一日一技：使用 Linux 自带的 logrotate 管理你的所有日志 转载

未闻Code 2021-09-17 12:02:38

文章标签 mongodb 日志文件 配置文件 产品经理 帮助文档 文章分类 运维 阅读数 129

我们在开发的过程中，会创建非常多的日志，对日志进行 rotate 是一个基本要求。

所谓的

```
rotate
```

，可以理解为对日志按照一定的规则进行切分。例如，每天晚上0点生成一个新的日志文件，并把老的文件归档。又或者每个日志文件超过多少 MB 以后就自动切分，并把老的内容单独存档或者压缩。存档以后的日志文件保存多少个。超过数量以后先删除老日志再删除新日志。

如果我们使用的是 Python，那么我们可以使用自带的

```
logging
```

模块或者第三方的

```
logoru
```

来写日志。但如果我们使用像是 MongoDB这种第三方的软件，那么要对日志进行 rotate 就非常麻烦了。如果不加以管理，MongoDB 的日志很容易就达到几十GB。

还有其他软件，他们的日志散落在系统的各个位置，我们应该如何替他们 rotate 呢？

如果你的系统是 Linux，那么主流发行版一般都会自带一个软件，叫做

```
logrotate
```

，通过简单的配置，就能让它帮你管理系统中各个地方的日志。

我们以 MongoDB 为例来进行说明。

MongoDB 的日志默认放在

```
/var/log/mongodb
```

文件夹中，如下图所示：

如果不干涉，那么所有的日志内容会无限制追加到这一个

```
mongod.log
```

文件中。并且，如果你的 MongoDB 正在运行，即使你使用

```
rm
```

命令删除了这个文件，它占用的空间也不会释放。

但使用

```
logrotate
```

来管理这个文件以后，它能通过

```
truncate
```

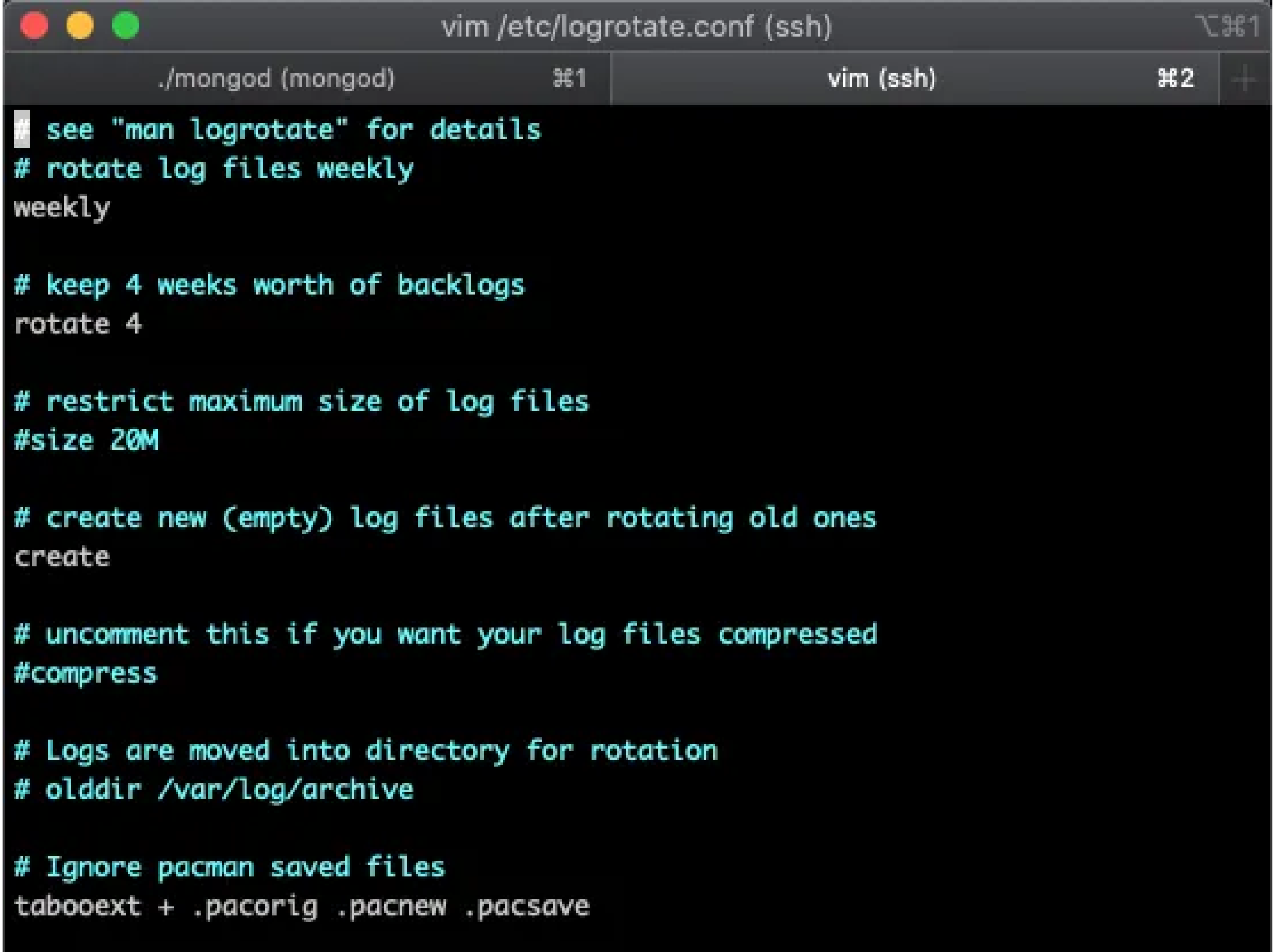
操作清空这个文件，从而实现释放空间的目的。

```
logrotate
```

