How to Install ELK Stack (Elasticsearch, Logstash, and Kibana) on Ubuntu

Introduction

The ELK stack is a set of applications for retrieving and managing log files.

It is a collection of three open-source tools, **Elasticsearch**, **Kibana**, and **Logstash**. The stack can be further upgraded with **Beats**, a lightweight plugin for aggregating data from different data streams.

In this tutorial, learn how to install the ELK software stack on Ubuntu 18.04 / 20.04.

Prerequisites

- A Linux system running Ubuntu 20.04 or 18.04
- Access to a terminal window/command line (**Search** > **Terminal**)
- A user account with **sudo** or **root** privileges

Java version 8 or 11 (required for Logstash)   •

Step 1: Install Dependencies
Install Java
The ELK stack requires Java 8 to be installed. Some components are compatible with Java 9, but not Logstash.

Note: To check your Java version, enter the following:
```
java -version
```
The output you are looking for is **1.8.x_xxx.** That would indicate that Java 8 is installed.
If you already have Java 8 installed, skip to Install Nginx.

1. If you don't have Java 8 installed, install it by opening a terminal window and entering the following:

```
sudo apt-get install openjdk-8-jdk
```

2. If prompted, type **y** and hit **Enter** for the process to finish.

```
dejan@dejan-phoenixnap:~$ sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
  openjdk-8-demo openjdk-8-source visualvm icedtea-8-plugin
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk
  openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 21 newly installed, 0 to remove and 80 not upgraded.
Need to get 43,3 MB of archives.
After this operation, 160 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Install Nginx
Nginx works as a web server and proxy server. It's used to configure password-controlled access to the Kibana dashboard.

1. Install Nginx by entering the following:

```
sudo apt-get install nginx
```

2. If prompted, type **y** and hit **Enter** for the process to finish.

```
dejan@dejan-phoenixnap:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream nginx nginx-common nginx-core
0 upgraded, 7 newly installed, 0 to remove and 80 not upgraded.
Need to get 602 kB of archives.
After this operation, 2134 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Note: For additional tutorials, follow our guides on installing Nginx on Ubuntu and setting up Nginx reverse proxy For Kibana

Step 2: Add Elastic Repository

Elastic repositories enable access to all the open-source software in the ELK stack. To add them, start by importing the GPG key.

1. Enter the following into a terminal window to import the PGP key for Elastic:

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

2. The system should respond with **OK**, as seen in the image below.

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

3. Next, install the **apt-transport-https** package:

sudo apt-get install apt-transport-https

4. Add the Elastic repository to your system's repository list:

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

```
dejan@dejan-phoenixnap:~$ sudo echo "deb https://artifacts.elastic.co/packages/7.x/apt
stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
dejan@dejan-phoenixnap:~$ sudo apt-get update
Get:1 http://repo.mysql.com/apt/ubuntu focal InRelease [12,2 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://rs.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://rs.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://rs.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [10,4 kB]
```

Step 3: Install Elasticsearch

1. Prior to installing Elasticsearch, update the repositories by entering:

```
sudo apt-get update
```

2. Install Elasticsearch with the following command:

```
sudo apt-get install elasticsearch
```


Configure Elasticsearch

1. Elasticsearch uses a configuration file to control how it behaves. Open the configuration file for editing in a text editor of your choice. We will be using nano:

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

2. You should see a configuration file with several different entries and descriptions. Scroll down to find the following entries:

```
#network.host: 192.168.0.1#http.port: 9200#
```

3. Uncomment the lines by deleting the **hash (#) sign** at the beginning of both lines and replace **192.168.0.1** with **localhost**.

It should read:

```
network.host: localhosthttp.port: 9200
```


4. Just below, find the *Discovery* section. We are adding one more line, as we are configuring a single node cluster:

```
discovery.type: single-node
```

.For further details, see the image below

```
# ---------------------------------- Network -----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: localhost    ◄──────────────
#
# Set a custom port for HTTP:
#
http.port: 9200    ◄──────────────
#
# For more information, consult the network module documentation.
#
# ---------------------------------- Discovery ---------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
# Single node Elastic stack
discovery.type: single-node    ◄──────────────
# For more information, consult the discovery and cluster formation module documentati>
#
```

By default, **JVM heap size** is set at 1GB. We recommend setting it to 5.
no more than half the size of your total memory. Open the following file
for editing:

```
sudo nano /etc/elasticsearch/jvm.options
```

