Threat Hunting: Log Monitoring Lab Setup with ELK

hackingarticles.in/threat-hunting-log-monitoring-lab-setup-with-elk

Raj August 13, 2020

Elastic Stack is formerly known as the ELK Stack.

Elk Stack is a collection of free opensource software from Elastic Company which is specially designed for centralized logging. It allows the searching, analyzing, and visualization of logs from different sources. in this guide, we will learn to install Elastic Stack on ubuntu.

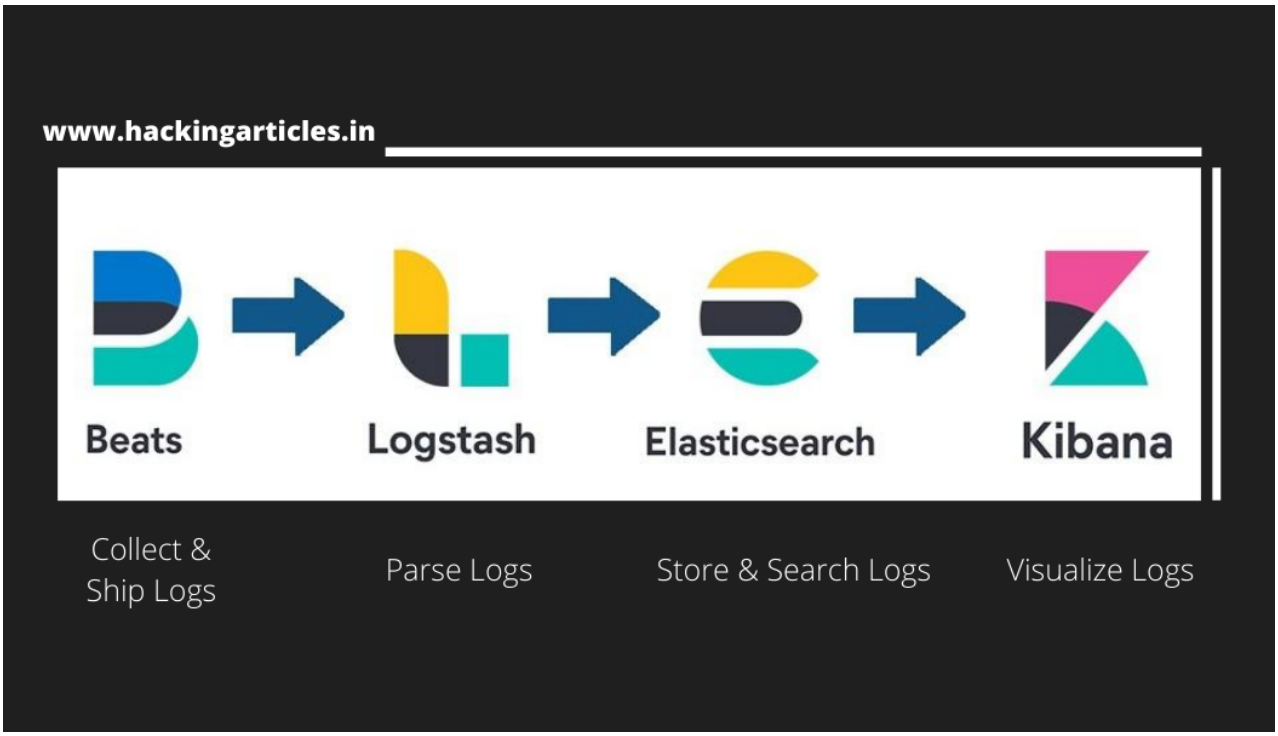
To configure ELK Stack in your Ubuntu platform, there are some prerequisites required for installation.

- Ubuntu 20.04
- Root Privileges

Table of Content

ELK Stack components

1. **Elasticsearch:** It is a restful search engine that stores or holds all of the collected Data.
2. **Logstash:** It is the Data processing component that sends incoming Data to Elasticsearch.
3. **Kibana:** A web interface for searching and visualizing logs.
4. **Filebeat:** A lightweight Single-purpose Data forwarder that can send data from thousands of machines to either Logstash or Elasticsearch.



Install Java and All Dependencies

Elasticsearch requires OpenJDK available in our machine. Install Java using the below command along with the HTTPS support and wget packages for APT.

```
apt install -y openjdk-11-jdk wget apt-transport-https curl
```

```
root@ubuntu:~# apt install -y openjdk-11-jdk wget apt-transport-https curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.20.3-1ubuntu1).
wget set to manually installed.
curl is already the newest version (7.68.0-1ubuntu2.1).
The following package was automatically installed and is no longer required:
  libllvm9
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
```

Now, we are going to import Elasticsearch public key into APT. To import the GPG key enter the following command:

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

```
root@ubuntu:~# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
```

Add Elastic repository to the directory sources.list.d by using the following command :

```
echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-6.x.list
```

```
root@ubuntu:~# 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
```

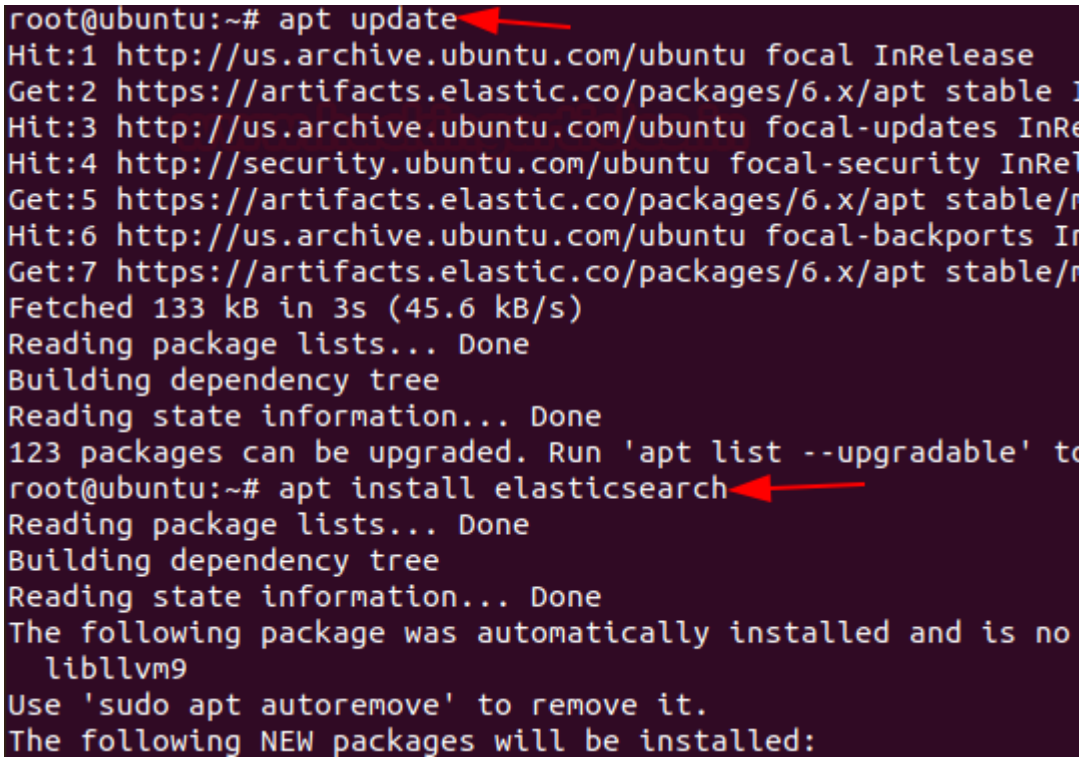
Install and configure Elasticsearch

Update the system repository

```
apt update
```

Install Elasticsearch by using the following command:

```
apt install elasticsearch
```

