日志和计划任务

日志和计划任务
- 日志
  - 日志的作用
  - 常用的日志
  - 日志的类型
  - 日志需要的服务和程序
  - rsyslog 的相关配置文件
  - /etc/rsyslog.conf
    - 设施 facility
    - 级别 loglevel
    - 动作 action
    - 示例
  - 重启服务
- logrotate 日志轮询
  - 参数和功能
  - 实验
- 计划任务
  - at
    - 服务的启动命令
    - at 命令的用法
    - 范例
  - crontab
    - crontab 服务的启动命令
    - crontab 命令的用法
    - 范例
- 日志和计划任务课后作业

日志

日志的作用

- 解决系统方面的错误
- 解决网络服务的问题
- 过往事件记录

常用的日志

日志	解释
/var/log/cron	crontab 计划任务
/var/log/dmesg	开机核心侦测信息
/var/log/lastlog	系统所有帐号最近一次登陆信息
/var/log/maillog	邮件往来信息
/var/log/messages	系统错误信息
/var/log/secure	涉及输入帐号密码的程序
/var/log/wtmp	正确登陆系统的账户信息
/var/log/btmp	错误登陆系统的账户信息
/var/log/httpd/*	
/var/log/samba/*	不同的网络服务会使用它们自己的登录文件来记载它们自己产生的各项信息

日志的类型

1> 可查看的 ASCII 的日志

• messages

• 与程序同名的目录下, 会记录和该程序相关的一些日志

2> 不可查看的 data 日志

需要调用某些命令才能去查看对应的日志

• wtmp <==last: 系统登陆登出信息

• btmp <==lastb: 错误的系统登陆登出情况

```
[#3#root@rhel6 ~]#file /var/log/wtmp
/var/log/wtmp: data
```

日志需要的服务和程序

rsyslogd 主要负责记录系统运作中，kernel 或应用程序产生的各种讯息，讯息被写入系统日志

logrotate 主要在进行日志文件的轮替功能

- rhel5 版本以前是 syslogd 服务，rhel6 之后是 rsyslogd 服务 (reliable and extended 可靠的和拓展的 syslogd 服务)

『 (1) 什么服务 (2) 什么等级信息 (3) 需要被记录在哪里 (设备或文件) 』

rsyslog 的相关配置文件

• /etc/rsyslog.conf

• /etc/rsyslog.d/*

/etc/rsyslog.conf

/etc/rsyslog.conf 文件中的一项配置记录由 “ 选项 ” (selector) 和 “ 动作 ” (action)两个部分组成,两者间用 tab 制表符进行分隔(使用空格间隔是无效的)。

“ 选项 ” (selector)由两部分组成:

- 设施 facility 和级别 loglevel,由点号.分隔,两部分都是大小写不敏感;
- 设施和级别都在 syslog(3)中有描述。各保留字段间用分号分隔。
- 如下行所示:

设施. 级别 [;设施.级别]TAB 动作

设施 **facility**

保留字段中的 “ 设施 ” (facility)代表信息产生的源头,可以是:

facility	设施
auth	认证系统,即询问用户名和口令
cron	系统定时系统执行定时任务时发出的信息
daemon	某些系统的守护程序的 syslog,如由 in.ftpd 产生的 log
kern	内核的 syslog 信息
lpr	打印机的 syslog 信息
mail	邮件系统的 syslog 信息
mark	定时发送消息的时标程序
news	新闻系统的 syslog 信息
user	本地用户应用程序的 syslog 信息
uucp	uucp 子系统的 syslog 信息,unix to unix copy, unix 主机之间相关的通讯
local0-7	种本地类型的 syslog 信息,这些信息可以又用户来定义
*	代表以上各种设备

级别 **loglevel**

保留字段中的 “ 级别 ” 代表信息的重要性,可以是:

num	loglevel	级别
0	emerg	紧急,处于 Panic 状态。通常应广播到所有用户;几乎要当机
1	alert	告警,当前状态必须立即进行纠正。例如,系统数据库崩溃;
2	crit	关键状态的警告。例如,硬件故障;
3	err	其它错误;
4	warn	警告;
5	notice	注意;非错误状态的报告,但应特别处理;
6	info	通报信息;
7	debug	调试程序时的信息;
	none	通常调试程序时用,指示带有 none 级别的类型产生的信息无需送出。如
	*.debug;mail.none	表示调试时除邮件信息外其它信息都送出。
	.xxx:	表示大于等于 xxx 级别的信息
	.=xxx:	表示等于 xxx 级别的信息
	.!xxx:	表示在 xxx 之外的等级的信息拓展:

```
[#19#root@rhel6 ~]#man syslog
```