Find the lines starting with **-Xms** and **-Xmx**. In the example below, the 6.
maximum (**-Xmx**) and minimum (**-Xms**) size is set to 512MB.

```
###########################################################
## IMPORTANT: JVM heap size
###########################################################
##
## You should always set the min and max JVM heap
## size to the same value. For example, to set
## the heap to 4 GB, set:
##
## -Xms4g
## -Xmx4g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html
## for more information
##
###########################################################

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms512m
-Xmx512m
```

Start Elasticsearch

:Start the Elasticsearch service by running a **systemctl** command .1

```
sudo systemctl start elasticsearch.service
```

It may take some time for the system to start the service. There will be no .output if successful

:Enable Elasticsearch to start on boot .2

```
sudo systemctl enable elasticsearch.service
```

```
dejan@dejan-phoenixnap:~$ sudo systemctl enable elasticsearch.service
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service →/li
b/systemd/system/elasticsearch.service.
dejan@dejan-phoenixnap:~$
```

Test Elasticsearch

:Use the **curl** command to test your configuration. Enter the following

```
curl -X GET "localhost:9200"
```

The name of your system should display, and **elasticsearch** for the cluster name. This indicates that Elasticsearch is functional and is .listening on **port 9200**

```
dejan@dejan-phoenixnap:~$ curl -X GET "localhost:9200"
{
  "name" : "dejan-phoenixnap",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Y-eObJS85LaWcZDY92n69g",
  "version" : {
    "number" : "7.7.1",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ad56dce891c901a492bb1ee393f12dfff473a423",
    "build_date" : "2020-05-28T16:30:01.040088Z",
    "build_snapshot" : false,
    "lucene_version" : "8.5.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
dejan@dejan-phoenixnap:~$
```

Step 4: Install Kibana

It is recommended to install Kibana next. Kibana is a graphical user .interface for parsing and interpreting collected log files

:Run the following command to install Kibana .1

```
sudo apt-get install kibana
```

.Allow the process to finish. Once finished, it's time to configure Kibana .2

Configure Kibana

:Next, open the **kibana.yml** configuration file for editing .1
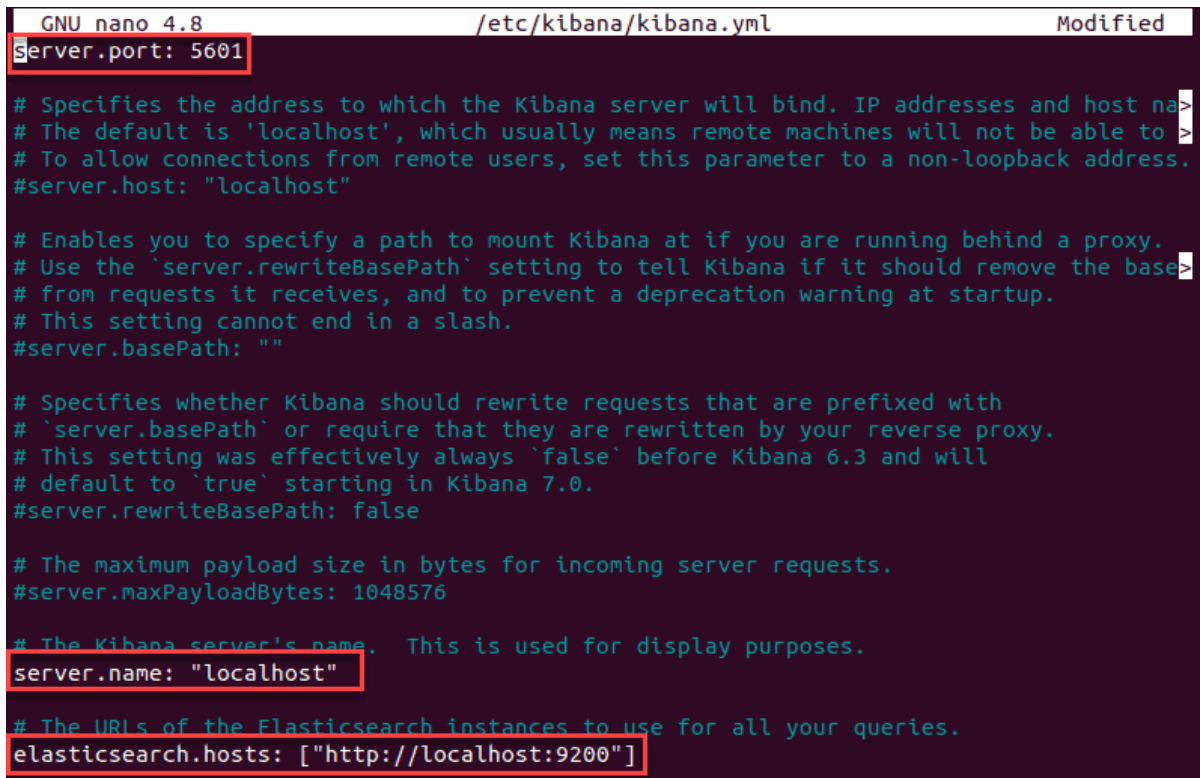
```
sudo nano /etc/kibana/kibana.yml
```