的配置文件地址有两个，第一个主配置文件地址为

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

，其内容如下：


vim /etc/logrotate.conf (ssh)
⌥⌘1
./mongod (mongod)
⌘1
vim (ssh)
⌘2
see "man logrotate" for details
rotate log files weekly
weekly
keep 4 weeks worth of backlogs
rotate 4
restrict maximum size of log files
#size 20M
create new (empty) log files after rotating old ones
create
uncomment this if you want your log files compressed
#compress
Logs are moved into directory for rotation
olddir /var/log/archive
Ignore pacman saved files
tabooext + .pacorig .pacnew .pacsave

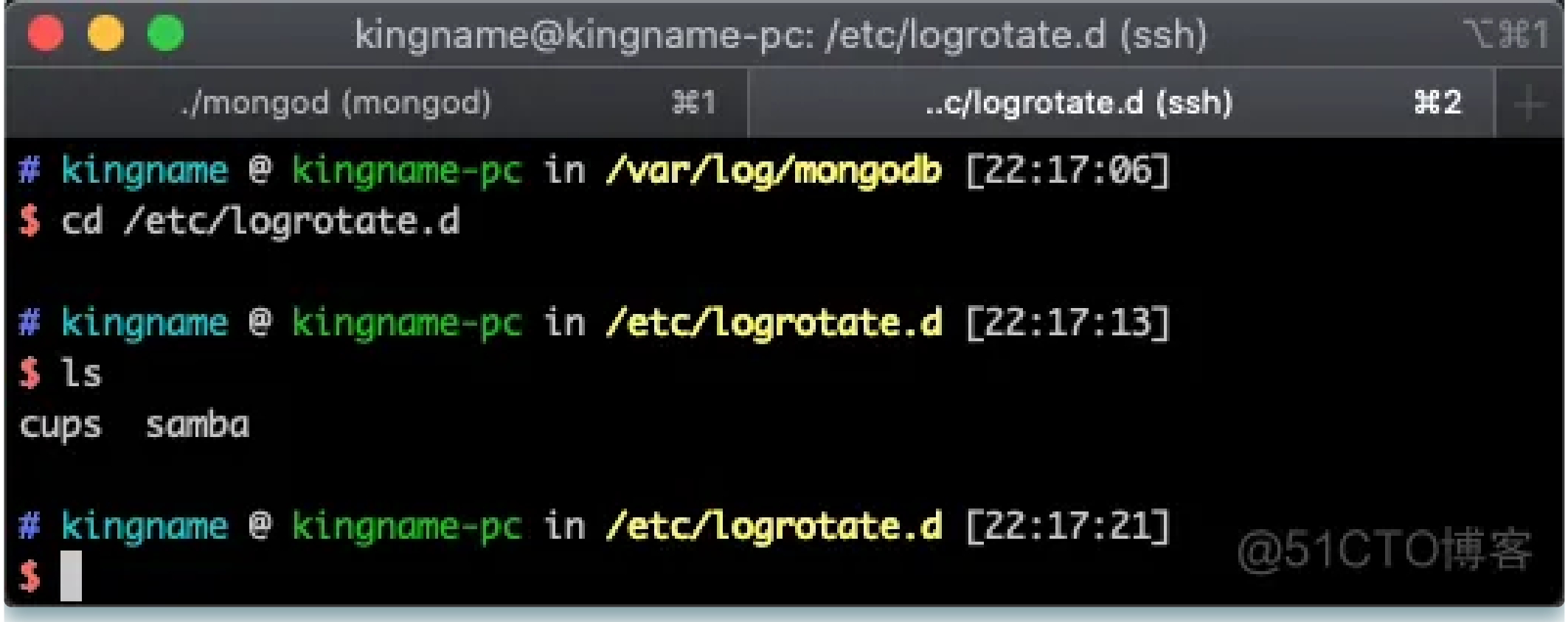
```
# Arch packages drop log rotation information into this directory
include /etc/logrotate.d

/var/log/wtmp {
    monthly
    create 0664 root utmp
    minsize 1M
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}
                                                    1,1           全部
```

