链滴

1- 系统日志管理 -rsyslog

作者：Carey

原文链接：https://ld246.com/article/1615546379631

来源网站：链滴

1 系统日志管理

1.1 系统日志介绍

将系统和应用发生的事件记录至日志中，以助于排错和分析使用：

日志记录的内容包括：

- 历史事件：时间，地点，人物，事件
- 日志级别：事件的关键性程度，Loglevel

1.1.1 sysklogd 系统日志服务

CentOS 5 之前版本采用的日志管理系统服务

- syslogd: system application 记录应用日志
- klogd: linux kernel 记录内核日志

事件记录格式：

- 日期时间 主机 进程[pid]: 事件内容
- C/S架构：通过TCP或UDP协议的服务完成日志记录传送，将分布在不同主机的日志实现集中管理

1.1.2 rsyslog 系统日志服务

rsyslog是CentOS 6 以后版本的系统管理服务.它提供了高性能，出色的安全性和模块化设计。 尽管rsyslog最初是常规的syslogd，但已发展成为一种瑞士军刀式的记录工具，能够接受来自各种来源的输

，并将其转换，然后输出到不同的目的地。

当应用有限的处理时，RSYSLOG每秒可以将超过一百万的消息传递到本地目的地。 即使在远程的目地和更精细的处理中，性能通常也被认为是“惊人的”。

官方网站:

https://www.rsyslog.com/

rsyslog 特性

- 多线程
- UDP, TCP, SSL, TLS, RELP
- MySQL, PGSQL, Oracle实现日志存储
- 强大的过滤器，可实现过滤记录日志信息中任意部分
- 自定义输出格式
- 适用于企业级中继链

1.1.3 ELK

ELK：由Elasticsearch, Logstash, Kibana三个软件组成

- 非关系型分布式数据库
- 基于apache软件基金会jakarta项目组的项目lucene
- Elasticsearch是个开源分布式搜索引擎，可以处理大规模日志数据，比如：Nginx、Tomcat、系日志等功能
- Logstash对日志进行收集、分析，过滤，并将其存储供以后使用
- Kibana 可以提供的日志分析友好的 Web 界面

1.2 rsyslog 管理

1.2.1 系统日志术语

- facility：设施，从功能或程序上对日志进行归类

```
#服务中定义的
auth, authpriv, cron, daemon,ftp,kern, lpr, mail, news, security(auth),user, uucp, syslog
#自定义的分类
local0-local7
```

- Priority 优先级别，从低到高排序

```
debug, info, notice, warn(warning), err(error), crit(critical), alert,emerg(panic)
```

- 参看帮助： man 3 syslog，man logger

```
[root@centos8 ~]#yum -y install man-pages
[root@centos8 ~]#man 3 rsyslog
```

1.2.2 rsyslog 相关文件

- 程序包：rsyslog
- 主程序：/usr/sbin/rsyslogd
- CentOS 6：/etc/rc.d/init.d/rsyslog {start|stop|restart|status}
- CentOS 7,8：/usr/lib/systemd/system/rsyslog.service
- 配置文件：/etc/rsyslog.conf，/etc/rsyslog.d/*.conf
- 库文件： /lib64/rsyslog/*.so

1.2.3 rsyslog配置文件

/etc/rsyslog.conf 配置文件格式：由三部分组成

- MODULES：相关模块配置
- GLOBAL DIRECTIVES：全局配置
- RULES：日志记录相关的规则配置

RULES配置格式：

facility.priority; facility.priority... target

facility格式：

```
*     #所有的facility
facility1,facility2,facility3,...        #指定的facility列表
```

priority格式：

```
*: 所有级别
none：没有级别，即不记录
PRIORITY：指定级别（含）以上的所有级别
=PRIORITY：仅记录指定级别的日志信息
```

target格式：

```
文件路径：通常在/var/log/，文件路径前的-表示异步写入
用户：将日志事件通知给指定的用户，* 表示登录的所有用户
日志服务器：@host，把日志送往至指定的远程UDP日志服务器 @@host 将日志发送到远程TCP日
服务器
管道： | COMMAND，转发给其它命令处理
```

