# 系统中常见的日志

|  |  |  |
| --- | --- | --- |
| **日志文件** | **查看命令** | **说明** |
| /var/log/cron |  | 定时任务相关日志 |
| /var/log/cups |  | 记录打印信息日志 |
| /var/log/dmesg | dmesg、vi | 记录了系统在开机时内核自检的信息，也可用dmesg命令直接查看内核自检信息 |
| /var/log/btmp | lastb | 记录错误登录日志，二进制文件不能用vi查看 |
| /var/log/lastlog | lastlog | 记录系统所以用户最后一次登录时间日志，二进制文件 |
| /var/log/mailo |  | 记录邮件信息 |
| /var/log/message |  | 记录系统重要信息的日志，记录linux系统绝大多数重要信息，如果系统出现问题，首要检查 |
| /var/log/secure |  | 记录验证和授权方面信息，只要涉及账户和密码的程序都会记录，例如ssh、su、sudo，甚至添加用户，和修改用户密码都会记录在这个日志文件中 |
| /var/log/wtmp | last | 永久记录所有用户的登录、注销信息，同时记录系统的启动、重启、关机事件，二进制文件 |
| /var/log/utmp | w、who、users | 记录当前已经登录的用户信息，这个文件会伴随着用户的登录和注销不断变化，只记录当前用户登录信息，二进制文件 |
| /var/log/ |  |  |

* 养成备份日志习惯，备份半年之内日志
* 有些日志是二进制文件， 需要专用命令查看，二进制文件为防止篡改，vi打开是乱码，无法修改查看，只能使用命令查看，保证其安全性
* 除系统默认日志外，采用RPM方式安装的系统服务也会默认把日志放在/var/log 不过这些日志不是由rsyslogd服务记录和管理而是各个服务使用自己的日志管理文档来记录，例如/var/log/httpd 、/var/log/samba
* 源码包安装的服务日志是在源码包指定的目录中

# 日志服务rsyslogd

## **1日志文件格式：**

只要是由日志服务rsyslogd记录的日志文件，他们的格式是一样的。基本日志格式包含以下四列：

* 事件产生的时间；
* 发生事件的服务器主机名；
* 产生事件的服务名或程序名；
* 事件的具体信息；

## 2 rsyslogd服务的配置文件

/etc/rsyslog.conf

例：authpriv.\* /var/log/secure --->服务名称[链接符号]日志等级 日志记录位置

解释：authpriv（认证相关服务）.\*（所有日志等级） 记录在/var/log/secure日志中

服务名称为：authpriv 连接符号为：. 日志等级为：\* 日志记录为：/var/log/secure

**服务名称：**rsyslogd服务可识别服务日志，也可理解为以下这些服务委托rsyslogd服务来代为管理日志。

|  |  |
| --- | --- |
| **常见服务名称** | **说明** |
| auth(LOG\_AUTH) | 安全和认证相关消息（不推荐使用authpriv替代） |
| authpriv(LOG\_AUTHPRIV) | 安全和认证相关消息（私有的） |
| cron(LOG\_CRON) | 系统定时任务cront和at产生的日志 |
| daemon(LOG\_DAEMON) | 和各个守护进程相关的日志 |
| ftp(LOG\_FTP) | ftp守护进程产生的日志 |
| kern(LOG\_KERN) | 内核产生的日志（不是用户进程产生的） |
| local0-local7(LOG\_LOCAL0-7) | 本地使用预留的服务 |
| lpr(LOG\_LPR) | 打印产生的日志 |
| mail(LOG\_MAIL) | 邮件收发信息 |
| news(LOG\_NEWS) | 与新闻服务器相关的日志 |
| syslog(LOG\_SYSLOG) | 有syslogd服务产生的日志信息（虽然服务名称已经改为rsyslogd,但是很多都还是沿用了syslogd的，这里并没有修改服务名） |
| user(LOG\_USER) | 用户等级类别的日志信息,用户进程 |
| uucp(LOG\_UUCP) | uucp子系统的日志信息，UUCP是早期linux系统进行数据传递的协议，后来也常用在新闻组服务中。 |
| ntp | 网络时间协议产生的消息 |
| console | 针对系统控制台的消息 |

**连接符号：**

例：authpriv.\* /var/log/secure --->服务名称[链接符号]日志等级 日志记录位置

连接符号为：“.”