@51CTO博客

另一个专门用于存放配置文件的地址为

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

，它里面的每一个文件都是一个配置文件，如下图所示：



我们要使用

```
logrotate
```

来管理 MongoDB，所以就在

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

文件夹中创建一个

```
mongodb
```

文件（文件名可以任意取），内容如下：

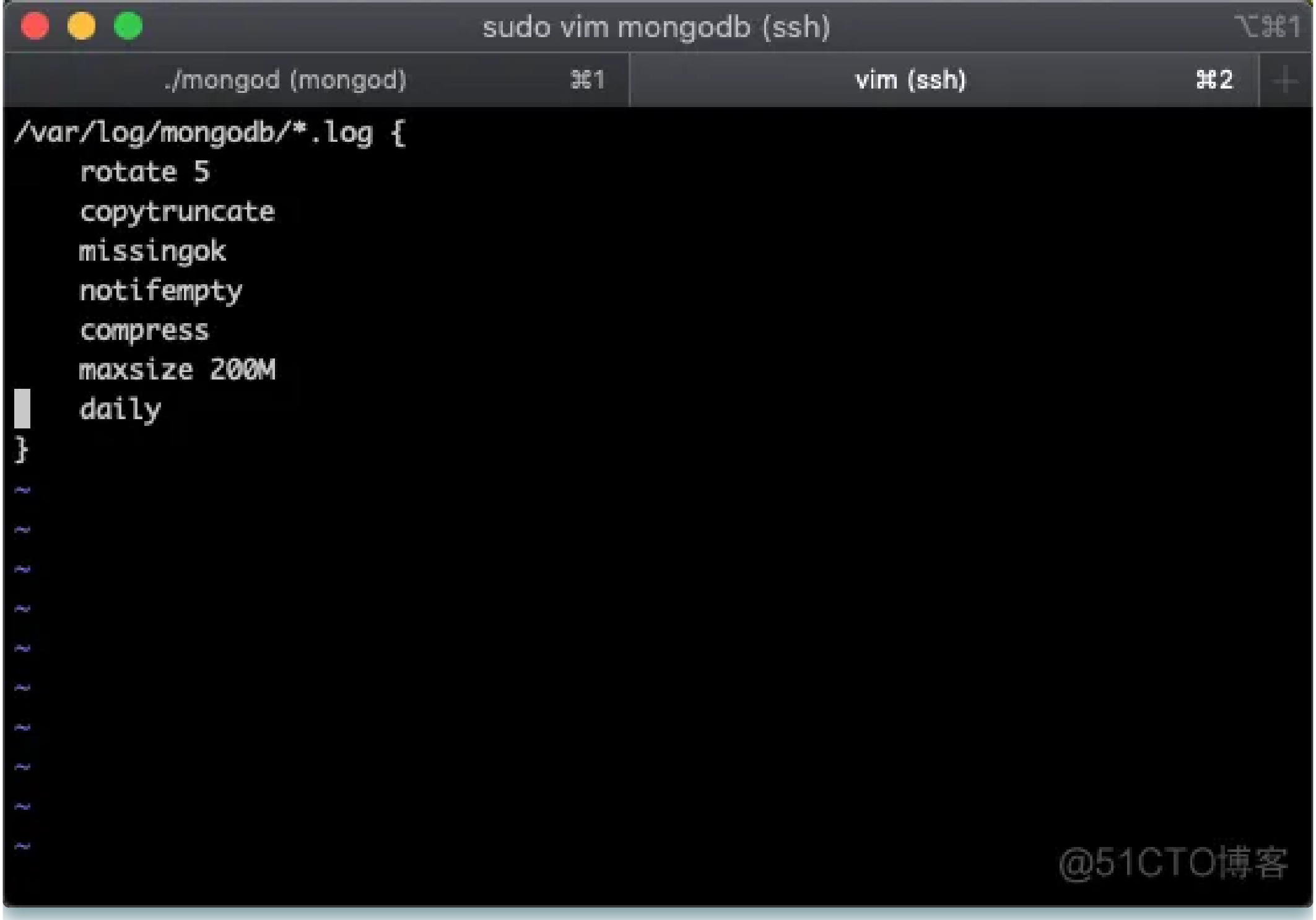
```
1.
```

```
/var/log/mongodb/*.log {
    rotate 5
    copytruncate
```

```
    missingok
    compress
    maxsize 200M
    daily
}
```

```
1.
```

如下图所示：



这个配置文件的意义如下：

*第一行用于指定 MongoDB 日志文件的地址为

```
/var/log/mongodb/
```

文件夹下面所有以

```
.log
```

结尾的文件。

- `rotate 5`

  表示保留5份日志文件

- copytruncate 表示，日志满足要求以后，先复制一份，然后把原来的日志文件清空
- missingok 表示如果这个文件夹下面为空，也没关系，自动忽略
- compress 表示要把日志进行压缩
- maxsize 200M 表示每个日志最多200MB，达到200MB 以后，就进行 rotate。
- daily 表示每天检查一次

保存这个文件，

```
logrotate
```

将会每天自动检查 MongoDB 的日志文件，如果发现它大小达到了200MB，那么首先会把这个日志复制出来一份（此时将会占用400MB 空间），然后把原来的日志文件内容清空（此时恢复200MB 空间占用）。被复制出来的日志文件会被压缩，并存放在当前文件夹中。当日志数量达到5个的时候，第六个日志文件生成，最开始压缩的那个文件被删除。从而保证 MongoDB 的日志及其备份总共占用空间不超过1GB。

```
logrotate
```

非常强大，还有非常多的配置参数用于实现各种日志管理的操作，大家可以执行命令

```
man logrotate
```

查看它的帮助文档。

CentOS7下日志轮转logrotate简单入门与实践

2022-12-04 257 发布于吉林

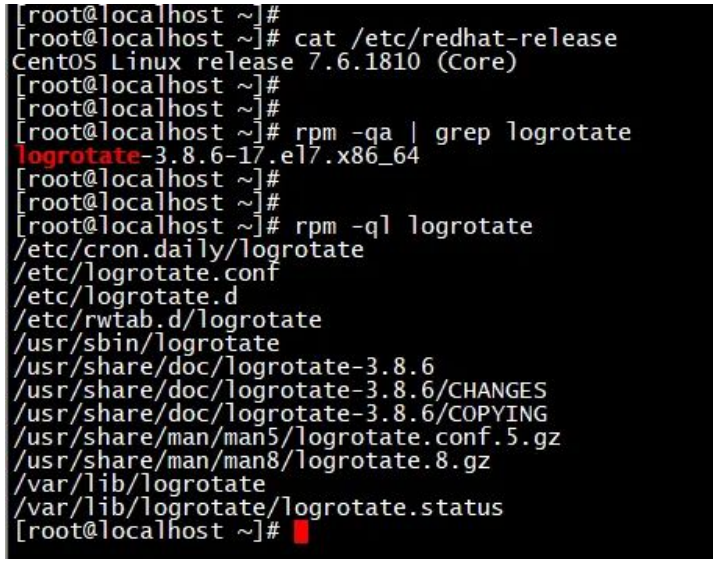
版权 ···

一、logrotate介绍

logrotate是一个日志文件管理工具。用来把旧文件轮转、压缩、删除，并且创建新的日志文件。我们可以根据日志文件的大小、天数等来转储，便于对日志文件管理，一般都是通过cron计划任务来完成的

