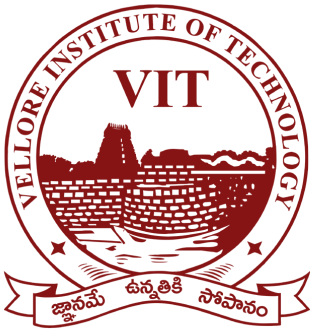
Report On Elastic Stack and FileBeats

Deploying ELK Stack and File Beats in Cloud and Local

Done By:
P Rama Ramana Sharma 20BCN7060
Thenipalli Chandu 20BCN7079
Kondasani Rama Sai 20BCN7049



VIT-AP UNIVERSITY

Project Statement

The goal of this project is to deploy an ELK (Elasticsearch, Logstash, Kibana) stack along with Filebeat to perform User Behaviour Analysis. User Behaviour Analysis is the process of analyzing user interactions and behaviors to gain insights that can be used to improve user experience and engagement.

To achieve this, we will set up Elasticsearch as the central repository for storing logs and data generated by the user interactions. Logstash will be used to collect, process, and filter the data generated by the user interactions, and Filebeat will be used to forward the logs from different sources to the Logstash.

We will use Kibana to visualize and analyze the data in real-time. Kibana will be used to create dashboards and visualizations that can be used to gain insights into user behavior, such as the most commonly accessed pages, the duration of visits, and the paths taken through the site.

The implementation of this project will enable the team to identify and analyze user behaviors and interactions with the site in real-time. This will help the team to identify areas of improvement and optimize the site's user experience, leading to increased user engagement and satisfaction.

What is UBA?

User behaviour analytics (UBA) involves collecting and analysing data on how users interact with a system, application, or website to gain insights into their behaviour patterns.
Here are the general steps for performing user behaviour analytics:

Define the goals and objectives: Determine what you want to achieve by analysing user behaviour. This may include improving user experience, increasing engagement, or identifying potential security threats.

Identify the data sources: Determine which data sources you will use to collect user behaviour data. This may include website analytics tools, server logs, user feedback surveys, or social media listening tools.

Collect and consolidate the data: Collect the data from the various sources and consolidate it into a centralised repository.

Analyse the data: Use data analysis tools to identify patterns and trends in the user behaviour data. Look for correlations between different data points and identify any outliers or anomalies.

Create user behaviour profiles: Use the insights gained from the analysis to create user behaviour profiles that describe the typical behaviour patterns of different user groups.

Identify opportunities for improvement: Use the user behaviour profiles to identify areas where user experience or engagement can be improved. This may involve tweaking website design, adding new features, or modifying existing processes.

Continuously monitor and improve: Continuously monitor user behaviour data to identify new trends or changes in behaviour patterns. Use this information to further refine the user behaviour profiles and identify new opportunities for improvement.

What are the tools?

There are several open source tools available for User Behaviour Analytics (UBA). Here are some of the popular ones:

Apache Metron: Apache Metron is a real-time, open-source cybersecurity platform that uses big data technologies to perform security analytics. It provides UBA capabilities by analysing user and entity behaviour to detect anomalies and threats

Wazuh: Wazuh is a free, open-source security monitoring solution that includes UBA capabilities. It uses machine learning algorithms to detect anomalies in user behaviour and identify potential security threats.

osquery: osquery is a free, open-source endpoint monitoring solution that includes UBA capabilities. It allows you to query and analyse data from your endpoints to detect anomalies in user behaviour and identify potential security threats.

Security Onion: Security Onion is a free, open-source network security monitoring solution that includes UBA capabilities. It uses machine learning algorithms to analyse user and entity behaviour to detect anomalies and threats.

Elastic Stack: Elastic Stack (formerly known as ELK Stack) is a free, open-source platform for real-time data analysis and visualisation. It includes a range of tools that can be used for UBA, including Elasticsearch, Logstash, and Kibana.

How ELK Stack?

The ELK stack is a collection of three open-source tools - Elasticsearch, Logstash, and Kibana - that work together to collect, store, and analyze data. Here is how each tool works:

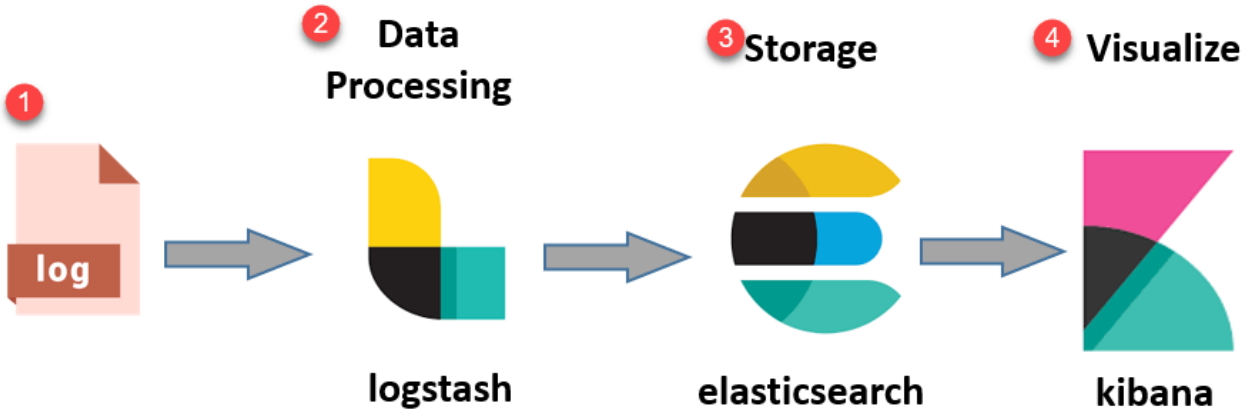
1. **Elasticsearch**: Elasticsearch is a search and analytics engine that provides a distributed, real-time search and analytics platform. It stores data in a distributed index, which allows for fast, real-time search and analysis of data. Elasticsearch is highly scalable, and it can handle a large volume of data with ease.

