Rsyslog系统日志管理

课程目标

- 了解日志的级别及作用
- 掌握rsyslog服务的本地日志和远程日志的管理；
- 能够使用logrotate程序根据需求对日志进行轮转

一、常见的系统日志

```
日志格式：文本日志/二进制日志/数据库日志
默认的相关日志文件：
/var/log/boot.log       系统引导日志，记录开机启动信息
/var/log/dmesg          核心的启动日志
/var/log/messages       系统的日志文件
/var/log/maillog        邮件服务的日志
/var/log/xferlog        ftp服务的日志
/var/log/secure         网络连接及系统登录的安全信息
/var/log/cron           定时任务的日志
/var/log/wtmp           记录所有的登入和登出     last -f 查看
/var/log/btmp           记录失败的登入尝试
```

二、日志管理程序

- 在RHEL6中，syslogd已经被rsyslog取代。它可以将日志写入数据库，并可以利用模块和插件控制输入输出。
- rsyslog程序管理本地和远程日志
  - 安装软件
  - 根据需求修改配置文件
  - 启动服务
  - 测试验证

三、日志级别

```
# man syslog
日志信息分为以下级别，从上到下级别依次降低
   none       <-- none 不算是一个等级，它表示不记录服务的所有信息
0  emerg      <-- 系统不可用
1  alert      <-- 特别留意的报警信息
2  crit       <-- 非常严重的状况
3  err        <-- 错误信息
4  warning    <-- 警告信息
5  notice     <-- 稍微需要注意的信息
6  info       <-- 正常信息
7  debug      <-- 调试信息，开发人员使用
```

四、日志配置

1. 日志定义相关符号

```
配置文件中常见的表示符号
.          <-- 用来分隔服务和级别       mail.info
*          <-- 任何服务，或者任何级别     *.info    mail.*
=          <-- 有等号表示等于某一级别，没有等号表示大于或者等于某一级别   mail.=info
!          <-- 排除操作，前面有相同服务的表达式，这个操作才有意义
               代表从前面表达式所包含的内容中排除某些内容
;          <-- 用于分隔不同的   服务.级别 组合      cron.=info;mail.info
,          <-- 用于分隔不同的服务              cron,mail.=info
-          <-- 用于指定目标文件时，代表异步写入   -/var/log/maillog
```

举例说明：

```
mail.=err
mail.err
cron.err;mail.=info
cron.info;cron.!=err
012456
cron.info;cron.!err
456
```

2. 了解配置文件

```
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
所有服务产生的日志，除了mail/验证/任务计划相关日志都记录/var/log/message
authpriv.*                                              /var/log/secure
记录所有跟验证有关日志
mail.*                                                  -/var/log/maillog
记录所有跟邮件有关的日志信息
cron.*                                                  /var/log/cron
记录跟任务计划查关的日志
*.emerg                                                 *
把所有级别为emerg的信息发送给所有登录到系统上的用户
uucp,news.crit                                          /var/log/spooler
local7.*                                                /var/log/boot.log
记录所有跟启动相关的日志信息
```

特别说明：

```
说明：
# man rsyslog.conf
The  facility  is one of the following keywords: auth, authpriv,
       cron, daemon, kern, lpr, mail, mark,  news,  security  (same  as
       auth), syslog, user, uucp and local0 through local7.

log facility  设备 设施：用来记录一种类型日志的日志设备

daemon
```

```
auth
authpriv
user
mail
lpr
news
uucp
ftp
local0-7
```

五、本地日志管理

需求1：

将本地邮件服务的日志记录到/var/log/test_mail.log里

思路：

1. 通过修改配置文件完成
2. 重启rsyslog服务
3. 测试验证

步骤：

```
1. 修改配置文件
vim rsyslog.conf
#mail.*                                                  -/var/log/maillog
mail.info                                               /var/log/test_mail.log

2. 启动服务
[root@log-server log]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
[root@log-server log]# ls test_mail.log
test_mail.log

3. 测试验证
   1) 发送一封邮件
[root@MissHou ~]# echo I love you | mail stu1

   2) 发个log消息测试
   logger用于往系统中写入日志，他提供一个shell命令接口到rsyslog系统模块

# logger -t "Loggertest" -p mail.info "Testing log info"
-t        指定标记记录
-p        指定输入消息的优先级，优先级可以是数字或者指定为 " facility.level" 的格式。
-i        逐行记录每一次logger的进程ID
```