1、CentOS7发行版上都默认安装有logrotate包

```
rpm -qa | grep logrotate
rpm -ql logrotate
```



2、logrotate的配置文件

logrotate的配置文件是/etc/logrotate.conf，通常不需要对它进行修改。
可以针对特定应用程序或服务日志文件的轮循设置在独立的配置文件中，放在/etc/logrotate.d/目录下

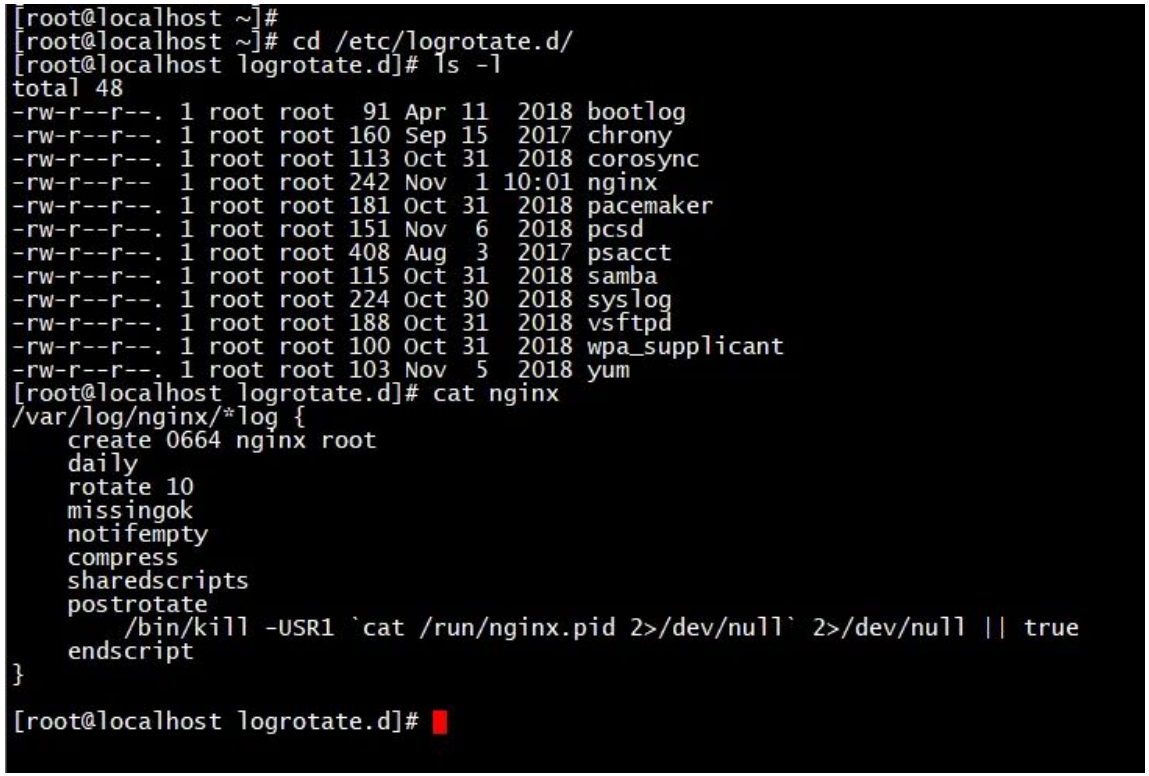
```
cat /etc/logrotate.conf
cd /etc/logrotate.d/
ls -l
```

```
[root@localhost ~]# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
        minsize 1M
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
[root@localhost ~]#
```

3、以nginx的日志轮转配置为例

```
/var/log/nginx/*log {
    create 0664 nginx root
    daily
```

```
    rotate 10
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
        /bin/kill -USR1 `cat /run/nginx.pid 2>/dev/null` 2>/dev/null || true
    endscript
}
```



参数说明：

- **create 0644 nginx root**: 以指定的权限创建全新的日志文件，同时logrotate也会重命名原始日志文件
- **daily**:日志文件将按日轮循
- **rotate 10**: 一次将存储10个归档日志。对于第11个归档，时间最久的归档将被删除
- **missingok**: 在日志轮循期间，任何错误将被忽略，例如“文件无法找到”之类的错误。
- **notifempty**: 如果日志文件为空，轮循不会进行。
- **compress**: 在轮循任务完成后，已轮循的归档将使用gzip进行压缩
- **postrotate/endscript:** 在所有其它指令完成后，postrotate和endscript里面指定的命令将被执行

4、logrotate日志轮转实操举例

```
[root@localhost ~]#
[root@localhost ~]# logrotate --help
Usage: logrotate [OPTION...] <configfile>
  -d, --debug               Don't do anything, just test (implies -v)
  -f, --force               Force file rotation
  -m, --mail=command        Command to send mail (instead of `/bin/mail')
  -s, --state=statefile     Path of state file
  -v, --verbose             Display messages during rotation
  -l, --log=STRING          Log file
  --version                 Display version information

Help options:
  -?, --help                Show this help message
  --usage                   Display brief usage message
[root@localhost ~]#
```