通常的日志文件的格式：

日志文件有很多，如： /var/log/messages,cron,secure等，基本格式都是类似的。格式如下

事件产生的日期时间 主机 进程(pid)：事件内容

范例：日志文件格式

```
[19:03:36 root@centos8 ~]#tail /var/log/messages
```

```
Mar  8 18:59:18 centos8 systemd[1]: run-r40e509c7b6424883bf892aaced46b127.service: Succ
eded.
Mar  8 19:03:10 centos8 systemd[1]: Starting dnf makecache...
Mar  8 19:03:10 centos8 dnf[1332]: CentOS-8 - Base - mirrors.aliyun.com           25 kB/s | 3.9
B     00:00
[19:04:03 root@centos8 ~]#tail /var/log/secure
Mar  1 14:30:18 centos8 sshd[1159]: pam_unix(sshd:session): session opened for user root by
uid=0)
Mar  8 18:53:07 centos8 sshd[932]: Server listening on 0.0.0.0 port 22.
Mar  8 18:53:07 centos8 sshd[932]: Server listening on :: port 22.
```

范例：将ssh服务的日志记录至自定义的local的日志设备

```
#修改sshd服务的配置
[19:04:05 root@centos8 ~]#vim /etc/ssh/sshd_config
#SyslogFacility AUTHPRIV
SyslogFacility local7

#修改rsyslog的配置
[19:06:35 root@centos8 ~]#vim /etc/rsyslog.d/ssh.conf
Local7.* /var/log/sshd.log
[19:07:43 root@centos8 ~]#systemctl restart rsyslog.service

#测试
[19:08:16 root@centos8 ~]#ssh 192.168.10.81
[19:07:56 root@centos8 ~]#tail /var/log/sshd.log
Mar  8 19:08:16 centos8 sshd[1491]: Accepted password for root from 192.168.10.1 port 6282
 ssh2
```

1.2.4 启用网络日志服务

启用网络日志服务功能，可以将多个远程主机的日志，发送到集中的日志服务器，方便统一管理。

功能模块：imudp，imtcp

范例：CentOS 8 启用网络日志功能

```
#接收端配置
[19:11:02 root@rsyslog ~]#vim /etc/rsyslog.conf
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

module(load="imtcp") # needs to be done just once
input(type="imtcp" port="514")
#在客户端指定将日志发送到远程的TCP、UDP的日志服务器
[19:12:02 root@web ~]#vim /etc/rsyslog.conf
#UDP协议
*.info;mail.none;authpriv.none;cron.none                @192.168.10.81:541
[19:12:20 root@mysql ~]#vim /etc/rsyslog.conf
#TCP协议
*.info;mail.none;authpriv.none;cron.none                @@192.168.10.81:514
```

范例：CentOS 7 和6 启用网络日志功能

```
vim /etc/rsyslog.conf
####MODULES####
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

1.2.5 常见日志文件

- /var/log/secure：系统安全日志，文本格式，应周期性分析
- /var/log/btmp：当前系统上，用户的失败尝试登录相关的日志信息，二进制格式，lastb命令进行看
- /var/log/wtmp：当前系统上，用户正常登录系统的相关日志信息，二进制格式，last命令可以查看
- /var/log/lastlog:每一个用户最近一次的登录信息，二进制格式，lastlog命令可以查看
- /var/log/dmesg：CentOS7 之前版本系统引导过程中的日志信息，文本格式，开机后的硬件变化不再记录，专用命令dmesg查看，可持续记录硬件变化的情况
- /var/log/boot.log 系统服务启动的相关信息，文本格式
- /var/log/messages ： 系统中大部分的信息
- /var/log/anaconda : anaconda的日志

范例：找到失败登录的IP

