.log-20131216这样,如果注释掉,切割出来是按数字递增,即前面说的 xxx.log-1这种格式
compress      //是否通过gzip压缩转储以后的日志文件，如xxx.log-20131216.gz ；如果不需要压缩，注释掉就行

include /etc/logrotate.d
# 将 /etc/logrotate.d/ 目录中的所有文件都加载进来

/var/log/wtmp {             //仅针对 /var/log/wtmp 所设定的参数
monthly                 //每月一次切割,取代默认的一周
minsize 1M              //文件大小超过 1M 后才会切割
create 0664 root utmp           //指定新建的日志文件权限以及所属用户和组
rotate 1                //只保留一个日志.
}
# 这个 wtmp 可记录用户登录系统及系统重启的时间
# 因为有 minsize 的参数，因此不见得每个月一定会执行一次喔.要看文件大小。
```

由这个文件的设定可以知道/etc/logrotate.d其实就是由/etc/logrotate.conf 所规划出来的目录，虽然可以将所有的配置都写入 /etc/logrotate.conf ，但是这样一来这个文件就实在是太复杂了，尤其是当使用很多的服务在系统上面时， 每个服务都要去修改 /etc/logrotate.conf 的设定也似乎不太合理了。

所以，如果独立出来一个目录，那么每个要切割日志的服务， 就可以独自成为一个文件，并且放置到 /etc/logrotate.d/ 当中。

其他重要参数说明：

compress	通过gzip 压缩转储以后的日志
nocompress	不做gzip压缩处理
copytruncate	用于还在打开中的日志文件，把当前日志备份并截断；是先拷贝再清空的方式，拷贝和清空之间有一个时间差，可能会丢失部分日志数据。
nocopytruncate	备份日志文件不过不截断
create mode owner group	轮转时指定创建新文件的属性，如create 0777 nobody nobody
nocreate	不建立新的日志文件
delaycompress	和compress 一起使用时，转储的日志文件到下一次转储时才压缩
nodelaycompress	覆盖 delaycompress 选项，转储同时压缩。
missingok	如果日志丢失，不报错继续滚动下一个日志
errors address	专储时的错误信息发送到指定的Email 地址
ifempty	即使日志文件为空文件也做轮转，这个是logrotate的缺省选项。
notifempty	当日志文件为空时，不进行轮转
mail address	把转储的日志文件发送到指定的E-mail 地址
nomail	转储时不发送日志文件
olddir directory	转储后的日志文件放入指定的目录，必须和当前日志文件在同一个文件系统
noolddir	转储后的日志文件和当前日志文件放在同一个目录下
sharedscripts	运行postrotate脚本，作用是在所有日志都轮转后统一执行一次脚本。如果没有配置这个，那么每个日志轮转后都会执行一次脚本
prerotate	在logrotate转储之前需要执行的指令，例如修改文件的属性等动作；必须独立成行
postrotate	在logrotate转储之后需要执行的指令，例如重新启动 (kill -HUP) 某个服务！必须独立成行
daily	指定转储周期为每天
weekly	指定转储周期为每周
monthly	指定转储周期为每月
rotate count	指定日志文件删除之前转储的次数，0 指没有备份，5 指保留5 个备份
dateext	使用当期日期作为命名格式
dateformat .%s	配合dateext使用，紧跟在下一行出现，定义文件切割后的文件名，必须配合dateext使用，只支持 %Y %m %d %s 这四个参数
size(或minsize) log-size	当日志文件到达指定的大小时才转储，log-size能指定bytes(缺省)及KB (sizek)或MB(sizem).

当日志文件 >= log-size 的时候就转储。 以下为合法格式：（其他格式的单位大小写没有试过）

size = 5 或 size 5 （>= 5 个字节就转储）

size = 100k 或 size 100k

size = 100M 或 size 100M

小示例：下面一个切割nginx日志的配置

```
[root@master-server ~]# vim /etc/logrotate.d/nginx
/usr/local/nginx/logs/*.log {
```

```
daily
rotate 7
missingok
notifempty
dateext
sharedscripts
postrotate
    if [ -f /usr/local/nginx/logs/nginx.pid ]; then
        kill -USR1 `cat /usr/local/nginx/logs/nginx.pid`
    fi
endscript
}
```

分享一例曾经使用过的nginx日志切割处理脚本：

1）logrotate日志分割配置

```
[root@bastion-IDC ~# vim /etc/logrotate.d/nginx
/data/nginx_logs/*.access_log
{
nocompress
daily
copytruncate
create
ifempty
olddir /data/nginx_logs/days
rotate 0
}
```

2）日志分割脚本