1)、排障过程中的最佳选择是使用‘-d’选项以预演方式运行logrotate

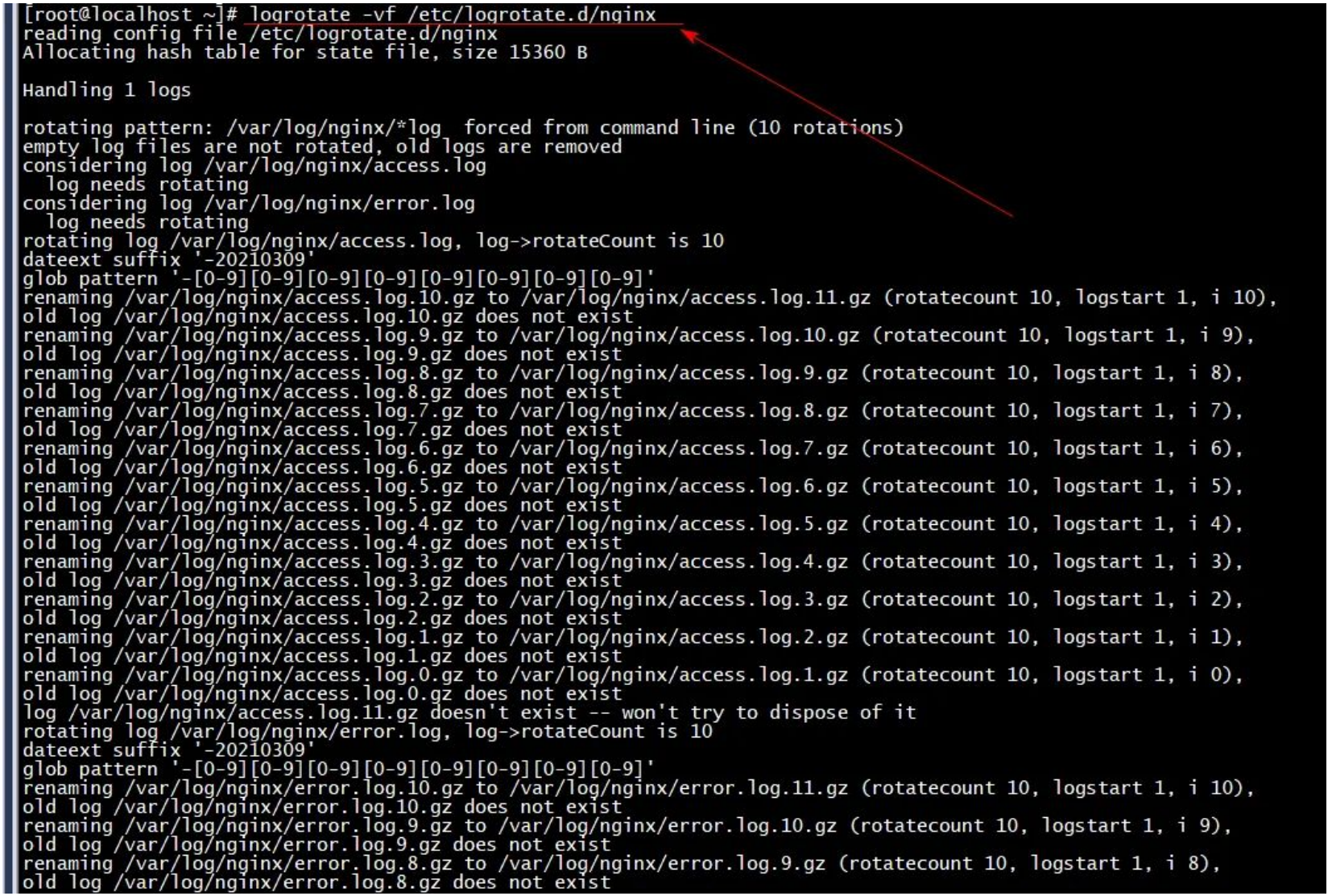
```
1 logrotate -d /etc/logrotate.d/nginx
```

```
[root@localhost ~]#
[root@localhost ~]# logrotate -d /etc/logrotate.d/nginx
reading config file /etc/logrotate.d/nginx
Allocating hash table for state file, size 15360 B

Handling 1 logs

rotating pattern: /var/log/nginx/*log  after 1 days (10 rotations)
empty log files are not rotated, old logs are removed
considering log /var/log/nginx/access.log
  log does not need rotating (log has been already rotated)considering log /var/log/nginx/error.log
  log does not need rotating (log has been already rotated)not running postrotate script, since no logs were rotated
[root@localhost ~]#
[root@localhost ~]#
```

从上面的输出结果可以看到的，logrotate判断该轮循是不必要的

2)、即使轮循条件没有满足，我们也可以通过使用‘-f’选项来强制logrotate轮循日志文件，‘-v’参数提供了详细的输出

```
1 logrotate -vf /etc/logrotate.d/nginx
```


[root@localhost ~]# logrotate -vf /etc/logrotate.d/nginx
reading config file /etc/logrotate.d/nginx
Allocating hash table for state file, size 15360 B