```
[root@centos8 ~]#awk '/Failed password/{print $(NF-3)}' /var/log/secure
192.168.39.7
192.168.39.18
192.168.39.18
```

范例：找出失败登录次数最多的前10个IP

```
[root@centos8 ~]#lastb -f btmp-test1 | awk '{print $3}'|sort | uniq -c|sort -
nr|head
8374 112.64.33.38
7041 221.125.235.4
6502 183.247.184.220
5970 203.190.163.125
5297 202.89.0.27
3062 119.163.122.32
2961 124.126.248.6
2921 92.222.1.40
2896 112.65.170.186
1955 118.97.213.118
[root@centos8 ~]#lastb -f btmp-test2 | awk '{ip[$3]++}END{for(i in ip){print
ip[i],i}}'|sort -nr|head
86294 58.218.92.37
43148 58.218.92.26
18036 112.85.42.201
10501 111.26.195.101
10501 111.231.235.49
```

```
10501 111.204.186.207
10501 111.11.29.199
10499 118.26.23.225
6288 42.7.26.142
4236 58.218.92.30
```

1.3 日志管理工具 journalctl

CentOS 7 以后版，利用Systemd 统一管理所有 Unit 的启动日志。带来的好处就是，可以只用journactl一个命令，查看所有日志（内核日志和应用日志）。

日志的配置文件：

/etc/systemd/journald.conf

journalctl命令格式

journalctl [OPTIONS...] [MATCHES...]

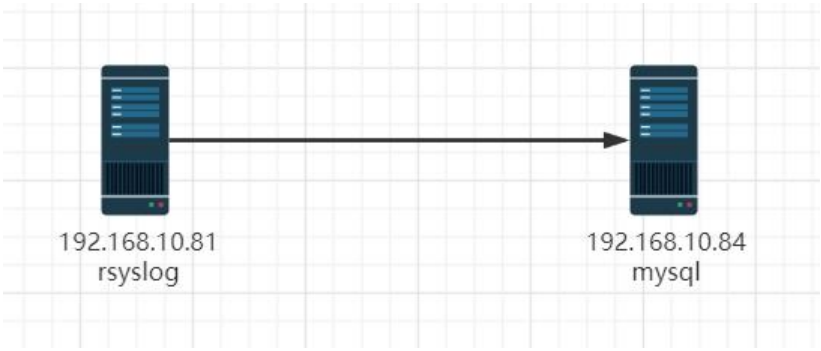
范例：journalctl用法

```
#查看所有日志（默认情况下，只保存本次启动的日志）
journalctl
#查看内核日志（不显示应用日志）
journalctl -k
#查看系统本次启动的日志
journalctl -b
journalctl -b -0
#查看上一次启动的日志（需更改设置）
journalctl -b -1
#查看指定时间的日志
journalctl --since="2017-10-30 18:10:30"
journalctl --since "20 min ago"
journalctl --since yesterday
journalctl --since "2017-01-10" --until "2017-01-11 03:00"
journalctl --since 09:00 --until "1 hour ago"
#显示尾部的最新10行日志
journalctl -n
#显示尾部指定行数的日志
journalctl -n 20
#实时滚动显示最新日志
journalctl -f
#查看指定服务的日志
journalctl /usr/lib/systemd/systemd
#查看指定进程的日志
journalctl _PID=1
#查看某个路径的脚本的日志
journalctl /usr/bin/bash
#查看指定用户的日志
journalctl _UID=33 --since today
#查看某个 Unit 的日志
journalctl -u nginx.service
journalctl -u nginx.service --since today
#实时滚动显示某个 Unit 的最新日志
```

