

Registration is open - Live, Instructor-led Online Classes - Elasticsearch in March - Solr in April - OpenSearch in May. [See all classes](#)

[SEMATEXT](#) > [BLOG](#) > [LOGGING](#)

Ubuntu Logs: How to Check and Configure Log Files

POSTED ON NOVEMBER 28, 2022 BY [JEAN-CHRISTOPHE DANSEREAU](#)

TABLE OF CONTENTS

- [What Are Ubuntu Logs?](#)
- [Ubuntu Log Files Location: Where Are They Stored?](#)
 - [System Logs](#)
 - [Application Logs](#)
 - [Other Useful Logs](#)
- [How to View Ubuntu Logs](#)
 - [GNOME Logs App](#)
 - [Log File Viewer](#)
 - [Terminal](#)
 - [Journalctl Command](#)
- [Ubuntu System Logging Daemon Configuration](#)
- [Ubuntu Log Rotation](#)
- [Useful Commands for Working with Ubuntu Logs](#)
 - [Change Log Directory](#)
 - [Search Log Files](#)
 - [Edit Log Files](#)
 - [Monitor Log Files](#)
- [Ubuntu Log Analysis and Monitoring with Sematext](#)
- [Conclusion](#)

Ubuntu provides extensive logging capabilities, so most of the activities happening in the system are tracked via logs. Ubuntu logs are valuable sources of information about the state of your Ubuntu operating system and the applications deployed on it.

The majority of the logs are in plain text ASCII format and easily readable. This makes them a great tool to use for troubleshooting and identifying the root causes associated with system failures or application errors.

Due to the wide variety of system and application logs available, choosing the appropriate log sources and locating them within your system can be a daunting task. In this post, I'll be taking you through the many **types of Ubuntu logs, as well as how to view and analyze them.**

What Are Ubuntu Logs?

Ubuntu logs record the events occurring in the hardware, software, and operating system, which aid in determining the underlying cause of any problems the system may experience.

-
- System logs that provide insights on operating system functionalities like access control and system processes.
 - Application logs are generated by the applications deployed in the system, offering information on their state.

Ubuntu Log Files Location: Where Are They Stored?

The root directory of the majority of log files is **/var/log**, while most system logs are generated by the [syslog log daemon](#) to capture activities. The OS will use syslog, whereas the majority of programs will log data to files in the /var/log subdirectory.

System Logs

System logs include details about the system and certain applications. Syslog usually stores its log files at **/var/log/messages** or **/var/log/syslog**.

Below are some key system logs and how to find them to help gain more insights into the health of your Ubuntu system and the applications running on it

Authorization Log

This log contains information about the authorization processes that happen in the system. This includes sudo commands, SSH logins, or any authentication modules used for collecting user credentials. You can find the authorization log file at **/var/log/auth.log**.

Daemon Log

Ubuntu systems have a number of daemons, which are effectively background services that offer certain functionalities, monitor the system, or support other processes. Some of the common daemons are systemd, logind, gcfsv, etc. They could also be application-specific daemons like mysqld. These logs are routed to the **/var/log/daemon.log** file and can help you investigate issues associated with their respective daemons.

Debug Log

Systems use the syslogd protocol to send log messages to a [central location](#). The debug log stores the debug messages sent to syslogd at the DEBUG level by applications or Ubuntu system components. You can find the debug log at **/var/log/debug**.

Kernel Log

This log captures events or messages related to the Linux kernel. The information available in this log helps administrators investigate kernel-level issues in Ubuntu and you can find it at **/var/log/kern.log**.

Kernel Ring Buffer

The kernel ring buffer is a data structure that lets you store information related to kernel operations in Ubuntu. It is not a log file but rather a constant-size buffer where the oldest messages are removed to make space for new messages. The messages can be queries using the following command:

```
dmesg | less
```

System Log

The system log contains exhaustive information about the state of an Ubuntu system. It's found at **/var/log/syslog**, the default location referred to by administrators when required information is not found in other log files. Applications or services that do not have their own log files use this file to store logging information. It also contains logs that used to be saved in **/var/log/messages** in older versions of Ubuntu.

Service Log

In Ubuntu, service logs are system logs that capture information about the state of your services. You can look into an event logged by the daemon systemd-journald if logs about a specific service are not found in /var/log. This daemon publishes a single

stream combining the log outputs from all services to the central journal in **`/run, var/log/journal/`**. To view data for a specific service, use the following command:

```
journalctl -f -u {service name}
```

Cron Log

The cron daemon reads crontab files and runs scheduled operations in the background. The logs it produces are useful for debugging and auditing cron jobs. The Ubuntu crontab log can be found in **`/var/log/syslog`**. The following command lets you view cron job-specific entries in the file:

```
grep CRON /var/log/syslog
```

Network Log

The NetworkManager service manages network connectivity in Ubuntu systems. By default, the log files generated by NetworkManager are streamed to **`/var/log/syslog`**. For the latest versions of Ubuntu, you can view the logs with the command:

```
journalctl /usr/sbin/NetworkManager
```

SEMATEXT LOGS MONITORING

The log data you need minus the headaches.