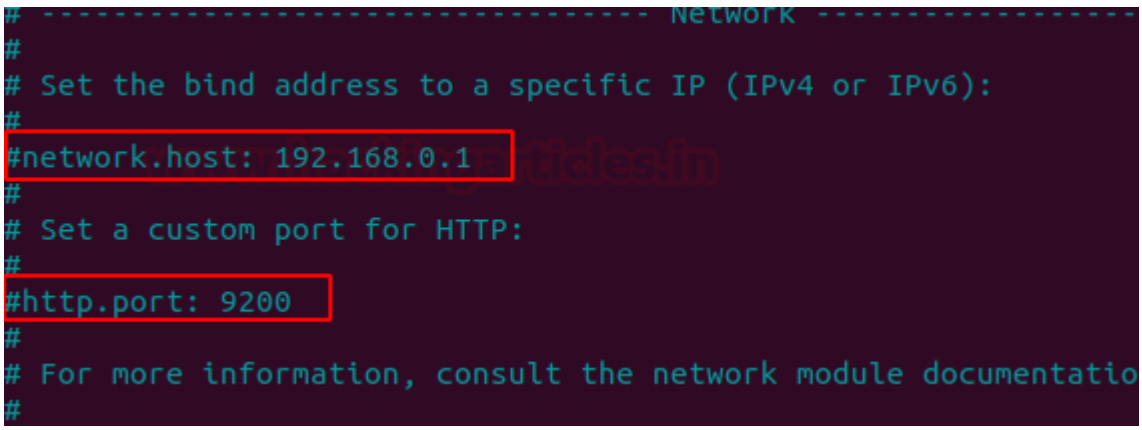


Next, we configure Elasticsearch.

Elasticsearch listens for traffic on port 9200. We are going to restrict outside access to our Elasticsearch instance so that outside parties cannot access data or shut down the elastic cluster through the REST API. Now we're going to do some modifications to the Elasticsearch configuration file – elasticsearch.yml.

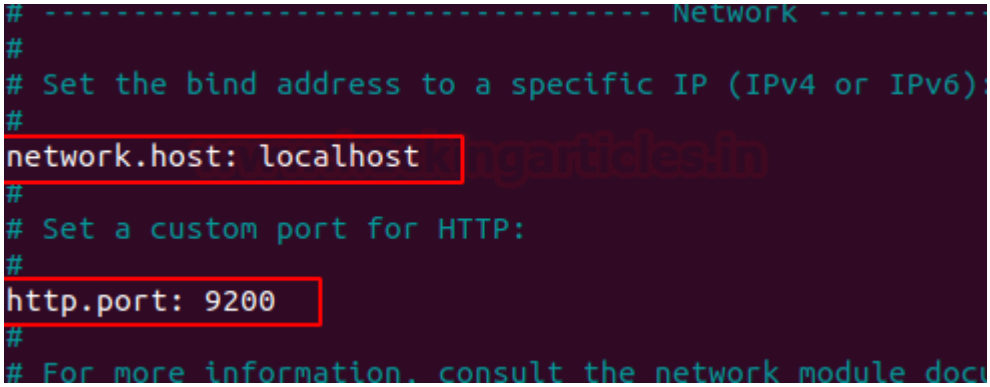
Enter the following command:

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



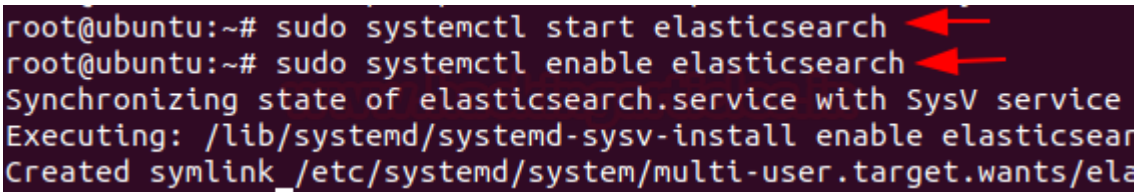
Find the line that specifies network.host attribute and uncomment it and add localhost as its value and also uncomment http.port attribute.

```
network.host: localhost
http.port: 9200
```



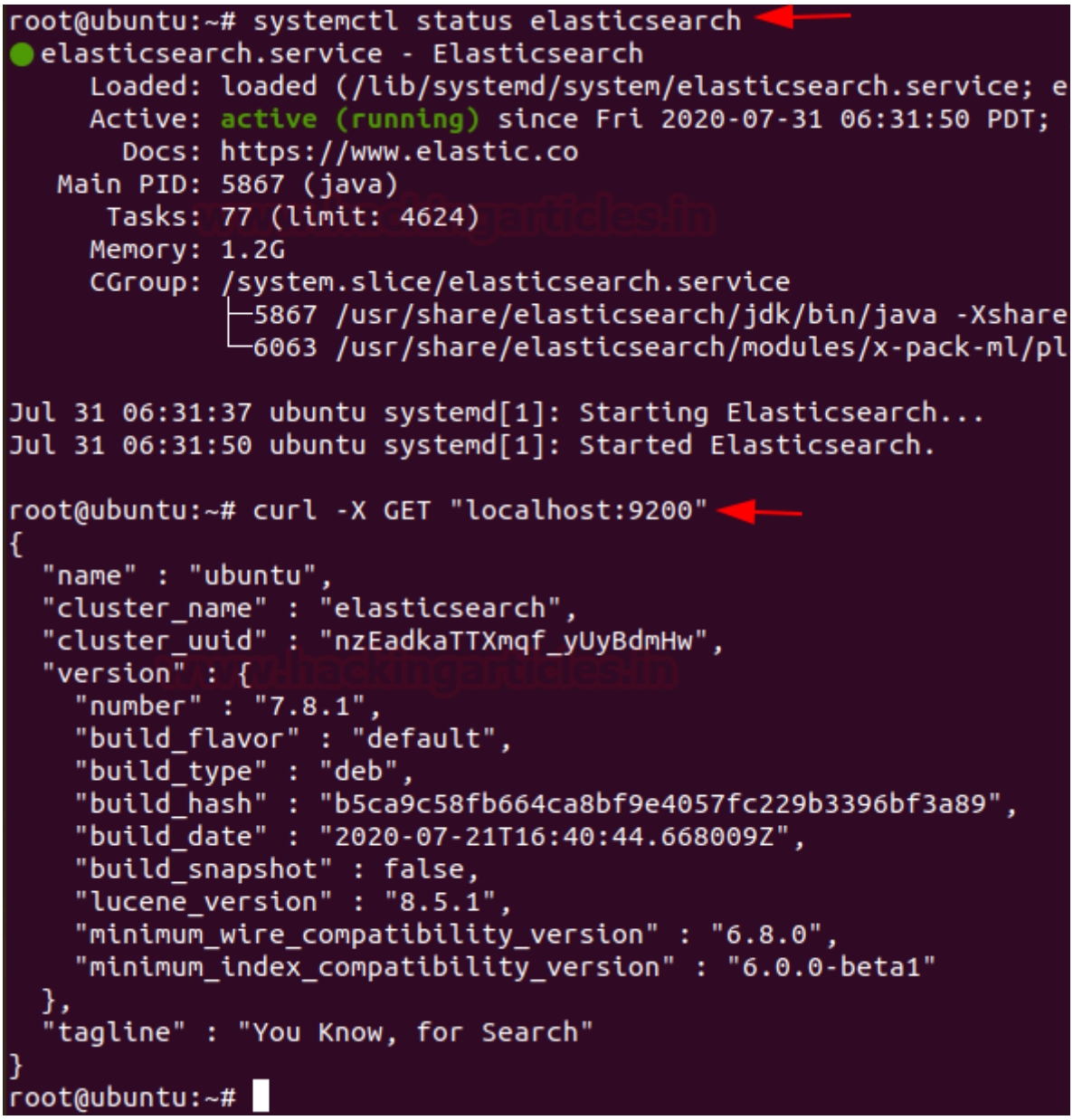
Now, start and enable Elasticsearch services.

```
systemctl start elasticsearch
systemctl enable elasticsearch
```

