

## Some logging and visualization tools available in the market with the preferred scenario to use one over other

There are quite a few open source log trackers and analysis tools available today, making choosing the right resources for activity logs easier than you think. The free and open source software community offers log designs that work with all sorts of sites and just about any operating system. Some of them are:

### Graylog

**Graylog** started in Germany in 2011 and is now offered as either an open source tool or a commercial solution. It is designed to be a centralized log management system that receives data streams from various servers or endpoints and allows you to browse or analyze that information quickly.

### Nagios

**Nagios** started with a single developer back in 1999 and has since evolved into one of the most reliable open source tools for managing log data. The current version of Nagios can integrate with servers running Microsoft Windows, Linux, or Unix.

### ELK Stack

**ELK Stack** is one of the most popular open source tools among organizations that need to sift through large sets of data and make sense of their system logs. Its primary offering is made up of three separate products: Elasticsearch, Kibana, and Logstash.

### LOGalyze

**LOGalyze** is an organization based in Hungary that builds open source tools for system administrators and security experts to help them manage server logs and turn them into useful data points. Its primary product is available as a free download for either personal or commercial use.

## Fluentd

**Fluentd** is a robust solution for data collection and is entirely open source. It does not offer a full frontend interface but instead acts as a collection layer to help organize different pipelines. It is used by some of the largest companies worldwide but can be implemented in smaller organizations as well.

## SolarWinds Papertrail

**SolarWinds Papertrail** is a hosted log management tool designed to help you collect and monitor logs from your servers, applications, databases, networking devices, syslog, cloud, and more.

## 10 best practises when logging and Necessity of Log Formatting

Within the last decade, the advancement of distributed systems has introduced new complexities in managing log data. As a result, log management has become a staple in modern IT operations, supporting a number of use cases including debugging, production monitoring, performance monitoring, support and troubleshooting. The 10 best practises when logging can be as follows:

### a) Setting up a Strategy. Don't log blindly

When developing your logging strategy, consider what is most important from your perspective and what value you want from your logs.

### b) Structure Our Log Data

In addition to developing a logging strategy, it's important to consider the format of your logs. Failing to understand effective logging formats makes it very difficult to identify and extract insights from your logs.

### **c) Separate and Centralize our Log Data**

Logs should always be automatically collected and shipped to a centralized location, separate from your production environment.

### **d) Practice End-to-end Logging**

In order to overcome common troubleshooting complexities and achieve a more holistic view of your application and systems, you should monitor and log across all system components.

### **e) Correlate Data Sources**

Correlating data enables you to quickly and confidently identify and understand events that are causing system malfunctions.

### **f) Using Unique Identifiers**

Unique identifiers can be useful for debugging, support and analytics. Identifiers allow you to track particular user sessions and pinpoint actions taken by individual users.

### **g) Adding Context**

When using logs as data, it's important to consider the context of each data point. Knowing a user clicked a button may not be as useful as knowing a user specifically clicked the "purchase" button.

### **h) Perform Real-time Monitoring**

Service disruptions can lead to a host of unfortunate outcomes, including unhappy customers, lost purchases and missing data. When production-level issues arise, a real-time monitoring solution can be crucial when every second counts.

### **i) Log in text format**

Log and look at the data manually in a file or the standard output then the planned logging will be more than fine.

## j) Use developer-friendly formats

The developer may be looking at the logs for troubleshooting or during debugging sessions. So the logs must be in developer friendly formats so that it's easier for them by simply looking at the logs.

**Preventing errors from ever getting to a production site** is often the most efficient and cost effective answer for the necessity of **log formatting**. Keeping an eye on the logs for all your applications will help ensure your end user has a better experience, and your hardware/applications are performing their best. Proper log formatting allows logs to be machine-readable and easily parsed so that it becomes easier for the developing side as well.