```
[root@bastion-IDC ~# vim /usr/local/sbin/logrotate-nginx.sh
#!/bin/bash
#创建转储日志压缩存放目录
mkdir -p /data/nginx_logs/days
#手工对nginx日志进行切割转换
/usr/sbin/logrotate -vf /etc/logrotate.d/nginx
#当前时间
time=$(date -d "yesterday" +"%Y-%m-%d")
#进入转储日志存放目录
cd /data/nginx_logs/days
```

```
#对目录中的转储日志文件的文件名进行统一转换
for i in $(ls ./ | grep "^\(.*\)\.[[:digit:]]$")
do
mv ${i} ./$(echo ${i}|sed -n 's/^\(.*\)\.\([[:digit:]]\)$/\1/p')-$(echo $time)
done
#对转储的日志文件进行压缩存放，并删除原有转储的日志文件，只保存压缩后的日志文件。以节约存储空间
for i in $(ls ./ | grep "^\(.*\)\-\([[:digit:]-]\+\)$")
do
tar jcvf ${i}.bz2 ./${i}
rm -rf ./${i}
done
#只保留最近7天的压缩转储日志文件
find /data/nginx_logs/days/* -name "*.bz2" -mtime 7 -type f -exec rm -rf {} \;
```

3）crontab定时执行

```
[root@bastion-IDC ~# crontab -e
#logrotate
0 0 * * * /bin/bash -x /usr/local/sbin/logrotate-nginx.sh > /dev/null 2>&1
```

手动执行脚本，测试下看看：

```
[root@bastion-IDC ~# /bin/bash -x /usr/local/sbin/logrotate-nginx.sh
[root@bastion-IDC ~# cd /data/nginx_logs/days
[root@bastion-IDC days# ls
huantest.access_log-2017-01-18.bz2
```

php脚本切割一例：

```
[root@huanqiu_web1 ~]# cat /etc/logrotate.d/php
/Data/logs/php/*log {
   daily
   rotate 365
   missingok
    notifempty
    compress
    dateext
    sharedscripts
    postrotate
      if [ -f /Data/app/php5.6.26/var/run/php-fpm.pid ]; then
```

```
        kill -USR1 `cat /Data/app/php5.6.26/var/run/php-fpm.pid`
    fi
  endscript
  postrotate
    /bin/chmod 644 /Data/logs/php/*gz
  endscript
}

[root@huanqiu_web1 ~]# ll /Data/app/php5.6.26/var/run/php-fpm.pid
-rw-r--r-- 1 root root 4 Dec 28 17:03 /Data/app/php5.6.26/var/run/php-fpm.pid

[root@huanqiu_web1 ~]# cd /Data/logs/php
[root@huanqiu_web1 php]# ll
total 25676
-rw-r--r-- 1 root   root        0 Jun  1  2016 error.log
-rw-r--r-- 1 nobody nobody    182 Aug 30  2015 error.log-20150830.gz
-rw-r--r-- 1 nobody nobody    371 Sep  1  2015 error.log-20150901.gz
-rw-r--r-- 1 nobody nobody    315 Sep  7  2015 error.log-20150907.gz
.........
.........
```

nginx日志切割一例

```
[root@huanqiu_web1 ~]# cat /etc/logrotate.d/nginx
/Data/logs/nginx/*/*log {
  daily
  rotate 365
  missingok
  notifempty
  compress
  dateext
  sharedscripts
  postrotate
  /etc/init.d/nginx reload
  endscript
}

[root@huanqiu_web1 ~]# ll /Data/logs/nginx/www.huanqiu.com/
..........
-rw-r--r-- 1 root root     1652 Jan  1 00:00 error.log-20170101.gz
-rw-r--r-- 1 root root     1289 Jan  2 00:00 error.log-20170102.gz
```

```
-rw-r--r-- 1 root root     1633 Jan  3 00:00 error.log-20170103.gz
-rw-r--r-- 1 root root     3239 Jan  4 00:00 error.log-20170104.gz
```

系统日志切割一例

```
[root@huanqiu_web1 ~]# cat /etc/logrotate.d/syslog
/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    sharedscripts
    postrotate
    /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
    endscript
}