Delete the **#** sign at the beginning of the following lines to activate .2
:them

```
server.port: 5601#server.host: "your-#
hostname"#elasticsearch.hosts: ["http://localhost:9200"]
```

:The above-mentioned lines should look as follows

```
server.port: 5601server.host: "localhost"elasticsearch.hosts:
["http://localhost:9200"]
```

.Save the file (Ctrl+**o**) and exit (Ctrl+ **x**) .3

```
  GNU nano 4.8                    /etc/kibana/kibana.yml                    Modified
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host na>
# The default is 'localhost', which usually means remote machines will not be able to >
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the base>
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
server.name: "localhost"

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://localhost:9200"]
```

Note: This configuration allows traffic from the same system
Elasticstack is configured on. You can set the **server.host** value to the
.address of a remote server
Start and Enable Kibana

:Start the Kibana service .1

```
sudo systemctl start kibana
```

.There is no output if the service starts successfully

:Next, configure Kibana to launch at boot .2

```
sudo systemctl enable kibana
```

```
dejan@dejan-phoenixnap:~$ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/system
d-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service →/etc/syste
md/system/kibana.service.
dejan@dejan-phoenixnap:~$
```

Allow Traffic on Port 5601

If the UFW firewall is enabled on your Ubuntu system, you need to **allow** **traffic on port 5601** to access the Kibana dashboard.

:In a terminal window, run the following command

```
sudo ufw allow 5601/tcp
```
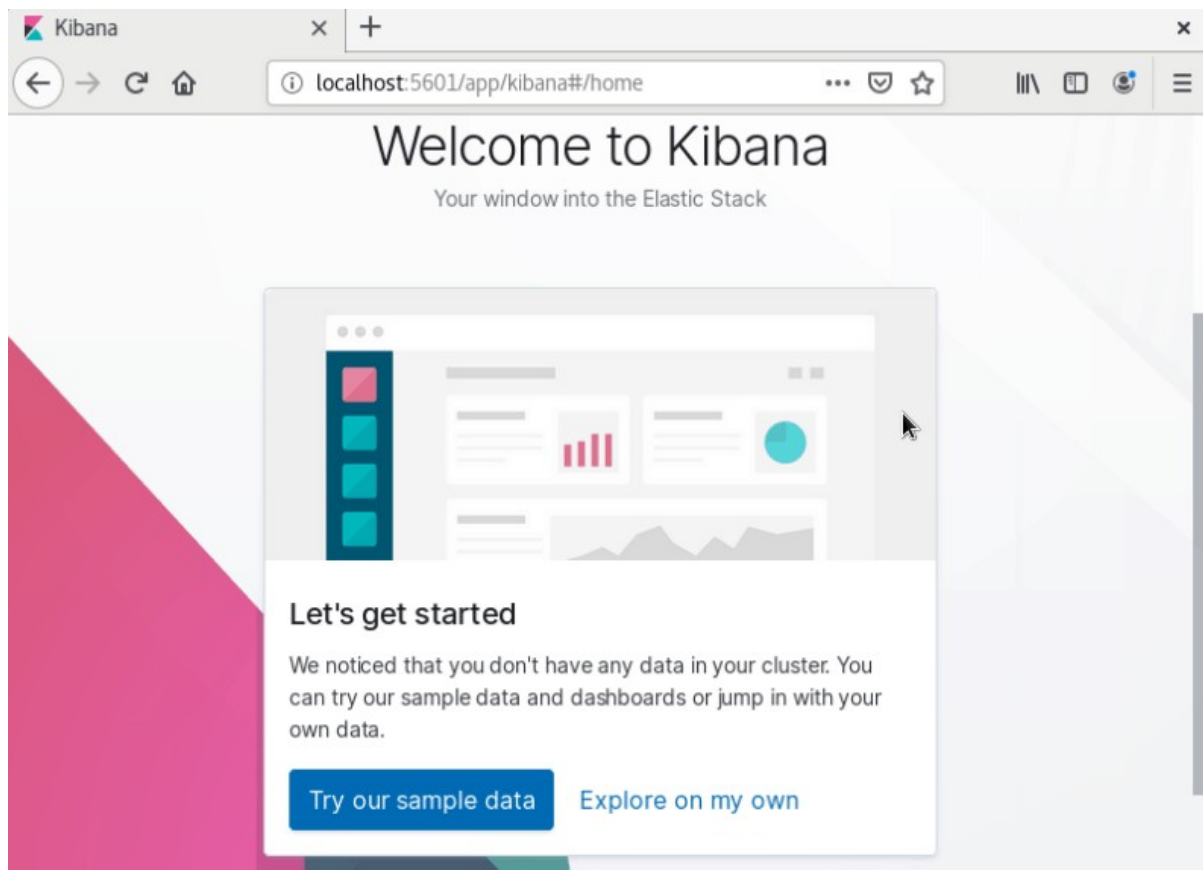
:The following output should display

```
dejan@dejan-phoenixnap:~$ sudo ufw allow 5601/tcp
Rules updated
Rules updated (v6)
dejan@dejan-phoenixnap:~$
```