* “.”代表只要比后面的等级高的（包含该等级）日志都记录下来，比如：“cron.info”代表cron服务产生的日志，只要日志等级大于等于info级别，就记录下来
* “.=”代表只记录所需等级的日志，其它等级的都不记录，比如：“\*.=emerg”代表任何日志服务产生的日志，只要等级是emerg登记就记录，这种方法少见，了解即可
* “.！”代表不等于，也就是除了该等级日志外，其它等级的日志都记录

**日志等级：**

|  |  |
| --- | --- |
| **日志等级** | **说明** |
| debug(LOG\_DEBUG) | 一般的调试信息说明，等级最低，一丁点操作都可能被记录 |
| info(LOG\_INFO) | 基本的通知信息 |
| notice(LOG\_NOTICE) | 普通信息，但是有一定的重要性 |
| warning(LOG\_WARNING) | 警告信息，但是还不会影响到服务或系统运行 |
| err(LOG\_ERR) | 错误信息，一般达到err等级的信息已经会影响到服务或者系统运行 |
| crit(LOG\_CRIT) | 临界状况信息，比err等级还要严重 |
| alert(LOG\_ALERT) | 警告状态信息，比crit等级还严重，必须立即采取行动 |
| emerg(LOG\_EMERG) | 疼痛等级信息，系统已经无法使用了，级别最高 |
| \* | 代表所有日志等级，比如“authpriv.\*”代表authpriv认证信息服务产生的日志，所有日志等级都记录 |
| None | 忽略这个日志服务，该服务所有日志都不再记录 |

# rsyslog configuration file

# For more information see /usr/share/doc/rsyslog-\*/rsyslog\_conf.html

# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### MODULES ####

# The imjournal module bellow is now used as a message source instead of imuxsock.

$ModLoad imuxsock # provides support for local system logging (e.g. via logger command)

#zdd# 加载imuxsock模块，为本地系统登录提供支持

$ModLoad imjournal # provides access to the systemd journal

#$ModLoad imklog # reads kernel messages (the same are read from journald)

#zdd# 加载imklog模块，为内核登录提供支持

#$ModLoad immark # provides --MARK-- message capability

#zdd# 加载immark模块，提供标记信息的能力

#zdd# 如果要将日志转发给日志服务器，要在此开启网络服务

# Provides UDP syslog reception

#$ModLoad imudp

#$UDPServerRun 514

#zdd# 加载UDP模块，允许使用UDP的514端口接受采用UDP协议转发的日志

# Provides TCP syslog reception

#$ModLoad imtcp

#$InputTCPServerRun 514

#zdd# 加载TCP模块，允许使用TCP的514端口接受采用TCP协议转发的日志

#zdd# 全局配置

#### GLOBAL DIRECTIVES ####

# Where to place auxiliary files

$WorkDirectory /var/lib/rsyslog

# Use default timestamp format

$ActionFileDefaultTemplate RSYSLOG\_TraditionalFileFormat

#zdd# 定义时间格式，默认使用的时间戳模式

# File syncing capability is disabled by default. This feature is usually not required,

# not useful and an extreme performance hit

#$ActionFileEnableSync on

#zdd# 文件同步功能，默认没有开启，是注释的。

# Include all config files in /etc/rsyslog.d/

$IncludeConfig /etc/rsyslog.d/\*.conf

#zdd# 包含/etc/rsyslog.d/目录中所有的.conf自配置文件，这个目录中的所有子配置文件也同时生效

# Turn off message reception via local log socket;

# local messages are retrieved through imjournal now.

$OmitLocalLogging on

# File to store the position in the journal

$IMJournalStateFile imjournal.state

#### 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

#zdd# mail、authpriv、cron除外所有的服务日志级别高于info的都将记录在messages文件中，

# The authpriv file has restricted access.

authpriv.\* /var/log/secure

#zdd# authpriv所有等级日志都将被记录在/var/log/secure

# Log all the mail messages in one place.

mail.\* -/var/log/maillog

#zdd# 将mail服务所有级别日志 -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

# Everybody gets emergency messages

\*.emerg :omusrmsg:\*

#zdd# 所有服务的emerg级别发送给指定用户，\*代表所有用户

# Save news errors of level crit and higher in a special file.

uucp,news.crit /var/log/spooler

#zdd# 将uucp与news服务高于crit级别的日志记录在/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)