**Create a file in your system. Whenever someone performs some action (read, write, execute) on that file, the event should be logged somewhere.**

Initially we create a file named **logfile** with,

```
$ sudo touch logfile
```

Give the permission to read, write and execute to the file with,,

```
$ sudo chmod +777 logfile
```

```
lostinserver@lostinserver:~$ sudo touch logfile
lostinserver@lostinserver:~$ sudo chmod +777 logfile
lostinserver@lostinserver:~$
```

Next we install the **iWatch** in our system.

**iWatch** is a realtime filesystem monitoring program, based on inotify, a file change notification system in the Linux kernel.

**\$ sudo apt install iwatch**

```
lostinserver@lostinserver:~$ sudo apt install iwatch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libevent-perl liblinux-inotify2-perl
  libmail-sendmail-perl libsys-hostname-long-perl libxml-libxml-perl
  libxml-namespacesupport-perl libxml-sax-base-perl libxml-sax-expat-perl
  libxml-sax-perl libxml-simpleobject-libxml-perl postfix
Suggested packages:
  sendxmpp yowsup-cli libxml-sax-expatxs-perl procmail postfix-mysql
  postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite
  sasl2-bin | dovecot-common resolvconf postfix-cdb postfix-doc
The following NEW packages will be installed:
  iwatch libcommon-sense-perl libevent-perl liblinux-inotify2-perl
  libmail-sendmail-perl libsys-hostname-long-perl libxml-libxml-perl
  libxml-namespacesupport-perl libxml-sax-base-perl libxml-sax-expat-perl
  libxml-sax-perl libxml-simpleobject-libxml-perl postfix
0 upgraded, 13 newly installed, 0 to remove and 98 not upgraded.
Need to get 1,971 kB of archives.
After this operation, 6,766 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 postfix amd64
3.4.13-0ubuntu1.2 [1,201 kB]
```

Now to monitor the file,

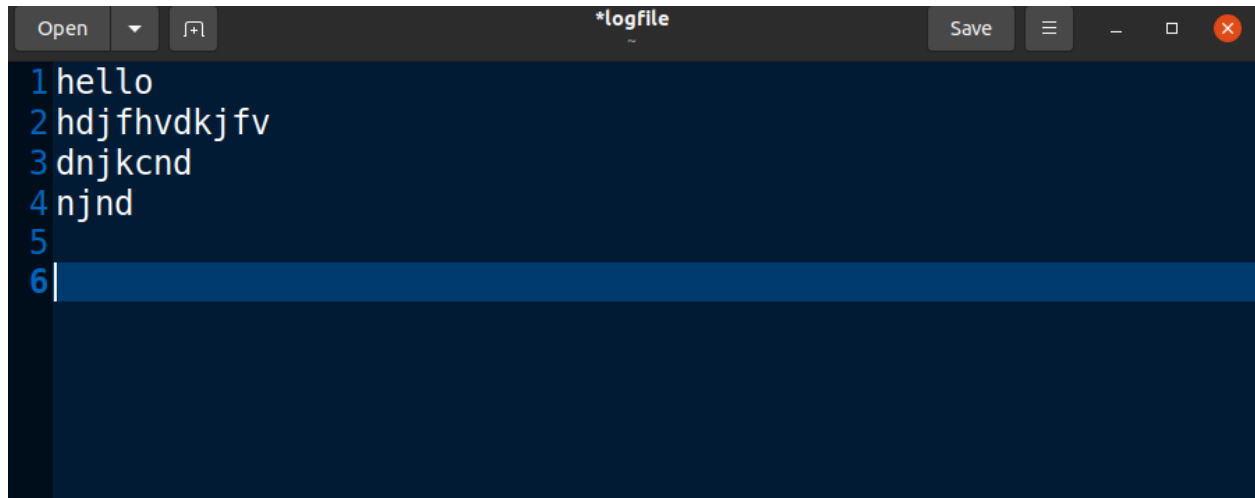
**\$ iwatch -e all\_events logfile &> finallogfile.txt**

Here, **all\_events** flag monitors all the events to file : write,read,execute,etc.

```
lostinserver@lostinserver:~$ iwatch -e all_events logfile &> finallogfile.txt
```