Test Kibana

To access Kibana, open a web browser and browse to the following :address

http://localhost:5601

.The Kibana dashboard loads

If you receive a *"Kibana server not ready yet"* error, check if the Elasticsearch and Kibana services are active.

Note: Check out our in-depth Kibana tutorial to learn everything you need to know visualization and data query.

Step 5: Install Logstash

Logstash is a tool that collects data from different sources. The data it collects is parsed by Kibana and stored in Elasticsearch.

Install Logstash by running the following command:

```
sudo apt-get install logstash
```

Start and Enable Logstash

1. Start the Logstash service:

```
sudo systemctl start logstash
```

2. Enable the Logstash service:

```
sudo systemctl enable logstash
```

3. To check the status of the service, run the following command:

```
sudo systemctl status logstash
```

```
dejan@dejan-phoenixnap:~$ systemctl status logstash
● logstash.service - logstash
    Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor pre>
    Active: active (running) since Mon 2020-07-06 11:50:34 CEST; 23min ago
  Main PID: 606 (java)
     Tasks: 31 (limit: 2319)
    Memory: 538.7M
    CGroup: /system.slice/logstash.service
            └─606 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInit>
```

Configure Logstash

Logstash is a highly customizable part of the ELK stack. Once installed, configure its **INPUT**, **FILTERS**, and **OUTPUT** pipelines according to your own individual use case.

All custom Logstash configuration files are stored in **.//etc/logstash/conf.d**
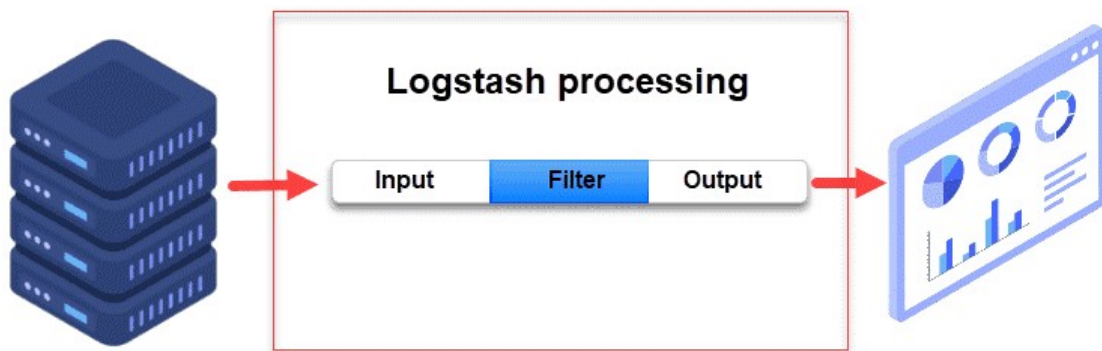
Note: Consider the following Logstash configuration examples and adjust the configuration for your needs.

Step 6: Install Filebeat

Filebeat is a lightweight plugin used to collect and ship log files. It is the most commonly used Beats module. One of Filebeat's major advantages is that it slows down its pace if the Logstash service is overwhelmed with data.

Install Filebeat by running the following command:

```
sudo apt-get install filebeat
```

Let the installation complete.

Note: Make sure that the Kibana service is up and running during the installation and configuration procedure.

Configure Filebeat

Filebeat, by default, sends data to Elasticsearch. Filebeat can also be configured to send event data to Logstash.