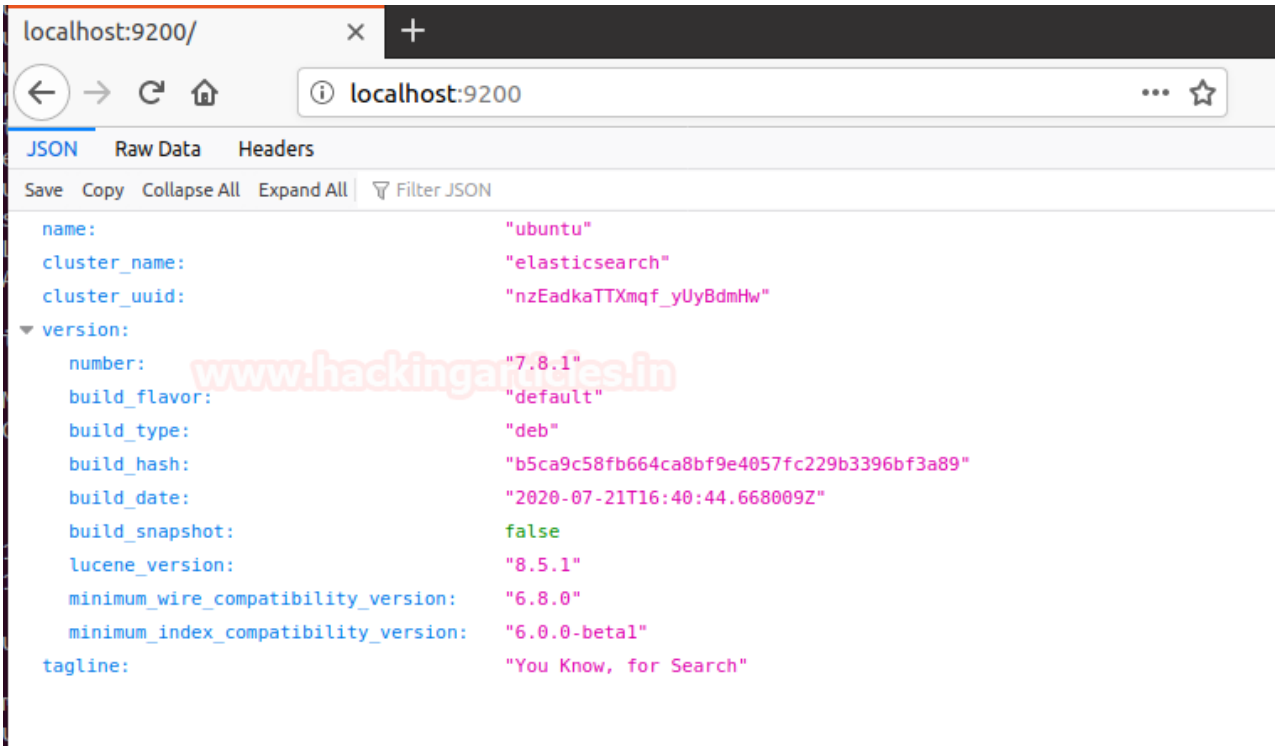


Let’s verify the status if Elasticsearch.

```
systemctl status elasticsearch
curl -X GET "localhost:9200"
```



By default Elasticsearch is listening on the port 9200 you can also verify it on your web browser by pinging https://localhost:9200



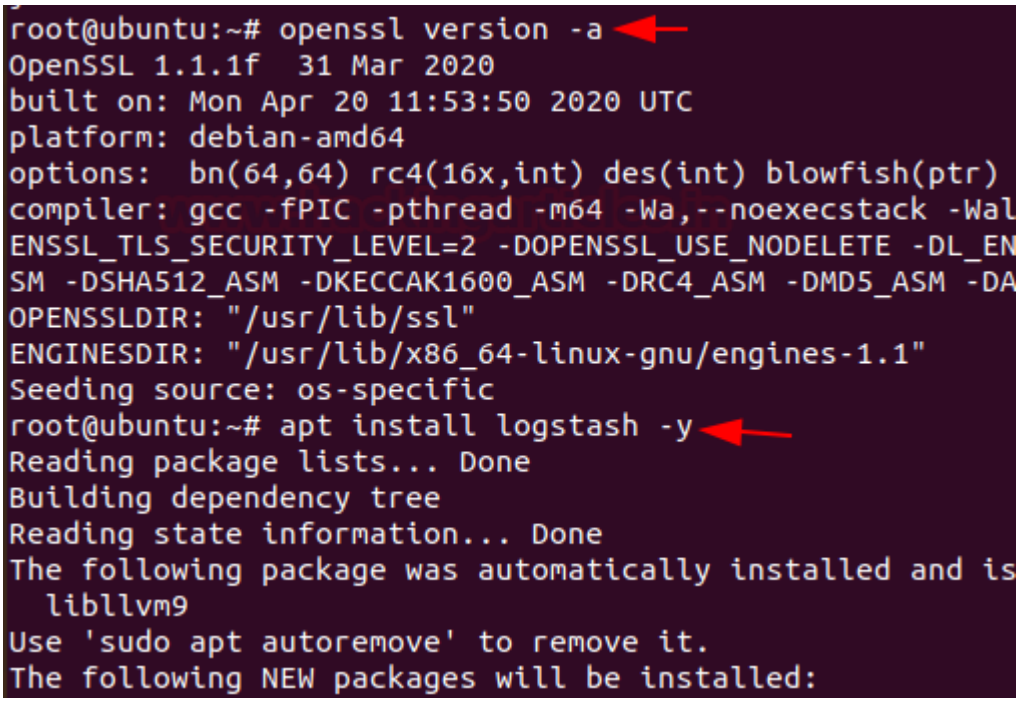
Now Elasticsearch is up and running.

Install and configure Logstash

Logstash used to collect and centralizing logs from different servers using filebeat

First Let's confirm OpenSSL is running and then install Logstash by running following command:

```
openssl version -a
apt install logstash -y
```



Edit the /etc/hosts file and add the following line

```
nano /etc/hosts
```

```
root@ubuntu:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       ubuntu
18.224.44.11    elk-master
```

Where 18.224.44.11 is ip address of server elk-master.

Let's generate an SSL certificate to secure the log data transfer from the client Rsyslog & Filebeat to the Logstash server.

To do this create a new SSL directory under Logstash configuration directory and navigate into that directory generate an SSL certificate by running following command:

```
mkdir -p /etc/logstash/ssl
cd /etc/logstash/
openssl req -subj '/CN=elk-master/' -x509 -days 3650 -batch -nodes -newkey
rsa:2048 -keyout ssl/logstash-forwarder.key -out ssl/logstash-forwarder.crt
```

```
root@ubuntu:~# mkdir -p /etc/logstash/ssl
root@ubuntu:~# cd /etc/logstash/
root@ubuntu:/etc/logstash# openssl req -subj '/CN=elk-master/' -x509 -days 3650 -ba
tch -nodes -newkey rsa:2048 -keyout ssl/logstash-forwarder.key -out ssl/logstash-fo
rwarder.crt
Generating a RSA private key
.................................+++++
.....................................................+++++
writing new private key to 'ssl/logstash-forwarder.key'
-----
```

Now, we are going to create new configuration files for Logstash named 'filebeat-input.conf' as input file from filebeat 'syslog-filter.conf' for system logs processing, and 'output-elasicsearch.conf' file to define Elasticsearch output.