动作 **action**

“ 动作 ” 域指示信息发送的目的地。可以是:

action	动作
/filename	日志文件。由绝对路径指出的文件名,此文件必须事先建立;
@host	远程主机; @符号后面可以是 ip,也可以是域名,默认在/etc/hosts 文件下 loghost 这个别名已经指定给了本机。
user1,user2	指定用户。如果指定用户已登录,那么他们将收到信息;
*	所有用户。所有已登录的用户都将收到信息。

示例

1. 记录到普通文件或设备文件

```
*.* /var/log/file.log
*.* /dev/pts/0
```

测试: logger -p local3.info ‘KadeFor is testing the rsyslog and logger ‘

logger 命令用于产生日志

1. 转发到远程

```
*.* @192.168.0.1 # 使用 UDP 协议转发到 192.168.0.1 的 514(默认)端口
*.* @@192.168.0.1:10514 # 使用 TCP 协议转发到 192.168.0.1 的 10514(默认)端口
```

1. 发送给用户(需要在线才能收到)

```
*.* root
*.* root,kadefor,up01
*.* *
```

- 使用,号分隔多个用户
- *号表示所有在线用户

1. 忽略,丢弃

```
local3.* # 忽略所有 local3 类型的所有级别的日志
```

1. 执行脚本

```
local3.* ^/tmp/a.sh
```

- ^号后跟可执行脚本或程序的绝对路径

1. 一个标准的简单的配置文件

```
#### RULES ####
规则部分
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*
/dev/console
#关于内核的所有日志都放到/dev/console(控制台)
# 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
# 记录所有日志类型的 info 级别以及大于 info 级别的信息到/var/log/messages,但是 mail
邮件信息,authpriv 验证方面的信息和 cron 时间任务相关的信息除外
# The authpriv file has restricted access.
authpriv.*
/var/log/secure
# authpriv 验证相关的所有信息存放在/var/log/secure
# Log all the mail messages in one place.
mail.*
-/var/log/maillog
# 邮件的所有信息存放在/var/log/maillog; 这里有一个-符号, 表示是使用异步的方式记录,
因为日志一般会比较大
# Log cron stuff
cron.*
/var/log/cron
# 计划任务有关的信息存放在/var/log/cron
# Everybody gets emergency messages
*.emerg
*
# 所有日志类型的 emerg 信息发给所有用户
# Save news errors of level crit and higher in a special file.
uucp,news.crit
/var/log/spooler
# 记录 uucp,news.crit 等存放在/var/log/spooler
# Save boot messages also to boot.log
local7.*
/var/log/boot.log
# 启动的相关信息存放在/var/log/boot.log
```

重启服务

service rsyslog restart <==rhel6

systermctl restart rsyslog <==rhel7

ps：rsyslog 的日志只要『被编辑过』就无法继续记录！需要重启日志恢复。

logrotate 日志轮询

/usr/sbin/logrotate 将旧的日志文件移动成旧文件，并重新建立一个新的空的档案来记录

/etc/cron.daily/logrotate 记录每天要进行的日志轮替的行为

/etc/logrotate.conf

/etc/logrotate.d/*

程序的配置文件

```
[#14#root@rhel6 ~]#grep -v "^#" /etc/logrotate.conf|grep -v "^$"
weekly
rotate 4
create
dateext
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
}
```

可以通过 `man logrotate` 来查看更多的一些定义。

参数和功能

compress 通过 gzip 压缩转储以后的日志

nocompress 不需要压缩时,用这个参数

copytruncate 用于还在打开中的日志文件,把当前日志备份并截断

nocopytruncate 备份日志文件但是不截断

create mode owner group 转储文件,使用指定的文件模式创建新的日志文件

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。

errors errors 转储时的错误信息发送到指定的 Email 地址

ifempty 即使是空文件也转储,这个是 logrotate 的缺省选项。

notifempty 如果是空文件的话,不转储

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

prerotate/endscript 在转储以前需要执行的命令可以放入这个对,这两个关键字必须单独成行

postrotate/endscript 在转储以后需要执行的命令可以放入这个对,这两个关键字必须单独成行

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

tabootext [+] list 让 logrotate 不转储指定扩展名的文件,缺省的扩展名是:.rpm-orig, .rpmsave, v, 和 ~

size size 当日志文件到达指定的大小时才转储,Size 可以指定 bytes (缺省)以及 KB (sizek)或者 MB (sizem).