1. To configure this, edit the **filebeat.yml** configuration file:

```
sudo nano /etc/filebeat/filebeat.yml
```

2. Under the *Elasticsearch output* section, comment out the following lines:

```
# output.elasticsearch:
  # Array of hosts to connect to.
  # hosts: ["localhost:9200"]
```
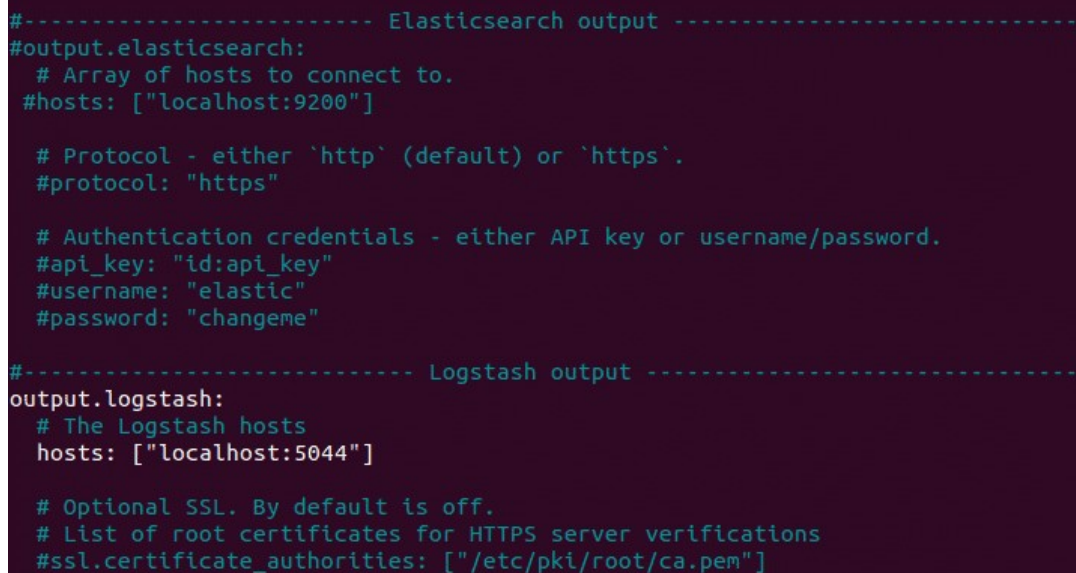
3. Under the *Logstash output* section, **remove the hash sign** (**#**) in the following two lines:

```
# output.logstash
  # hosts: ["localhost:5044"]
```

It should look like this:

```
output.logstash
  hosts: ["localhost:5044"]
```

For further details, see the image below.

4. Next, enable the **Filebeat system** module, which will examine local system logs:

```
sudo filebeat modules enable system
```

The output should read **Enabled system**.

5. Next, load the index template:

```
sudo filebeat setup --index-management -E
output.logstash.enabled=false -E
''output.elasticsearch.hosts=["localhost:9200"]
```

The system will do some work, scanning your system and connecting to your Kibana dashboard.

```
dejan@dejan-phoenixnap:~$ sudo filebeat setup --index-management -E output.logstash.ena
bled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite:true` for enabling.

Index setup finished.
```

Start and Enable Filebeat

Start and enable the Filebeat service:

```
sudo systemctl start filebeatsudo systemctl enable filebeat
```

Verify Elasticsearch Reception of Data

Finally, verify if Filebeat is shipping log files to Logstash for processing. Once processed, data is sent to Elasticsearch.

```
curl -XGET http://localhost:9200/_cat/indices?v
```

```
dejan@dejan-phoenixnap:~$ curl -XGET http://localhost:9200/_cat/indices?v
health status index                          uuid                    pri rep docs.coun
t docs.deleted store.size pri.store.size
green  open   .apm-custom-link               OuHTpMUlTseqvwzywmFgRg   1   0
0          0      208b           208b
yellow open   filebeat-7.7.1-2020.07.06-000001 kVZRXBamQUSVetxFi5ijjQ  1   1
0          0      208b           208b
green  open   .kibana_task_manager_1         raXydUuxS1aR5GJkkHOPhQ   1   0
5          6      64.9kb         64.9kb
green  open   .apm-agent-configuration       t_7itihaTEuX7FI-1NV5_Q   1   0
0          0      208b           208b
green  open   .kibana_1                      aU7h3QW0QMqQQGWGAHK3Kg   1   0
4          0      49.4kb         49.4kb
dejan@dejan-phoenixnap:~$
```

Note: For further details on health status indicators, please see Elastic's Cluster Health documentation.

Conclusion

Now you have a functional ELK stack installed on your Ubuntu system. We recommend defining your requirements and start adjusting ELK for your needs. This powerful monitoring tool can be customized for individual use cases.

Customize data streams with Logstash, use different Beats modules to gather various types of data, and utilize Kibana for easy browsing through log files.