Linux下的日志系统

- 前言：

其实这节应该学点网络知识基础，上一节涉及了一部分网络知识，但立志运维Linux的话，必须得把计网学好（它们之间的密切关系着），可惜内容太多，在这里写不如直接拿一本计网入门，所以我就不写了，直接开始日志系统。说到日志，相信大家都熟悉。日志是系统维护中非常重要的一个组成部分，这一节我们来讲讲Linux下的日志。

- 正文：

1、什么是日志

日志就是Linux下的普通的文本文件，里面的内容可能是系统、软件、服务等工作时写进去的，也可能是用户自己用脚边编程记录进去的。总而言之，日志就是记录系统等的允许情况和记录，时刻方便我们查看系统状况。

2、日志在哪和日志的一般格式

```
[root@localhost ~]# ls /var/log
anaconda  btmp    dmesg      grubby_prune_debug  maillog   rhsm    spooler   wtmp
audit     chrony  dmesg.old  httpd               messages  sa      tallylog  yum.log
boot.log  cron    firewalld  lastlog             qemu-ga   secure  tuned
[root@localhost ~]# tail /var/log/cron
Sep 20 16:01:01 localhost run-parts(/etc/cron.hourly)[1480]: finished 0anacron
Sep 20 16:06:50 localhost crond[793]: (CRON) INFO (Shutting down)
Sep 20 18:21:19 localhost crond[797]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 29% if us
ed.)
Sep 20 18:21:22 localhost crond[797]: (CRON) INFO (running with inotify support)
Sep 20 18:30:01 localhost CROND[1457]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Sep 20 18:40:01 localhost CROND[1544]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Sep 23 16:24:15 localhost crond[789]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 28% if us
ed.)
Sep 23 16:24:22 localhost crond[789]: (CRON) INFO (running with inotify support)
Sep 23 16:30:01 localhost CROND[1578]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Sep 23 16:40:02 localhost CROND[1835]: (root) CMD (/usr/lib64/sa/sa1 1 1)
[root@localhost ~]# cat /var/log/yum.log | tail -10
Sep 14 19:41:39 Installed: rest-0.8.1-2.el7.x86_64
Sep 14 19:41:39 Installed: libwmf-lite-0.2.8.4-44.el7.x86_64
Sep 14 19:41:39 Installed: json-glib-1.4.2-2.el7.x86_64
Sep 14 19:41:39 Installed: libtool-ltdl-2.4.2-22.el7_3.x86_64
Sep 14 19:41:39 Installed: ImageMagick-6.9.10.68-5.el7_9.x86_64
Sep 14 19:41:40 Installed: xkeyboard-config-2.24-1.el7.noarch
Sep 14 19:41:40 Installed: libxkbcommon-0.7.1-3.el7.x86_64
Sep 14 19:41:41 Installed: gtk3-3.22.30-6.el7.x86_64
Sep 14 19:41:41 Installed: 1:emacs-24.3-23.el7.x86_64
Sep 20 10:33:11 Installed: iptables-services-1.4.21-35.el7.x86_64
[root@localhost ~]#
```

CentOS Linux有一个默认的日志路径—— /var/log/ ，这里面有不少日志文件，都是系统一些服务命令专属的。我们随意看看其中两个，一个是cron，即是crontab执行计划任务的记录（虽然我没有设置任务，所以没什么记录）；另一个是yum.log，一眼看出这是安装软件时留下来的记录。

不同日志有不同的格式，但一般都是这样的：

记录日期——记录时间——哪个进程产生（CROND）——哪个用户（root）——执行的什么命令（CMD）

主要体现以下三点：

①日志是某个软件、服务、系统功能，每当发生一次事件时，就记录下的内容；

②每一行内容代表发生的一次事件记录；

③每一行内容按列来细分具体内容。

3、/var/log/messages 系统综合日志文件

```
[root@localhost ~]# tail /var/log/messages
Sep 23 16:25:11 localhost NetworkManager[813]: <warn>  [1632385511.1897] dhcp6 (enp0s3): request tim
ed out
Sep 23 16:25:11 localhost NetworkManager[813]: <info>  [1632385511.1910] dhcp6 (enp0s3): state chang
ed unknown -> timeout
Sep 23 16:25:11 localhost NetworkManager[813]: <info>  [1632385511.2059] dhcp6 (enp0s3): canceled DH
CP transaction, DHCP client pid 1145
Sep 23 16:25:11 localhost NetworkManager[813]: <info>  [1632385511.2060] dhcp6 (enp0s3): state chang
ed timeout -> done
Sep 23 16:26:47 localhost chronyd[774]: Source 193.182.111.14 replaced with 193.182.111.142
Sep 23 16:30:01 localhost systemd: Started Session 2 of user root.
Sep 23 16:39:34 localhost systemd: Starting Cleanup of Temporary Directories...
Sep 23 16:39:34 localhost systemd: Started Cleanup of Temporary Directories.
Sep 23 16:40:01 localhost systemd: Started Session 3 of user root.
Sep 23 16:50:01 localhost systemd: Started Session 4 of user root.
[root@localhost ~]# _
```

日志系统有个万金油文江，当遇到某种软件、服务等不知道去哪里找它们日志时，可以尝试在/var/log/messages里找。

4、真正的爹——journalctl

不知道小伙伴们有没有留意到，有时候一些命令执行失败了，系统会提示see "journalctl -xe" for details。这个journalctl是用来查看日志的命令，方便排查问题用的。参数 -e 代表跳到日志末尾， -x 是在日志中加入注解，这个命令最好在刚出事的时候立刻使用查看较好。