实验

要求:

1. 记录所有日志类型的 info 级别以及大于 info 级别的信息,保存到/var/log/test,但是 mail 邮件信息,authpriv 验证方面的信息和 cron 时间任务相关的信息除外
2. /var/log/test 日志轮询方式为:

• 每天轮询一次;

• 保留 4 个文件;

• 以时间命名;

• 创建与原日志同名的新文件。

第一步:修改 rsyslog 的配置文件/etc/rsyslog.conf

`*.info;mail.none;authpriv.none;cron.none`

第二步:创建空文件/var/log/test

`touch /var/log/test`

第三步:重新启动服务

`service rsyslog restart`

第四步:修改 logrotate 配置文件/etc/logrotate.conf

```
/var/log/test { <== 轮循的日志是谁
daily <== 轮询周期多久
rotate 4 <== 保留几个带时间戳的文件
dateext <== 是否以时间戳为文件更名格式
create <== 是否需要创建一个与原日志文件同名的新文件
```

第五步:测试 `logrotate -vf /etc/logrotate.conf`

```
[#41#root@rhel6 ~]#logrotate -vf /etc/logrotate.conf
[#42#root@rhel6 ~]#ls /var/log/|grep test
test
test-20160630
/var/log/test
```

计划任务

每个人或多火烧都有一些约会或者是工作,有的工作是例行性的, 例如每年一次的加薪、每个月一次 的工作报告、每周一次的午餐会报、每天需要的打卡等等; 有的工作则是临时发生的, 例如刚好总公 司有高官来访,需要你准备演示器材等等! 用在生活上面, 例如每年的爱人的生日、每天的起床时间 等等、还有突发性的计算机大降价 (啊!真希望天天都有!) 等等啰。

像上面这些例行性工作,通常你得要记录在日历上面才能避免忘记!不过, 由于我们常常在计算机前 面的缘故,如果计算机系统能够主动的通知我们的话,那么不就轻松多了!嘿嘿! 这个时候 Linux 的 计划任务就可以派上场了! 例如: 每一天早上 8:00 钟要服务器连接上音响,并自动播放音乐来唤你起床; 而中午 12:00 希望 Linux 可以发一封信到你的邮件信箱,提醒你可以去吃午餐了; 另外, 在每年的你爱人生 日的前一天,先发封信提醒你, 以免忘记这么重要的一天。

那么 Linux 的例行性工作是如何进行计划任务的呢? 咱 们的 Linux 是透过 crontab 和 at 这两个东西!这两个玩意儿有啥异同?就让我们来瞧瞧先!

	一次性计划任务	周期性计划任务
软件	at-3.1.10-43.el62.1.x8664	cronie-1.4.4-12.el6.x86_64
服务	atd	crond
命令	at	crontab
服务存放文件	/etc/init.d/atd	/etc/init.d/crond
系统配置文件	/etc/at.deny	/etc/cron.deny /etc/cron.d/*
程序缓存文件	/var/spool/at	/var/spool/cron/* /var/log/cron

at

当执行 at 程序并在终端输入命令后, 首先系统会到配置文件中寻找 at 相关的文档

- /etc/at.allow <== 记录了允许使用 at 命令的用户名
- /etc/at.deny <== 记录了不允许使用 at 命令的用户名

若以上两个文件都没有, 则系统默认只有 root 可以使用 at

输入的内容将被保存到以下目录中 /var/spool/at/*

服务的启动命令

rhel6

service atd start 启动

service atd restart 重启

service atd status 查看

rhel7

systermctl start atd

systermctl restart atd

systermctl status atd

at 命令的用法

```
at [-mldv] TIME

at -c 工作号码

选项与参数 :

-m 当 at 工作完成后 , 即使没有输出讯息 , 亦以 email 通知使用者该工作已完成。
-l at -l =atq, 列出目前系统上面的所有该用户的 at 计划任务 ;
-d at -d =atrm , 可以取消一个在 at 计划任务中的工作 ;
-v 可以使用较明显的时间格式显出 at 计划任务中的任务列表 ;
-c 可以列出后面接的该项工作的实际指令内容。
```