Get Started

Schedule a Demo

Audit Log

Audit information about Ubuntu systems is provided by [auditd](#), the Linux audit daemon. The default location of the auditd log file is **`/var/log/audit/audit.log`**. This also allows you to view commands like `ausearch` and `aureport` to parse and analyze audit logs.

Startup Log

System boot logs are located in **/var/log/boot.log**. To read the most recent boot log messages on Ubuntu operating systems that use systemd, use the `journalctl` command as below:

```
journalctl -b
```

Crash Log

The Ubuntu Kernel Crash Dump mechanism collects the state and memory of the kernel when a system crashes. This data is saved to **/var/crash** and helps in identifying the root cause of the crash.

Firewall Log

Firewall logs are necessary for identifying odd network activities, spotting attacks, and debugging firewall rules. The Ubuntu firewall logs associated with the default firewall service UFW can be found in the **/var/log** folder. You can manage the location of the logs by editing the Ubuntu syslog config file available at **/etc/syslog.conf**.

Journal Log

Journald is the log management service daemon in Ubuntu. Go ahead and edit the location of the logging file in **/etc/systemd/journald.conf**. To view the logging data captured by journald, use the `journalctl` command.

Error Log

Error logs provide useful information about issues happening in application and system components. [Error log files](#) specific to an application or service are available in the **/var/log** folder.

SSH Log

Information about user logins and usage of sudo commands in SSH is also stored in the authorization logs. You can search the files to get information about SSH logs using the following command:

```
grep sshd /var/log/auth.log | less
```

Application Logs

I usually recommend you analyze application logs as the first step when attempting to identify the source of an application error or, in general, to review your app's health. Let's explore the logging capabilities of some common applications.

Apache Log

Apache generates two types of logs to help provide valuable insights into the health of the Apache server, its resource utilization patterns, and associated metrics:

- Access logs include the server's response codes, resource access information, time of access, source IP address, and other details. In Ubuntu, Apache access logs can be found at **`/var/log/apache2/access.log`**.
- Error logs log any errors that happen while the Apache server processes requests. These are very helpful to troubleshoot Apache server issues and can contain insights into the root cause. In Ubuntu machines, Apache error logs are available at **`/var/log/apache2/error.log`**.

Read the [Apache logs guide](#) to learn more.

NGINX Log

NGINX provides multiple log files that the administrator can configure to provide insights into activities on the server:

- Access logs provide information about the resource access pattern on the server, i.e., response code, user agent, source IP address, etc. The default log location is **`/var/log/nginx/access.log`**.
- Error logs contain all NGINX errors, related to both configuration or access. You can find them at **`/var/log/nginx/error.log`**.

If you're using NGINX, read [NGINX logging guide](#) to learn more about how to work with NGINX logs.

Docker Log

As [Docker containers](#) are ephemeral, capturing the logs requires additional configuration.

The logs sent by containers to stdout or stderr should be forwarded by a logging driver to the log destination. The logs are streamed to the **`/var/lib/docker/containers/`** directory of the container host machines. Access them using the following command:

```
$ docker logs [OPTIONS] <CONTAINER-NAME OR ID>
```

Read our blog post on Docker logs for a deep dive into Docker logging.

HAProxy Log

HAproxy logs provide useful information to troubleshoot the errors encountered by clients accessing the applications with an HAProxy frontend. In the event of an error, a corresponding log will be written in HAProxy log files. Ubuntu systems, you can locate this log at **/var/log/haproxy.log**.

Check out the [HAProxy logging guide](#) if HAProxy is part of your technology stack.

PostgreSQL Log

By default, PostgreSQL logs are streamed to stderr. The logs will, however, be written to the default OS log directory if the `logging_collector` parameter of the service is enabled. Log files of the PostgreSQL server in Ubuntu can be found at `/var/log/postgresql/postgresql-x.x.main.log`. X.x.

If you're running PostgreSQL on Ubuntu, read our post on [PostgreSQL logging](#) to learn how to best handle its logs.

Other Useful Logs

In addition to the abovementioned logs, there are some additional Ubuntu server logs that the system generates but are not human-readable. These logs are also stored in the **/var/log** folder.

Login Failure Log

These logs contain information about the latest login failures in the system. The logs files are located at **/var/log/faillog**. The data here can be parsed using the `faillog` command to view the latest login failures.