#

# An on-disk queue is created for this action. If the remote host is

# down, messages are spooled to disk and sent when it is up again.

#$ActionQueueFileName fwdRule1 # unique name prefix for spool files

#$ActionQueueMaxDiskSpace 1g # 1gb space limit (use as much as possible)

#$ActionQueueSaveOnShutdown on # save messages to disk on shutdown

#$ActionQueueType LinkedList # run asynchronously

#$ActionResumeRetryCount -1 # infinite retries if host is down

# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional

#\*.\* @@remote-host:514

# ### end of the forwarding rule ###

## 3日志记录位置：

* 日志记录位置就是当前日志输出到哪个日志文件中保存，当然也可以把日志输出到打印机打印，或者输出到远程日志服务器上（日志服务器要允许接收）
* 日志文件的绝对路径，最常见的日志保存方法
* 系统设备文件，如“/devlp0”代表第一台打印机，如果日志保存位置是打印机设备的，当有日志时，就会在打印机打印，此方法不太符合可持续发展观，耗费大量纸质资源
* 转发给远程主机，因为可以选择TCP协议和UDP协议传输日志信息，所以有两种发送格式，如“@192.168.10.222:514”就会把日志内容使用UDP协议发送到192.168.10.222的UDP 514端口上。“@@192.168.10.222:514”就会把日志内容使用TCP协议发送到192.168.10.222的UDP 514端口上。514为日志服务默认端口，192.168.10.222主机只要同意接收日志，就可以把日志保存在日志服务器上
* 转发给用户，如“root”将日志发送给root用户，root用户要在线，否则收不到日志信息。使用“ \* ”代表发送给所有在线用户。如果需要把日志发送给多个在线用户，用户名之间用“，”分隔。
* 忽略或丢弃日志，如果接受日志的对象是“~”，代表这个日志不会记录，而被直接丢弃。如“local3.\* ~” 代表忽略local3服务类型所有的日志都不记录。
* \*.emerg \* 解释：无论什么服务产生了emerg等级日志，将给所有用户发送日志

# logrotate日志轮替管理：

分为：切割 和 轮替功能 系统自带的 logrotate

执行文件：/usr/sbin/logrotate

主配置文件：/etc/logrotate.conf

自定义配置文件：/etc/logrotate.d/\*.conf

/etc/logrotate.d/ - 通常一些第三方软件包，会把自己私有的配置文件，也放到这个目录下。如yum，zabbix-agent，syslog等。

修改配置文件后，并不需要重启服务。

由于logrotate实际上只是一个可执行文件，不是以daemon运行。

logrotate 的执行由crond服务实现。在/etc/cron.daily目录中，有个文件logrotate，它实际上是个shell script，用来启动logrotate。logrotate程序每天由cron在指定的时间（/etc/crontab）启动。因此，使用ps是无法查看到logrotate的。如果它没有起来，就要查看一下crond服务有没有在运行。在执行logrotate时，需要指定其配置文件/etc/logrotate.conf。

1日志文件的命名规则

日志轮替最主要的作用就是把旧的日志文件移动并改名，同事建立新的空日志文件，当旧日志文件超出保存范围之后，就会进行删除，旧文件改名依靠/etc/logrotate.conf配置文件中的“dateext”参数。

* 如果配置文件中拥有“deteext”参数，那么日志会用日期来作为日志文件的后缀，日志文件将不会重复，只需保存指定的日志个数，删除多余日志文件即可
* 如果配置文件中没有“dateext”参数，那么日志文件就需要进行改名了，例如：secure日志，当第一次进行日志轮替时，当前secure日志会自动加上后缀.1为sevure.1，然后新建新的同名文件secure来保存新的日志。第二次进行轮替时，secure.1会改为secure.2 当前的secure会改为secure.1再新建secure文件