Handling 1 logs

rotating pattern: /var/log/nginx/*log forced from command line (10 rotations)
empty log files are not rotated, old logs are removed
considering log /var/log/nginx/access.log
  log needs rotating
considering log /var/log/nginx/error.log
  log needs rotating
rotating log /var/log/nginx/access.log, log->rotateCount is 10
dateext suffix '-20210309'
glob pattern '-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
renaming /var/log/nginx/access.log.10.gz to /var/log/nginx/access.log.11.gz (rotatecount 10, logstart 1, i 10),
old log /var/log/nginx/access.log.10.gz does not exist
renaming /var/log/nginx/access.log.9.gz to /var/log/nginx/access.log.10.gz (rotatecount 10, logstart 1, i 9),
old log /var/log/nginx/access.log.9.gz does not exist
renaming /var/log/nginx/access.log.8.gz to /var/log/nginx/access.log.9.gz (rotatecount 10, logstart 1, i 8),
old log /var/log/nginx/access.log.8.gz does not exist
renaming /var/log/nginx/access.log.7.gz to /var/log/nginx/access.log.8.gz (rotatecount 10, logstart 1, i 7),
old log /var/log/nginx/access.log.7.gz does not exist
renaming /var/log/nginx/access.log.6.gz to /var/log/nginx/access.log.7.gz (rotatecount 10, logstart 1, i 6),
old log /var/log/nginx/access.log.6.gz does not exist
renaming /var/log/nginx/access.log.5.gz to /var/log/nginx/access.log.6.gz (rotatecount 10, logstart 1, i 5),
old log /var/log/nginx/access.log.5.gz does not exist
renaming /var/log/nginx/access.log.4.gz to /var/log/nginx/access.log.5.gz (rotatecount 10, logstart 1, i 4),
old log /var/log/nginx/access.log.4.gz does not exist
renaming /var/log/nginx/access.log.3.gz to /var/log/nginx/access.log.4.gz (rotatecount 10, logstart 1, i 3),
old log /var/log/nginx/access.log.3.gz does not exist
renaming /var/log/nginx/access.log.2.gz to /var/log/nginx/access.log.3.gz (rotatecount 10, logstart 1, i 2),
old log /var/log/nginx/access.log.2.gz does not exist
renaming /var/log/nginx/access.log.1.gz to /var/log/nginx/access.log.2.gz (rotatecount 10, logstart 1, i 1),
old log /var/log/nginx/access.log.1.gz does not exist
renaming /var/log/nginx/access.log.0.gz to /var/log/nginx/access.log.1.gz (rotatecount 10, logstart 1, i 0),
old log /var/log/nginx/access.log.0.gz does not exist
log /var/log/nginx/access.log.11.gz doesn't exist -- won't try to dispose of it
rotating log /var/log/nginx/error.log, log->rotateCount is 10
dateext suffix '-20210309'
glob pattern '-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
renaming /var/log/nginx/error.log.10.gz to /var/log/nginx/error.log.11.gz (rotatecount 10, logstart 1, i 10),
old log /var/log/nginx/error.log.10.gz does not exist
renaming /var/log/nginx/error.log.9.gz to /var/log/nginx/error.log.10.gz (rotatecount 10, logstart 1, i 9),
old log /var/log/nginx/error.log.9.gz does not exist
renaming /var/log/nginx/error.log.8.gz to /var/log/nginx/error.log.9.gz (rotatecount 10, logstart 1, i 8),
old log /var/log/nginx/error.log.8.gz does not exist

```
renaming /var/log/nginx/error.log.1.gz to /var/log/nginx/error.log.2.gz (rotatecount 10, logstart 1, i 1),
old log /var/log/nginx/error.log.1.gz does not exist
renaming /var/log/nginx/error.log.0.gz to /var/log/nginx/error.log.1.gz (rotatecount 10, logstart 1, i 0),
old log /var/log/nginx/error.log.0.gz does not exist
log /var/log/nginx/error.log.11.gz doesn't exist -- won't try to dispose of it
renaming /var/log/nginx/access.log to /var/log/nginx/access.log.1
creating new /var/log/nginx/access.log mode = 0664 uid = 995 gid = 0
renaming /var/log/nginx/error.log to /var/log/nginx/error.log.1
creating new /var/log/nginx/error.log mode = 0664 uid = 995 gid = 0
running postrotate script
compressing log with: /bin/gzip
compressing log with: /bin/gzip
[root@localhost ~]# cd /var/log/nginx/
[root@localhost nginx]# ll -trh
total 8.0K
-rw-r--r-- 1 root  root 207 Mar  9 10:51 error.log.1.gz
-rw-r--r-- 1 root  root 287 Mar  9 10:51 access.log.1.gz
-rw-rw-r-- 1 nginx root   0 Mar  9 11:00 error.log
-rw-rw-r-- 1 nginx root   0 Mar  9 11:00 access.log
[root@localhost nginx]#
```

二、crontab实现定时任务

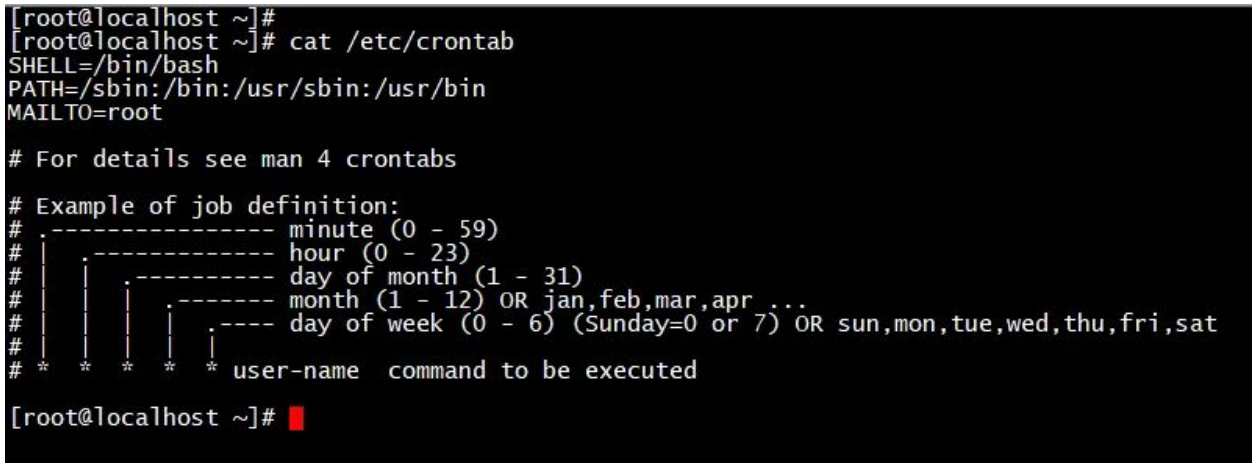
linux内置的cron进程能帮我们实现定时执行某些任务的需求，cron搭配shell脚本，非常复杂的指令也没有问题。