```
journalctl -u nginx.service -f
#合并显示多个 Unit 的日志
journalctl -u nginx.service -u php-fpm.service --since today
#查看指定优先级（及其以上级别）的日志，共有8级
0: emerg
1: alert
2: crit
3: err
4: warning
5: notice
6: info
7: debug
journalctl -p err -b
#日志默认分页输出，--no-pager 改为正常的标准输出
journalctl --no-pager
#日志管理journalctl
#以 JSON 格式（单行）输出
journalctl -b -u nginx.service -o json
#以 JSON 格式（多行）输出，可读性更好
journalctl -b -u nginx.serviceqq -o json-pretty
#显示日志占据的硬盘空间
journalctl --disk-usage
#指定日志文件占据的最大空间
journalctl --vacuum-size=1G
#指定日志文件保存多久
journalctl --vacuum-time=1years
```

2 实战案例

2.1 实战案例1：利用mysql存储日志信息



2.1.1 目标

利用rsyslog日志服务，将收集的日志记录于MySQL中

2.1.2 环境准备

两台主机
一台：rsyslog日志服务器，IP：192.168.10.81
一台：mysql数据库服务器，IP：192.168.10.84

2.1.3 实现步骤

2.1.3.1 在rsyslog服务器上安装连接mysql模块相关的程序包

```
[19:46:29 root@rsyslog ~]#yum install -y rsyslog-mysql
[19:33:21 root@rsyslog ~]#rpm -ql rsyslog-mysql
/usr/lib/.build-id
/usr/lib/.build-id/b1
/usr/lib/.build-id/b1/435a976b2dfddfb19d0d1517964f615d510402
/usr/lib64/rsyslog/ommysql.so
/usr/share/doc/rsyslog/mysql-createDB.sql
#查看sql脚本文件内容
[19:33:29 root@rsyslog ~]#cat /usr/share/doc/rsyslog/mysql-createDB.sql
#将sql脚本复制到数据库服库上
[19:33:48 root@rsyslog ~]#scp /usr/share/doc/rsyslog/mysql-createDB.sql 192.168.10.84:
```

2.1.3.2 准备MySQL Server

```
[19:35:26 root@mysql ~]#yum install -y mysql-server
#在mysql数据库服务器上创建相关数据库和表，并授权rsyslog能连接至当前服务器
[19:37:05 root@mysql ~]#systemctl enable --now mysqld.service
[19:37:22 root@mysql ~]#mysql  create user syslog@'192.168.10.%' identified by '123456';
mysql> grant all on Syslog.* to 'syslog'@'192.168.10.%';
```

2.1.3.3 配置日志服务器将日志发送至指定数据库

```
#配置rsyslog将日志保存到mysql中
[19:23:32 root@rsyslog ~]#vim /etc/rsyslog.conf
#在 MODULES 语言下面，如果是 CentOS 8 加下面行
module(load="ommysql")
#在 MODULES 语言下面，如果是 CentOS 7，6 加下面行
$ModLoad ommysql

#在RULES语句块加下面行的格式
#facility.priority   :ommysql:DBHOST,DBNAME,DBUSER, PASSWORD
*.info              :ommysql:192.168.10.84,Syslog,syslog,123456
[19:42:50 root@rsyslog ~]#systemctl restart rsyslog.service
```

2.1.3.4 测试

```
#在日志服务器上生成日志
[19:54:42 root@rsyslog ~]#logger "zhangzhuo"

