

## Tomcat8 日志过大解决方案

### 说明

本日志rollup方案以脚本和 crontab 定时任务为主，主要是保留N天的日志数据，并且对大日志按照 20MB（可以自定义）大小切割。对于重要的日志数据，可以通过脚本将历史日志文件加密后扔到存储桶中（如：AWS S3），可以通过本方案的脚本进行扩展。

1 场景	3
2 解决方案	3
3 方案细则	3
3.1 检查crontab系统是否安装	3
3.2 脚本内容	3
3.3 配置 crontab 定时任务	5

作者	版本	密级	备注
姜鹏	V0.1	仅内部使用	日志rollup解决方案



## 1 场景

Tomcat8 catalina.out 日志过大，容易导致磁盘空间不够；不方便查看。

## 2 解决方案

通过shell命令删除前N天的历史日志文件，只保留N天前的日志；将保留的日志按照文件大小切割，每20M切割。将脚本配置在crontab中，每分钟执行一次。

## 3 方案细则

### 3.1 检查crontab系统是否安装

一般系统都会自带 crontab。输入： crontab -l 出现响应就证明已安装。

```
[[root@izwz97ihkr7f1efzto963sz ~]# crontab -l
no crontab for root
```

如果没有，下载安装包解压执行安装：

```
$yum install crontab
```

或者：

```
$/configure --prefix=/bankapp/crontab
```

```
$make & make install
```

### 3.2 脚本内容

保留前N天的日志数据，所有的日志文件超过20M就开始rollup 。

创建脚本文件：

```
$touch /bankapp/scripts/splash_log.sh
```

开始编辑：

```
$vi /bankapp/scripts/splash_log.sh
```

内容如下：

```
#!/bin/sh
maxsize=$((1024*1024*20))
keep_days=2
log_path=/usr/home/softwares/tomcat/logs
ymd=`date +%Y-%m-%d`
hms=`date +%H-%M-%S`

#rm -rf ${log_path}/manager*.log
#rm -rf ${log_path}/host-manager*.log
#rm -rf ${log_path}/localhost*.log

# start delete the expired files
j=0

cd $log_path

declare -a files

for i in `ls -l|awk '{print $9}'`
do
    files[j]=$i
    #echo $i
    j=`expr $j + 1`
done

for file in ${files[@]}
do
    cols=`echo $file | awk -F"." '{print NF}'`
    echo $file

    if [ $cols -eq 3 ]
    then
        time=`echo $file|awk -F"." '{print $2}'`
        #echo $time
        file_time=`date -d "${time}" +%s`
        now_time=`date -d "${ymd}" +%s`
        stampDiff=`expr $now_time - $file_time`
        dayDiff=`expr $stampDiff / 86400`
        if [ $dayDiff -gt 3 ]
        then
            echo $file 'expired, is being deleted '
            rm -rf $file
        fi
    fi
done
```

5

```
    fi
  fi
done

# end delete the expired files


#start splash the remain logfiles

declare -a remain_files

for i in `ls -l|awk '{print $9}`

do
  remain_files[j]=$i
  #echo $i
  j=`expr $j + 1`
done


for file in ${remain_files[@]}
do
  file_size=`ls -l $file | awk '{ print $5 }'`
  #echo 'file_size:'$file_size
  diff=`expr $file_size - $maxsize`
  if [ ${file_size} -ge ${maxsize} ];then
    cd ${log_path} && cp $file ${log_path}/file.${hms}.log && cat /dev/null > $file
  fi
done

#end splash the remain logfiles

echo 'logfiles rotate done'
```

### 3.3 配置 crontab 定时任务

每分钟执行一次脚本splash\_log.sh

\$crontab -e

配置如下任务：

```
*/1 * * * * /bankapp/scripts/splash_log.sh
```

查看任务执行情况：

\$tail -f /var/log/cron

```
[root@ip-172-31-20-28 ec2-user]# tail -f /var/log/cron
Jan  9 06:01:01 ip-172-31-20-28 CROND[26694]: (root) CMD (run-parts /etc/cron.hourly)
Jan  9 06:01:01 ip-172-31-20-28 run-parts(/etc/cron.hourly)[26694]: starting 0anacron
Jan  9 06:01:01 ip-172-31-20-28 run-parts(/etc/cron.hourly)[26703]: finished 0anacron
Jan  9 07:01:01 ip-172-31-20-28 CROND[9153]: (root) CMD (run-parts /etc/cron.hourly)
Jan  9 07:01:01 ip-172-31-20-28 run-parts(/etc/cron.hourly)[9153]: starting 0anacron
Jan  9 07:01:01 ip-172-31-20-28 run-parts(/etc/cron.hourly)[9162]: finished 0anacron
Jan  9 07:44:40 ip-172-31-20-28 crontab[12132]: (ec2-user) BEGIN EDIT (ec2-user)
Jan  9 07:45:38 ip-172-31-20-28 crontab[12132]: (ec2-user) REPLACE (ec2-user)
Jan  9 07:45:38 ip-172-31-20-28 crontab[12132]: (ec2-user) END EDIT (ec2-user)
Jan  9 07:46:01 ip-172-31-20-28 CROND[12159]: (ec2-user) CMD (/home/ec2-user/splash_log.sh)
Jan  9 07:47:01 ip-172-31-20-28 CROND[12284]: (ec2-user) CMD (/home/ec2-user/splash_log.sh)
```

从日志可以看出， 每间隔一分钟执行了一次 splash\_log.sh

Done66