---------/etc/logrotate.conf 配置详细-----------

# see "man logrotate" for details

# rotate log files weekly

weekly #zdd# 按每周进行轮替

# keep 4 weeks worth of backlogs

rotate 4 #zdd# #保存4个

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

create #zdd# #轮替时改名，并新建一个

# use date as a suffix of the rotated file

dateext #zdd# #使用日期作为日志轮替文件的后缀

# uncomment this if you want your log files compressed

#compress #zdd# #默认注释的，如取消注释，日志则会在转储的同时进行压缩

#zdd# 以上日志配置为默认配置，如果没有单独设定参数，将默认遵守以上参数设置

#zdd# 如果轮替日志配置了独立参数，那么独立参数优先

# RPM packages drop log rotation information into this directory

include /etc/logrotate.d #zdd# 加载logrotate.d 目录中的所有配置文件

# no packages own wtmp and btmp -- we'll rotate them here

/var/log/wtmp {

monthly #如果没有定义这个独立参数，将启用前面的weekly

create 0664 root utmp

minsize 1M

rotate 1 #如果没有定义这个独立参数，将启用前面rotate 4

}

/var/log/btmp {

Missingok #如果日志不存在，则忽略日志信息

monthly

create 0600 root utmp

rotate 1

}

# system-specific logs may be also be configured here.

------logrotate.d 目录中的syslog配置文件--------

/var/log/cron

/var/log/maillog

/var/log/messages

/var/log/secure

/var/log/spooler

{

missingok

sharedscripts #要执行脚本，必须添加这个

postrotate #日志轮替之后执行下面脚本

/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true #脚本内容，重启syslogd

endscript #脚本结束

}

相关参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| daily | 日志轮替周期是每天 |
| wekkly | 日志轮替周期是每周 |
| monthly | 日志轮替周期是每月 |
| routate 2 | 保留日志文件个数为2，0指没有备份 |
| compress | 日志轮替时，旧的日志进行压缩 |
| create mode owner group | 建立新日志，同时指定新日志的权限与所有者和所属组，如create 0600 root utmp |
| mail address | 当日志轮替时，输出内容通过邮件发送指定的邮件地址，如mail shenc@lamp.net |
| missingok | 如果日志不存在，则忽略该日志的警告信息 |
| notifempty | 如果日志为空文件，则不进行日志轮替 |
| minsize 1M | 日志轮替的最小值为：1M 也就是日志一定要达到这个最小值，才会轮替，否则就算轮替时间到达也不轮替 |
| size 1M | 日志轮替指定文件大小为：1M 日志大于指定大小才进行日志轮替，而不是按照时间轮替 |
| dateext | 使用日期作为日志轮替文件后缀 |
| sharedscripts | 在此关键字之后的脚本只执行一次 |
| prerotate/endscript | 在日志轮替之前执行脚本命令，endscript标示prerotate脚本结束 |
| postrotate/endscript | 在日志轮替之后执行脚本命令，endscript标示postrotate脚本结束 |
|  |  |

## 平滑重启 /bin/kill -HUP：

/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true

当我们发送一个挂起信号(1 或 HUP)时，大多数服务器进程[所有常用的进程]都会进行复位操作并重新加载它们的配置文件。在对配置文件作必要的更改后，发出该命令以动态更新服务配置。

如果想要更改配置而不需停止并重新启动服务，请使用该命令。

在对配置文件作必要的更改后，发出该命令以动态更新服务配置。

HUP(1)是让进程挂起，睡眠; 2> /dev/null” 代表忽略掉错误提示信息。

示例1：

向所有正在运行的 Web 服务器进程发送挂起信号，告诉 Web 服务器重新加载其配置文件并对文件进行复位操作。

[root@localhost]# ps -A | grep httpd | grep -v grep | awk '{print $1}' | xargs -L 1 kill -HUP

上面的命令中包含了很多的操作，所以让我们来仔细地看一下管道的每个部分。

ps 和 grep 命令用来在所有的进程中搜索 httpd(并且忽略用来搜索 httpd 进程的 grep 进程)。

接下来，awk 只显示输出结果中的进程 ID，并将它传递给 xargs。然后，xargs 命令接受每个进程 ID(因为使用了 -L 1 以便一次提取一行内容)，并使用kill -HUP 向相应的进程发送一个挂起信号。

