ELK Installation Guide

ELK Installation

Prerequisites

- A Linux system running Ubuntu 20.04 or 18.04
- A user account with sudo or root previleges
- Java Version 8 or 11 (required for Logstash)

> 💡 ELK stack need to be the same version for it to function properly. For this guide, we will be using version 7.17.18.

Step 1: Install Dependencies

1. Installing the default JRE/JDK

a. Update the package index:

```
sudo apt update
```

c. Next, check if Java is already installed

```
java -version
```

d. If Java is not installed, you will see the following output:

```
pth@ubuntu-soc:~$ java -version

Command 'java' not found, but can be installed with:

sudo apt install default-jre              # version 2:1.11-72, or
sudo apt install openjdk-11-jre-headless  # version 11.0.20.1+1-0ubuntu1~20.04
sudo apt install openjdk-16-jre-headless  # version 16.0.1+9-1~20.04
sudo apt install openjdk-17-jre-headless  # version 17.0.8.1+1~us1-0ubuntu1~20.04
sudo apt install openjdk-8-jre-headless   # version 8u382-ga-1~20.04.1
sudo apt install openjdk-13-jre-headless  # version 13.0.7+5-0ubuntu1~20.04
```

e. Java Runtime Environment (JRE) from OpenJDK 11 will allow you to run almost all Java software. Java Development Kit (JDK) may be needed in addition to the JRE in order to compile and run some specific Java-based software. To install both JDK and JRE, execute the following command:

```
sudo apt install default-jdk
```

e. Verify the installation with the following commands:

```
java -version
```

```
javac -version
```

Similar output should be seen:

```
pth@ubuntu-soc:~$ java -version
openjdk version "11.0.22" 2024-01-16
OpenJDK Runtime Environment (build 11.0.22+7-post-Ubuntu-0ubuntu220.04.1)
OpenJDK 64-Bit Server VM (build 11.0.22+7-post-Ubuntu-0ubuntu220.04.1, mixed mode, sharing)
pth@ubuntu-soc:~$ javac -version
javac 11.0.22
```

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. Import PGP KEY

a. Enter the following command to import the PGP key for Elastic:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsear
ch | sudo apt-key add -
```

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

```
pth@ubuntu-soc:~$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
pth@ubuntu-soc:~$
```

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

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

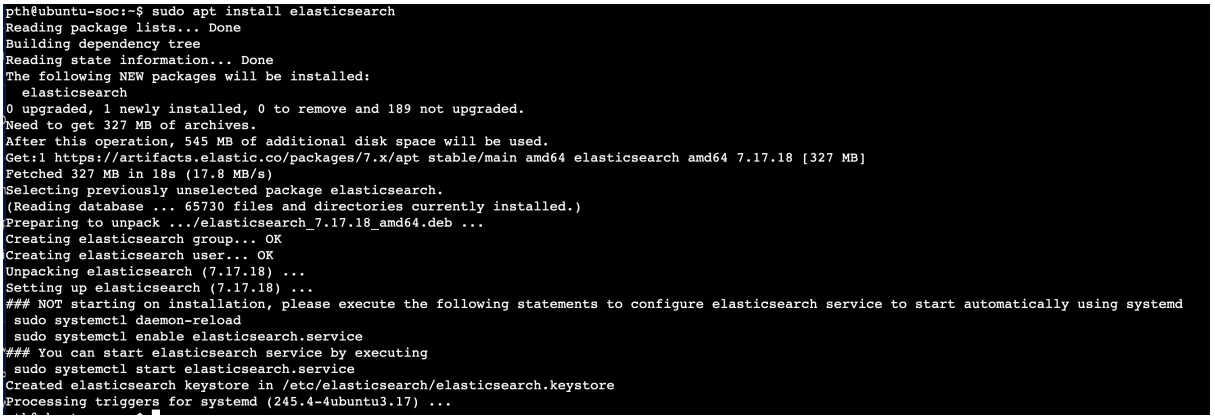
d. Update the repositories:

```
sudo apt update
```

Step 3: Install Elasticsearch

1. Install Elasticsearch with the following command:

```
sudo apt install elasticsearch
```



2.Configure Elasticsearch

a. Elasticsearch uses a configuration file to manage the different setup options. Open the configuration file for editing in a text editor of your choice.

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

> 💡 **Note:** Elasticsearch's configuration file is in YAML format, which means that we need to maintain the indentation format. Be sure that you do not add any extra spaces as you edit this file.

The `elasticsearch.yml` file provides configuration options for your cluster, node, paths, memory, network, discovery, and gateway. Most of these options are preconfigured in the file but you can change them according to your needs. For this installation, we will only adjust the settings for the network host.

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

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

c. Activate the lines by deleting the **hash** (`#`) **sign** at the beginning of both lines and replace `192.168.0.1` with the IP address of your Kibana server.

> 💡 `localhost` in this guide refers to the IP address of your Kibana server

d. Insert the following command under the `Discovery` section

```
discovery.seed_hosts: ["localhost". "127.0.0.1", "[0.0.0.
0]"]
```

The changes should look like this at the end:

```
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: "localhost"
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
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: ["localhost", "127.0.0.1", "[0.0.0.0]"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
```

e. By default, **JVM heap size** is set at 4GB. The recommended setting is no more than half the size of your total memory. Open the following file for editing:

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

f. Find the lines starting with `-Xms` and `-Xmx`. In the example below, the maximum (`-Xmx`) and minimum (`-Xms`) size is set to 512MB. Set it accordingly based on half the size of your total memory. Note that the value of `-Xms` and `-Xmx` **MUST** be the **SAME.**

3.Start Elasticsearch

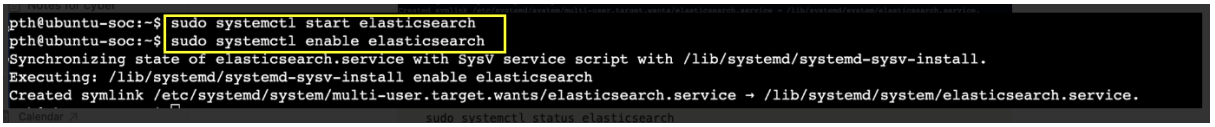
a. Start the Elasticsearch service by running a `systemctl` command:

```
sudo systemctl start elasticsearch
```

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

b. Enable Elasticsearch to start on boot:

```
sudo systemctl enable elasticsearc
```

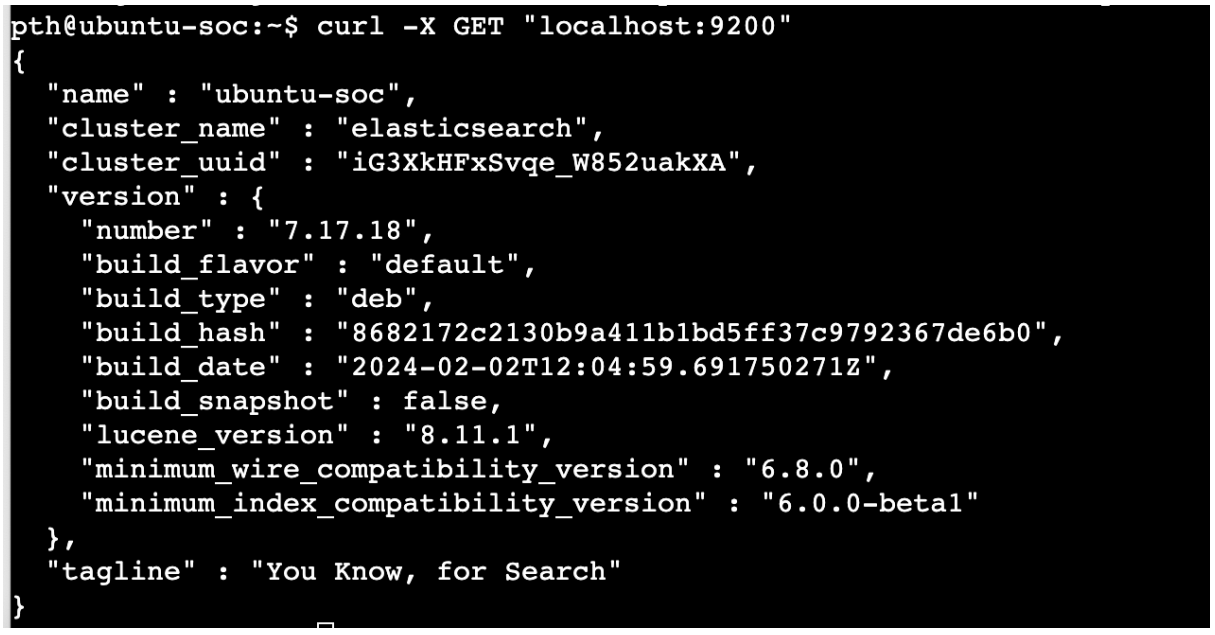


c. Check status of Elasticsearch :

```
sudo systemctl status elasticsearch
```

d. Test Elasticsearch

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



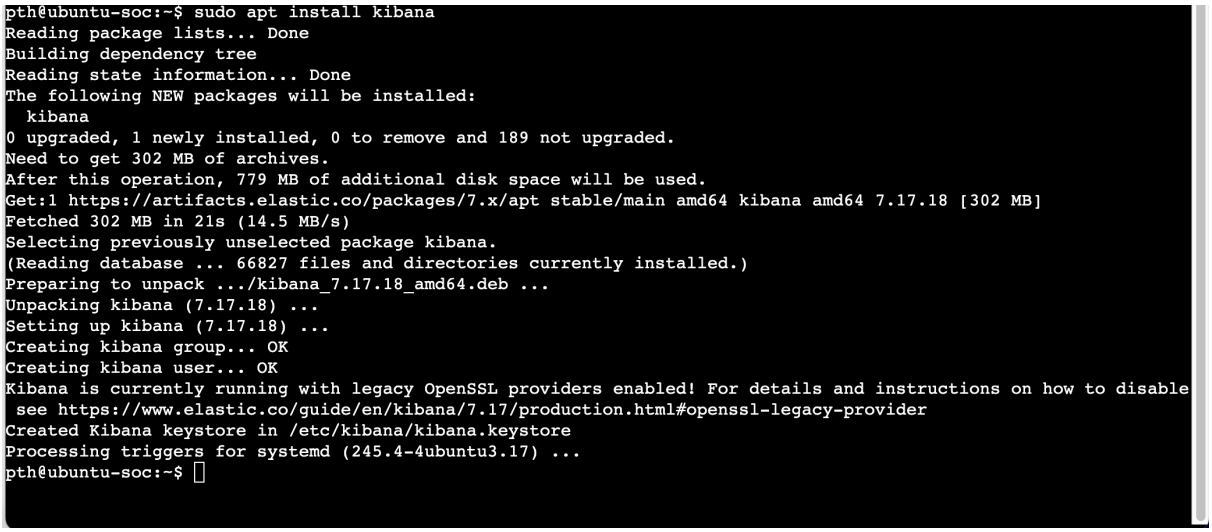
Step 4: Install Kibana

Kibana is a graphical user interface for parsing and interpreting collected log files.

1. Install Kibana with the following command:

```
sudo apt install kibana
```

a. Open the **kibana.yml** configuration file for editing:



2. Configure Kibana

```
sudo nano /etc/kibana/kibana.yml
```

a. Delete the `#` sign at the beginning of the following lines to activate them:

```
#server.port: 5601
```

```
#server.host: "localhost"
```

```
#elasticsearch.hosts: ["http://localhost:9200"]
```

The above-mentioned lines should look as follows:

```
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# 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 basePath
# 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

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 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"]
```

b. Save the file (Ctrl+ `S`) and exit (Ctrl+ `X`).

3. Start Kibana

a. Start Kibana service

```
sudo systemctl start kibana
```

There is no output if the service starts successfully.

b. Next, configure Kibana to launch at boot:

```
sudo systemctl enable kibana
```

```
pth@ubuntu-soc:~$ sudo systemctl start kibana
pth@ubuntu-soc:~$ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
```

c. Next, verify status of Kibana:

```
pth@ubuntu-soc:~$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-03-08 15:19:53 UTC; 30s ago
       Docs: https://www.elastic.co
   Main PID: 3012 (node)
      Tasks: 11 (limit: 1131)
     Memory: 270.2M
     CGroup: /system.slice/kibana.service
             └─3012 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest
```

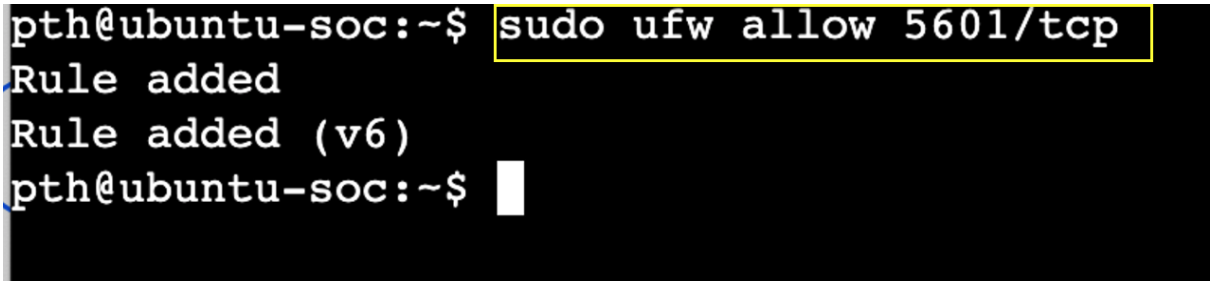
4. Allow Traffic on Port 5601

We need to **allow traffic on port 5601** to access the Kibana dashboard.

Run the following command:

```
sudo ufw allow 5601/tcp
```

The following should display:

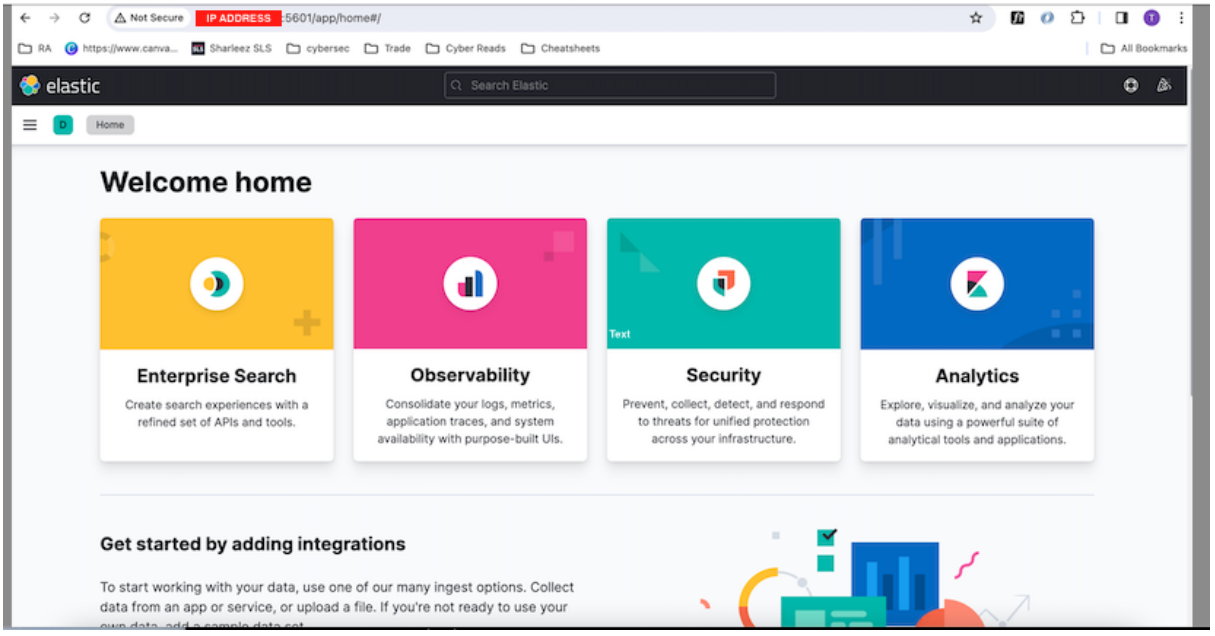


5. Test Kibana

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

```
http://localhost:5601
```

The Kibana dashboard should loads as seen below:



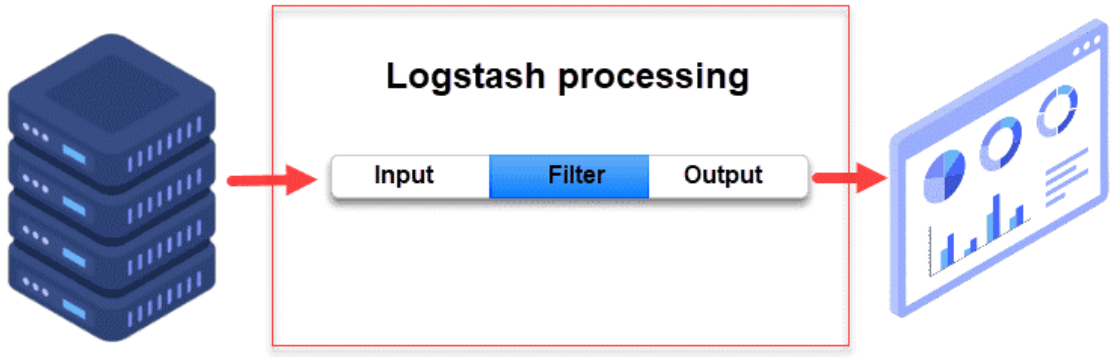
With the above, we have successfully installed and configured Kibana and Elasticsearch.

Step 5: Install Logstash

1. Install Logstash with the following command:

```
sudo apt install logstash
```

Logstash is like a pipeline which takes in data at one end, processes it and sends it out to its destination (Elasticsearch). A Logstash pipeline has two required elements, `input` and `output`, and one optional element, `filter`. The input plugins consume data from a source, the filter plugins process the data, and the output plugins write the data to a destination.



2. Configure Logstash

a. Create a configuration file called `02-beats-input.conf` where you will set up your Filebeat input:

```
sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

b. Insert the following configuration. This specifies a `beats` input that will listen on TCP port `5044`.

/etc/logstash/conf.d/02-beats-input.conf

```
input {
  beats {
    port => 5044
  }
}
```

Save and close the file.

c. Next, create a configuration file called `30-elasticsearch-output.conf`:

```
sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

d. Insert the following `output` configuration. This output configures Logstash to store the Beats data in Elasticsearch, which is running at `localhost:9200`, in an index named after the Beat used. The Beat used in this guide is Filebeat:

/etc/logstash/conf.d/30-elasticsearch-output.conf

```
output {
  if [@metadata][pipeline] {
    elasticsearch {
    hosts => ["localhost:9200"]
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-
%{+YYYY.MM.dd}"
    pipeline => "%{[@metadata][pipeline]}"
    }
  } else {
    elasticsearch {
    hosts => ["localhost:9200"]
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-
%{+YYYY.MM.dd}"
    }
  }
}
```

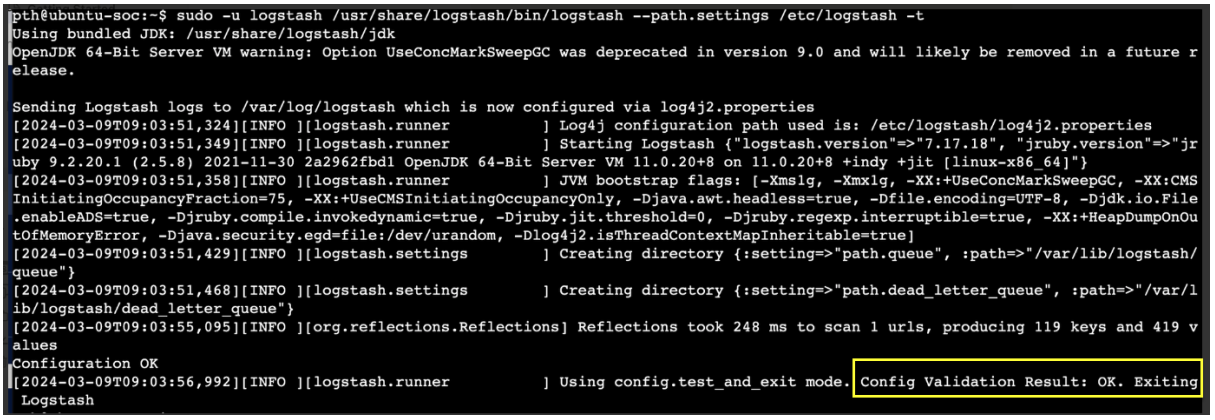
Save and close the file.

3. Test your Logstash configuration with this command:

```
sudo -u logstash /usr/share/logstash/bin/logstash --path.se
ttings /etc/logstash -t
```

If there are no syntax errors, your output will display `Config Validation Result: OK. Exiting Logstash` as seen below. If you don't see this, check for any errors noted in your output config file and amend accordingly.

Note that you will receive warnings from OpenJDK, but they should not cause any problems and can be ignored.



4. Start Logstash

a. If your configuration test is successful, start the Logstash service

```
sudo systemctl start logstash
```

There is no output if the service starts successfully.

b. Next, configure Logstash to launch at boot:

```
sudo systemctl enable logstash
```

c. Next, check status of Logstash :

Now that Logstash is running correctly and is fully configured, we would have successfully set up a ELK stack.

Step 6: Enhance Elasticsearch security

By default, the login page is not enabled for elasticsearch. Thus we need to enhance the security by enabling the login page for elasticsearch.

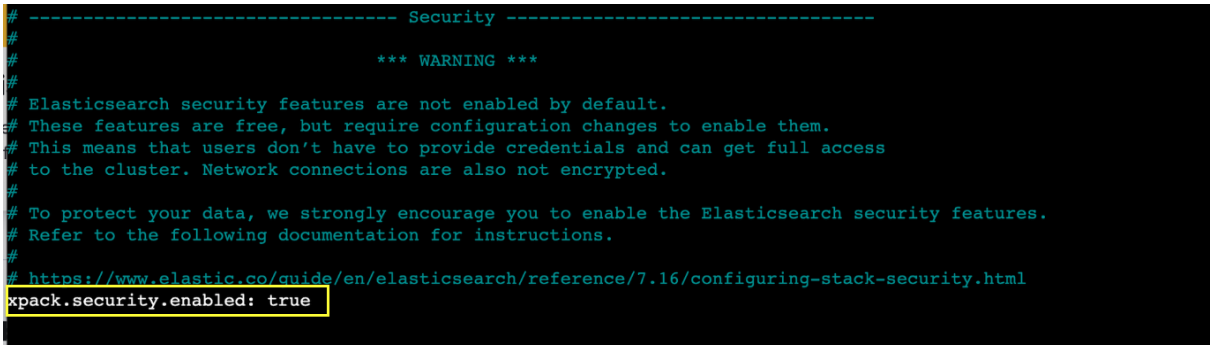
1. Enable the security for Elasticsearch

a. Open the elasticsearch yml file with the following command:

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

b. Insert the following line at the security section:

```
xpack.security.enabled: true
```



c. Restart Elasticsearch service to update the change in security setting:

```
sudo systemctl restart elasticsearch
```

d. Set up passwords by using the below command:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwor
```

You will be prompted to set passwords for various systems and enter the credentials accordingly:

```
pth@ubuntu-soc:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beat
s_system,remote_monitoring_user.
You will be prompted to enter passwords as the process progresses.
Please confirm that you would like to continue [y/N]y


Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Changed password for user [apm_system]
Changed password for user [kibana_system]
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
```

> 💡 The password will be required for future use so please keep a record of the credentials

e. Open the kibana yml file with the following command:

```
sudo nano /etc/kibana/kibana.yml
```

f. Set the value of `elasticsearch.username` to `“kibana_system”`

Set the value of `elasticsearch.password` to `“<password>”` (the value set in step d)

```
# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "kibana_system"
elasticsearch.password: "<password>"
# Kibana can also authenticate to Elasticsearch via "service account tokens".
# If may use this token instead of a username/password.
# elasticsearch.serviceAccountToken: "my_token"
```

e. Add password for the kibana_system user to the Kibana keystore:

```
sudo /usr/share/kibana/bin/kibana-keystore add elasticsearch.
```

```
pth@ubuntu-soc:~$ sudo /usr/share/kibana/bin/kibana-keystore add elasticsearch.password
Enter value for elasticsearch.password: *********
```

When prompted, enter the password set in step d above.

> 💡 By default, the kibana keystore is already created. If its not, create the keystore with the following command:
>
> `sudo /usr/share/kibana/bin/kibana-keystore create`

2. Enable Transport SSL on Elasticsearch

After setting the passwords, we need to configure the Transport Layer Security (TLS). to ensure that a malicious node cannot join the cluster and exchange data with other nodes. TLS enhances the security of communication between the nodes.

a. Open the elasticsearch.yml configuration file again and add the following lines:

```
xpack.security.transport.ssl.enabled: true
```

```
# ---------------------------------- Security ----------------------------------
#
#                                *** WARNING ***
#
# Elasticsearch security features are not enabled by default.
# These features are free, but require configuration changes to enable them.
# This means that users don't have to provide credentials and can get full access
# to the cluster. Network connections are also not encrypted.
#
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.
# Refer to the following documentation for instructions.
#
# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-stack-security.html
xpack.security.enabled: true
xpack.security.transport.ssl.enabled: true
```

b. Next, move to the '*/usr/share/elasticsearch/bin*' directory to generate the certificates used for encrypting internal communications between the services or clusters using the following command:

```
cd /usr/share/elasticsearch/bin
./elasticsearch-certutil ca
```

💡 press 'enter' when prompted for inputs

c. This creates a certificate authority that contains the public certificate and private key used to sign the certificates:

```
pth@ubuntu-soc:/usr/share/elasticsearch/bin$ ls ..
NOTICE.txt  README.asciidoc  bin  elastic-stack-ca.p12  jdk  lib  modules  plugins
```

d. Run the following command to generate a certificate and private key:

```
./bin/elasticsearch-certutil cert -ca elastic-stack-ca.p12
```

💡 press 'enter' when prompted for input

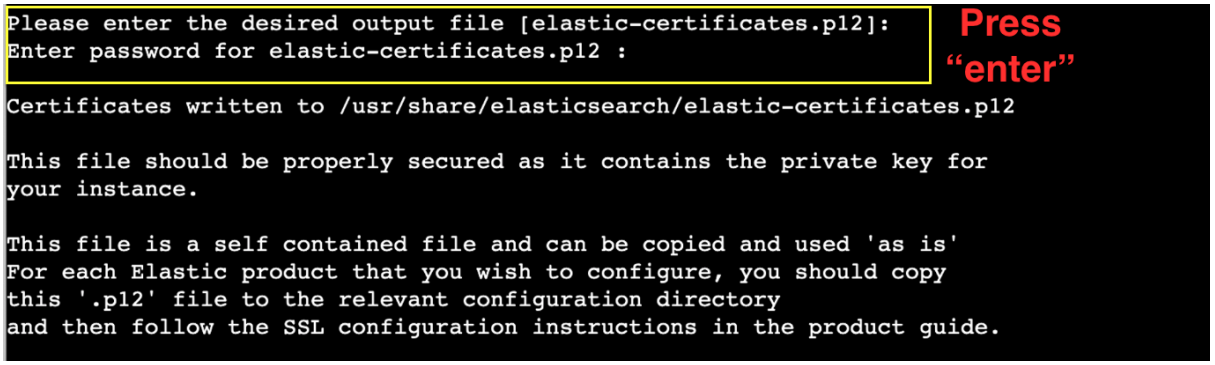
```
pth@ubuntu-soc:~$ cd /usr/share/elasticsearch/bin
pth@ubuntu-soc:/usr/share/elasticsearch/bin$ ./elasticsearch-certutil ca
./elasticsearch-env: line 87: /etc/default/elasticsearch: Permission denied
pth@ubuntu-soc:/usr/share/elasticsearch/bin$ sudo !!
sudo ./elasticsearch-certutil ca
This tool assists you in the generation of X.509 certificates and certificate
signing requests for use with SSL/TLS in the Elastic stack.

The 'ca' mode generates a new 'certificate authority'
This will create a new X.509 certificate and private key that can be used
to sign certificate when running in 'cert' mode.

Use the 'ca-dn' option if you wish to configure the 'distinguished name'
of the certificate authority

By default the 'ca' mode produces a single PKCS#12 output file which holds:
    * The CA certificate
    * The CA's private key

If you elect to generate PEM format certificates (the -pem option), then the output will
be a zip file containing individual files for the CA certificate and private key
```