cron介绍

cron守护进程是一个由实用程序和配置文件组成的小型子系统，在几乎所有类 UNIX 系统上都可以找到某种风格的cron，我们可以用ps aux|grep cron找到crond这个守护进程。我们经常使用的是crontab命令是cron table的简写，它是cron的配置文件，也可以叫它作业列表，我们可以在以下文件夹内找到相关配置文件。

- /var/spool/cron/ 目录下存放的是每个用户包括root的crontab任务，每个任务以创建者的名字命名
- /etc/crontab 这个文件负责调度各种管理和维护任务。
- /etc/cron.d/ 这个目录用来存放任何要执行的crontab文件或脚本。

我们还可以把脚本放在/etc/con.hourly、/etc/con.daily、/etc/con.weekly、/etc/con.monthly目录中，让它每小时/天/星期、月执行一次。



crontab的使用

我们常用的命令如下：

```
crontab [-u username]    //省略用户表表示操作当前用户的crontab
    -e      (编辑工作表)
    -l      (列出工作表里的命令)
    -r      (删除工作)
```

我们用crontab -e进入当前用户的工作表编辑，是常见的vim界面。每行是一条命令。crontab的命令构成为 时间+动作，其时间有分、时、日、月、周五种，操作符有

```
* 取值范围内的所有数字
/ 每过多少个数字
- 从X到Z
,  散列数字
```

aaaa

以下是几个例子

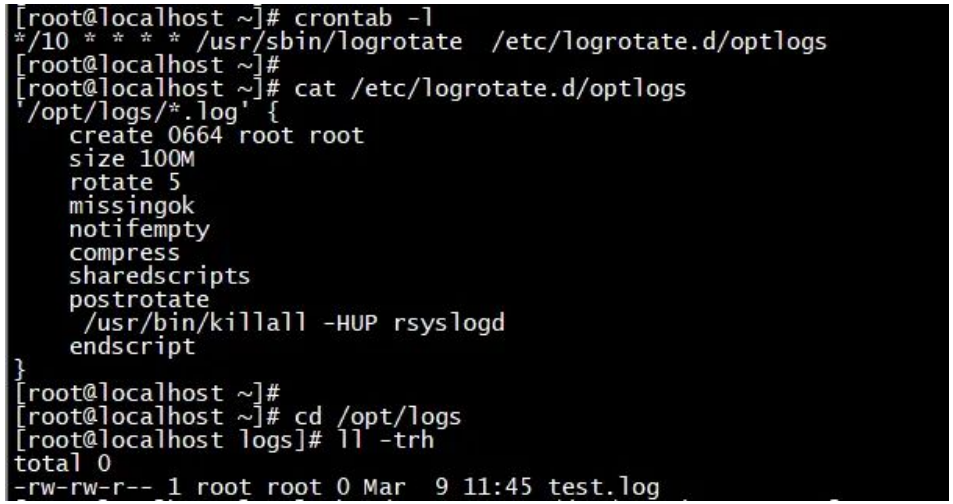
```
时间                    注释
0 0 25 12 *     //在12月25日的0时0分
```

