Journalctl

On systemd systems the journal is implemented with the journald daemon, which handles all of the messages produced by the kernel, initrd, services, etc.The journald daemon collects data from all available sources and stores them in a binary format for easy and dynamic manipulation.

Setting the System Time

One option of the binary journal is the ability to view log records in UTC or local time. By default, systemd will display results in local time. Because of this, make sure the timezone is set up correctly. The systemd suite actually comes with a tool called timedatectl that can help with this. First, see what timezones are available with the list-timezones option:

```
timedatectl list-timezones
```

This will list the timezones available on your system. When you find the one that matches the location of your server, you can set it by using the set-timezone option:

```
timedatectl set-timezone 'zone'
timedatectl set-timezone Europe/Amsterdam
```

To ensure that your machine is using the correct time now, use the timedatectl command alone, or with the status option. The display will be the same:

```
timedatectl status
```

Output

```
Local time: Fri 2021-07-09 14:44:30 EDT
Universal time: Fri 2021-07-09 18:44:30 UTC
RTC time: Fri 2021-07-09 18:44:31
Time zone: America/New_York (EDT, -0400)
System clock synchronized: yes
      NTP service: active
      RTC in local TZ: no
```

The first line should display the correct time.

Log Viewing

Standard

```
journalctl
```

Boot messages

```
journalctl -b
journalctl -b -1  (previous boot)
Displaying Kernel Messages
```

Kernel messages

Usually found in dmesg output, can be retrieved from the journal as well. To display only these messages, we can add the -k or –dmesg flags to our command:

```
journalctl -k
```

By default, this will display the kernel messages from the current boot. To get the messages from five boots ago, you could type:

```
journalctl -k -b -5
```

Time period

```
journalctl --since "2015-01-10 17:15:00"
journalctl --since "2015-01-10" --until "2015-01-11 03:00"
journalctl --since yesterday
journalctl --since 09:00 --until "1 hour ago"
```

By Unit

```
journalctl -u nginx.service
journalctl -u nginx.service --since today
journalctl -u nginx.service -u php-fpm.service --since today
```

By Process, User, or Group ID

At other times, you may wish to show all of the entries logged from a specific user or group. This can be done with the _UID or _GID filters. For instance, if your web server runs under the www-data user, you can find the user ID by typing:

```
journalctl _PID=8088
```

```
id -u www-data

Output
33
```

```
journalctl _UID=33 --since today
```

Other Attributes

The systemd journal has many fields that can be used for filtering. Some of those are passed from the process being logged and some are applied by journald using information it gathers from the system at the time of the log.

```
man systemd.journal-fields
```

The -F option can be used to show all of the available values for a given journal field. For instance, to see which group IDs the systemd journal has entries for, you can type:

```
journalctl -F _GID

Output
32
99
102
133
81
84
100
```

By Component Path

We can also filter by providing a path location. If the path leads to an executable, journalctl will display all of the entries that involve the executable in question. For instance, to find those entries that involve the bash executable, you can type:

```
journalctl /usr/bin/bash
```

By Priority

You can use journalctl to display only messages of a specified priority or above by using the -p option. This allows you to filter out lower priority messages. For instance, to show only entries logged at the error level or above, you can type:

```
journalctl -p err -b
```

In order of highest to lowest priority, these are:

```
0: emerg
1: alert
2: crit
3: err
```

```
4: warning
5: notice
6: info
7: debug
```

Active Process Monitoring

The journalctl command imitates how many administrators use tail for monitoring active or recent activity. This functionality is built into journalctl, allowing you to access these features without having to pipe to another tool. To display a set amount of records, you can use the -n option, which works exactly as tail -n. By default, it will display the most recent 10 entries:

```
journalctl -n
```

You can specify the number of entries you'd like to see with a number after the -n:

```
journalctl -n 20
```

Following Logs

To actively follow the logs as they are being written, you can use the -f flag. Again, this works as you might expect if you have experience using tail -f:

```
journalctl -f
```