Last Logins Log

This log lists the most recent logins to the system. The log file is located at **/var/log/lastlogin**. To parse the log and view a list of last logins to the system, use the following command:

```
lastlog | less
```

Login Records Log

This log stores information about users currently logged into the system. The log file is located at **/var/log/wtmp** and can be parsed by command line utilities such as the `who` command.

How to View Ubuntu Logs

Logs provide a wealth of information to help troubleshoot issues or errors in the Ubuntu system or applications deployed in it. But to view and analyze the log files, additional tools are required, some of which I'll cover here.

GNOME Logs App

GNOME is a commonly used desktop GUI interface for Ubuntu. In the latest versions of Ubuntu GNOME, a log viewer utility named "Logs" is available out of the box. It's a GUI utility that helps you view a number of log files including your hardware, system, application, security, etc.

In the Ubuntu Dash, search for "Logs" to open the utility. The utility contains different tabs to view the logs. For example, you can click on the "System" tab to view the system logs. You can also view additional details about specific logs by clicking on them or search for logs using keywords.

Log File Viewer

The log file viewer is another GUI-based tool that lets you view and analyze logs and is available in older versions of Ubuntu. To access the utility, search for "Log Viewer" in the Ubuntu Dash. It has several categories like syslog, auth.log, kern.log, etc. listed on the left panel. You can use these to explore respective log types. You can also search for specific logs using keywords.

Terminal

Administrators comfortable with the command-line mode of analyzing logs can use the Ubuntu terminal. There are several commands available to view the log files in Ubuntu:

- `tail` to view the last few lines of the log file
- `dmesg` to view log messages from Kernel's buffer
- `cat` to open and view log files
- `more` for when you want to browse through the log file one screen at a time; use space bar to go to the next page
- `less` also displays content of log files one screen at a time but has additional options to scroll forward, backward, and search
- `grep` to search for specific keywords within a log; helpful if you are looking for information on specific incidents/errors in the log file, for example, `dmesg | grep`

[keyword]

Journalctl Command

In earlier sections, I talked about `journald`, which is the `systemd`'s logging service or, in other words, the log management daemon in Ubuntu. `Journalctl` is the utility available for viewing and querying these logs since they're stored in binary format by `journald`. You can use it without any parameters to view all the content of the `systemd` journal:

```
journalctl
```

As all the logs in the journal will be overwhelming to comprehend, use appropriate filters for the output. For instance, to view only messages within a specified timeframe, input the following command:

```
journalctl --since "2 hours ago"
```

To view logs associated with a specific `systemd` unit, use the `-u` switch. For example, to view logs from the MySQL service:

```
journalctl -u mysql.service
```

You can also view log messages related to activities of a specific user with the UID of the user as a switch, as shown below:

```
journalctl _UID=209
```

You can use the `--since` and `--until` switches to view logs within a time period:

```
journalctl --since "2022-07-21 23:15:00" --until "2022-07-21 23:20:00"
```

Ubuntu System Logging Daemon Configuration

`Syslogd`, also known as `sysklogd`, is the default system logging daemon in Ubuntu. It performs the task of receiving messages from different system sources and sends them to the defined log file destination. Besides log messages, additional useful metadata like timestamps for when the logs are generated, source system hostnames, etc. are also captured in these log files.

You can configure the log files destination and the priority level of logs by editing the **/etc/syslog.conf** file. The selector field in the configuration file specifies the logging facility and the required log priority level. For example, you can choose the auth logging facility and the priority as "info" or "warning".

The log destination file is specified by the action field. This could be the default Ubuntu server log files location, i.e., **/var/log/syslog**, or the name of a centralized logging server.

Ubuntu Log Rotation

With multiple logs generated every minute by different system and application components, the log file data can easily consume the entire disk space in your machine. So it's important to compress and archive log files periodically. This process is called log rotation. Initially, logs are renamed after a specified timeframe and a new log file gets created for the latest logs. The logs are then compressed to save disk space. The utility used to manage this log rotation process is called "logrotate".

Using Logrotate

Configure the logrotate utility via the file **/etc/logrotate.conf**. With the default settings in logrotate.conf, Ubuntu rotates log files weekly for files owned by the Ubuntu syslog server group and root user, with a default retention of four logs at a time.

When current logs are rotated, new empty log files are created to which the logs get streamed. The logrotate utility is invoked from the cron script **/etc/cron.daily/logrotate**.

You can add a log rotation configuration specific to applications to **/etc/logrotate.d/**. This is also where you can define the log rotation settings of system tools like APT, syslog, and dpkg or for specific applications like MySQL and Apache2.

Useful Commands for Working with Ubuntu Logs

Let's look at some of the basic commands you can use in an Ubuntu terminal to explore log files.