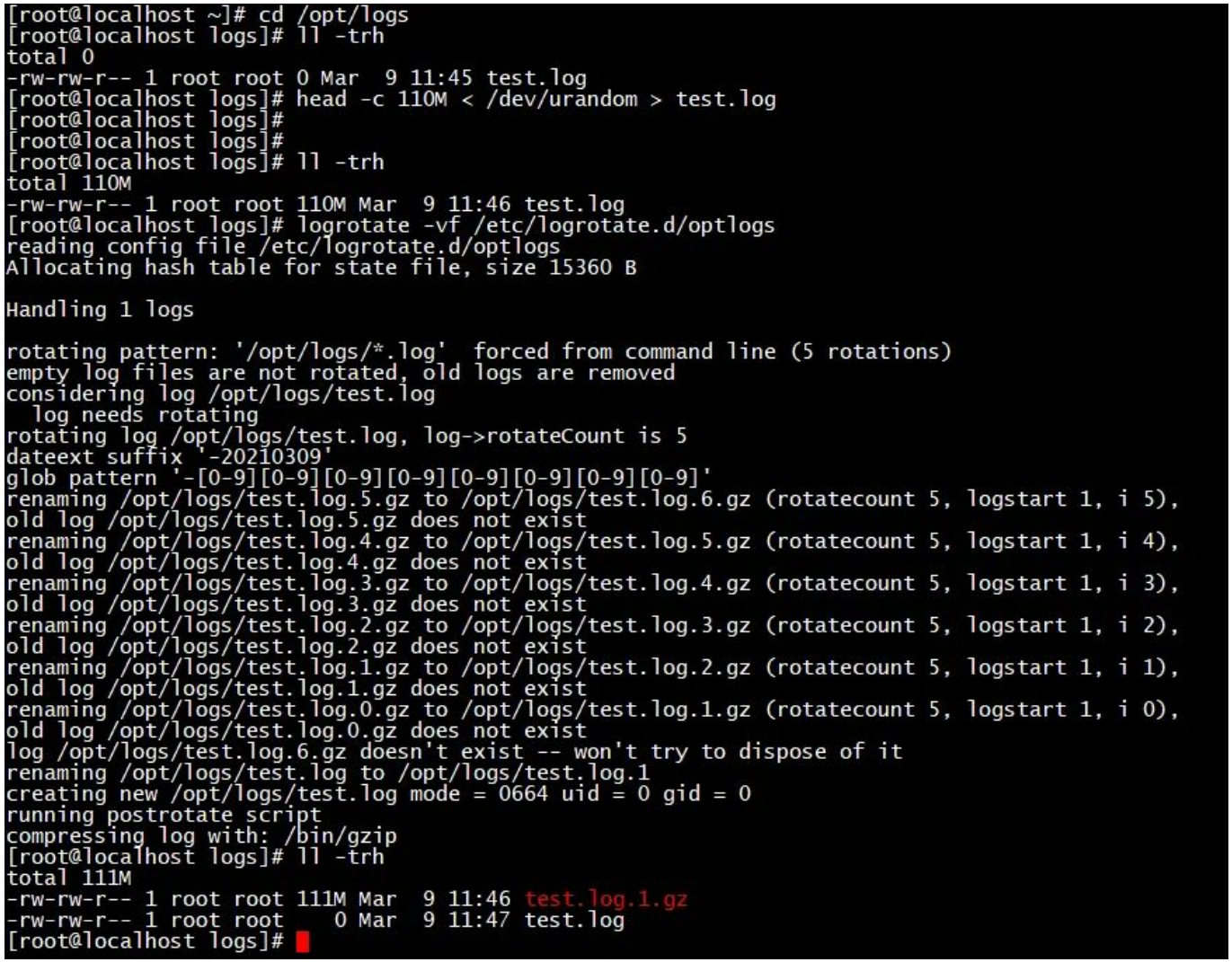
```
*/5 * * * *     //每过5分钟
* 4-6 * * *     //每天的4 5 6点
* * * * 2，5     //每周二和周五
```

三、crontab与logrotate配合使用举例

例如：每10分钟轮转/opt/logs/下的日志文件 当日志文件大小达到100M时进行轮转，最多保留5个

```
crontab -l
*/10 * * * * /usr/sbin/logrotate  /etc/logrotate.d/optlogs
cat /etc/logrotate.d/optlogs
'/opt/logs/*.log' {
    create 0664 root root
    size 100M
    rotate 5
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
     /usr/bin/killall -HUP rsyslogd
    endscript
}
```




[root@localhost ~]# cd /opt/logs
[root@localhost logs]# ll -trh
total 0
-rw-rw-r-- 1 root root 0 Mar 9 11:45 test.log
[root@localhost logs]# head -c 110M < /dev/urandom > test.log
[root@localhost logs]#
[root@localhost logs]#
[root@localhost logs]# ll -trh
total 110M
-rw-rw-r-- 1 root root 110M Mar 9 11:46 test.log
[root@localhost logs]# logrotate -vf /etc/logrotate.d/optlogs
reading config file /etc/logrotate.d/optlogs
Allocating hash table for state file, size 15360 B

Handling 1 logs

rotating pattern: '/opt/logs/*.log' forced from command line (5 rotations)
empty log files are not rotated, old logs are removed
considering log /opt/logs/test.log
  log needs rotating
rotating log /opt/logs/test.log, log->rotateCount is 5
dateext suffix '-20210309'
glob pattern '-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
renaming /opt/logs/test.log.5.gz to /opt/logs/test.log.6.gz (rotatecount 5, logstart 1, i 5),
old log /opt/logs/test.log.5.gz does not exist
renaming /opt/logs/test.log.4.gz to /opt/logs/test.log.5.gz (rotatecount 5, logstart 1, i 4),
old log /opt/logs/test.log.4.gz does not exist
renaming /opt/logs/test.log.3.gz to /opt/logs/test.log.4.gz (rotatecount 5, logstart 1, i 3),
old log /opt/logs/test.log.3.gz does not exist
renaming /opt/logs/test.log.2.gz to /opt/logs/test.log.3.gz (rotatecount 5, logstart 1, i 2),
old log /opt/logs/test.log.2.gz does not exist
renaming /opt/logs/test.log.1.gz to /opt/logs/test.log.2.gz (rotatecount 5, logstart 1, i 1),
old log /opt/logs/test.log.1.gz does not exist
renaming /opt/logs/test.log.0.gz to /opt/logs/test.log.1.gz (rotatecount 5, logstart 1, i 0),
old log /opt/logs/test.log.0.gz does not exist
log /opt/logs/test.log.6.gz doesn't exist -- won't try to dispose of it
renaming /opt/logs/test.log to /opt/logs/test.log.1
creating new /opt/logs/test.log mode = 0664 uid = 0 gid = 0
running postrotate script
compressing log with: /bin/gzip
[root@localhost logs]# ll -trh
total 111M
-rw-rw-r-- 1 root root 111M Mar 9 11:46 test.log.1.gz
-rw-rw-r-- 1 root root   0 Mar 9 11:47 test.log
[root@localhost logs]#