[root@huanqiu_web1 ~]# ll /var/log/messages*
-rw------- 1 root root 34248975 Jan 19 18:42 /var/log/messages
-rw------- 1 root root 51772994 Dec 25 03:11 /var/log/messages-20161225
-rw------- 1 root root 51800210 Jan  1 03:05 /var/log/messages-20170101
-rw------- 1 root root 51981366 Jan  8 03:36 /var/log/messages-20170108
-rw------- 1 root root 51843025 Jan 15 03:40 /var/log/messages-20170115
[root@huanqiu_web1 ~]# ll /var/log/cron*
-rw------- 1 root root 2155681 Jan 19 18:43 /var/log/cron
-rw------- 1 root root 2932618 Dec 25 03:11 /var/log/cron-20161225
-rw------- 1 root root 2939305 Jan  1 03:06 /var/log/cron-20170101
-rw------- 1 root root 2951820 Jan  8 03:37 /var/log/cron-20170108
-rw------- 1 root root 3203992 Jan 15 03:41 /var/log/cron-20170115
[root@huanqiu_web1 ~]# ll /var/log/secure*
-rw------- 1 root root  275343 Jan 19 18:36 /var/log/secure
-rw------- 1 root root 2111936 Dec 25 03:06 /var/log/secure-20161225
-rw------- 1 root root 2772744 Jan  1 02:57 /var/log/secure-20170101
-rw------- 1 root root 1115543 Jan  8 03:26 /var/log/secure-20170108
-rw------- 1 root root  731599 Jan 15 03:40 /var/log/secure-20170115
[root@huanqiu_web1 ~]# ll /var/log/spooler*
-rw------- 1 root root 0 Jan 15 03:41 /var/log/spooler
-rw------- 1 root root 0 Dec 18 03:21 /var/log/spooler-20161225
-rw------- 1 root root 0 Dec 25 03:11 /var/log/spooler-20170101
-rw------- 1 root root 0 Jan  1 03:06 /var/log/spooler-20170108
```

```
-rw------- 1 root root 0 Jan  8 03:37 /var/log/spooler-20170115
```

tomcat日志切割一例

```
[root@huanqiu-backup ~]# cat /etc/logrotate.d/tomcat
/Data/app/tomcat-7-huanqiu/logs/catalina.out {
rotate 14
daily
copytruncate
compress
notifempty
missingok
}

[root@huanqiu-backup ~]# ll /Data/app/tomcat-7-huanqiu/logs/catalina.*
-rw-r--r--. 1 root root     0 Jan 19 19:11 /Data/app/tomcat-7-huanqiu/logs/catalina.out
-rw-r--r--. 1 root root 95668 Jan 19 19:11 /Data/app/tomcat-7-huanqiu/logs/catalina.out.1.gz
```

早期用过的nginx日志处理一例

```
[root@letv-backup ~]# vim /letv/sh/cut_nginx_log.sh
#!/bin/bash
# 你的日志文件存放目录
logs_path="/letv/logs/"
# 日志文件的名字，多个需要空格隔开
logs_names=(error access pv_access)
dates=`date -d "yesterday" +"%Y%m%d"`
mkdir -p ${logs_path}$dates/
num=${#logs_names[@]}
for((i=0;i<num;i++));do
mv ${logs_path}${logs_names[i]}.log ${logs_path}$dates/${logs_names[i]}.log
done
#nginx平滑重启
kill -USR1 `cat /letv/logs/nginx/nginx.pid`

结合crontab定时执行
[root@letv-backup ~]# crontab -e
#nginx日志切割
00 00 * * * cd /letv/logs;/bin/bash /letv/sh/cut_nginx_log.sh > /dev/null 2>$1
```

3）尝试解决logrotate无法自动轮询日志的办法

现象说明：

使用logrotate轮询nginx日志，配置好之后，发现nginx日志连续两天没被切割，这是为什么呢？？

然后开始检查日志切割的配置文件是否有问题，检查后确定配置文件一切正常。

于是怀疑是logrotate预定的cron没执行，查看了cron的日志，发现有一条Dec 7 04:02:01 www crond[18959]: (root) CMD (run-parts /etc/cron.daily)这样的日志，证明cron在04:02分时已经执行/etc/cron.daily目录下的程序。

接着查看/etc /cron.daily/logrotate（这是logrotate自动轮转的脚本）的内容：

```
[root@huanqiu_test ~]# cat /etc/cron.daily/logrotate
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

没有发现异常，配置好的日志轮转操作都是由这个脚本完成的，一切运行正常，脚本应该就没问题。

直接执行命令：

```
[root@huanqiu_test ~]# /usr/sbin/logrotate /etc/logrotate.conf
```

这些系统日志是正常轮询了，但nginx日志却还是没轮询。

接着强行启动记录文件维护操作，纵使logrotate指令认为没有需要，应该有可能是logroate认为nginx日志太小，不进行轮询。

故需要强制轮询，即在/etc/cron.daily/logrotate脚本中将 -t 参数替换成 -f 参数

```
[root@huanqiu_test ~]# cat /etc/cron.daily/logrotate
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
```

```
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -f logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

最后最后重启下cron服务：

```
[root@huanqiu_test ~]# /etc/init.d/crond restart
Stopping crond: [ OK ]
Starting crond: [ OK ]
```


阅读原文