示例2

更改/etc/ssh/sshd\_config，将里面的Port改为新端口，如28，然后 kill -HUP `cat /var/run/sshd.pid` 就行了。

现有连接自己不会断，因为kill -HUP `cat /var/run/sshd.pid` 只是HUP监听的那个，已经建立的连接（不同的 pid）不会断。

然后你在现有账号下面exit，在用新的端口登陆就可以了。

## logrotate 命令

logrotate [选项] 配置文件名 #一般不需要执行

Logrotate -d 配置文件名 #详细显示指令执行过程

logrotate -v 配置文件名 #显示日志轮替过程

logrotate -vf 配置文件名 #强制进行日志轮替，且显示轮替过程

选项：如果不带选项，则会按照配置文件中的条件进行日志轮替

-v : 显示日志轮替过程

-f : 强制进行日志轮替，不管日志轮替条件是否已经符合，强制配置文件中所有的日志进行轮替

-usage 　显示指令基本用法。

Chattr +a /var/log/alert.log +a属性，只允许往alert.log文件中追加数据，不能修改

## 自定义日志文件加入日志轮替：

1. 确保/etc/logrotate.conf 配置文件中 inclue 参数定义加载配置文件目录（一般定义为/etc/logrotate.d目录）
2. 在logrotate.d目录中创建一个新文件，如下图、



1. 编辑文件，添加相关配置，参考如下

例如：syslog

/var/log/cron

/var/log/maillog

/var/log/messages

/var/log/secure

/var/log/spooler

{

missingok

sharedscripts

postrotate

/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true

endscript

}

例如：nginx

/date/logs/nginx/access.log /date/logs/nginx/access/default.log{

daily

rotate 5

sharedscripts

postrotate

#重启rsyslog服务

/bin/kill -HUP $(/bin/cat /var/run/syslogd.pid) &>/dev/null

#重启Nginx服务

/bin/kill -HUP $(/bin/cat /usr/local/nginx/logs/nginx.pid) &>/dev/null

endscript

}

例如：rsync日志

/var/run/rsyncd.log{

daily

rotate 7

notifempty

dateext

}

[root@server2 run]# logrotate -d /etc/logrotate.d/rsync #如无报错即正常

[root@server2 run]# logrotate -vf /etc/logrotate.d/rsync #立即强制执行



# journal 日志

journalctl 用来查询 systemd-journald 服务收集到的日志。systemd-journald 服务是 systemd init 系统提供的收集系统日志的服务。

journalctl工具是CentOS-7才有的工具，在Systemd出现之前，Linux系统及各应用的日志都是分别管理的，Systemd开始统一管理了所有Unit的启动日志，这样带来的好处就是可以只用一个 journalctl命令，查看所有内核和应用的日志。

日志的配置文件/etc/systemd/journald.conf

日志服务要么将日志持久存储在 /var/log/journal 目录中、 要么将日志临时存储在 /run/log/journal/ 目录中(关机即丢失)。默认情况下， 如果在启动期间存在 /var/log/journal/ 目录，那么日志将会被持久存储在此目录中， 否则将会临时存储在 /run/log/journal/ 目录中。可以配置使用 Storage= 选项 设置日志的存储位置(无论 /var/log/journal/ 目录是否存在)。

默认journalctl的日志都是保存到内存中。要保存到文件，只需创建/var/log/journal目录，重启journald服务即可

mkdir -p /var/log/journal

systemctl restart systemd-journald

如果 /var/log/journal/ 目录不存在， 同时又将日志配置为持久存储(这是 journald.conf 的默认配置)，那么将会尽可能自动创建该目录， 并为其设置正确的权限与拥有者：

mkdir -p /var/log/journal

systemd-tmpfiles --create --prefix /var/log/journal

## 1基本用法：

[root@devops-101 ~]# systemctl status systemd-journald #查看状态

[root@devops-101 ~]# journalctl #查看所有日志(默认显示本次启动的所有日志)