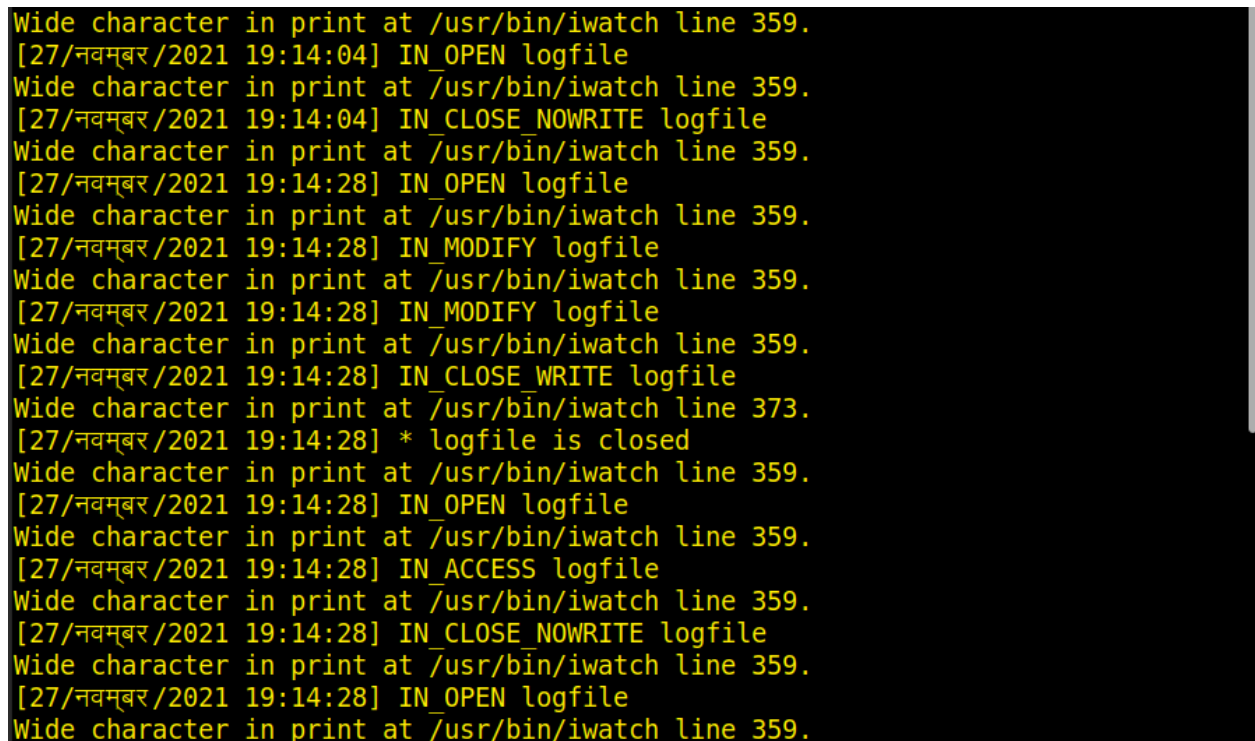
We can see here that no logs are printed.

Now, we open the file and make some changes in the file,



```
Open [icon] *logfile Save [icon] [icon] [icon]
1 hello
2 hdjfhvdkjfv
3 dnjkcnd
4 njnd
5
6 |
```

Now, in our terminal we can see the events logged in there.



```
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:04] IN_OPEN logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:04] IN_CLOSE_NOWRITE logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_OPEN logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_MODIFY logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_MODIFY logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_CLOSE_WRITE logfile
Wide character in print at /usr/bin/iwatch line 373.
[27/नवम्बर/2021 19:14:28] * logfile is closed
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_OPEN logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_ACCESS logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_CLOSE_NOWRITE logfile
Wide character in print at /usr/bin/iwatch line 359.
[27/नवम्बर/2021 19:14:28] IN_OPEN logfile
Wide character in print at /usr/bin/iwatch line 359.
```