e. Now create a directory named 'certs' in the elasticsearch directory and copy the certificates into the 'certs' directory (need to switch user to root to be able to perform the following actions):

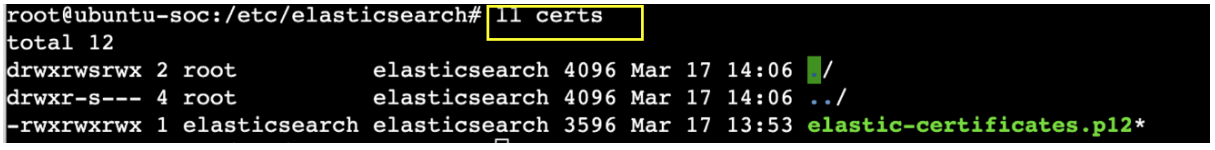
```
cd /etc/elasticsearch
```

```
mkdir certs
```

```
mv /usr/share/elasticsearch/elastic-certificates.p12 /etc/ela
```

f. Check that the certificates have the right user permissions.

```
ll certs
```



If the results are different from the above, modify user permissions and ownership to allow Elasticsearch access to it:

```
sudo chmod 777 -R certs/
```

```
sudo chown elasticsearch: certs/elastic-certificates.p12
```

g. Open the Elasticsearch configuration file again and add the following to enable internode communication and provide access to the node's certificate.

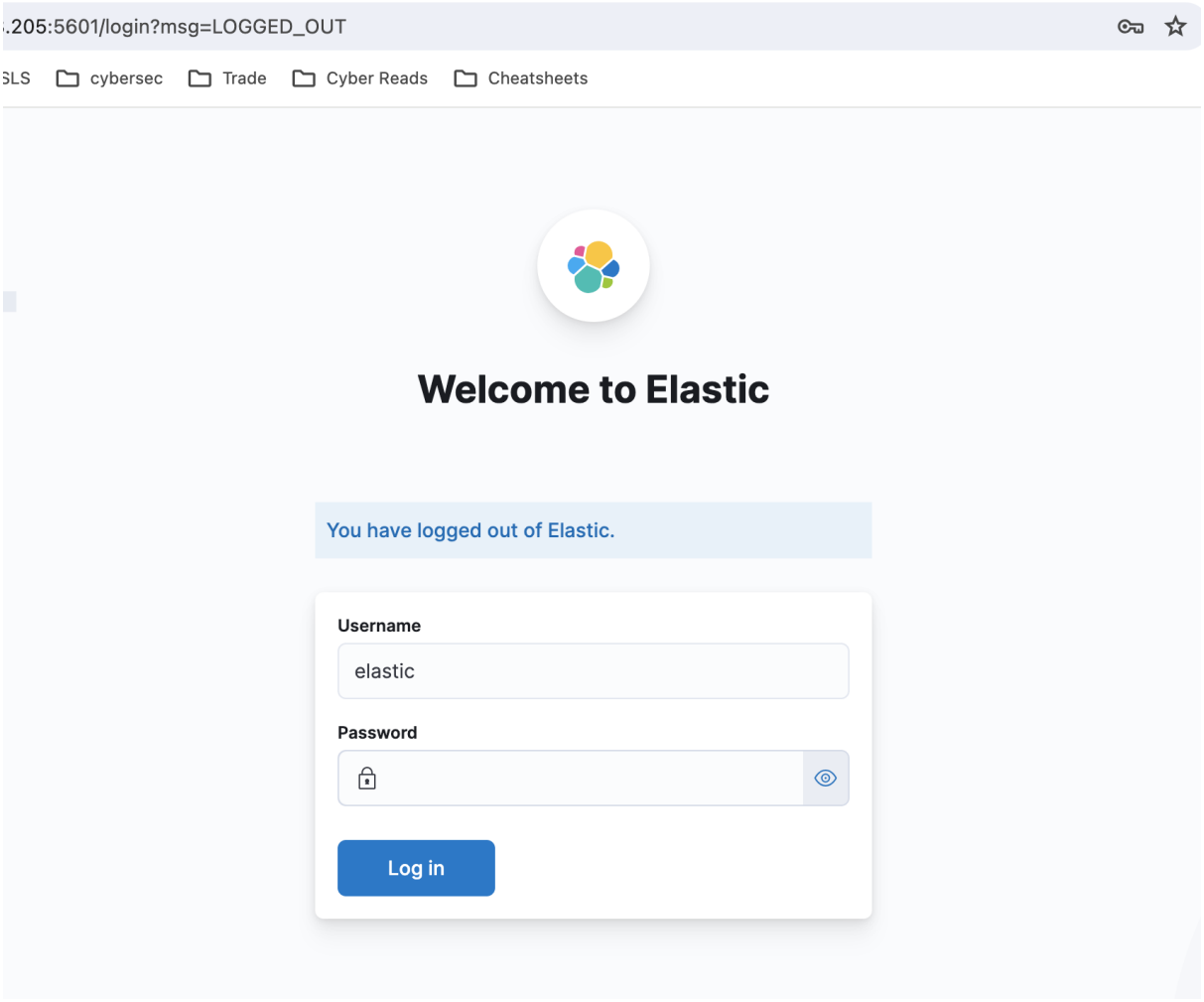
```
xpack.security.transport.ssl.verification_mode: certificate
```

```
xpack.security.transport.ssl.keystore.path: certs/elastic-cer
```

```
xpack.security.transport.ssl.truststore.path: certs/elastic-c
```

```
# ---------------------------------- Security ----------------------------------
#
#                                *** WARNING ***
#
# Elasticsearch security features are not enabled by default.
# These features are free, but require configuration changes to enable them.
# This means that users don't have to provide credentials and can get full access
# to the cluster. Network connections are also not encrypted.
#
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.
# Refer to the following documentation for instructions.
#
# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-stack-security.html
xpack.security.enabled: true
xpack.security.http.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: certs/elastic-certificates.p12
xpack.security.transport.ssl.truststore.path: certs/elastic-certificates.p12
```

h. Open your browser and go to localhost:5601. This time, you should see the below if the xpack was setup properly. Log into Kibana as the elastic user. Password will be the one created in step d.