Navigate to Logstash directory create a file 'filebeat-input.conf' in conf.d directory by running command

```
cd /etc/logstash/
nano conf.d/filebeat-input.conf
```

and paste the following configuration

```
input {
  beats {
    port => 5443
    type => syslog
    ssl => true
    ssl_certificate => "/etc/logstash/ssl/logstash-forwarder.crt"
    ssl_key => "/etc/logstash/ssl/logstash-forwarder.key"
  }
}
```

```
input {
  beats {
    port => 5443
    type => syslog
    ssl => true
    ssl_certificate => "/etc/logstash/ssl/logstash-forwarder.crt"
    ssl_key => "/etc/logstash/ssl/logstash-forwarder.key"
  }
}
```

For the system log data processing, we are going to use a filter plugin named ‘grok’. Create a new conf. file ‘syslog-filter.conf in the same directory

```
nano conf.d/syslog-filter.conf
```

and paste the following configuration lines

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %
{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

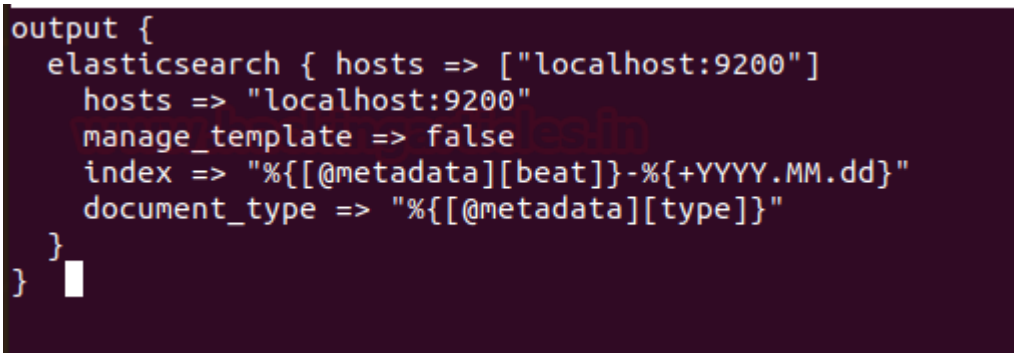
```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:sys
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

And at last create a configuration file ‘output-elasticsearch.conf’ for the output of elasticsearch.

nano conf.d/output-elasticsearch.conf

and paste the following configuration

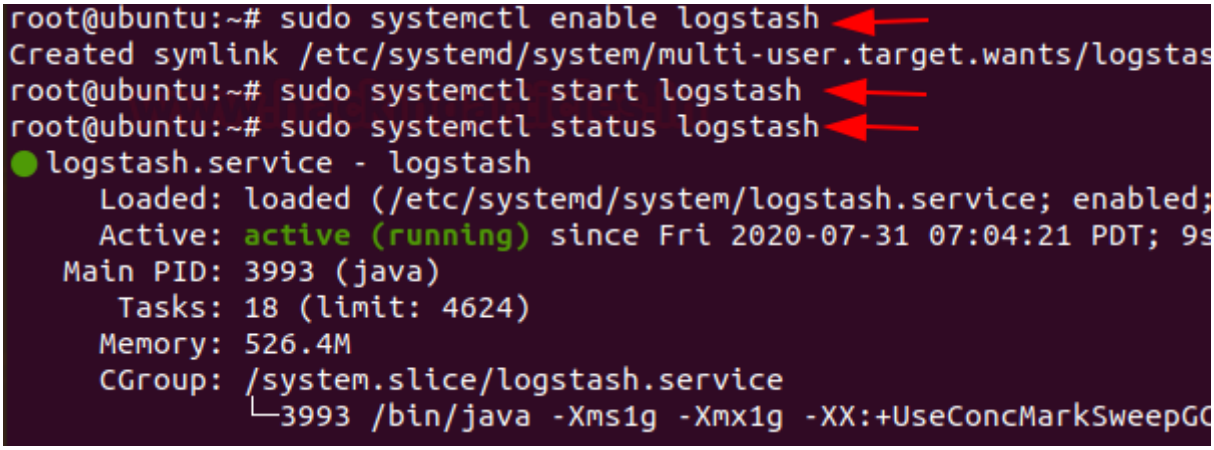
```
output {
  elasticsearch { hosts => ["localhost:9200"]
    hosts => "localhost:9200"
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
  }
}
```



And at last, save and exit.

Now start, enable & verify the status of Logstash service.

```
systemctl start logstash
systemctl enable logstash
systemctl status logstash
```



Install and configure Kibana

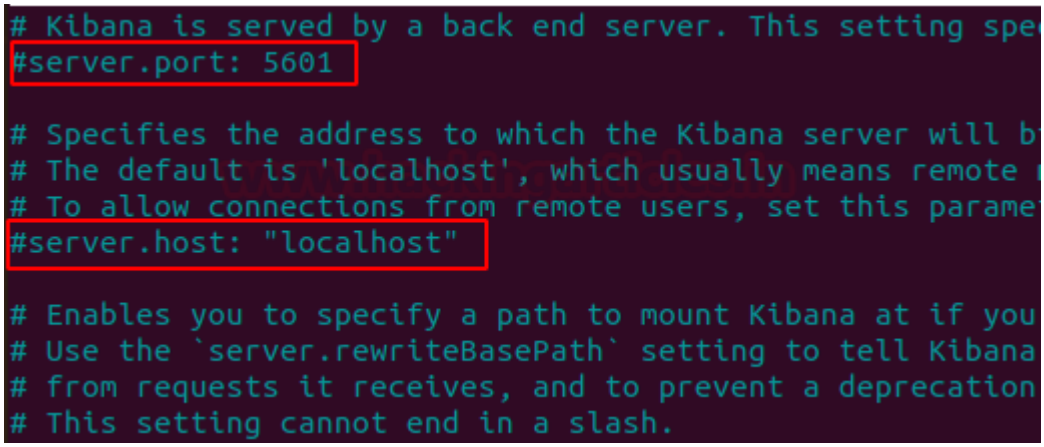
Install Kibana by using the following command