Also as we have given command to save the log in **finallogfile.txt** , the logs are written to this file. If the file exists, it appends the data to it otherwise the file is newly created and then the data(logs) is written as,

```
154 Wide character in print at /usr/bin/iwatch line 359.
155 Wide character in print at /usr/bin/iwatch line 359.
156 [27/नवम्बर/2021 21:32:59] IN_OPEN logfile
157 [27/नवम्बर/2021 21:32:59] IN_ACCESS logfile
158 [27/नवम्बर/2021 21:32:59] IN_CLOSE_NOWRITE logfile
159 [27/नवम्बर/2021 21:33:00] IN_OPEN logfile
160 [27/नवम्बर/2021 21:33:00] IN_ACCESS logfile
161 [27/नवम्बर/2021 21:33:00] IN_CLOSE_NOWRITE logfile
162 [27/नवम्बर/2021 21:33:00] IN_OPEN logfile
163 [27/नवम्बर/2021 21:33:00] IN_ACCESS logfile
164 [27/नवम्बर/2021 21:33:00] IN_CLOSE_NOWRITE logfile
165 [27/नवम्बर/2021 21:33:01] IN_OPEN logfile
166 [27/नवम्बर/2021 21:33:01] IN_OPEN logfile
167 [27/नवम्बर/2021 21:33:01] IN_ACCESS logfile
168 [27/नवम्बर/2021 21:33:01] IN_CLOSE_NOWRITE logfile
169 [27/नवम्बर/2021 21:33:01] IN_OPEN logfile
170 [27/नवम्बर/2021 21:33:01] IN_ACCESS logfile
171 [27/नवम्बर/2021 21:33:01] IN_CLOSE_NOWRITE logfile
172 [27/नवम्बर/2021 21:33:01] IN_ACCESS logfile
173 [27/नवम्बर/2021 21:33:01] IN_CLOSE_NOWRITE logfile
```

We can see the opening,closing,modifying,etc. events in the above real time log.

Install logstash in your system. download a sample nginx log, parse the logs using logstash. The parsed output must contain the geographical information like country, state etc. that the request is originating from. save the parsed output to a file in your system.

[https://github.com/elastic/examples/blob/master/Common%20Data%20Formats/nginx\\_logs/nginx\\_logs](https://github.com/elastic/examples/blob/master/Common%20Data%20Formats/nginx_logs/nginx_logs)

For installing logstash in the system,

```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

```
$ sudo apt-get install apt-transport-https
```

```
$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

```
$ sudo apt-get update && sudo apt-get install logstash
```

```
lostinserver@lostinserver:~$ sudo apt-get update && sudo apt-get install logstash
Hit:1 http://np.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://np.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://np.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.6 kB]
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Get:6 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,346 kB]
Hit:7 http://ppa.launchpad.net/linuxuprising/java/ubuntu focal InRelease
Hit:8 https://dl.google.com/linux/chrome/deb stable InRelease
Get:9 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [64.9 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:11 https://deb.nodesource.com/node_14.x focal InRelease
Get:12 http://ppa.launchpad.net/pipewire-debian/pipewire-upstream/ubuntu focal InRelease [24.4 kB]
Get:14 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [84.6 kB]
```



Now, download the nginx logs using,

**\$wget**

[https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/nginx\\_logs/nginx\\_logs](https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/nginx_logs/nginx_logs)

```
lostinservice@lostinservice:~$ wget https://raw.githubusercontent.com/elastic/exam
ples/master/Common%20Data%20Formats/nginx_logs/nginx_logs
--2021-11-27 22:20:09-- https://raw.githubusercontent.com/elastic/examples/mast
er/Common%20Data%20Formats/nginx_logs/nginx_logs
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.1
33, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.
133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6991577 (6.7M) [text/plain]
Saving to: 'nginx_logs'

nginx_logs          100%[=====>]    6.67M   728KB/s   in 23s

2021-11-27 22:20:38 (297 KB/s) - 'nginx_logs' saved [6991577/6991577]

lostinservice@lostinservice:~$
```