|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 配置项 | 参数值 | 默认参数值 | 示例数值 | 参数作用 |
| Storage | none/volatile/persistent/auto | auto | none | 它有四个可能的值： “none”：这有效地关闭了日记功能。收到的任何日志消息都将被删除。但是，任何重定向到控制台或syslog或内核日志缓冲区仍然有效。 “volatile”：日志数据保存在内存中，暂时可在/ run /log/journal目录下使用。如果目录不存在，将创建该目录。 “persistent”：日志数据持久保存在/var/log/ journal目录下的磁盘上。如果目录不存在，将创建该目录。如果磁盘卷不可访问或可写，则将在/run/log/journal下创建文件。 “auto”：存储模式类似于持久性 - 数据将写入磁盘;但是，如果/var/log/journal目录不存在，它将在/run/log/journal下创建。 |
| Compress | yes/no | yes | none | 如果启用此参数，则在写入磁盘之前将压缩存储在日志中的大于阈值的数据。默认情况下，该选项处于启用状态。 |
| Seal | yes/no | yes | yes | 为日志添加序列号 |
| SplitMode |  |  | uid | 每个用户分别记录日志 |
| SyncIntervalSec |  |  | 1m | 日志同步到磁盘的间隔，高级别的日志，如：CRIT、ALERT、EMERG 三种总是实时同步 |
| RateLimitInterval |  |  | 30s | 即制日志的最大流量，此处指 30s 内最多记录 100000 条日志，超出的将被丢弃 |
| RateLimitBurst |  |  | 100000 | 与 RateLimitInterval 配合使用 |
| SystemMaxUse |  |  | 64G | 此参数控制日志在持久时可以使用的最大磁盘空间。默认为磁盘空间的10％。 |
| SystemKeepFree |  |  | 1G | 这是控制日志可以增长到多大的几个参数之一。如果systemd将日志保存在/ var / log / journal目录下，则此参数适用。它指定systemd-journald守护程序将为托管日记的文件系统中的其他应用程序留出多少磁盘空间。默认值为15％。 |
| SystemMaxFileSize |  |  | 128M | 这指定了持久存储的日志文件的最大大小。默认值为SystemMaxUse参数大小的八分之一。 |
| RuntimeMaxUse |  |  | 64G | 与SystemMaxUse相同，适用于易失性日志存储（文件保存在/ run / log / journal下）。同样，默认值为10％。 |
| RuntimeKeepFree |  |  | 128M | 与SystemKeepFree相同，但这在日记存储选项设置为“volatile”时适用，这意味着日志文件在/ run / log / journal下创建。 |
| RuntimeMaxFileSize |  |  | 128M | 与SystemMaxFileSize相同：适用于/ run / log / journal下的文件。 |
| MaxRetentionSec |  | 0 | 100year | 在日记中存储条目的最长时间。包含早于此期间记录的日记文件将自动删除。但是，该值不必仅以秒为单位指定。它可以以“年”，“月”，“周”，“日”，“h”或“m”为后缀。默认值为0.  此参数不需要从默认值0设置，因为SystemMaxUse参数将确保日志不会增长并填满整个磁盘空间。 |
| MaxFileSec |  |  | 1day | 这与MaxRetentionSec参数相同，但这适用于单个日志文件。此参数的默认值为一个月，并控制将日记帐分录保留在单个日记帐文件中的最长时间。同样，可以使用值0关闭此值，因为参数SystemMaxFileSize可以控制单个文件的最大大小。 |
| ForwardToSyslog | yes/no | yes | yes | 此参数指定systemd-journald守护程序接收的日志消息是否也将转发到syslog守护程序。默认值为yes，但如果没有进程从套接字读取，则不会发生任何事情。 |
| ForwardToKMsg | yes/no | no | no | 如果systemd-journald收到的日志消息也将转发到内核日志缓冲区。默认为no。 |
| ForwardToConsole | yes/no | no | no | 如果systemd-journald收到的日志消息也将转发到系统控制台。 如果启用此参数，则另一个参数TTYPath确定要向其发送消息的控制台TTY。该参数的默认值是/ dev / console。 |
| ForwardToWall | yes/no | yes | yes | 如果systemd-journald收到的日志消息也将作为挂起消息发送给所有登录用户。默认为yes。 |
| TTYPath |  |  | /dev/console | 是否转发符合条件的日志到所有登陆用户的终端 |
| MaxLevelStore | emerg/alter/crit/err/warnign/notice/info/debug | debug | info | 此参数可以采用以下任何值：   * 0 OR “EMERG” * 1 OR “ALERT” * 2 OR “CRIT” * 3 OR “ERR” * 4 OR “WARNING” * 5 OR “NOTICE” * 6 OR “INFO” * 7 OR “DEBUG”   所有等于或低于指定级别的消息都将存储在磁盘上。默认值为“debug”，表示从“emerg”到“debug”的所有日志消息。 |
| MaxLevelSyslog | emerg/alter/crit/err/warnign/notice/info/debug | debug | err | 控制转发到Syslog的最大日志消息级别。同样，默认值为“debug”。 |
| MaxLevelKMsg | emerg/alter/crit/err/warnign/notice/info/debug | notice | notice | 控制转发到内核日志缓冲区的最大日志消息级别。默认值为“notice”。 |
| MaxLevelConsole | emerg/alter/crit/err/warnign/notice/info/debug | info | info | 控制转发到系统控制台的最大日志消息级别，默认值为“info”。 |
| MaxLevelWall | emerg/alter/crit/err/warnign/notice/info/debug | emerg | emerg | 控制转发到登录用户墙的最大日志消息级别。默认为“emerg” - 意味着如果启用，将立即通知用户紧急事件。 |