[root@devops-101 ~]# journalctl -b #查看本次启动的所有日志

[root@devops-101 ~]# journalctl -k #查看内核日志（不显示应用日志）

查看指定时间的日志

过滤任意时间限制显示指定条件之前（--until）、之后（--since）或之间的日志。

[root@devops-101 ~]# journalctl --since="2018-09-21 10:00:00" #20180921 10点之后日志

查询一个时间段范围内的日志

[root@devops-101 ~]# journalctl --since="2018-09-21 10:21:00" --until="2018-09-21 10:22:00"

[root@devops-101 ~]# journalctl -since=yesterday #查看昨天的日志

[root@devops-101 ~]# journalctl --since 09:00 --until "1 hour ago" #查看9点之前一个小时日志

[root@devops-101 ~]# journalctl --since "20 min ago" #查看20分钟之前的

[root@devops-101 ~]# journalctl -f #实时监控日志，类似于tail -f命令

[root@devops-101 ~]# journalctl -n20 #查看最后10条

[root@devops-101 ~]# journalctl --disk-usage #查看当前日志占用磁盘空间的总大小

[root@devops-101 ~]# journalctl --vacuum-size=1G #指定日志文件最大空间

[root@devops-101 ~]# journalctl --vacuum-time=1years #指定日志文件保存多久

## 2根据不同的主题进行过滤筛选

根据服务筛选

[root@devops-101 ~]# journalctl -u kubelet.service

[root@devops-101 ~]# journalctl -u kubelet

[root@devops-101 ~]# journalctl -u httpd.service

[root@devops-101 ~]# journalctl -u httpd.service --since today

根据进程ID查询

如果进程使用了systemd托管日志，则可以通过以下命令查找进程对应的日志。

[root@devops-101 ~]# journalctl \_PID=33

[root@devops-101 ~]# journalctl \_PID=33 --since today

Systemd journal 有很多可以用来过滤的字段，可以通过 man systemd.journal-fields 查看所有可以用来过滤的字段。

对于用来筛选的字段，可以使用-F参数来查看所有可以用来过滤的值，

例如：journalctl -F \_GID。

按优先级

系统提供了从0 (emerg) 到 7 (debug) 一共7个级别的日志，可以配合-p参数分别查看对应级别的日志。

7个级别的含义为

0: emerg

1: alert

2: crit

3: err

4: warning

5: notice

6: info

7: debug

[root@devops-101 ~]# journalctl -p 5 -u httpd.service

[root@devops-101 ~]# journalctl -p err -u httpd.service

[root@devops-101 ~]# journalctl -p err -b #-b本次启动所有日志

## 3调整显示输出

如果大家需要对journal条目进行处理，则可能需要使用更易使用的格式以简化数据解析工作。幸运的是，journal能够以多种格式进行显示，只须添加-o选项加格式说明即可。

[root@devops-101 ~]# journalctl -b -u httpd -o json

默认情况，journal输出进入分页模式，用户可以在终端上调整显示的内容，如果要不需要分页，需要加上--no-pager参数。

[root@devops-101 ~]# journalctl -p 5 --no-pager -o json

通过-o参数，可以设置为json格式输出，这对于其他接收json格式的日志分析工具非常友好。

[root@devops-101 ~]# journalctl -p 5 --no-pager -o json-pretty

使用json-pretty则对于管理员查看日志非常易读。

支持的各种格式如下：

cat: 只显示信息字段本身。

export: 适合传输或备份的二进制格式。

json: 标准JSON，每行一个条目。

json-pretty: JSON格式，适合人类阅读习惯。

json-sse: JSON格式，经过打包以兼容server-sent事件。

short: 默认syslog类输出格式。

short-iso: 默认格式，强调显示ISO 8601挂钟时间戳。

short-monotonic: 默认格式，提供普通时间戳。

short-precise: 默认格式，提供微秒级精度。

verbose: 显示该条目的全部可用journal字段，包括通常被内部隐藏的字段。

1.6 活动日志跟踪

journalctl也支持类似tail的功能，如通过-n参数指定显示最近的多少行，默认为10行，通过-f参数持续监控日志输出。

## 4. 维护

2.1 查看日志占用的磁盘空间

[root@devops-101 ~]# journalctl --disk-usage

Archived and active journals take up 8.0M on disk.

2.2 设置日志占用的空间

root@devops-101 ~]# journalctl --vacuum-size=500M

Vacuuming done, freed 0B of archived journals on disk.