范例

1. 再过五分钟后, 将 /root/.bashrc 寄给 root 自己
2. 查看 at 计划任务中的工作
3. 查看该计划任务的实际指令内容

```
[#60#root@rhel6 ~]#at now + 5 minutes
at> /bin/mail root -s "testing at job" < /root/.bashrc
at> <EOT>
job 1 at 2016-06-30 00:40
[#61#root@rhel6 ~]#at -l
1
2016-06-30 00:40 a root
[#62#root@rhel6 ~]#date
Thu Jun 30 00:37:21 CST 2016
[#63#root@rhel6 ~]#at -c 1
#!/bin/sh
# atrun uid=0 gid=0
# mail root 0
umask 22
.........
此处省略
OLDPWD=/var/log; export OLDPWD
cd /root || {
echo 'Execution directory inaccessible' >&2
exit 1
}
${SHELL:-/bin/sh} << 'marcinDELIMITER00a5c603'
/bin/mail root -s "testing at job" < /root/.bashrc
marcinDELIMITER00a5c6034. 2016 年 10 月 20 日 12:00 广播一条信息 “ Happy birthday to me!”;取消该计划任务。
[#68#root@rhel6 ~]#at 12:00 2016-10-20
at> echo "Happy birthday to me!"|wall
at> <EOT>
job 2 at 2016-10-20 12:00
[#69#root@rhel6 ~]#atq
2
2016-10-20 12:00 a root
[#70#root@rhel6 ~]#at -l
2
2016-10-20 12:00 a root
[#71#root@rhel6 ~]#at -d 2
[#72#root@rhel6 ~]#atq
```

1. 由于机房预计划 2016/07/18 停电, 我想要在 2016/07/17 23:00 关机?

```
[#73#root@rhel6 ~]#at 23:00 2016-07-17
at> shutdown -h now
at> <EOT>
job 3 at 2016-07-17 23:00
[#74#root@rhel6 ~]#atq
3
2016-07-17 23:00 a root
```

crontab

crontab 服务的启动命令

rhel6

service crond start 启动

service crond restart 重启

service crond status 查看

rhel7

systermctl start crond

systermctl restart crond

systermctl status crond

crontab 命令的用法

```
crontab [-u username] [-l|-e|-r]

选项与参数 :
-u 只有 root 使用 , 亦即帮其他使用者建立 / 移除 crontab 计划任务;
-e 编辑 crontab 工作内容
-l 查阅 crontab 工作内容
-r 移除所有 crontab 的工作内容 , 若仅要移除一项 , 请用 -e 去编辑
```

crontab -e 编辑的格式说明:

代表意义	分钟	小时	日期	月份	周
数字范围	0-59	0-23	1-31	1-12	0-7

特殊字	代表意义
*(星号)	代表任何时刻
,(逗号)	代表分隔时段
-(减号)	代表一段时间范围
/n(斜线)	n 代表数字 , 『每隔 n 单位间隔』 , 例如每五分钟进行一次

`man 5 crontab` 查看具体用法帮助

范例

1. student 每天 12 点发广播给自己提醒要吃饭拉!

```
crontab -e
vi 编辑画面 每项工作都是一行。
0 12 * * * echo “Lunch time!!!!”|wall
分 时 日 月 周 |<======= 命令 ===========|
```

1. 每个月的第一天下午 2 点 15 分,将/etc 目录打包压缩成/tmp/etc.tar.bz2 文件。

```
15 14 1 * * tar -jcf /tmp/etc.tar.bz2 /etc
```

1. 周一到周五的晚上 10 点,将/var 目录打包压缩成/tmp/var.tar.bz2 文件。

```
0 22 * * 1-5 tar -jcf /tmp/var.tar.bz2 /var
```

1. 每天 0 点 23 分,2 点 23 分,4 点分...22 点 23 分,就输出 “ 休息一会 ” 到终端上。

```
23 0-23/2 * * * echo "have a rest"
```

1. 每周日的 4 点 5 分提醒自己去跑步。

```
5 4 * * sun echo "run at 5 after 4 every sunday"
```

日志和计划任务课后作业

1. 记录所有日志类型的 info 级别以及大于 info 级别的信息,保存到/var/log/test,但是 mail 邮件信息,authpriv 验证方面的信息和 cron 时间任务相关的信息除外
2. /var/log/test 日志轮询方式为:

- 每周轮询一次;
- 保留 6 个文件;
- 以时间命名;
- 创建与原日志同名的新文件。

---

1. 再过 10 分钟后, 将 /root/.bashrc 寄给 root 自己
2. 查看 at 计划任务中的工作
3. 查看该计划任务的实际指令内容
4. 由于机房预计划 2016/09/18 停电, 我想要在 2016/09/17 23:00 关 机?

---

1. student 每天上午 11:50 发广播给自己提醒要吃饭拉!
2. 每个月的第一天下午 5 点 30 分,将/etc 目录打包压缩成/tmp/etc.tar.bz2 文件。
3. 周一到周六的晚上 9 点,将/var 目录打包压缩成/tmp/var.tar.bz2 文件。
4. 每天 1 点 22 分,3 点 22 分,5 点 22 分...23 点 22 分,就输出 “ 休息一会 ” 到终端上。
5. 每周六的 6 点 20 分提醒自己去跑步。