Elasticsearch Logstash Kibana

Step 1: Install Dependencies

Install Java

The ELK stack requires Java 8 to be installed. Certain components are compatible with Java 9 but logstash requires java 8.

To check you java version:

```
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 not, than install java 8 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.

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.

Step 2: Create 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.

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
```

Step 3: Install Elasticsearch

1. Before 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 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
```

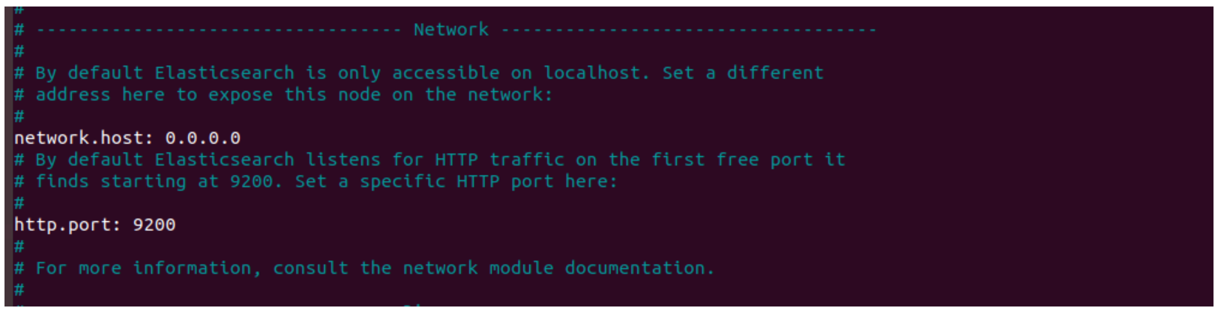
```
#http.port: 9200
```

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: 0.0.0.0
```

```
http.port: 9200
```

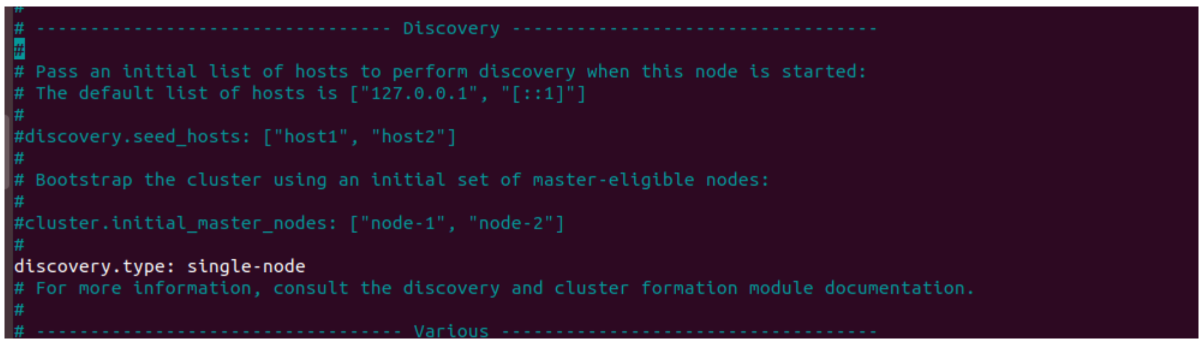


```
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
```

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.

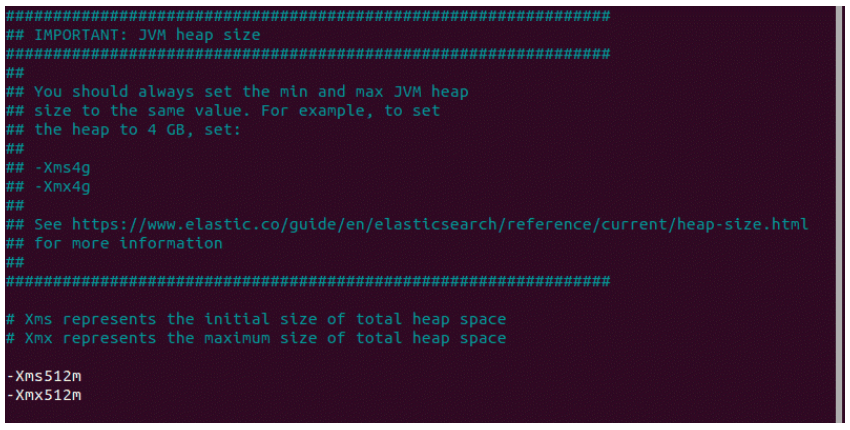


```
# ---------------------------------- Discovery -----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
discovery.type: single-node
# For more information, consult the discovery and cluster formation module documentation.
#
# ---------------------------------- Various -----------------------------------
```

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

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

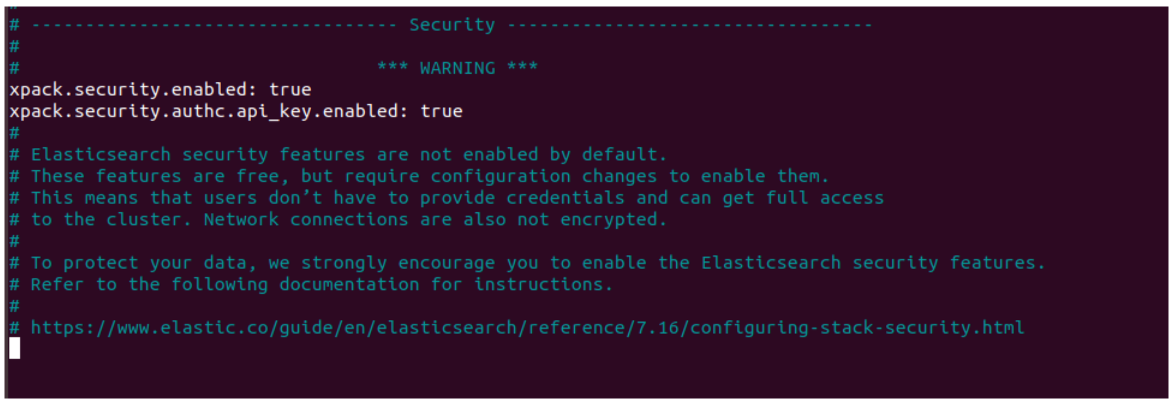
6. Find the lines starting with `-Xms` and `-Xmx`. In the example below, the maximum (`-Xmx`) and minimum (`-Xms`) size is set to 512MB.



```
################################################################
## 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
```