```
apt install kibana
```

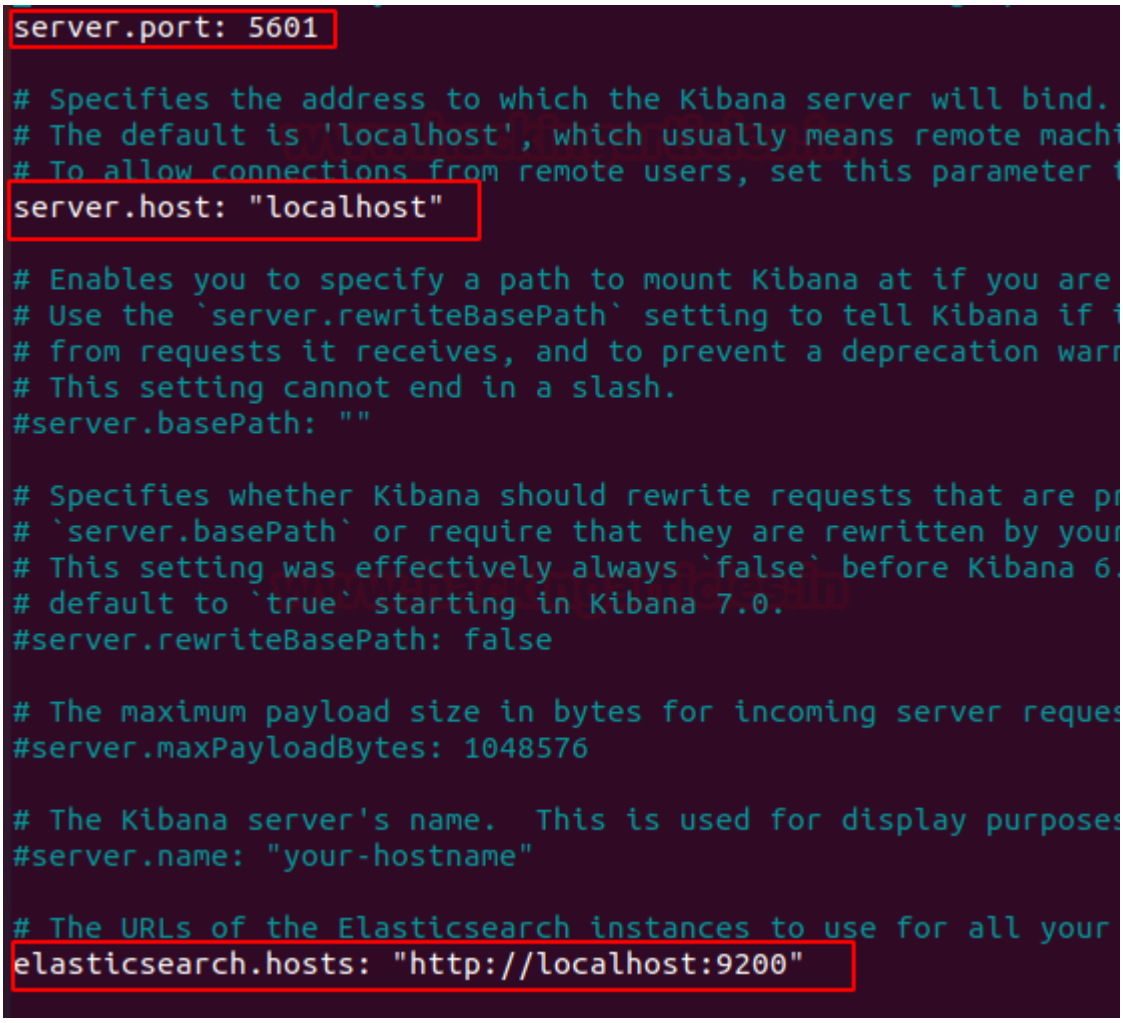
```
root@ubuntu:~# apt install kibana
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed a
  libllvm9 linux-headers-5.4.0-26 linux-headers-5.4.0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 0 not
Need to get 346 MB of archives.
```

We are going to do some modifications to the kibana configuration file.

`nano /etc/kibana/kibana.yml`

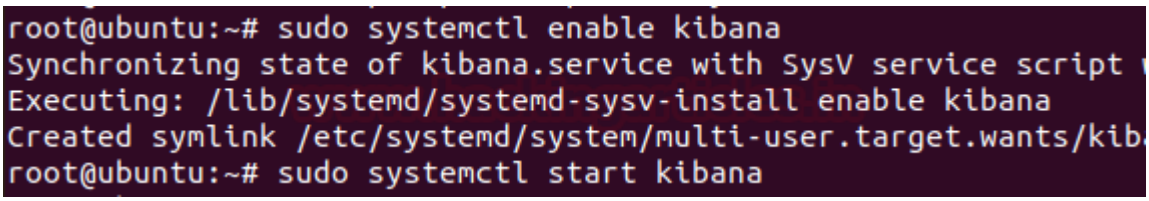


Locate and uncomment the following Attributes



Now start & enable the kibana service:

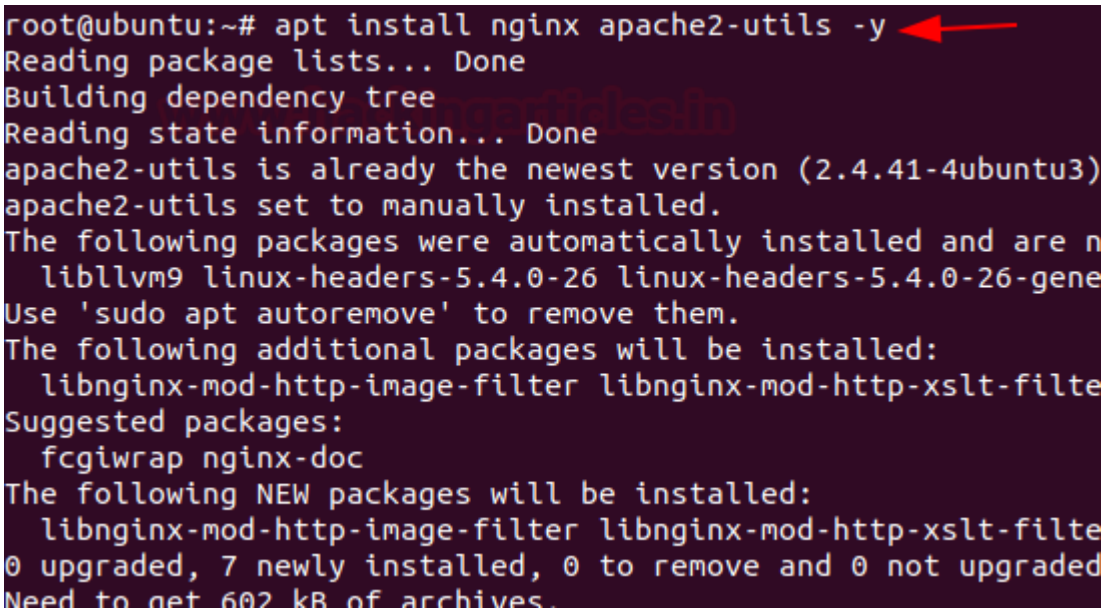
```
systemctl enable kibana
systemctl start kibana
```



Install and configure NGINX

Install Nginx and ‘Apache2-utlis’

```
apt install nginx apache2-utils -y
```



Now, create a new virtual host file named Kibana.

```
nano /etc/nginx/sites-available/kibana
```

and paste the following configuration Into the file.

```
server {
    listen 80;
    server_name localhost;
    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/.kibana-user;
    location / {
        proxy_pass https://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
server {
    listen 80;

    server_name localhost;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/.kibana-user;

    location / {
        proxy_pass https://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Let's create authentication for the Kibana Dashboard and activate the Kibana virtual host configuration and test Nginx configuration after that enable & restart the Nginx service by using the following command.

```
sudo htpasswd -c /etc/nginx/.kibana-user elastic
ln -s /etc/nginx/sites-available/kibana /etc/nginx/sites-enabled/
nginx -t
systemctl enable nginx
systemctl restart nginx
```

```
root@ubuntu:~# sudo htpasswd -c /etc/nginx/.kibana-user elastic
New password:
Re-type new password:
Adding password for user elastic
root@ubuntu:~# ln -s /etc/nginx/sites-available/kibana /etc/nginx/sites-enabled/
root@ubuntu:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu:~# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/syste
Executing: /lib/systemd/systemd-sysv-install enable nginx
```

Install and configure Filebeat

We're going to configure filebeat data shippers on our elk-master server. This will be used to collect data from various sources and transport them to Logstash and Elasticsearch.

Download & Install filebeat by running the following command.

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.8.11-amd64.deb
```

```
root@ubuntu:~# curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.8.11-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.5M  100 11.5M    0     0   745k      0  0:00:15  0:00:15 --:--:-- 1217k
```

Let's repackage the downloaded file by using the following command:

```
sudo dpkg -i filebeat-6.8.11-amd64.deb
```

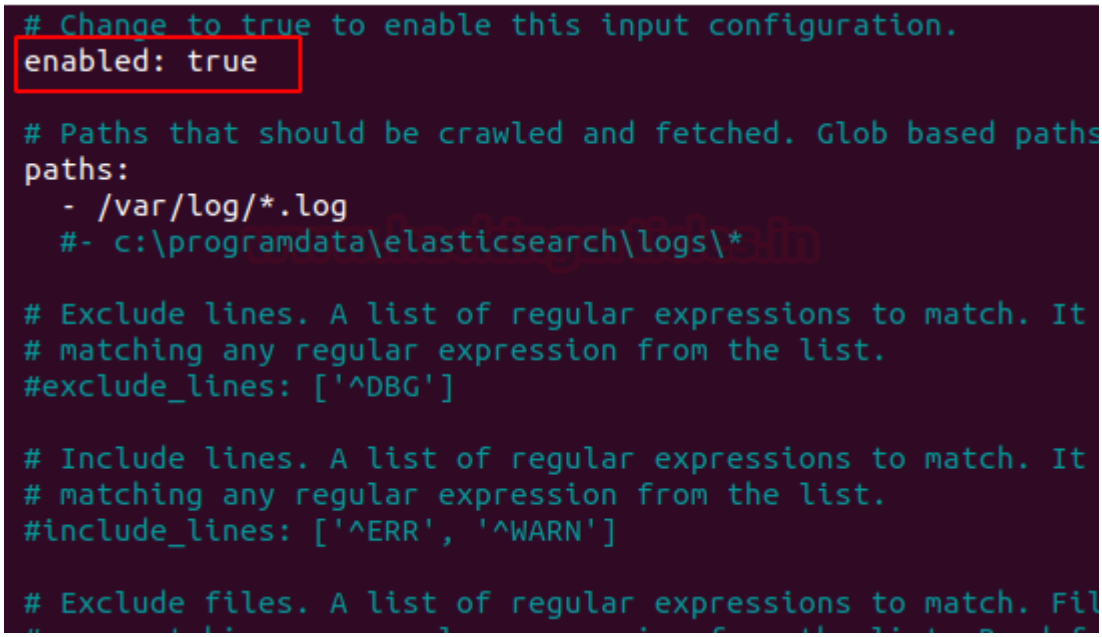
```
root@ubuntu:~# sudo dpkg -i filebeat-6.8.11-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 267530 files and directories currently installed.)
Preparing to unpack filebeat-6.8.11-amd64.deb ...
Unpacking filebeat (6.8.11) ...
Setting up filebeat (6.8.11) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
root@ubuntu:~#
```

Next, open the filebeat configuration file named ‘filebeat.yml’

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

Edit the configuration file:

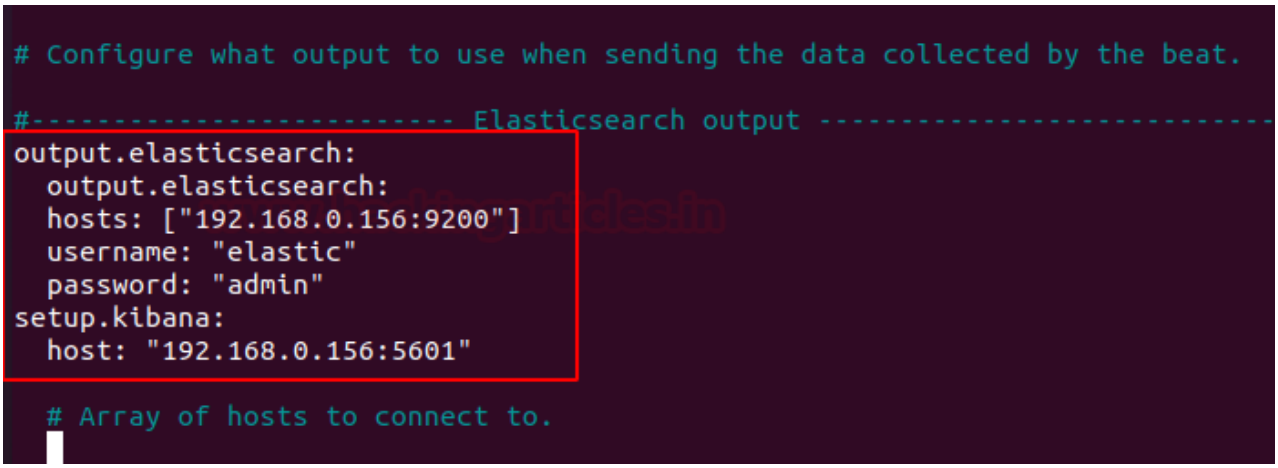
we’re going to use Elasticsearch to perform additional processing on data collected by filebeat. Therefore, Enable the filebeat prospectors by changing the ‘enabled’ line value to ‘true’.



Next head to the Elasticsearch output section and add the following lines

output.elasticsearch:

```
hosts: ["192.168.0.156:9200"]
username: "elastic"
password: "123"
setup.kibana:
host: "192.168.0.156:5601"
```



Enable and configure the Elasticsearch module by running following command

sudo filebeat modules enable elasticsearch

Let's start filebeat

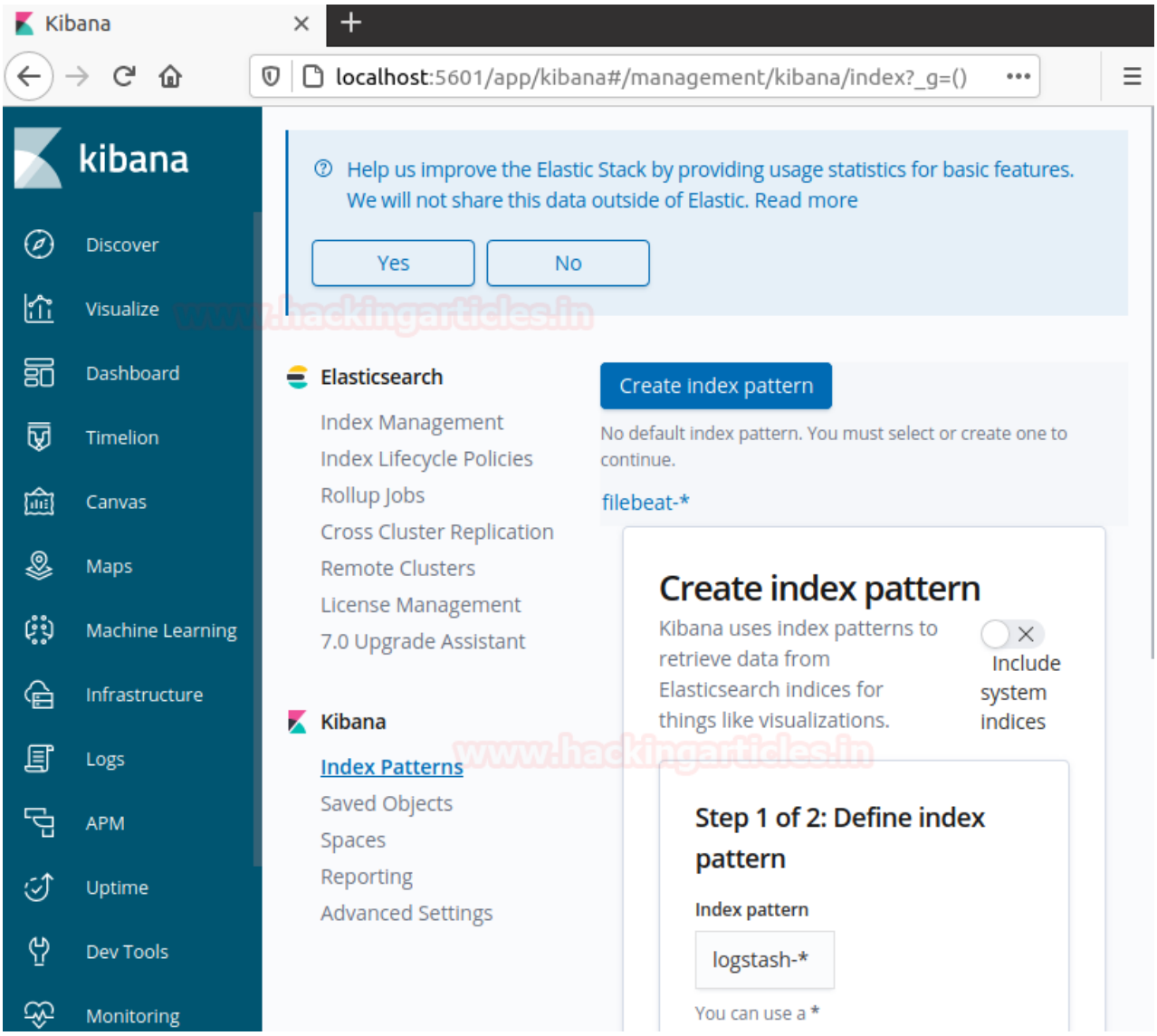
```
sudo filebeat setup
sudo service filebeat start
```

And at last copy the Logstash certificate file – logstash-forwarder.crt – to /etc/filebeat directory by running following command:

```
cp /etc/logstash/ssl/logstash-forwarder.crt /etc/filebeat/
sudo service filebeat restart
```

To test ELK stack open your browser and browse your server ip address followed by port 5601

```
https://localhost:5601
```



Routing Linux Logs to Elasticsearch

We're routing logs from rsyslog to Logstash and these logs transferred to Elasticsearch automatically

Routing From Logstash To Elasticsearch

Before routing logs from rsyslog to Logstash firstly we need to set up log forwarding between Logstash and Elasticsearch.

To do this we're going to create a configuration file for Logstash. To create configuration file head over towards the directory /etc/logstash/conf.d and create a logstash.conf file

```
cd /etc/logstash/conf.d
nano logstash.conf
```

paste the following configuration into the logstash.conf file

```
input
{

  udp
{

    host =>
"127.0.0.1"
    port => 10514

    codec =>
"json"
    type =>
"rsyslog"
  }

}


# The Filter pipeline stays empty here, no formatting is done.
filter { }



# Every single log will be forwarded to ElasticSearch. If you are using another
port, you should specify it
here.

output
{

  if [type] == "rsyslog"
{
    elasticsearch {

      hosts => [ "127.0.0.1:9200"
]

}

  }
  }
```

```
root@ubuntu:~# cd /etc/logstash/conf.d
root@ubuntu:/etc/logstash/conf.d# cat logstash.conf
input {
  udp {
    host => "127.0.0.1"
    port => 10514
    codec => "json"
    type => "rsyslog"
  }
}


# The Filter pipeline stays empty here, no formatting is done.
filter { }


# Every single log will be forwarded to ElasticSearch. If you are using

output {
  if [type] == "rsyslog" {
    elasticsearch {
      hosts => [ "127.0.0.1:9200" ]
    }
  }
}
root@ubuntu:/etc/logstash/conf.d# systemctl restart logstash
```

Restart the Logstash service.