Change Log Directory

The first step for accessing logs would be to switch to the log file directly using the following command:

```
cd /var/log
```

Search Log Files

Log files can contain a lot of information, so you might sometimes have to use the search option to drill down to the relevant log content, or you can use the grep command. For example, to search for all lines with the word "error" in a log file:

```
grep "error" kern.log
```

Edit Log Files

Simple text editors like nano let you edit and make changes in the log files. Note that to save changes, you will have to use the sudo command, which is not a recommended practice. A sample command for editing the log file is:

```
nano test.log
```

Monitor Log Files

As log files are continuously getting updated, use the following command to monitor those changes in real time:

```
tail -f error.log
```

While the utilities and tools discussed above can be helpful in analyzing logs in individual servers, when it comes to large-scale infrastructure deployments you need to consider custom-built solutions for Ubuntu log management and analysis like Sematext Logs. You may have to correlate events and logs across multiple systems, and the manual [analysis of logs](#) could become cumbersome.

We have also created a comprehensive guide to the [best Ubuntu server monitoring tools](#) that you may find handy.

Ubuntu Log Analysis and Monitoring with Sematext

Sematext Logs is a [log management tool](#) that can provide out-of-the-box integration for [monitoring Ubuntu logs](#). It's a full-spectrum solution, capable of collecting logs and events in real time from your Ubuntu systems, then parsing and analyzing them to give deeper visibility into your Ubuntu hosts. It allows you to set up threat or [anomaly detection alerts](#) to be warned about any potential issues upfront, which helps bring down troubleshooting time.

Simply install the Sematext agent for an intuitive UI that lets you select the type of logs you want to monitor to cut out the noise and focus on relevant information. For example, the agent is capable of collecting logs from important system components like systemd-journal as well as specific applications like NGINX.



Sematext – Ubuntu logs overview



Sematext – Ubuntu journald log view

You can configure the Ubuntu logging options, input, output, and parser in the agent configuration file as per your [log monitoring](#) requirements. Moreover, being part of the [Sematext Cloud monitoring suite](#), it allows you to also correlate the data from your logs with additional context from performance metrics for faster troubleshooting and debugging. Sematext Cloud even helps you build an actionable dashboard for a bird's-eye view of the state of your Ubuntu system.

Watch the video below to learn more about what Sematext Logs can do for you or jumpstart your logging journey by trying out the 14-day free trial!

Sematext Logs Product Overview | Centralized Logging for all of your Applica



capture the level of detail your organization requires. Ubuntu's built-in tools for viewing logs can help with the first level of analysis. This approach works well if you have only a few systems to monitor.

However, large-scale log ingestion and management needs enterprise-scale specialized tools like [Sematext Logs](#). When your Ubuntu landscape is expansive, this requirement becomes even more prevalent, especially due to the sheer volume of logs generated. Without the right tools at your disposal, finding the source of outages and correcting them could prove to be a difficult undertaking.

Author Bio

Jean-Christophe Dansereau

Jean-Christophe Dansereau is a Canadian software engineer and tech writer, specializing in backend and cloud engineering. He helps tech companies develop and scale applications and write technical blogs to allow organizations to communicate their message.

[Start Free Trial](#)

SEMATEXT IS HIRING

- [Product Manager, Sematext Cloud](#)
- [Agent Engineer](#)
- [Backend Engineer](#)
- [Full Stack Developer](#)
- [Frontend Developer](#)

[See all jobs](#)

DO YOU HAVE A COOL STORY TO SHARE?

[Write for us](#)

Stay up to date

Get tips, how-tos, and news about Elastic / ELK Stack, Observability, Solr, and Sematext Cloud news and updates.

Email *

☐ I agree to receive digital communications pursuant to the terms of [privacy policy](#). I can opt-out at any time using the unsubscribe link in Sematext emails. *

[Subscribe](#)



Production Support

Solr, Elasticsearch, OpenSearch, Logging Consulting

Advanced Training


PRODUCTS

Sematext Cloud
Infrastructure Monitoring
Log Management
Real User Monitoring
Synthetic Monitoring
APM / Tracing






SERVICES

Consulting
Support
Training

ABOUT

Company
Blog
Jobs
Customers
Status 
Awards

CONTACT

+1 347-480-1610
info@sematext.com
Brooklyn, NY USA 
Twitter 
Facebook 
GitHub 
LinkedIn 

© **Sematext Group. All rights reserved**

[Terms Of Service](#) · [Privacy Policy](#)

Apache Lucene, Apache Solr and their respective logos are trademarks of the Apache Software Foundation. Elasticsearch, Kibana, Logstash, and Beats are trademarks of Elasticsearch BV, registered in the U.S. and in other countries. Sematext Group, Inc. is not affiliated with Elasticsearch BV.