Security Configuration

```
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
```



```
# ---------------------------------- Security ----------------------------------
#
#                                  *** WARNING ***
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
#
# 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
```

Start Elasticsearch

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

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

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

2. Enable Elasticsearch to start on boot:

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

3. Create password for every service

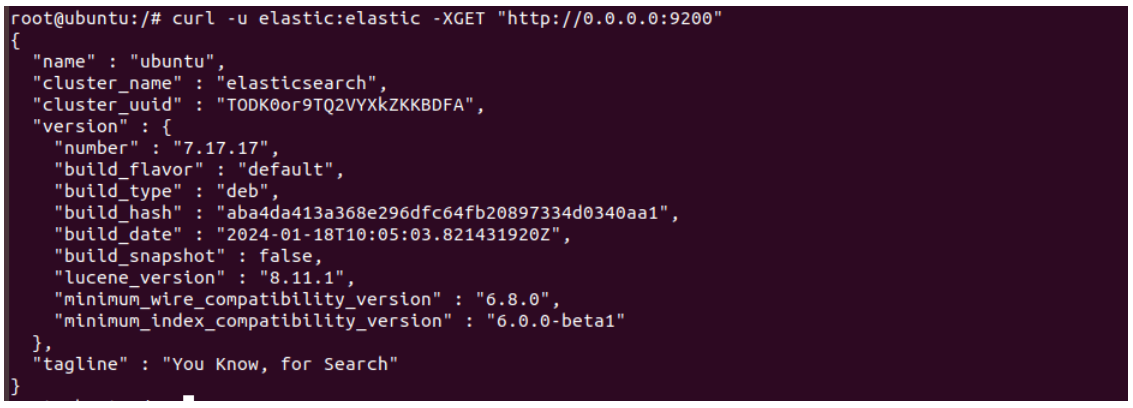
```
./usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive
```

Test Elasticsearch

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

```
curl -u username:password -X GET 0.0.0.0:9200
```

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**.



```
root@ubuntu:/# curl -u elastic:elastic -XGET "http://0.0.0.0:9200"
{
  "name" : "ubuntu",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "TODK0or9TQ2VYXkZKKBDFA",
  "version" : {
    "number" : "7.17.17",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "aba4da413a368e296dfc64fb20897334d0340aa1",
    "build_date" : "2024-01-18T10:05:03.821431920Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

#note: replace elastic:elastic with you username and password for elasticsearch

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:

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

Configure Kibana

1. Next, open the **kibana.yml** configuration file for editing:

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

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

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

```
#server.host: "your-hostname"
```

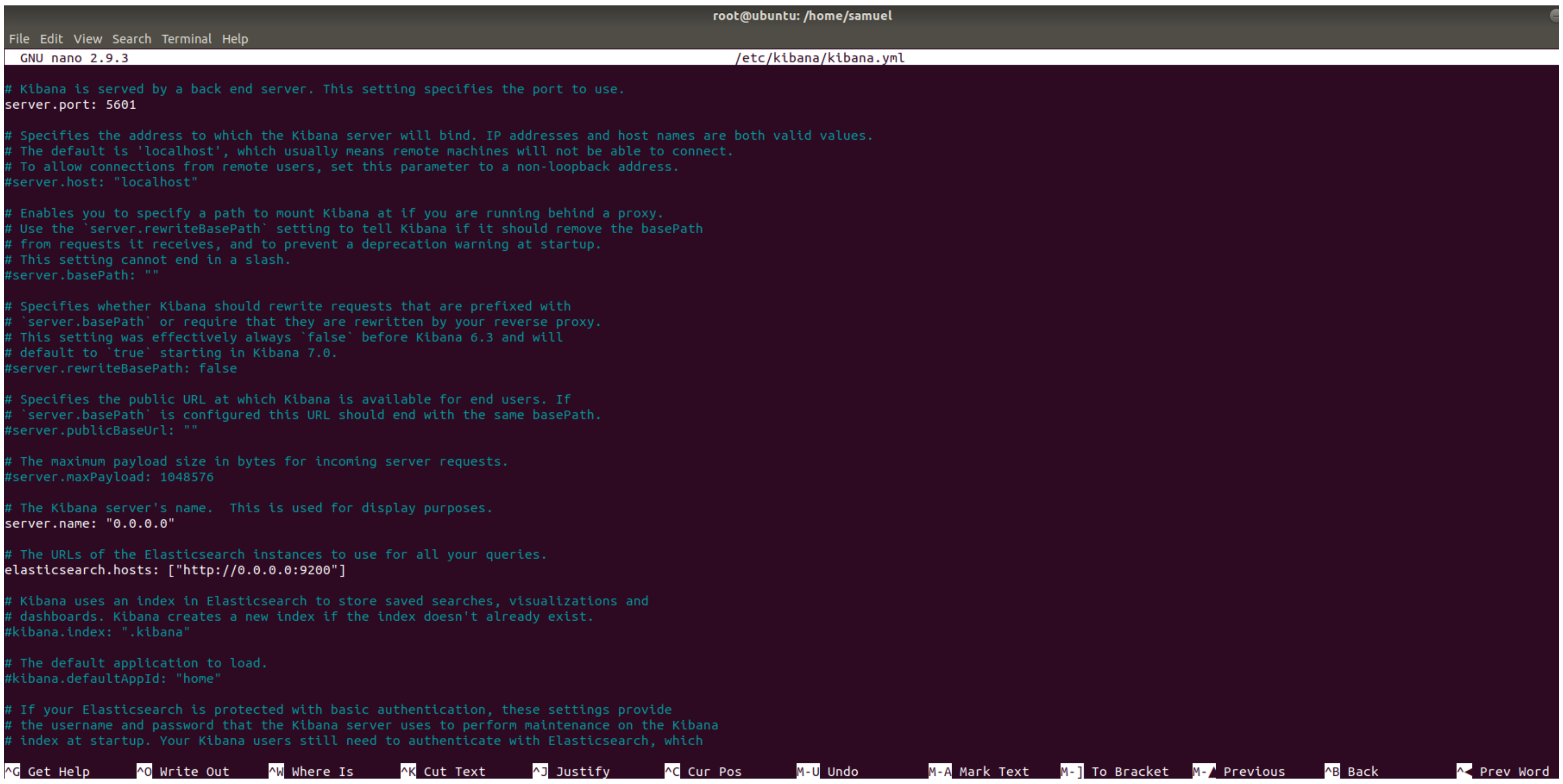
```
#elasticsearch.hosts: ["http://localhost:9200"]
```

The above-mentioned lines should look as follows:

```
server.port: 5601
```

```
server.host: "0.0.0.0"
```

```
elasticsearch.hosts: ["http://0.0.0.0:9200"]
```



```
# 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: "0.0.0.0"

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://0.0.0.0:9200"]

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "home"

# 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
```

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

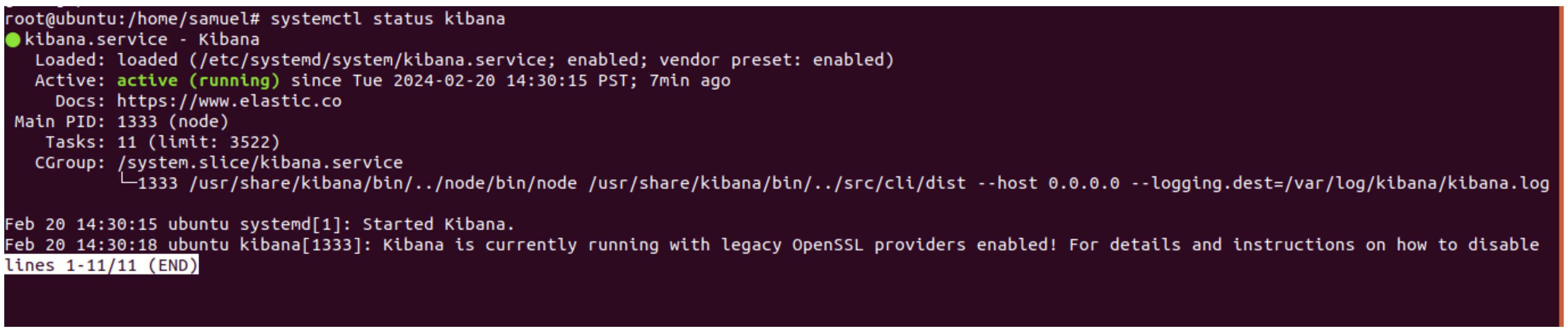
Note: This configuration allows traffic from the same system Elasticstack is configured on. You can set the `server.host` value to the address of a remote server.

Start and Enable Kibana

1. Start the Kibana service:

```
sudo systemctl start kibana
```

There is no output if the service starts successfully.



```
root@ubuntu:/home/samuel# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-02-20 14:30:15 PST; 7min ago
     Docs: https://www.elastic.co
 Main PID: 1333 (node)
    Tasks: 11 (limit: 3522)
   CGroup: /system.slice/kibana.service
           └─1333 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --host 0.0.0.0 --logging.dest=/var/log/kibana/kibana.log

Feb 20 14:30:15 ubuntu systemd[1]: Started Kibana.
Feb 20 14:30:18 ubuntu kibana[1333]: Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable
lines 1-11/11 (END)
```

2. Next, configure Kibana to launch at boot:

```
sudo systemctl enable kibana
```

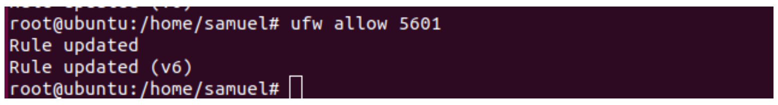
Allow Traffic on Port 5601

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:



```
root@ubuntu:/home/samuel# ufw allow 5601
Rule updated
Rule updated (v6)
root@ubuntu:/home/samuel# 
```

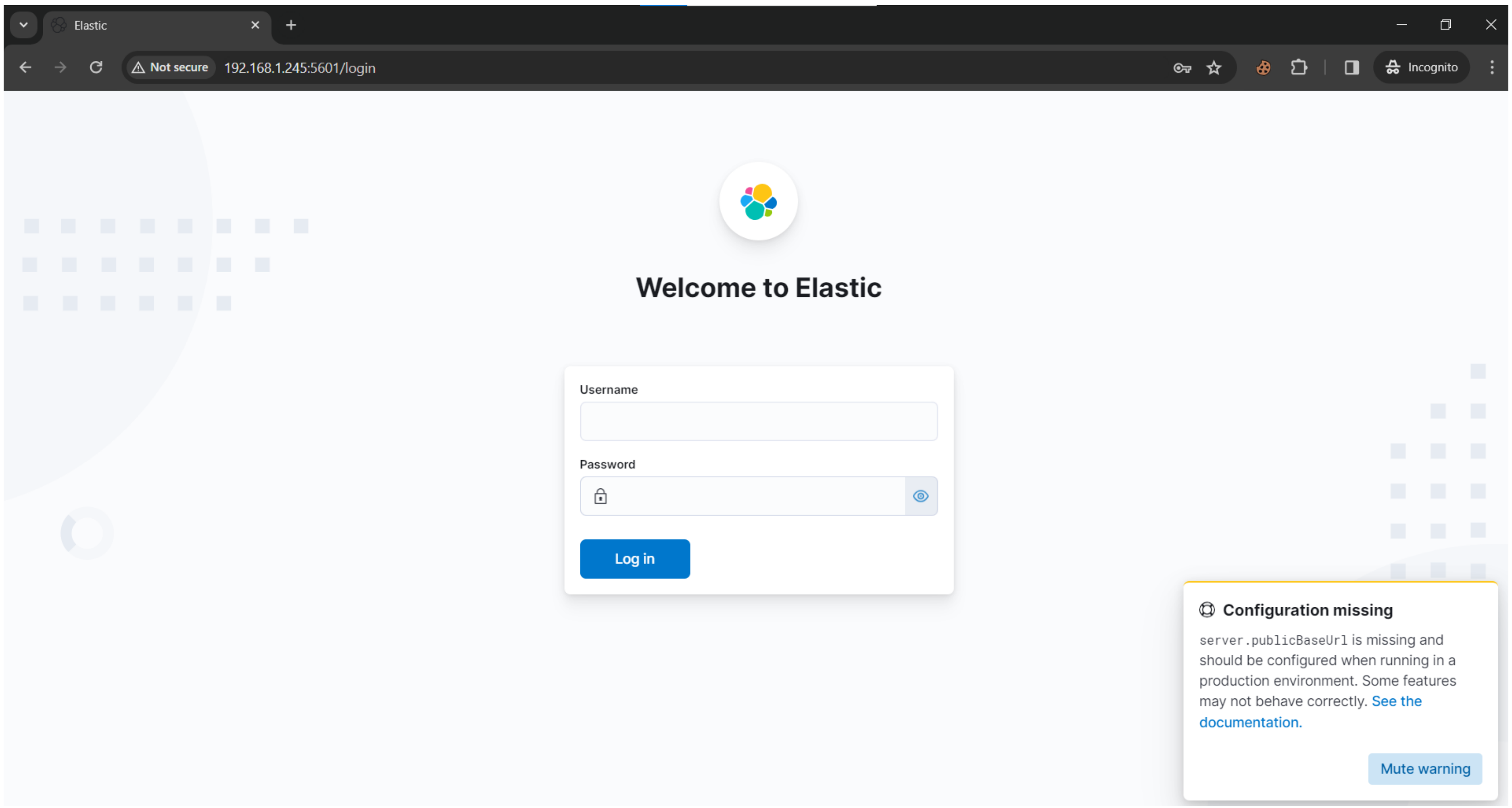
Test Kibana

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