完全参数配置模板：/etc/systemd/journald.conf

|  |
| --- |
| [Journal]  #日志存储到磁盘  Storage=persistent  #压缩日志  Compress=yes  #为日志添加序列号  Seal=yes  #每个用户分别记录日志  SplitMode=uid  #日志同步到磁盘的间隔，高级别的日志，如：CRIT、ALERT、EMERG 三种总是实时同步  SyncIntervalSec=1m  #即制日志的最大流量，此处指 30s 内最多记录 100000 条日志，超出的将被丢弃  RateLimitInterval=30s  #与 RateLimitInterval 配合使用  RateLimitBurst=100000  #限制全部日志文件加在一起最多可以占用多少空间，默认值是10%空间与4G空间两者中的较小者  SystemMaxUse=64G  #默认值是15%空间与4G空间两者中的较大者  SystemKeepFree=1G  #单个日志文件的大小限制，超过此限制将触发滚动保存  SystemMaxFileSize=128M  #日志滚动的最大时间间隔，若不设置则完全以大小限制为准  MaxFileSec=1day  #日志最大保留时间，超过时限的旧日志将被删除  MaxRetentionSec=100year  #是否转发符合条件的日志记录到本机的其它日志管理系统，如：rsyslog  ForwardToSyslog=yes  ForwardToKMsg=no  #是否转发符合条件的日志到所有登陆用户的终端  ForwardToWall=yes  MaxLevelStore=debug  MaxLevelSyslog=err  MaxLevelWall=emerg  ForwardToConsole=no  #TTYPath=/dev/console  #MaxLevelConsole=info  #MaxLevelKMsg=notice |

操作步骤如下：

在CentOS 7开始使用的systemd使用了journal日志，这个日志的管理方式和以往使用syslog的方式不同，可以通过管理工具维护。

使用df -h检查磁盘文件，可以看到/run目录下有日志目录/run/log/journal，占用了数G空间

在日志目录下有很多历史累积的日志。

检查当前journal使用磁盘量

journalctl --disk-usage

清理方法可以采用按照日期清理，或者按照允许保留的容量清理

journalctl --vacuum-time=2d

journalctl --vacuum-size=500M

如果要手工删除日志文件，则在删除前需要先轮转一次journal日志

systemctl kill --kill-who=main --signal=SIGUSR2 systemd-journald.service

要启用日志限制持久化配置，可以修改 /etc/systemd/journald.conf

SystemMaxUse=16M

ForwardToSyslog=no

然后重启

systemctl restart systemd-journald.service

检查journal是否运行正常以及日志文件是否完整无损坏

journalctl --verify

每次修改过系统配置参数以后需要执行命令：

systemctl restart systemd-journald.service