需求2：

把ssh服务的日志指定记录到/var/log/ssh下

思路：

1. 单独指定ssh服务的日志载体（ssh服务需要配置——>local0）
2. rsyslog程序将来自于local0设备载体上的日志单独记录（通过rsyslog.conf配置）
3. 重启相关的服务
4. 测试验证

步骤：

```
1. 指定ssh服务的日志设备载体为local0
vim /etc/ssh/sshd_config
#SyslogFacility AUTHPRIV
SyslogFacility LOCAL0

2. 重启sshd服务

3. 修改rsyslog.conf配置文件，来管理ssh服务的日志
[root@log-server ~]# vim /etc/rsyslog.conf
local0.*                                                /var/log/ssh.log

4. 重启rsyslog服务
```

课堂练习：

搭建FTP服务，并且将ftp服务的上传下载日志单独记录到/var/log/ftp.log里

```
方法1：
vim /etc/vsftpd/vsftpd.conf
...
syslog_enable=YES     表示日志通过rsyslog程序管理

vim /etc/rsyslog.conf
...
ftp.*    /var/log/ftp.log

方法2：
vim /etc/xinetd.d/vsftpd
...
log_type = SYSLOG ftp


vim /etc/rsyslog.conf
...
ftp.*    /var/log/ftp.log
```

六、远程日志管理

目的：把多台服务器的日志远程记录到其中一台日志服务器集中化管理，方便对其统一分析和管理

需求：

将ssh应用服务器的ssh服务的日志远程记录到日志管理服务器上

环境：

log-server:10.1.1.1

ssh-server：10.1.1.3

client:10.1.1.2

思路：

1. 在ssh-server上将ssh服务的日志单独记录
2. 在ssh-server上通过修改rsyslog.conf文件来将本地ssh服务日志远程传送到log-server服务器上
3. 在log-server上打开514端口等待客户来传送日志
4. 测试验证

步骤：

```
以下内容在ssh-server完成（10.1.1.3）
1. 将ssh服务日志单独记录
#SyslogFacility AUTHPRIV
SyslogFacility LOCAL0

2. 重启ssh服务
[root@ssh-server ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]

3. 将local0设备载体上的日志远程发送到log-server日志管理服务器上
vim /etc/rsyslog.conf
...
local0.*        @@10.1.1.1:514
4. 重启rsyslog服务
[root@ssh-server ~]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]

以下操作在log-server上完成（10.1.1.1）：
1. 修改rsyslog.conf配置文件，加载相应的模块，并且打开514端口

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

2. 重启服务
[root@log-server ~]# service rsyslog restart
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]

3. 检查端口是否监听
[root@log-server ~]# netstat -nltp|grep 514
tcp        0      0 0.0.0.0:514                 0.0.0.0:*                   LISTEN
 31512/rsyslogd
```

```
tcp        0      0 :::514                      :::*                        LISTEN
 31512/rsyslogd


在客户端10.1.1.2上测试验证:
[root@client ~]# ssh 10.1.1.3
root@10.1.1.3's password:
Last login: Thu Jul 12 11:51:36 2018 from 10.1.1.2
[root@ssh-server ~]# exit
logout
Connection to 10.1.1.3 closed.

log-server查看:
[root@log-server ~]# tail -f /var/log/messages
Jul 12 11:54:47 ssh-server sshd[3578]: Accepted password for root from 10.1.1.2 port 36305 ssh2
Jul 12 12:00:16 ssh-server sshd[3578]: Received disconnect from 10.1.1.2: 11: disconnected by
user
```

思考:

如果日志服务器管理多台服务器，那么如何区分不同的客户端?