```
http://your-ip-address:5601
```

#note: Replace '`your-ip-address`' with the IP address of the machine on which the ELK stack is running.

The Kibana dashboard loads.



Configuration missing

`server.publicBaseUrl` is missing and should be configured when running in a production environment. Some features may not behave correctly. See the documentation.

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

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:

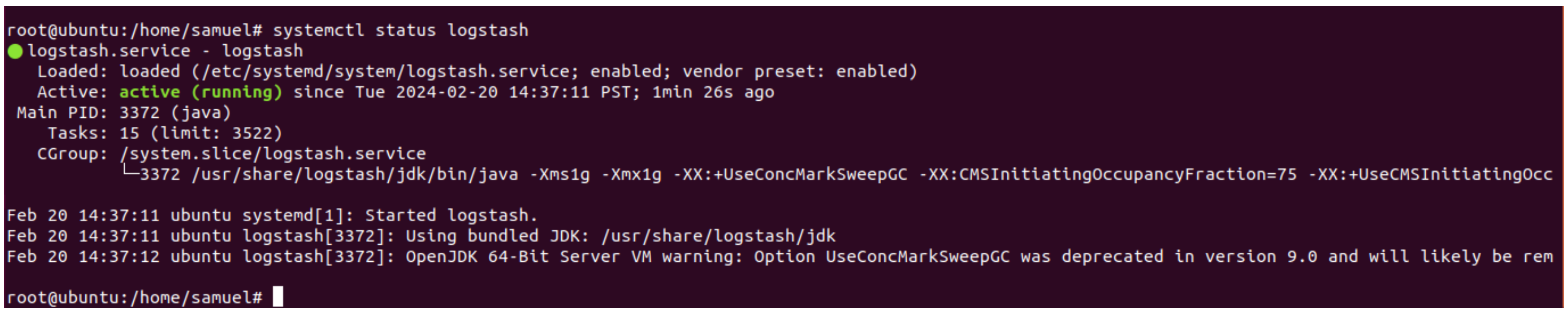
```
sudo systemctl start logstash
```

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
```



```
root@ubuntu:/home/samuel# systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-02-20 14:37:11 PST; 1min 26s ago
 Main PID: 3372 (java)
    Tasks: 15 (limit: 3522)
   CGroup: /system.slice/logstash.service
           └─3372 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOcc

Feb 20 14:37:11 ubuntu systemd[1]: Started logstash.
Feb 20 14:37:11 ubuntu logstash[3372]: Using bundled JDK: /usr/share/logstash/jdk
Feb 20 14:37:12 ubuntu logstash[3372]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be rem

root@ubuntu:/home/samuel#
```

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"]
```

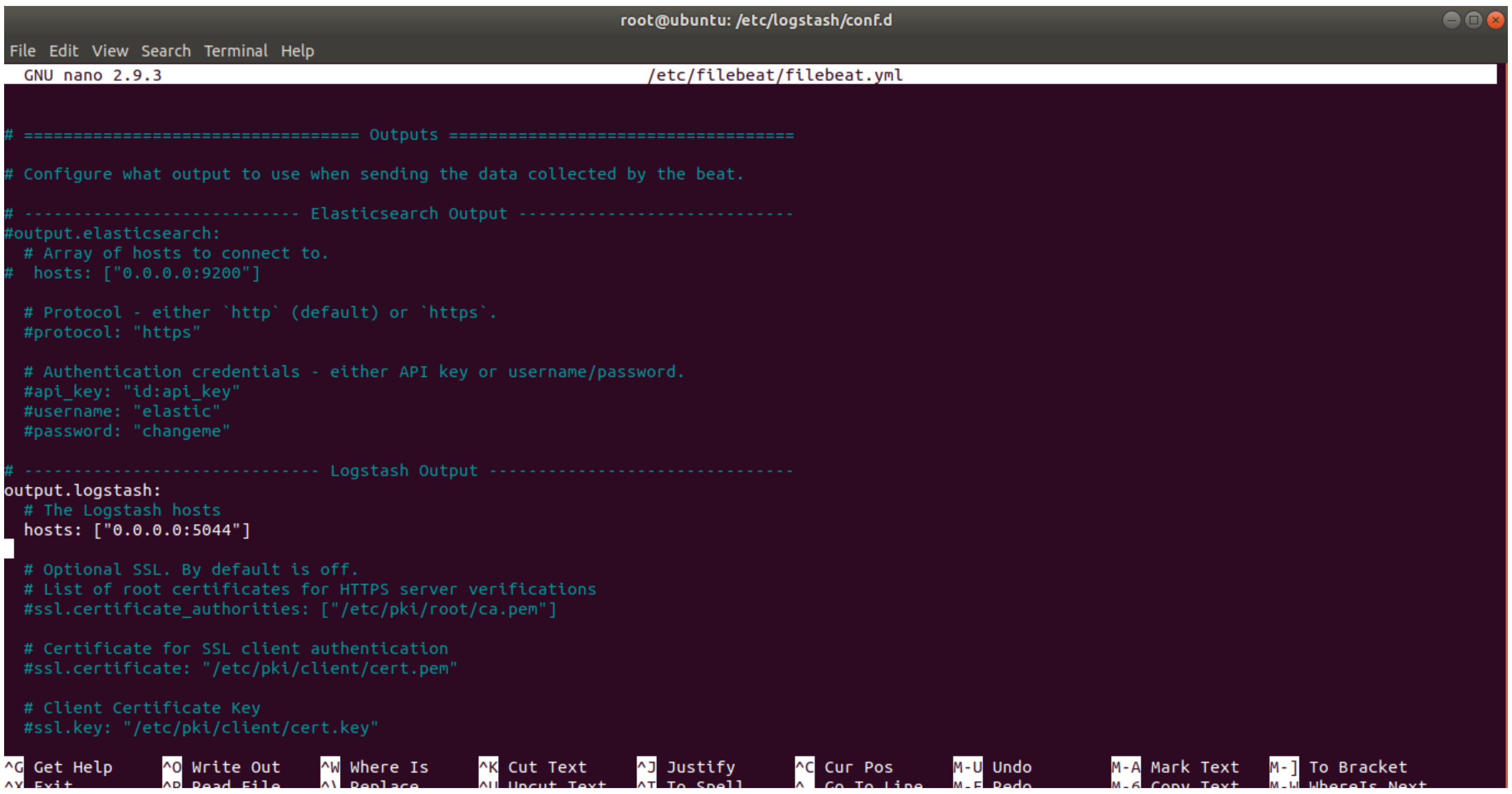
3. Under the *Logstash output* section, **remove the hash sign** (`#`) in the following two lines and replace "`localhost`" with "`0.0.0.0":`

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

It should look like this:

```
output.logstash
     hosts: ["0.0.0.0:5044"]
```

For further details, see the image below.



root@ubuntu: /etc/logstash/conf.d
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                                  /etc/filebeat/filebeat.yml


# ================================== Outputs ===================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
#  hosts: ["0.0.0.0:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["0.0.0.0:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"
```

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

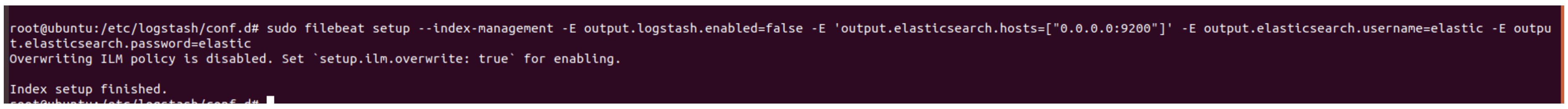
```
sudo filebeat modules enable systema
```

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=["0.0.0.0:9200"]' -E output.elasticsearch.username=your
_elastic_username -E output.elasticsearch.password=your_elastic_password
```

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



```
root@ubuntu:/etc/logstash/conf.d# sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["0.0.0.0:9200"]' -E output.elasticsearch.username=elastic -E outpu
t.elasticsearch.password=elastic
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
root@ubuntu:/etc/logstash/conf.d#
```

Start and Enable Filebeat

Start and enable the Filebeat service:

```
sudo systemctl start filebeat
```

```
sudo 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.

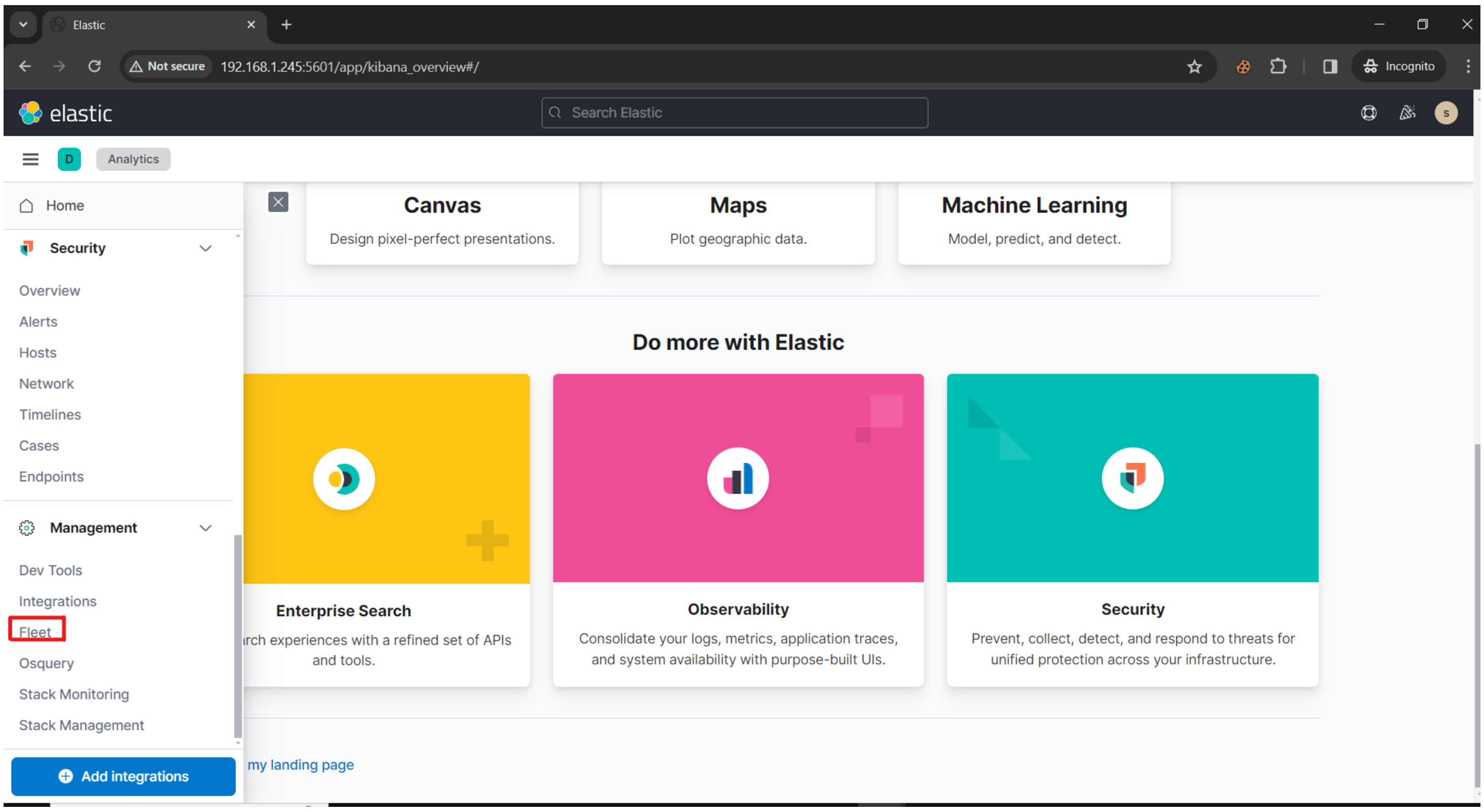
```
curl -XGET http://localhost:9200/_cat/indices?v
```

Enrolling Elastic Agents

This step involves enrolling the agents i.e. the endpoint/host that are to be monitored.