```
systemctl restart logstash
```

Let's check that everything is running correctly issue the following command:

```
netstat -na | grep 10514
```

```
root@ubuntu:~# systemctl restart logstash
root@ubuntu:~# netstat -na | grep 10514
tcp6       0      0 127.0.0.1:10514         :::*                    LISTEN
```

Routing from rsyslog to Logstash

Rsyslog has the capacity to transform logs using templates in order to forward logs in rsyslog, head over to the directory /etc/rsylog.d and create a new file named 70-output.conf

```
cd /etc/rsyslog.d
nano 70-output.conf
```

And paste the following configuration into the 70-output.conf file

```
# This line sends all lines to defined IP address at port 10514
# using the json-template format.
*.*                         @127.0.0.1:10514;json-template
```

```
root@ubuntu:~# cd /etc/rsyslog.d
root@ubuntu:/etc/rsyslog.d# nano 70-output.conf
root@ubuntu:/etc/rsyslog.d# cat 70-output.conf
# This line sends all lines to defined IP address at port 10514
# using the json-template format.

*.*                         @127.0.0.1:10514;json-template

root@ubuntu:/etc/rsyslog.d#
```

Now we have log forwarding, create a 01-json-template.conf file in the same folder

```
nano 01-json-template.conf
```

And paste the following configuration into the 01-json-template.conf file

```
template(name="json-template"
  type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")     property(name="timereported"
dateFormat="rfc3339")
      constant(value="\",\"@version\":\"1")
      constant(value="\",\"message\":\"")     property(name="msg" format="json")
      constant(value="\",\"sysloghost\":\"")  property(name="hostname")
      constant(value="\",\"severity\":\"")    property(name="syslogseverity-text")
      constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
      constant(value="\",\"programname\":\"") property(name="programname")
      constant(value="\",\"procid\":\"")      property(name="procid")
    constant(value="\"}\n")
}
```

Restart rsyslog service and verify that logs are correctly forwarded into Elasticsearch.