#在数据库上查询到上面的测试日志
mysql> select * from SystemEvents\G
```

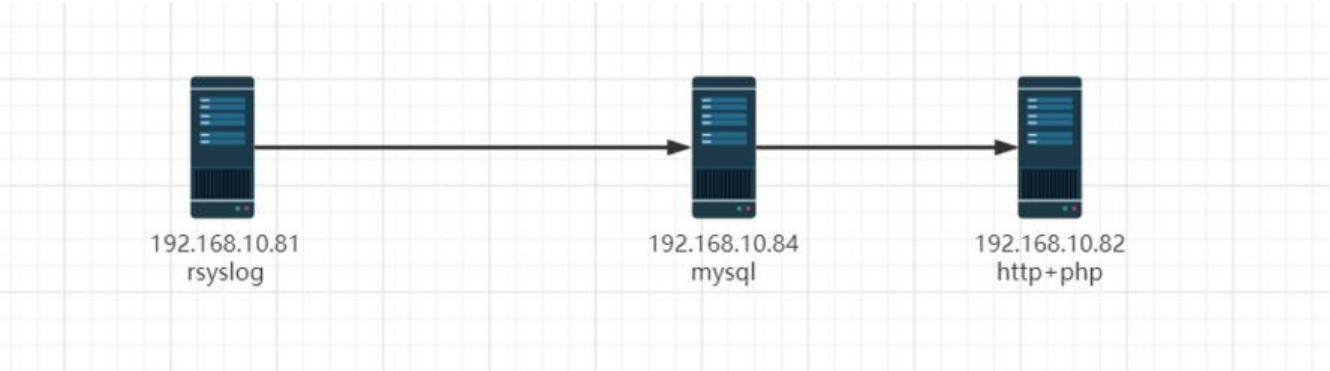
2.2 实战案例2：通过 loganalyzer 展示数据库中的日志

loganalyzer是用 php 语言实现的日志管理系统，可将MySQL数据库的日志用丰富的WEB方式进行示

官网：https://loganalyzer.adiscon.com

2.2.1 目标

通过 loganalyzer 展示数据库中的日志



2.2.2 环境准备

三台主机

- 一台日志服务器，利用上一个案例实现，IP：192.168.10.81
- 一台数据库服务器，利用上一个案例实现，IP：192.168.10.84
- 一台当httpd+php 服务器，并安装loganalyzer展示web图形，IP：192.168.10.82

2.2.3 步骤

2.2.3.1 安装 php和相关软件包

在192.168.10.82主机上安装php和相关软件包

```
[20:04:33 root@web ~]#yum -y install httpd php-fpm php-mysqlnd php-gd
[20:04:42 root@web ~]#systemctl restart httpd php-fpm
```

2.2.3.2 安装 LogAnalyzer

在192.168.10.82主机上安装LogAnalyzer

```
#从http://loganalyzer.adiscon.com/downloads/ 下载loganalyzer-4.1.10.tar.gz
[20:07:48 root@web ~]#tar xvf loganalyzer-4.1.11.tar.gz
[20:12:23 root@web ~]#mv loganalyzer-4.1.11/src /var/www/html/log
[20:09:40 root@web ~]#touch /var/www/html/log/config.php
[20:10:15 root@web ~]#chmod 666 /var/www/html/log/config.php
```

2.2.3.3 基于 web 页面初始化

访问http://192.168.10.82/log 实现初始化

选择：MySQL Native, Syslog Fields, Monitorware

2.2.3.4 安全加强

[20:20:20 root@web ~]#chmod 644 /var/www/html/log/config.php

3 logrotate日志转储

3.1 logrotate 介绍

logrotate 程序是一个日志文件管理工具。用来把旧的日志文件删除，并创建新的日志文件，称为日转储或滚动。可以根据日志文件的大小，也可以根据其天数来转储，这个过程一般通过 cron 程序来行

3.2 logrotate 配置

软件包：logrotate

相关文件

- 计划任务：/etc/cron.daily/logrotate
- 程序文件：/usr/sbin/logrotate
- 配置文件： /etc/logrotate.conf
- 日志文件：/var/lib/logrotate/logrotate.status

配置文件主要参数如下：

配置参数	说明
compress	通过gzip压缩转储以后的日志
nocompress	不压缩
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，使用指定的权限，所有者，所属组创建新的日志文件
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 address	专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，此为默认选项
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，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，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
tabooext [+] 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，bytes(缺省)及KB或MB
sharedscripts	默认，对每个转储日志运行prerotate和postrotate脚本，日志文件的绝对路径作为第一个参数传递给脚本。这意味着单个脚本可以针对与多个文件匹配的日志文件条目多次运行（例如/ var / log / news / *.example）。如果指定此项sharedscripts，则无论有多少个日志与通配符模式匹配，脚本都只会运行一次
nosharedscripts	针对每一个转储的日志文件，都执行一次prerotate 和 postrotate 脚本，此为默认值

配置参数	说明
missingok	如果日志不存在，不提示错误，继续处理下一个
nomissingok	如果日志不存在，提示错误，此为默认值

3.3 logroate 配置范例

范例： 设置nginx的日志转储