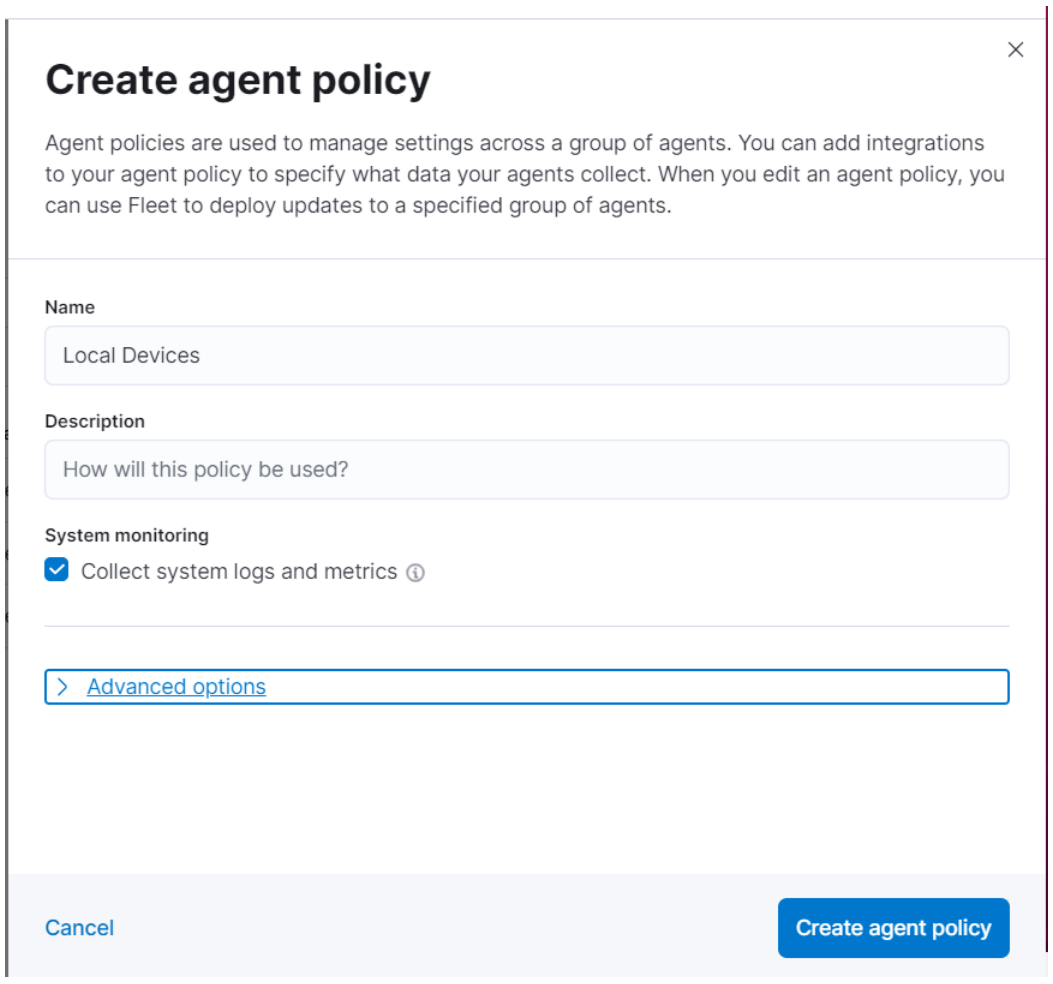
Head over to your elastic dashboard:
replace the "
`your_elastic_server_ip`" with IP address of the machine on which your ELK stack is running. If you are running you ELK stack on a machine that is running on a remote location such virtual cloud instance, make sure that you use the public IP address of the machine and properly configured firewall rules

```
http://your_elastic_server_ip:5601
```


Elastic
Not secure
192.168.1.245:5601/app/kibana_overview#/
Incognito
elastic
Search Elastic
D
Analytics
Home
Security
Overview
Alerts
Hosts
Network
Timelines
Cases
Endpoints
Management
Dev Tools
Integrations
Fleet
Osquery
Stack Monitoring
Stack Management
Add integrations
Canvas
Design pixel-perfect presentations.
Maps
Plot geographic data.
Machine Learning
Model, predict, and detect.
Do more with Elastic
Enterprise Search
rch experiences with a refined set of APIs and tools.
Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.
Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.
my landing page

Step 1: Create New Agent Policy

1. After opening the fleet management console, click on '`Agent Policies`'
2. Click on `Create New Policy` and give the policy a name of your choice:



Create agent policy

Agent policies are used to manage settings across a group of agents. You can add integrations to your agent policy to specify what data your agents collect. When you edit an agent policy, you can use Fleet to deploy updates to a specified group of agents.

Name: Local Devices

Description: How will this policy be used?