1. 这件事情需要再日志管理服务器端完成
2. 将接收过来的日志单独管理保存称为本地日志管理
3. 可以通过定义模板的方式完成

```
解决办法:
可以通过定义模板来保存不同的日志文件:
[root@log-server log]# vim /etc/rsyslog.conf

在该文件的最后面加入以下内容:
//定义一个模板DynFile，将日志保存在/var/log里，文件名为system-客户端的主机名.log
$template DynFile,"/var/log/system-%HOSTNAME%.log"
//动态加载调用上面的模板
local0.* ?DynFile
```

课堂练习:

将FTP服务的日志远程保存到log-server日志管理服务器上。

七、日志轮转

日志轮循（轮转）：日志轮转，切割，备份，归档

- 为什么要日志轮转?

1、避免日志过大占满/var/log的文件系统

2、方便日志查看

3、 将丢弃系统中最旧的日志文件，以节省空间

4、日志轮转的程序是logrotate

5、 logrotate本身不是系统守护进程，它是通过计划任务crond每天执行

- 如何进行日志轮转?
  - 了解相关配置文件

```
[root@MissHou ~]# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly
#以7天为一个周期(每周轮转)
rotate 4
#每4周备份日志文件（保留4份日志文件）
create
#当老的转储文件被归档后,创建一个新的空的转储文件重新记录,权限和原来的转储文件权限一样.
dateext
#用日期来做轮转之后的文件的后缀名
#compress
#指定不压缩转储文件,如需压缩去掉注释就可以了.通过gzip压缩
include /etc/logrotate.d #加载外部目录
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly     表示此文件是每月轮转，而不会用到上面的每周轮转
    create 0664 root utmp    轮转之后创建新文件，权限是0664，属于root用户和utmp组
   minsize 1M      文件大于1M，而且周期到了，才会轮转
    rotate 1        保留1分日志文件，每1个月备份一次日志文件
}

/var/log/btmp {
    missingok               如果日志文件不存在，不报错
    monthly
    create 0600 root utmp
    rotate 1
}

[root@MissHou ~]# cat /etc/logrotate.d/syslog
//这个子配置文件，没有指定的参数都会以默认方式轮转

/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    sharedscripts       不管有多少个文件待轮转，prerotate 和 postrotate 代码只执行一次
    postrotate           轮转完后执行postrotate 和 endscript 之间的shell代码
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true  这一句话表示
轮转后对rsyslog的pid进行刷新（但pid其实不变）
    endscript
```

```
}
```

思考：
为什么轮转后需要对rsyslog的pid进行刷新呢？

需求1：ssh服务的日志单独保存到了/var/log/ssh里，如何进行轮转？

要求：

1. 每天进行轮转，保留5天的日志文件
2. 日志文件大小大于5M进行轮转

思路：

1. 修改logrotate程序的主配置文件增加/var/log/ssh文件的轮转

或者

2. 在/etc/logrotate.d/目录创建一个文件

步骤：

方法1：修改主配置文件

```
vim /etc/logrotate.conf
...
/var/log/ssh {
  daily
  rotate 5
  size      5M
  create
}
```

测试验证：

方法2：创建子配置文件

```
[root@log-server logrotate.d]# pwd
/etc/logrotate.d
[root@log-server logrotate.d]# vim ssh
/var/log/ssh {
  missingok
  daily
  rotate        5
  size          5M
  create
  nodateext    //不以日期作为后缀
}
```

测试验证：

讨论为什么轮转后需要对rsyslog的pid进行刷新呢?

```
sharedscripts
   prerotate
      轮转前执行脚本
   endscript

sharedscripts
   postrotate
      轮转后执行脚本
   endscript
```

为什么轮转后要做上面的kill -HUP? HUP是一个信号，这个信号的默认操作为终止进程

小实验:
准备环境:
不以时间作为后缀去轮转

```
# logger -t "哈哈" "你好"
# tail -3 /var/log/messages   --默认这条信息在当前的messages里
```

注释掉以日期为后缀测试:

修改子配置文件 /etc/logrotate.d/syslog

```
{
    sharedscripts
    postrotate
        logger -t "呵呵" "再见! "
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
    endscript
}
```

强制轮转

```
# logrotate -f /etc/logrotate.conf
```

请问这条信息在轮转后在哪个文件里?

```
messages.1
```

结论: 先写日志再刷新PID; 刷新之前写的日志文件是老的日志文件。
再次编辑子配置文件

```
{
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
        logger -t "呵呵" "我又回来啦! "
    endscript
}
```