2.3 设置日志保存的时间

[root@devops-101 ~]# journalctl --vacuum-time=1month

Vacuuming done, freed 0B of archived journals on disk.

## 5. 配置

上面的一些维护动作，也可以通过配置参数设置，配置文件位于/etc/systemd/journald.conf。

<http://www.jinbuguo.com/systemd/journald.conf.html> #参考资料

Storage=persistent #启用持久记录

Compress=yes

Seal=yes

SplitMode=uid

SyncIntervalSec=5m

RateLimitInterval=30s

RateLimitBurst=1000

SystemMaxUse= #指定journal所能使用的最高持久存储容量。

SystemKeepFree= #指定journal在添加新条目时需要保留的剩余空间。

SystemMaxFileSize= #控制单一journal文件大小，符合要求方可被转为持久存储。

RuntimeMaxUse= #指定易失性存储中的最大可用磁盘容量（/run文件系统之内）。

RuntimeKeepFree= #指定向易失性存储内写入数据时为其它应用保留的空间量（/run文件系统之内）

RuntimeMaxFileSize= #指定单一journal文件可占用的最大易失性存储容量（/run文件系统之内）。

通过设置上述值，可以控制journald对服务器空间的消耗及保留方式。

MaxRetentionSec=

MaxFileSec=1month

ForwardToSyslog=yes

ForwardToKMsg=no

ForwardToConsole=no

ForwardToWall=yes

TTYPath=/dev/console

MaxLevelStore=debug

MaxLevelSyslog=debug

MaxLevelKMsg=notice

MaxLevelConsole=info

MaxLevelWall=emerg

# 学习一个服务过程

* 服务的概述：含名字、功能、特点、端口号
* 安装：rpm安装、模块安装、源码安装、配置文件的位置
* 服务启动关闭方法
* 常用命令总结
* 修改配置文件，实战
* 排错

# Rsync：

官方网站：<https://rsync.samba.org/>

Rsync=remote sync 远程同步

## 服务的概述

Rsync 与 scp比较：

scp 无法备份大量数据，类似windows的复制，

rsync：变量复制，变统计，边比较，复制变换的类容

rsync开源、快速，多功能，可实现全量增量，可本地、可远程备份数据。一个rsync相当于scp、cp、rm，并且优于它们的命令。

在同步数据时，默认情况下，rsync通过其独特的“quick check”算法：

* 可仅同步大小或者最后修改时间变化的文件或者目录，
* 也可根据权限、属主等属性变化同步，但需要指定相应的参数，
* 至只同步一个文件里面有变化的内容部分，可实现快速的同步备份数据

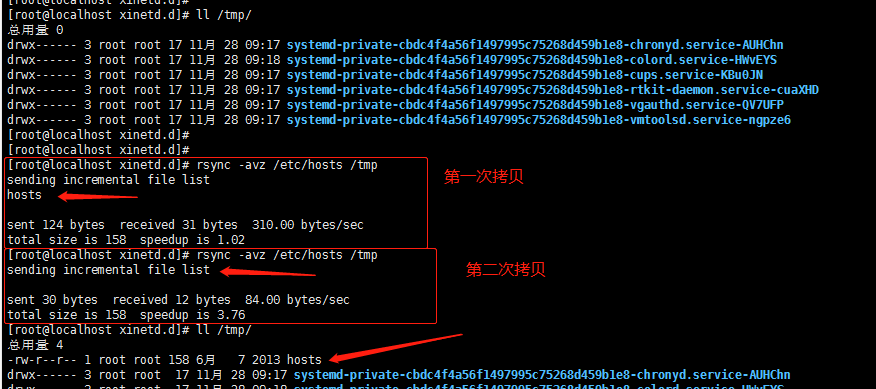
它的特性如下：

* 可以镜像保存整个目录树和文件系统。
* 可以很容易做到保持原来文件的权限、时间、软硬链接等等。
* 无须特殊权限即可安装。
* 快速：第一次同步时 rsync 会复制全部内容，但在下一次只传输修改过的文件。rsync 在传输数据的过程中可以实行压缩及解压缩操作，因此可以使用更少的带宽。
* 安全：可以使用rcp、rsh、ssh等方式来传输文件，当然也可以通过直接的socket(进程方式)连接。
* 支持匿名传输，以方便进行网站镜像。
* 采用C/S模式，就是一个点到点的传输
* 默认端口：873