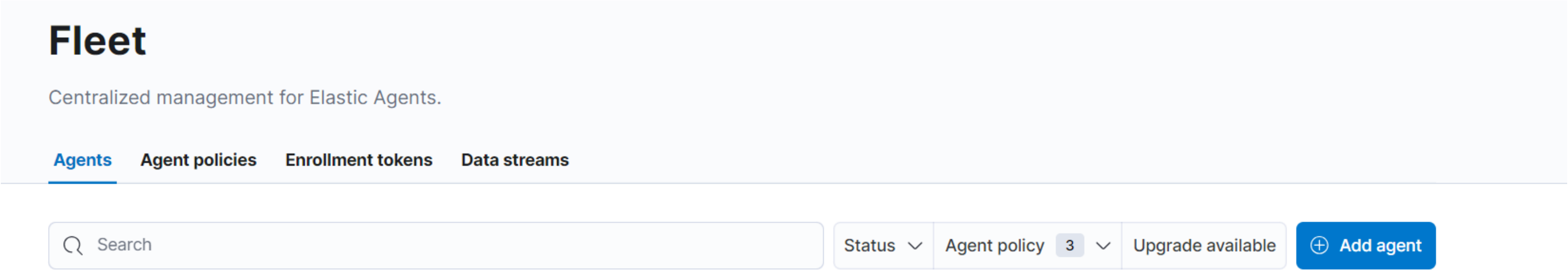
System monitoring

☑ Collect system logs and metrics

3. Click '`Create agent policy`'

step 2: Adding New Agents

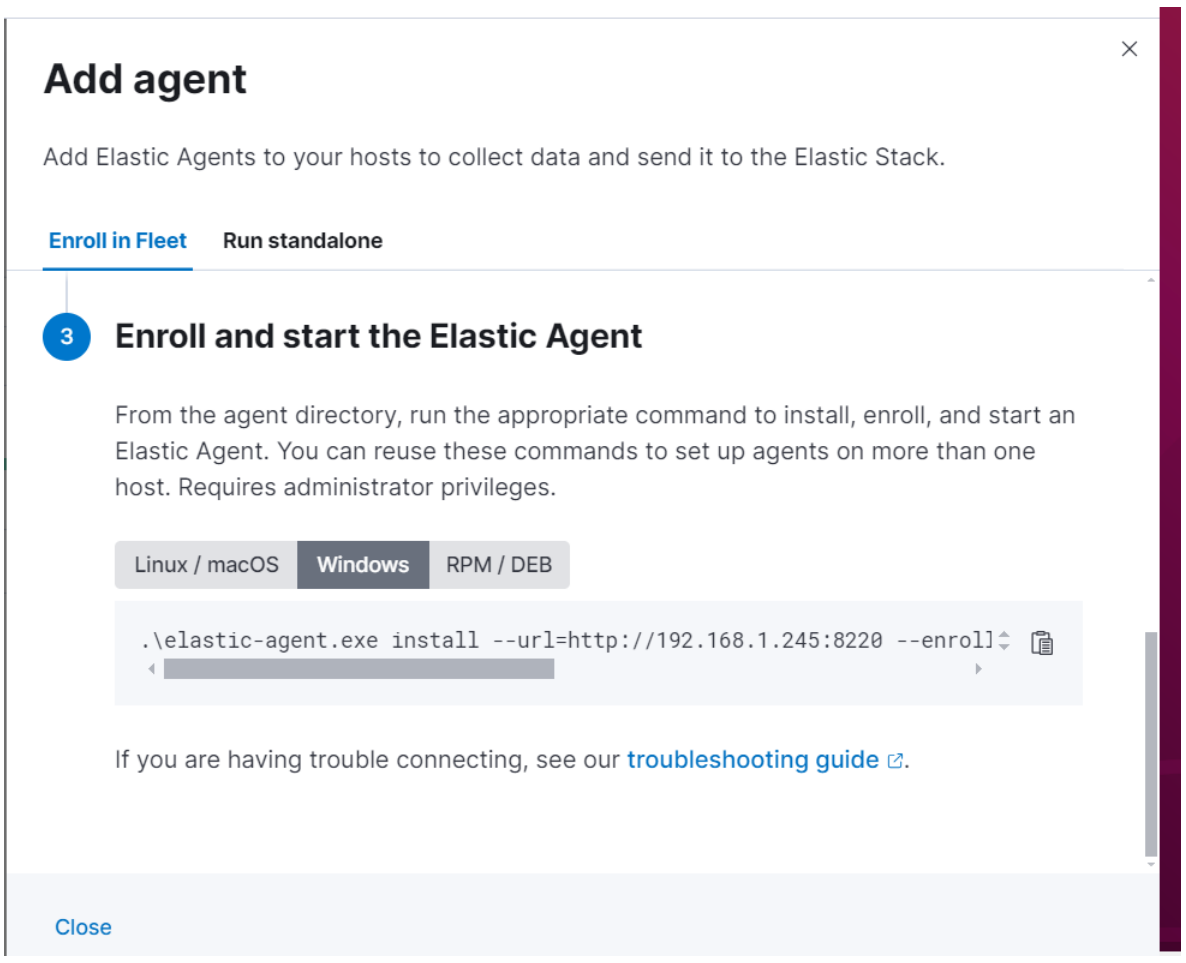
1. Click on Agents



Fleet

Centralized management for Elastic Agents.

2. Click on '`Add Agent`' and follow the steps and install **Elastic Agent** on you machine.
   Depending upon you machine execute the elastic agent enrollment command provided by your fleet. Refer the screenshot below:



Add agent

Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack.

Enroll and start the Elastic Agent

From the agent directory, run the appropriate command to install, enroll, and start an Elastic Agent. You can reuse these commands to set up agents on more than one host. Requires administrator privileges.

```
.\elastic-agent.exe install --url=http://192.168.1.245:8220 --enroll
```

If you are having trouble connecting, see our troubleshooting guide.

In my case, I am trying to enroll a windows machine as an agent:

3. Open terminal/powershell in directory in which the the Elastic Agent is downloaded

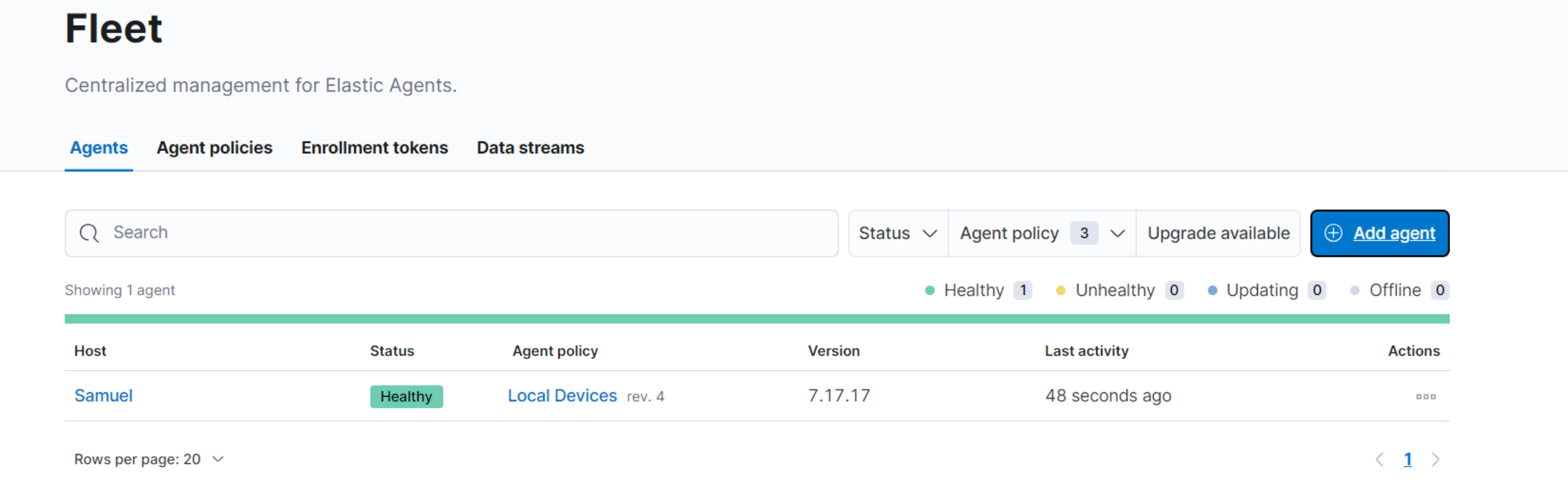
```
#for Windows system

.\elastic-agent.exe install --url=http://192.168.1.245:8220 --enrollment-token=ak1RTmFZMEJKWHV5SzZERFF1dF86M2xIMzFTNmxTdC1RRDhyU3lKWlpLUQ==
```

```
#for Linux/macOS
sudo ./elastic-agent install --url=http://192.168.1.245:8220 --enrollment-token=aXNUdWFJMEJKWHV5SzZERFktc0g6UGYtM2dILU5SUFdpb0VwM1NFcGhjQQ==
```

Replace '`192.168.1.245`' with the IP address of the machin on which your ELK stack is running.

#note: The command for the each operating system might differ, thus make sure that appropriate command is executed.

After executing the command, the agent shoould be enrolled.



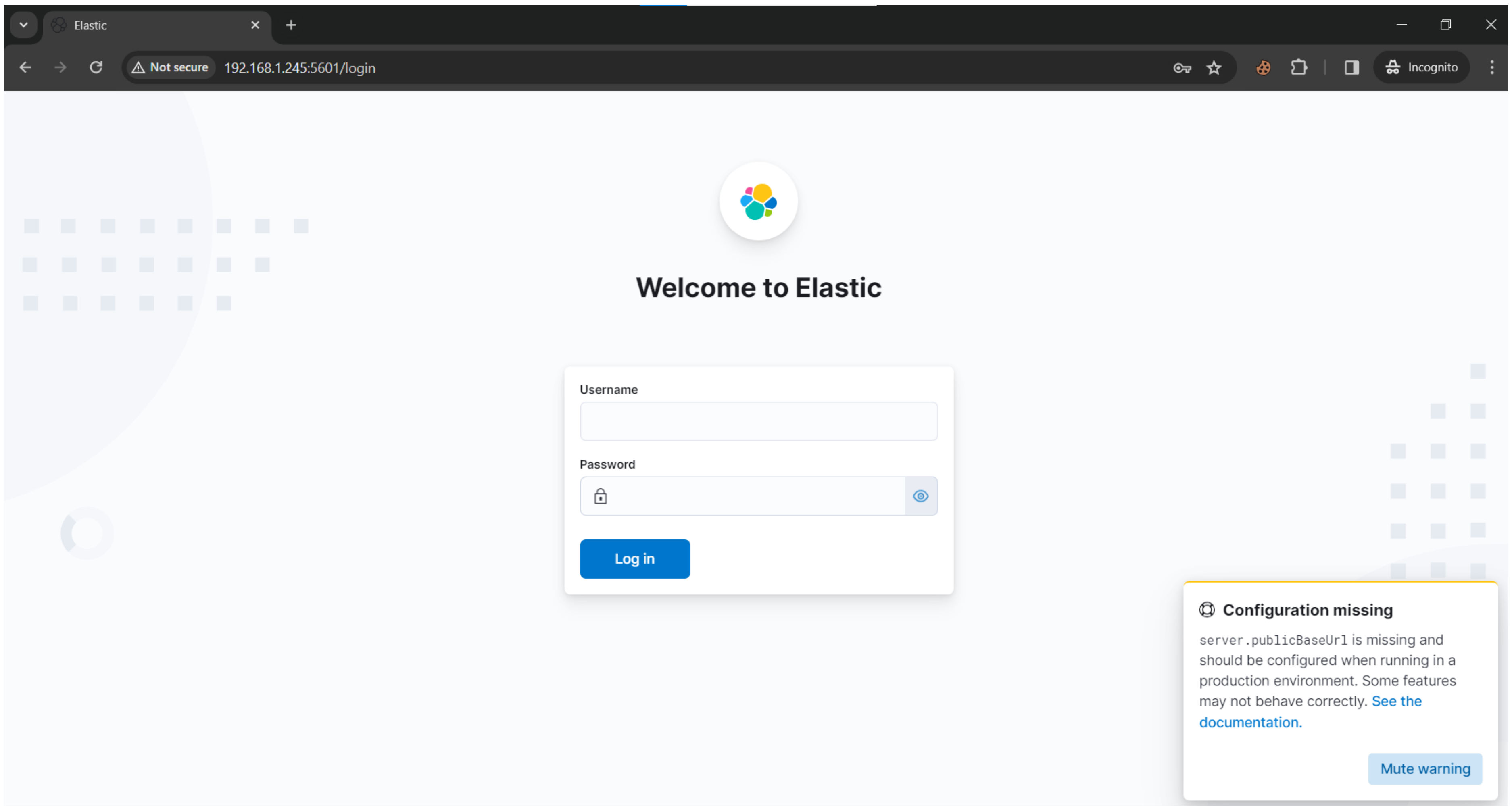
Fleet

Centralized management for Elastic Agents.

Showing 1 agent

● Healthy 1 ● Unhealthy 0 ● Updating 0 ● Offline 0

Host	Status	Agent policy	Version	Last activity	Actions
Samuel	Healthy	Local Devices rev. 4	7.17.17	48 seconds ago	···



Welcome to Elastic

Username

Password

Configuration missing

`server.publicBaseUrl` is missing and should be configured when running in a production environment. Some features may not behave correctly. See the documentation.