强制轮转

```
# logrotate -f /etc/logrotate.conf
```

请问这条信息在轮转后在哪个文件里?

```
messages
```

结论: 先刷新PID再写日志; 日志就会写到新的日志文件里。

常见的一些参数：

```
常用的指令解释，这些指令都可以在man logrotate 中找得到。
daily                指定转储周期为每天
monthly              指定转储周期为每月
weekly                   <-- 每周轮转一次(monthly)
rotate 4                 <-- 同一个文件最多轮转4次，4次之后就删除该文件
create 0664 root utmp    <-- 轮转之后创建新文件，权限是0664，属于root用户和utmp组
dateext                  <-- 用日期来做轮转之后的文件的后缀名
compress                 <-- 用gzip对轮转后的日志进行压缩
minsize 30K              <-- 文件大于30K，而且周期到了，才会轮转
size 30k                 <-- 文件必须大于30K才会轮转，而且文件只要大于30K就会轮转
                             不管周期是否已到
missingok                <-- 如果日志文件不存在，不报错
notifempty               <-- 如果日志文件是空的，不轮转
delaycompress            <-- 下一次轮转的时候才压缩
sharedscripts            <-- 不管有多少个文件待轮转，prerotate 和 postrotate 代码只执行一次
prerotate                <-- 如果符合轮转的条件
                             则在轮转之前执行prerotate和endscript 之间的shell代码
postrotate               <-- 轮转完后执行postrotate 和 endscript 之间的shell代码
```

作业

作业1：将authpriv设备日志记录到/var/log/auth.log

```
vim /etc/rsyslog.conf
...
#authpriv.*                                      /var/log/secure
authpriv.*                                      /var/log/auth.log
```

作业2：改变应用程序sshd的日志设备为local5, 并定义local5设备日志记录到/var/log/local5.local

```
1. vim /etc/ssh/sshd_config
#SyslogFacility AUTHPRIV
SyslogFacility LOCAL5

2. 重启sshd服务
service sshd restart

3. vim /etc/rsyslog.conf
local5.*    /var/log/local5.local

4. service rsyslog restart
```

作业3: 要求如下：

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

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

2. /var/log/test 日志轮询方式为:

1> 每天轮询一次; 2> 保留 4 个文件; 3> 以时间命名; 4> 创建与原日志同名的新文件。

```
方法1:
vim /etc/logrotate.conf
...

/var/log/test {
   daily
   rotate 4
   dateext
   create
}
方法2:
vim /etc/logrotate.d/test
/var/log/test {
   daily
   rotate 4
   dateext
   create
}
```

课堂作业1：

将远程接收过来的ftp的日志按照以下要求进行轮转：

1. 每周轮转1次，保留1个月的日志文件
2. 日志大小超过100M并且到了轮转周期再进行轮转
3. 以时间作为后缀轮转老的文件

课堂作业:2： 备份etc目录，要求：

1. 每天4:00备份/etc目录到/var/back
2. 将备份命令写在脚本中，如/root/back.sh，加执行权限
3. 每天备份的文件名包含当天的日期，如2016-11-09_etc.tar.gz

4. 计划任务执行时，屏幕不产生任何输出
5. 只保留最近5天的备份

第一种版本

```
tar -czf /var/back/`date +%F`_etc.tar.gz /etc
find /var/back -mtime +5 -exec rm -rf {} \;
```

第二种版本

```
#!/bin/bash
filename=`date +%F`_etc.tar.gz
back_dir=/var/back

# 判断备份文件存放目录是否存在
if [ ! -d $back_dir ];then
        mkdir -p $back_dir
fi

#备份
tar -czf ${back_dir}/$filename /etc &>/dev/null

#删除修改时间超过5天的文件
find ${back_dir} -mtime +5 |xargs rm -rf
```

2、手动测试脚本

3、配置cron执行脚本

4、测试cron

课后作业：

1. 将ftp服务的日志保存在/var/log/ftp.log（至少2种方法）
2. 按照以下要求轮转/var/log/ftp/ftp.log日志文件
   1> 每个月轮转一次
   2> 保留1个月的日志
   3> 文件大小超过10M并且到了轮转周期再轮转