Rsync 三个工作方式：

* 方式一：单个主机本地之间的数据传输（类似于cp命令的功能）

[root@localhost xinetd.d]# rsync -avz /etc/hosts /tmp #具体选项请看下文常用命令



Rsync删除功能：

--delete参数，比较两边的文件，删除目的目录中的源目录中没有的所有文件



* 方式二：借助rcp、ssh等通道来传输数据（类似于scp命令的功能）
* 方式三：以守护进程（socket）的方式传输数据（这是rsync自身的重要功能）

服务名词解释：

* 发起端：负责发起rsync同步操作的客户机，通知服务器我要备份你的数据
* 备份源：负责响应来自客户机rsync同步操作的服务器所在，需要备份的服务器
* 服务端：运行rsync服务，需要备份的服务器
* 客户端：备份备份数据

常见的备份：

* 完整备份：每次备份都是从备份源将所有文件或目录备份到目的地
* 差异备份：备份上次完全备份以后有变化的数据（他针对的上次的完全备份，备份过程中不清除存档）
* 增量备份：备份上次完全备份以后有变化的数据（他才不管是哪种类型的备份，有变化的数据就备份，备份过程中清楚存档备份）

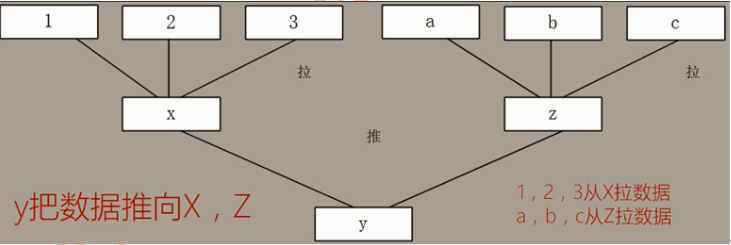
数据同步方式： 两种方式rsync都有对应的命令来实现

推push:一台主机负责把数据传送给其他主机，服务器开销大，比较适合后端服务器少的情况

推：目的主机配置为rsync服务器，源主机周期性使用rsync命令把要同步的目录推送过去

拉pull:所有主机定时去找一主机拉取数据，可能导致数据缓慢

拉：源主机配置为rsync服务器，目的主机周期性使用rsync命令把要同步的目录拉取过来

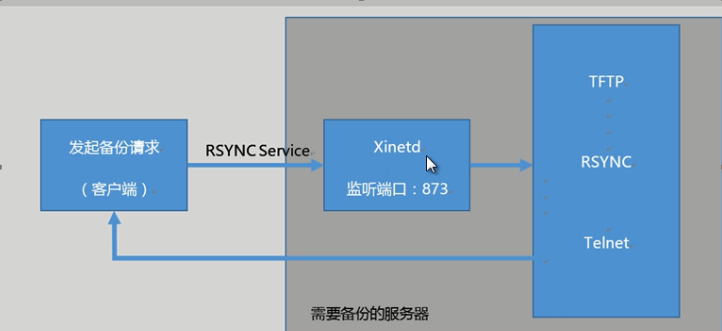


rsync工作原理：

Rsync是Xinetd服务的一小部分，rsync由Xinetd服务管理

xinetd服务接受rsync server 的请求，唤醒众多服务中的rsyncf服务（众多服务包含http、rsync、telnet等），唤醒rsync后，再与发起备份请求的客户端去沟通，进行备份。

总结：使用rsync同步就是通过xinetd监听873端口，有873端口请求，xinetd会通知它所管辖的rsync服务来做回应，接下来就是rsync服务之间的通讯。



## 安装

可rpm安装，可源码安装

Rpm安装：

因为由xinetd安装，先得安装xinetd

下载xinetd-2.\*\*\*\*.rpm包 下载rsync\*\*\*.rpm包

[root@server1 ~]# rpm -ivh xinetd-2.\*\*\*\*.rpm

## 配置文件

## 启停办法

## 常用命令

[root@server1 ~]# rsync --version #查看版本

## 配置文件