```
systemctl restart rsyslog
curl -XGET 'http://localhost:9200/logstash-*/_search?q=*&pretty'
```

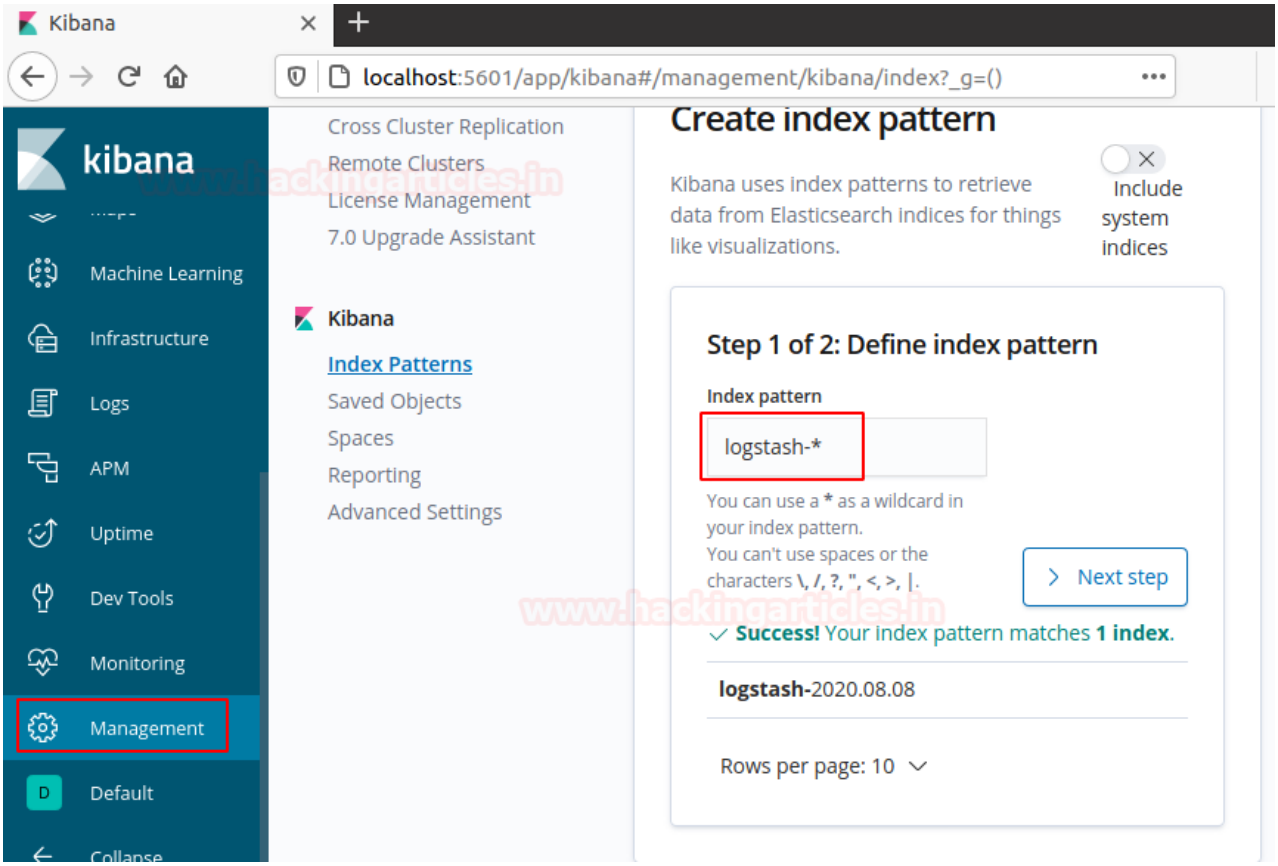
```
root@ubuntu:~# cd /etc/rsyslog.d
root@ubuntu:/etc/rsyslog.d# nano 01-json-template.conf
root@ubuntu:/etc/rsyslog.d# systemctl restart rsyslog
root@ubuntu:/etc/rsyslog.d# cat 01-json-template.conf
template(name="json-template"
  type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
      constant(value="\",\"@version\":\"1")
      constant(value="\",\"message\":\"")     property(name="msg" format="json")
      constant(value="\",\"sysloghost\":\"")  property(name="hostname")
      constant(value="\",\"severity\":\"")    property(name="syslogseverity-text")
      constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
      constant(value="\",\"programname\":\"") property(name="programname")
      constant(value="\",\"procid\":\"")      property(name="procid")
    constant(value="\"}\n")
}
root@ubuntu:/etc/rsyslog.d# systemctl restart rsyslog
root@ubuntu:/etc/rsyslog.d#
```

Note:- Logs will be forwarded in an index named logstash-*.

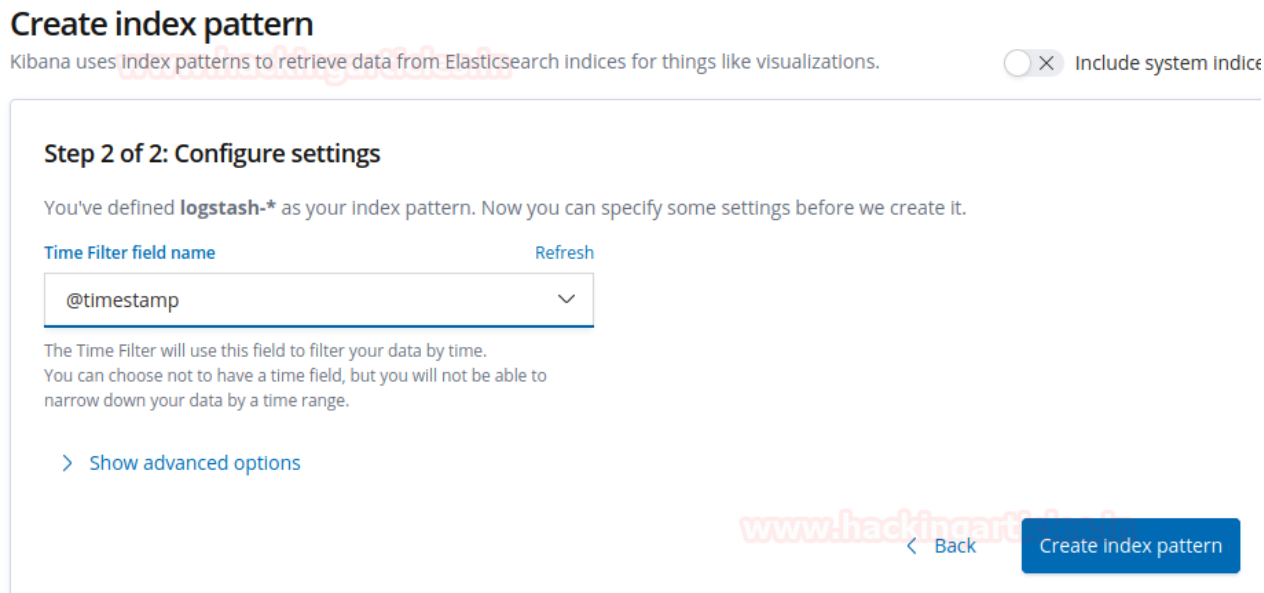
Create a Log Dashboard in Kibana

Open your browser and head over to https://localhost:5601 and you should see the following screen.

Go to the management section and create an index pattern called logstash-* and proceed for the next step.



we've defined logstash-* as our index pattern. Now we can specify some settings before we create it. In the field of time filter field name choose @timestamp and create an index pattern



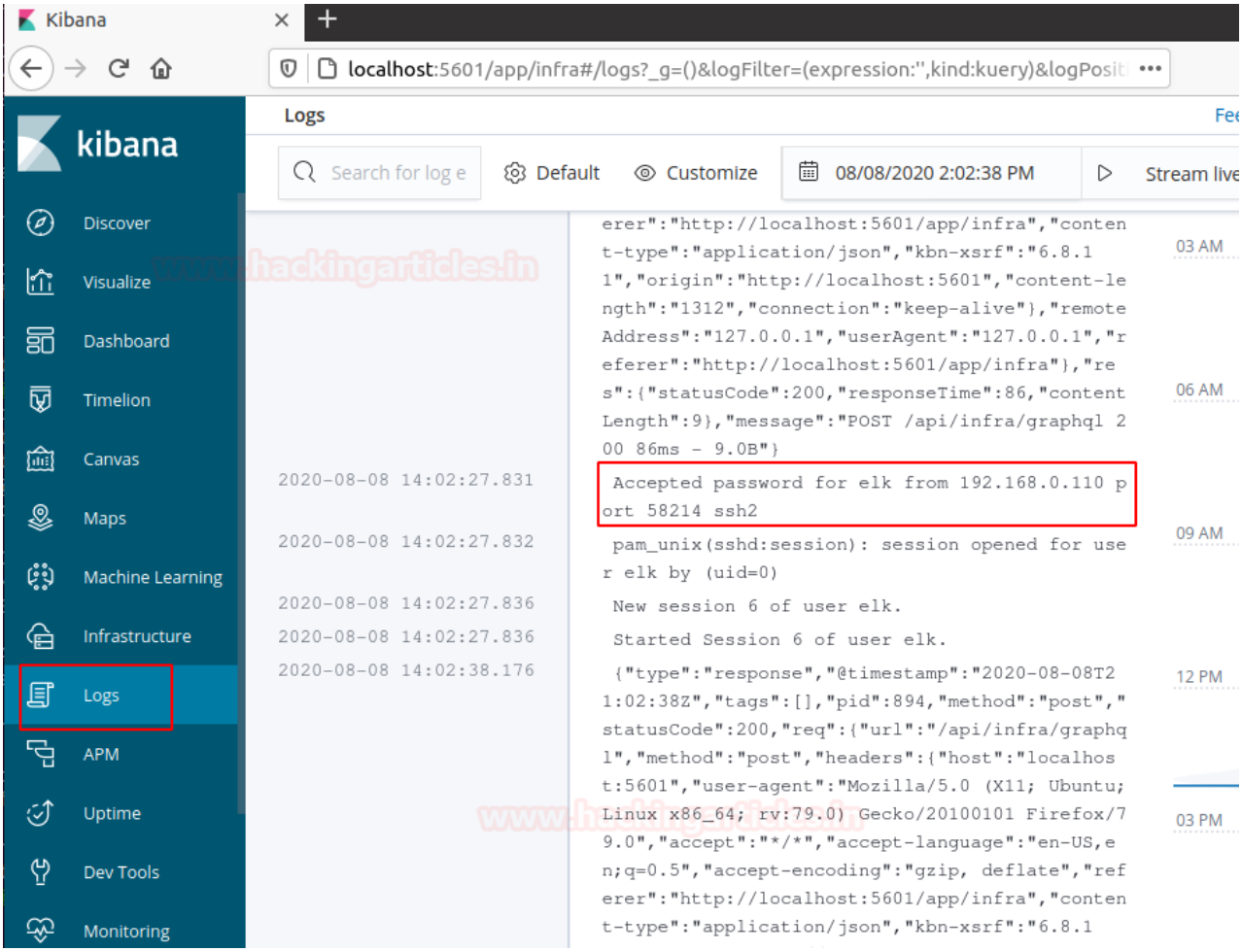
Monitoring SSH entries

This one is a little bit special, as we can go into the “Discover” tab in order to build our panel.

When entering the discover tab, select logstash-*

From there, in the fiterbar, put a query filter “programename:ssh*”.

Now we can see every log related to the SSHd service in our machine.



As we can see, now we have direct access to every log related to the SSHd service. we can for example track illegal access attempts or wrong logins.

Similarly, we can monitor various illegal access attempts or wrong logins like ftp, telnet etc…

For example, I took Telnet access to my server from a different machine.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.102  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe7b:ddc6  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:7b:dd:c6  txqueuelen 1000  (Ethernet)
        RX packets 106  bytes 13897 (13.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 95  bytes 7296 (7.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 796 (796.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 796 (796.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# telnet 192.168.0.156
Trying 192.168.0.156 ...
Connected to 192.168.0.156.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ubuntu login: elk
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic

   https://microk8s.io/ has docs and details.

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Aug  8 14:12:42 PDT 2020 from 192.168.0.102 on pts/2
elk@ubuntu:~$
```

Let’s check what happens on the Kibana dashboard.

Hold tight!

	- 9.0B"}
2020-08-08 14:12:37.288	connect from 192.168.0.102 (192.168.0.102)
2020-08-08 14:12:37.366	Server returned error NXDOMAIN, mitigating potential DNS
2020-08-08 14:12:40.936	pam_unix(login:auth): Couldn't open /etc/securetty: No s
2020-08-08 14:12:42.088	pam_unix(login:auth): Couldn't open /etc/securetty: No s
2020-08-08 14:12:42.212	pam_unix(login:session): session opened for user elk by
2020-08-08 14:12:42.223	New session 8 of user elk.
2020-08-08 14:12:42.226	Started Session 8 of user elk.

Nice! Now your panel is included in your dashboard.

Author – Vijay is a Certified Ethical Hacker, Technical writer and Penetration Tester at Hacking Articles. Technology and Gadget freak. Contact **Here**