```
cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
daily
rotate 100
missingok
compress
delaycompress
notifempty
create 644 ngnix nginx
postrotate
if [ -f /app/nginx/logs/nginx.pid ]; then
kill -USR1 `cat /app/nginx/logs/nginx.pid`
```

```
fi
endscript
}
```

范例：对指定日志手动执行日志转储

```
#生成测试日志
[20:27:39 root@rsyslog ~]#dd if=/dev/zero of=/var/log/test1.log bs=2M count=1
1+0 records in
1+0 records out
2097152 bytes (2.1 MB, 2.0 MiB) copied, 0.00150028 s, 1.4 GB/s
[20:28:42 root@rsyslog ~]#dd if=/dev/zero of=/var/log/test2.log bs=2M count=1
1+0 records in
1+0 records out
2097152 bytes (2.1 MB, 2.0 MiB) copied, 0.0018783 s, 1.1 GB/s
#针对不同的日志创建转储配置文件
[20:32:20 root@rsyslog ~]#cat /etc/logrotate.d/test1
/var/log/test1.log {
    daily          每天执行
    rotate 5       只保存5个
    compress       压缩转储的日志
    delaycompress 下次压缩
    missingok      日志不存在不提示错误
    size 1M        大小大于1M才转储
    notifempty     空文件不转储
    create 640 bin nobody   转储后创建新文件
    postrotate      执行完毕后执行的脚本
    echo `date +%F_%T` >>/var/log/test1.log
    endscript
}

/var/log/test2.log {
  daily
  rotate 5
  compress
  delaycompress
  missingok
  size 1M
  notifempty
  create 644 root root
  postrotate
echo `date +%F_%T` >> /var/log/test2.log
  endscript
}
#针对一个测试日志，手动执行日志转储
[20:35:44 root@rsyslog ~]#logrotate /etc/logrotate.d/test1
[20:36:37 root@rsyslog ~]#ll /var/log/test*
-rw-r----- 1 bin  nobody       0 Mar  8 20:36 /var/log/test1.log
-rw-r--r-- 1 root root   2097152 Mar  8 20:28 /var/log/test1.log.1
-rw-r--r-- 1 root root   2097152 Mar  8 20:29 /var/log/test2.log
#对所有日志进行手动转储
[root@centos8 ~]#logrotate   /etc/logrotate.conf
[root@centos8 ~]#ll /var/log/test*
-rw-r--r-- 1 bin nobody       0 Nov 12 14:00 /var/log/test1.log
```

```
-rw-r--r-- 1 root root 2097152 Nov 12 13:59 /var/log/test1.log.1
-rw-r--r-- 1 root root       0 Nov 12 14:01 /var/log/test2.log
-rw-r--r-- 1 root root 2097152 Nov 12 13:59 /var/log/test2.log-20191112
[root@centos8 ~]#ls /data
test1.log test2.log
[root@centos8 ~]#cat /data/test1.log
2019-11-12_14:01:51
#对所有日志进行手动转储
[20:37:55 root@rsyslog ~]#logrotate /etc/logrotate.conf
[20:38:52 root@rsyslog ~]#ll /var/log/test*
-rw-r----- 1 bin  nobody       0 Mar  8 20:36 /var/log/test1.log
-rw-r--r-- 1 root root   2097152 Mar  8 20:28 /var/log/test1.log.1
-rw-r--r-- 1 root root         0 Mar  8 20:38 /var/log/test2.log
-rw-r--r-- 1 root root   2097152 Mar  8 20:29 /var/log/test2.log-20210308
```