另外不止报错信息，服务的正常关闭和启动信息一样会体现在journalctl里，它记录的信息很全面。

```
-- Unit session-4.scope has finished starting up.
--
-- The start-up result is done.
Sep 23 16:50:01 localhost.localdomain CROND[2098]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Sep 23 16:53:06 localhost.localdomain polkitd[764]: Registered Authentication Agent for unix-process
Sep 23 16:53:06 localhost.localdomain sshd[1136]: Received signal 15; terminating.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopping OpenSSH server daemon...
-- Subject: Unit sshd.service has begun shutting down
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit sshd.service has begun shutting down.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
-- Subject: Unit sshd.service has finished shutting down
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit sshd.service has finished shutting down.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
-- Subject: Unit sshd.service has begun start-up
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit sshd.service has begun starting up.
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on 0.0.0.0 port 22.
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on :: port 22.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
-- Subject: Unit sshd.service has finished start-up
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit sshd.service has finished starting up.
--
-- The start-up result is done.
Sep 23 16:53:06 localhost.localdomain polkitd[764]: Unregistered Authentication Agent for unix-proce
lines 2175-2210/2210 (END)
```

例如，我使用了 systemctl restart sshd 重启sshd服务，journalctl 里就记录了详细的关闭开启信息。

说白了，systemctl能管理所有的systemd的units，那么journalctl就能查看所有units的日志。

5、journalctl的拓展用法

① -u + 某个服务： 单独查看某个服务（unit）的日志

② -f -u + 某个服务： 单独跟踪某个服务（unit）的日志

```
[root@localhost ~]# journalctl -u sshd.service
-- Logs begin at Thu 2021-09-23 16:24:05 CST, end at Thu 2021-09-23 17:01:09 CST. --
Sep 23 16:24:25 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 23 16:24:25 localhost.localdomain sshd[1136]: Server listening on 0.0.0.0 port 22.
Sep 23 16:24:25 localhost.localdomain sshd[1136]: Server listening on :: port 22.
Sep 23 16:24:25 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopping OpenSSH server daemon...
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on 0.0.0.0 port 22.
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on :: port 22.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost ~]# journalctl -f -u sshd.service
-- Logs begin at Thu 2021-09-23 16:24:05 CST. --
Sep 23 16:24:25 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 23 16:24:25 localhost.localdomain sshd[1136]: Server listening on 0.0.0.0 port 22.
Sep 23 16:24:25 localhost.localdomain sshd[1136]: Server listening on :: port 22.
Sep 23 16:24:25 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopping OpenSSH server daemon...
Sep 23 16:53:06 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on 0.0.0.0 port 22.
Sep 23 16:53:06 localhost.localdomain sshd[2188]: Server listening on :: port 22.
Sep 23 16:53:06 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
_
```

③ --disk-usage 可查看日志文件大小 与 --vacuum-size=容量(如1G) 可指定日志文件的大小： 前者就是查看文件大小，后者会一直删除旧记录直到记录所占容量符合要求。

④ --since 和 --until ： 限定时间和日期的起点和终点

```
[root@localhost ~]# journalctl --disk-usage
Archived and active journals take up 6.1M on disk.
[root@localhost ~]# journalctl --vacuum-size=1G
Vacuuming done, freed 0B of archived journals on disk.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# journalctl --since "2021-09-23 10:00:00"^C
[root@localhost ~]# journalctl --since 09:00 --until "1 hour ago"^C
[root@localhost ~]# journalctl --since yesterday^C
[root@localhost ~]# _
```

由于内容太多不好展示命令我就Ctrl+C了。

6、rsyslog——爹中爹

rsyslog日志管理系统，是journalctl这个强大的日志查询工具的爹。在CentOS里，各种自带的服务先把自己的日志信息统一交给rsyslog进行汇总，然后rsyslog为每一个服务生成单独的日志文件，最后journalctl把通过journal进程收集的所有unit的日志展示出来。可以说，rsyslog最先收集日志，journalctl进行二次加工。

rsyslog的配置文件路径是—— /etc/rsyslog.conf。我们打开来看看吧：

```
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# 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

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log


# ### begin forwarding rule ###
# 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)
[root@localhost ~]# _
```

我把最重要的最需要关注的RULES展示出来了，这里定义了rsyslog具体把哪些服务收集成日志。

```
*.info;mail.none;authpriv.none;cron.none        /var/log/messages

mail.*                                          /var/log/maillog
```

*.info是把info或更高级别的信息送到messages文件里。

mail.none是mail的日志不发到messages文件里。

mail.*是把所有mail的日志都送到maillog文件里。

同理后面的cron.*是把所有的cron计划任务的日志都送到cron文件里。

等等等等……都同理。

另外，rsyslog除了可以收集本机日志外，还可在两台服务器之间相互传递收集日志。

7、日志信息级别字段说明

①Ebug：有调试信息的，日志信息最多。

②Info：一般信息的日志，最常用。

③Notice：最具有重要性的普通条件的信息。

④Warning：警告级别。

⑤Err：错误级别，阻止某个功能或者模块不能正常工作的信息。

⑥Crit：严重级别，阻止整个系统或者整个软件不能正常工作的信息。

⑦Alert：需要立刻修改的信息。

⑧Emerg：内核崩溃等严重信息。

⑨None：什么都不记录。

- 后记：

本节看的都是系统自身或者自带的服务日志，现实环境中其实很多都是要自定义日志文件去收集某些外来软件运行时的日志，那就不能依靠journalctl这样的服务了，所有到后面我们要学会shell脚本编程来自动收集和分析那些日志。