At this point, we would have successfully set up the basic security feature for our ELK stack.

Next, we need to set up the Secure Sockets Layer (SSL) connection on Logstash.

3. Set Up SSL on Logstash

In order for Logstash to connect to Elasticsearch now, it must have a copy of the certificate as Elasticsearch to pass the security validation.

a. Now create a sub-directory named ‘certs’ in the logstash config directory and copy the certificates into the ‘certs’ directory (need to switch user to root to be able to perform the following actions):

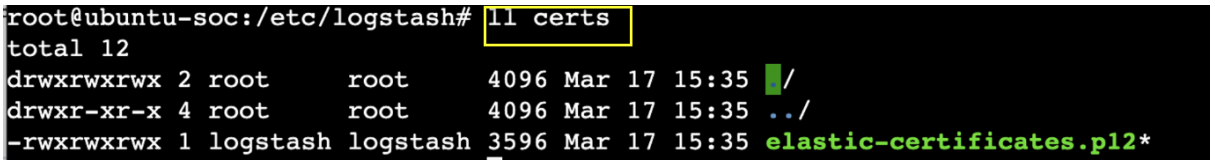
```
cd /etc/logstash
```

```
mkdir certs
```

```
cp /etc/elasticsearch/certs/elastic-certificates.p12 certs/el
```

b. Check that the certificates have the right user permissions.

```
ll certs
```



If the results are different from the above, run the following to modify user permissions and ownership to allow Elasticsearch access to it:

```
sudo chmod 777 -R certs/
```

```
sudo chown logstash: certs/elastic-certificates.p12
```

c. Update the logstash output configuration file with the newly added cert and login credentials.

```
sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

```
output {
  if [@metadata][pipeline] {
        elasticsearch {
        hosts => ["localhost:9200"]
        cacert => '/etc/logstash/certs/elastic-certificates.p12'
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        pipeline => "%{[@metadata][pipeline]}"
        user => "elastic"
        password => "password"
        }
  } else {
        elasticsearch {
        hosts => ["localhost:9200"]
        cacert => '/etc/logstash/certs/elastic-certificates.p12'
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        user => "elastic"
        password => "password"
        }
  }
}
```

d. Restart Logstash. Login to your Kibana Dashboard to check that the logs are being sent successfully.

Installing Filebeat on desired server

Step 1: Add Elastic Repository

To install filebeat on a server, we need to first install the Elastic repositories to 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:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsear
ch | sudo apt-key add -
```

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

```
thp@ubuntu-hp:~$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
thp@ubuntu-hp:~$
```

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

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

4. Update the repositories:

```
sudo apt update
```

Step 2: Install Filebeat

The Elastic Stack uses data shippers called Beats to collect data from various sources and transport them to Logstash or Elasticsearch. We will be using Filebeat to forward local logs from your desired server to logstash.

1. Install Filebeat with the following command:

```
sudo apt install filebeat
```

2. Configure Filebeat

a. Configure Filebeat to connect to Logstash. In this guide, we will modify the configuration file that comes with Filebeat.

Open the Filebeat configuration file::

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

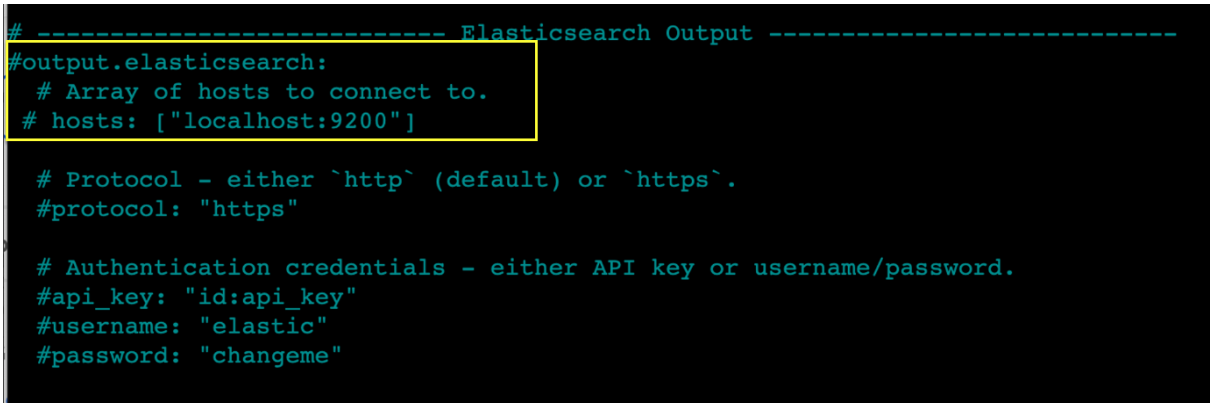
> 💡 **Note:** As with Elasticsearch, Filebeat's configuration file is in YAML format. This means that proper indentation is crucial, so be sure to use the same number of spaces that are indicated in these instructions.

b. We will use Logstash to perform additional processing on the data collected by Filebeat hence Filebeat will not need to send any data directly to

Elasticsearch, so let's disable that output. To do so, find the `output.elasticsearch` section and comment out the following lines by adding a `#` at the beginning of the line.

/etc/filebeat/filebeat.yml

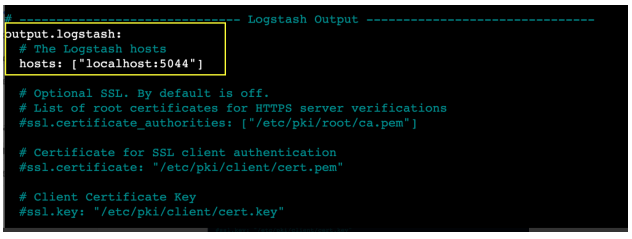
```
...
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]
...
```



c. Next, configure the `output.logstash` section. Remove the `#` in the lines `output.logstash:` and `hosts: ["localhost:5044"]`. This will configure Filebeat to connect to Logstash on your Elastic Stack server at port `5044`, which we specified a Logstash input earlier:

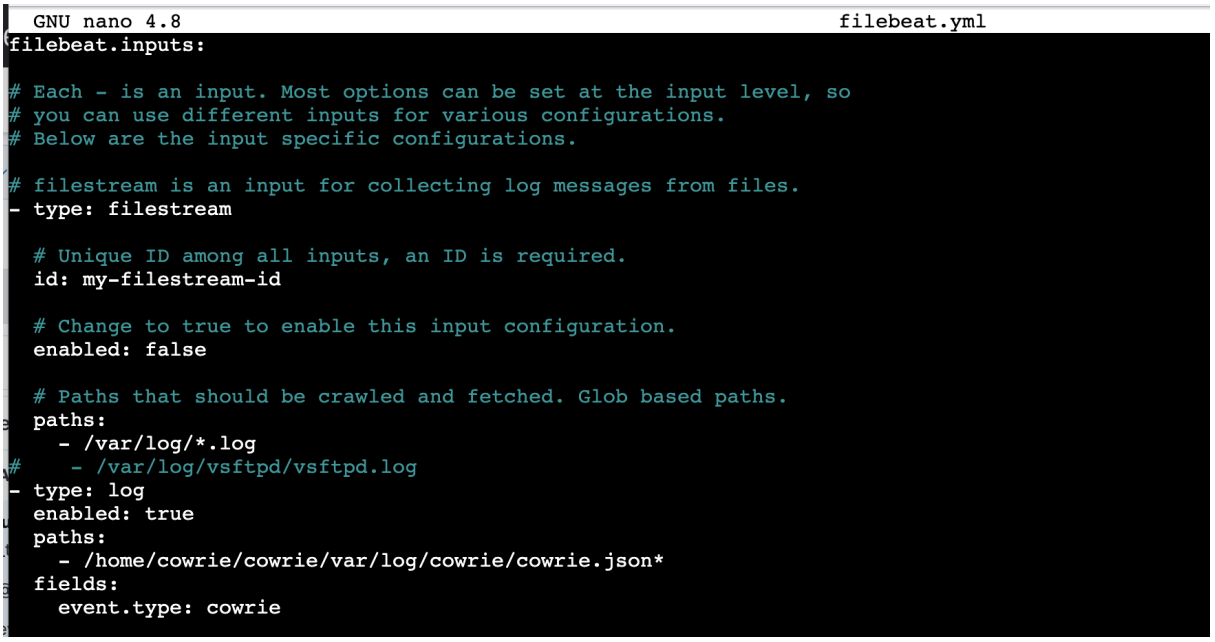
/etc/filebeat/filebeat.yml

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



Then, add the following under filebeat.inputs to collect logs from the cowrie honeypot.

```
filebeat.inputs:
  - type: log
  enabled: true
  paths:
    - /home/cowrie/cowrie/var/log/cowrie/cowrie.json*
  fields:
    event.type: cowrie
```



Save and close the file.

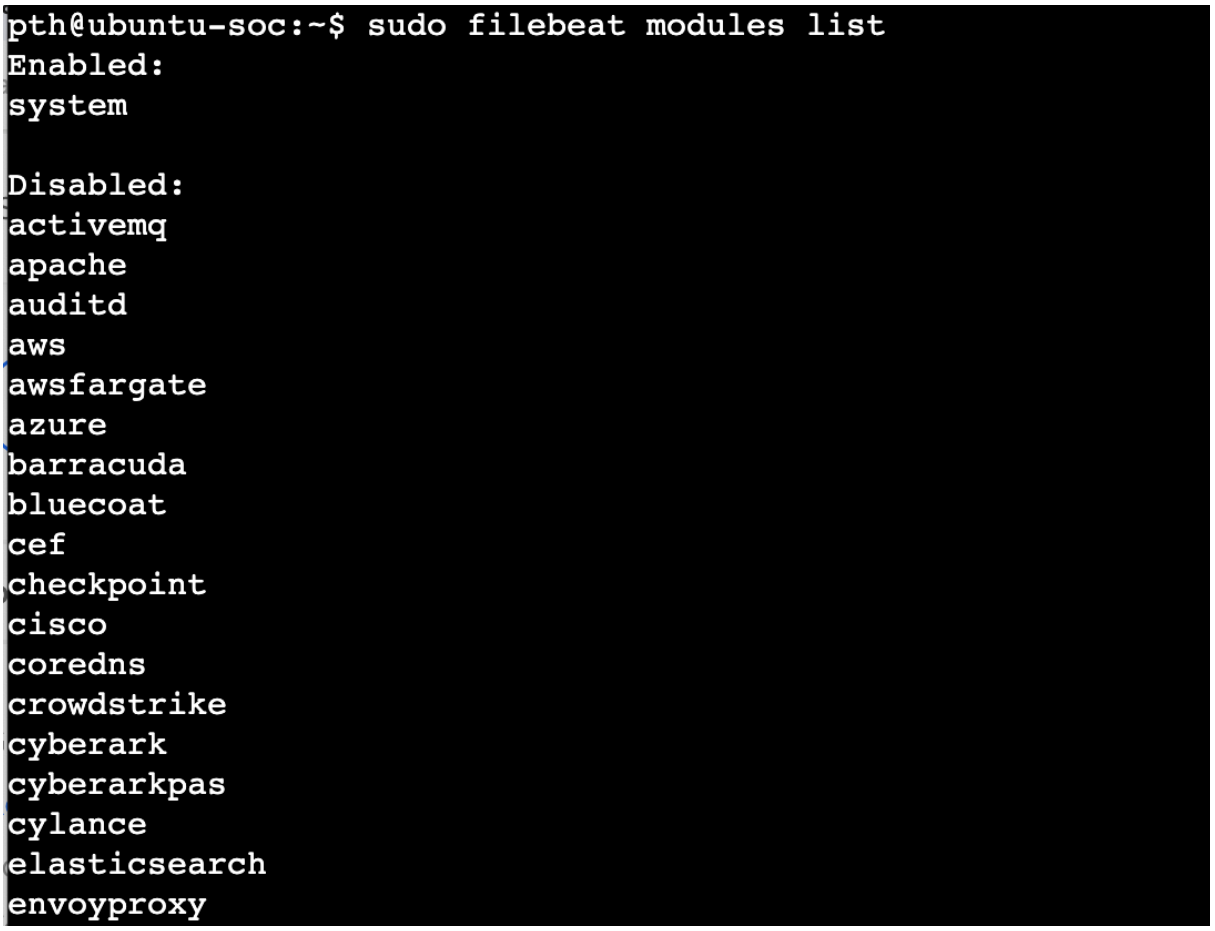
d. The functionality of Filebeat can be extended with Filebeat modules. We will use the system module, which collects and parses logs created by the system logging service of common Linux distributions.

To enable the system module:

```
sudo filebeat modules enable system
```

e. You can see a list of enabled and disabled modules by running:

```
sudo filebeat modules list
```



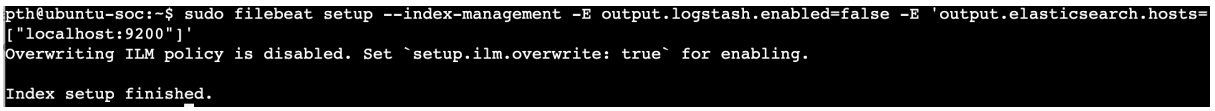
f. Next, we need to set up the Filebeat ingest pipelines, which parse the log data before sending it through logstash to Elasticsearch. To load the ingest pipeline for the system module, enter the following command:

```
sudo filebeat setup --pipelines --modules system
```

g. Load the index template into Elasticsearch. An *Elasticsearch index* is a collection of documents that have similar characteristics. Indexes are identified with a name, which is used to refer to the index when performing various operations within it. The index template will be automatically applied when a new index is created.

