Centos7 日志查看工具

1 概述

日志管理工具journalctl是centos7上专有的日志管理工具，该工具是从message这个文件里读取信息。Systemd统一管理所有Unit的启动日志。带来的好处就是，可以只用journalctl一个命令，查看所有日志（内核日志和应用日志）。

日志的配置文件是: /etc/systemd/journald.conf

journalctl功能强大，用法非常多。

本文将介绍journalctl的相关使用方法。

2 journalctl 使用方法

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,如上次系统崩溃，需要查看日志时，就要看上一次的启动日志。

```
journalctl  -b -1
```

查看指定时间的日志

```
journalctl --since="2012-10-3018:17:16"
journalctl --since "20 minago"
journalctl --since yesterday
journalctl --since"2015-01-10" --until "2015-01-11 03:00"
journalctl --since 09:00 --until"1 hour ago"
journalctl --since"15:15" --until now
```

显示尾部的最新10行日志

```
journalctl  -n
```

显示尾部指定行数的日志查看的是/var/log/messages的日志，
但是格式上有所调整，如主机名格式不一样而已

```
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

日志优先级共有8级

- 0: emerg
- 1: alert
- 2: crit
- 3: err
- 4: warning
- 5: notice
- 6: info

- 7: debug

显示不同级别的日志：

```
journalctl  -p err  -b
journalctl  -p err..alert -b
```

不分页标准输出日志
默认分页输出--no-pager改为正常的标准输出

```
journalctl  --no-pager
```

以JSON格式（单行）输出

JSON(JavaScript Object Notation)是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。它基于JavaScriptProgramming Language, Standard ECMA-262 3rd Edition - December 1999的一个子集。JSON采用完全独立于语言的文本格式，但是也使用了类似于C语言家族的习惯（包括C, C++, C#, Java,JavaScript, Perl, Python等）。这些特性使JSON成为理想的数据交换语言。
JSON建构于两种结构：

“名称/值”对的集合（A collection ofname/value pairs）：不同的语言中，它被理解为对象（object），纪录（record），结构（struct），字典（dictionary），哈希表（hash table），有键列表（keyed list），或者

关联数组（associativearray）。

值的有序列表（An ordered list of values）：在大部分语言中，它被理解为数组（array）。

这些都是常见的数据结构。事实上大部分现代计算机语言都以某种形式支持它们。这使得一种数据格式在同样基于这些结构的编程语言之间交换成为可能。

例子:

以JSON格式（单行）输出

```
journalctl  -b -u httpd.service  -o json
```

以JSON格式（多行）输出，可读性更好，建议选择多行输出

```
journalctl  -b -u httpd.service  -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
```