To load the template, use the following command:

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

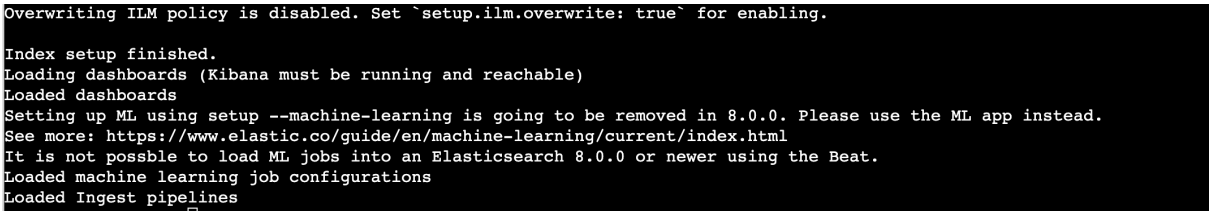


h. Filebeat comes packaged with sample Kibana dashboards that allow you to visualize Filebeat data in Kibana. Before you can use the dashboards, you need to create the index pattern and load the dashboards into Kibana.

As the dashboards load, Filebeat connects to Elasticsearch to check version information. To load dashboards when Logstash is enabled, you need to disable the Logstash output and enable Elasticsearch output with the following command:

```
sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=localhost:5601
```

You should see a output similar to this:



3. Start Filebeat

a. If your configuration test is successful, start the Filebeat service

```
sudo systemctl start filebeat
```

There is no output if the service starts successfully.

b. Next, configure Filebeat to launch at boot:

```
sudo systemctl enable filebeat
```

```
pth@ubuntu-soc:~$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
```

c. Next, check status of Filebeat :

```
pth@ubuntu-soc:~$ sudo systemctl start filebeat
pth@ubuntu-soc:~$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
pth@ubuntu-soc:~$ systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-03-09 14:31:06 UTC; 1min 30s ago
       Docs: https://www.elastic.co/beats/filebeat
   Main PID: 2146 (filebeat)
      Tasks: 7 (limit: 2339)
     Memory: 71.5M
     CGroup: /system.slice/filebeat.service
             └─2146 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /us
```

If youve set up your Elastic Stack correctly, Filebeat will begin shipping your syslog and authorization logs to Logstash, which will then load that data into Elasticsearch.

4. Test your filebeat configuration with this command:

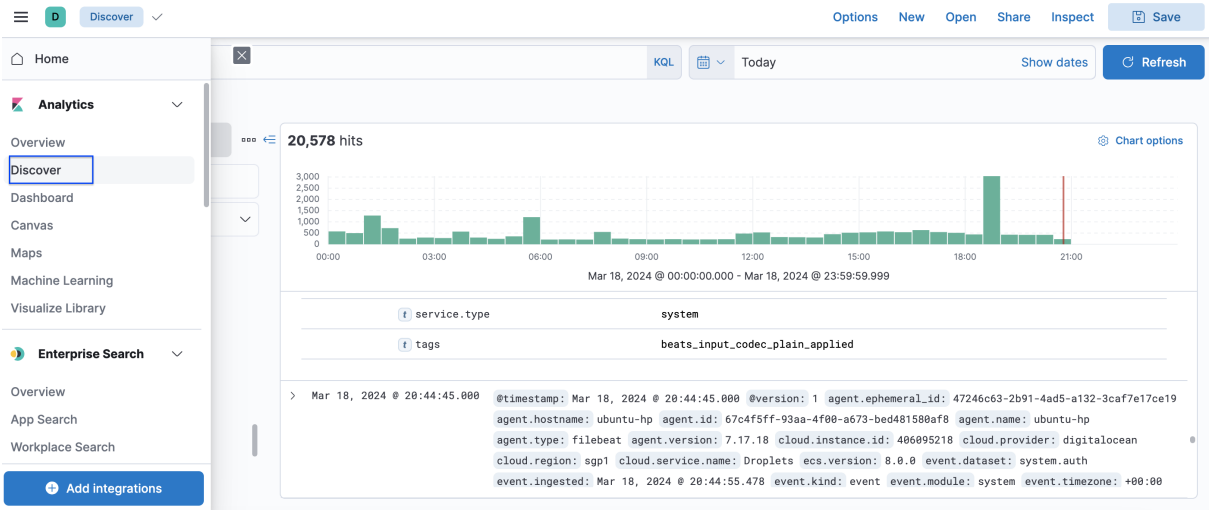
```
curl -XGET 'http://localhost:9200/filebeat-*/_search?prett
y'
```

You should receive output similar to this:

```
  ],
  "cloud" : {
    "instance" : {
      "id" : "405264217"
    },
    "provider" : "digitalocean",
    "service" : {
      "name" : "Droplets"
    },
    "region" : "sgp1"
  },
  "input" : {
    "type" : "log"
  },
  "@timestamp" : "2024-03-08T15:58:25.000Z",
  "system" : {
    "auth" : { }
  },
  "ecs" : {
    "version" : "8.0.0"
  },
  "related" : {
    "hosts" : [
      "ubuntu-soc"
    ]
  },
  "service" : {
    "type" : "system"
  },
  "host" : {
    "hostname" : "ubuntu-soc",
    "os" : {
      "kernel" : "5.4.0-122-generic",
      "codename" : "focal",
      "name" : "Ubuntu",
      "type" : "linux",
      "family" : "debian",
      "version" : "20.04.4 LTS (Focal Fossa)",
      "platform" : "ubuntu"
```


💡 If your output shows 0 total hits, Elasticsearch is not loading any logs under the index you searched for, and you will need to review your setup for errors. If you received the expected output, continue to the next step, in which we will see how to navigate through some of Kibana's dashboards.

Go to your browser and go to localhost:5601. Click on the menu button at the top left and go to Analytics → Discover tab. You should see the logs appearing like below:


Discover
Options
New
Open
Share
Inspect
Save
Home
KQL
Today
Show dates
Refresh
Analytics
Overview
Discover
Dashboard
Canvas
Maps
Machine Learning
Visualize Library
20,578 hits
Chart options
3,000
2,500
2,000
1,500
1,000
500
0
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
Mar 18, 2024 @ 00:00:00.000 - Mar 18, 2024 @ 23:59:59.999
service.type
system
tags
beats_input_codec_plain_applied
Enterprise Search
Overview
App Search
Workplace Search
Add integrations
Mar 18, 2024 @ 20:44:45.000
@timestamp: Mar 18, 2024 @ 20:44:45.000 @version: 1 agent.ephemeral_id: 47246c63-2b91-4ad5-a132-3caf7e17ce19
agent.hostname: ubuntu-hp agent.id: 67c4f5ff-93aa-4f00-a673-bed481580af8 agent.name: ubuntu-hp
agent.type: filebeat agent.version: 7.17.18 cloud.instance.id: 406095218 cloud.provider: digitalocean
cloud.region: sgp1 cloud.service.name: Droplets ecs.version: 8.0.0 event.dataset: system.auth
event.ingested: Mar 18, 2024 @ 20:44:55.478 event.kind: event event.module: system event.timezone: +00:00