2. **Logstash**: Logstash is a data pipeline that collects, filters, and transforms data from different sources before sending it to Elasticsearch. It can collect data from different sources such as log files, databases, and message queues. Logstash can also perform data transformation and filtering to ensure that only the relevant data is sent to Elasticsearch.

3. **Kibana**: Kibana is a data visualization tool that allows users to create dashboards, visualizations, and reports based on data stored in Elasticsearch. Kibana provides a web interface that allows users to interact with the data and visualize it in various formats such as tables, graphs, and maps.

When used together, Elasticsearch, Logstash, and Kibana form a complete data analysis solution. Logstash collects data from different sources and filters and transforms it before sending it to Elasticsearch for storage.

Elasticsearch stores the data and provides fast, real-time search and analysis capabilities. Kibana provides a user-friendly interface to visualize and analyze the data stored in Elasticsearch.
Overall, the ELK stack is a powerful tool for collecting, storing, and analyzing data in real-time. It can be used for various use cases such as log analysis, security analysis, and business intelligence.



What are Beats

Beats is a platform for lightweight data shippers that are designed to send various types of data to the Elasticsearch or Logstash for processing, indexing, and visualization. Beats is part of the Elastic Stack and is used to collect data from various sources such as logs, metrics, and network packets. Beats are easy to install, lightweight, and have a small footprint, making them ideal for use in various distributed systems.
Beats consists of four different types of data shippers, each designed for a specific purpose:

1. **Filebeat**: Filebeat is used to collect log data from various sources, including log files and the standard output of applications. Filebeat is lightweight and efficient and can be used to send data to Elasticsearch or Logstash for processing and analysis.
2. **Metricbeat**: Metricbeat is used to collect metric data from various sources, including servers, applications, and operating systems. Metricbeat is designed to collect real-time metrics and can be used to monitor system performance and identify issues.
3. **Packetbeat**: Packetbeat is used to monitor network traffic and capture network data in real-time. It can be used to identify and diagnose network performance issues, track application usage, and detect security threats.

4. **Auditbeat**: Auditbeat is used to collect audit data from various sources, including operating systems, applications, and security logs. It can be used to monitor and track system activity, detect suspicious behavior, and comply with regulations and standards.

Overall, Beats is an important component of the Elastic Stack and provides a lightweight, efficient way to collect and ship data from various sources. With its different types of data shippers, Beats can be used for various use cases such as log analysis, system monitoring, network analysis, and security analysis.

How ELK Stack Works with Beats

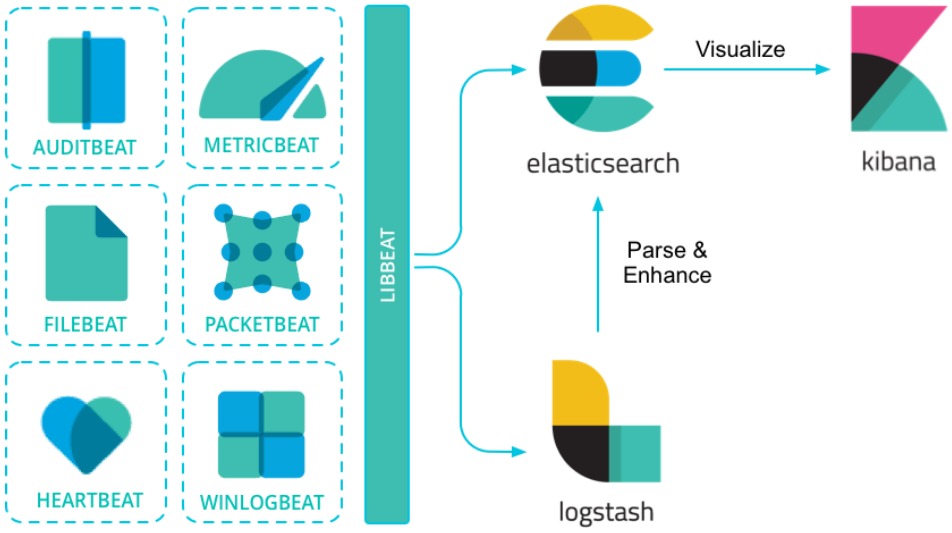
ELK stack works with Beats to collect, process, and analyze data from various sources in real-time. Beats are lightweight data shippers that collect and send data to Logstash or Elasticsearch for further processing, indexing, and visualization.

Beats can be configured to collect data from various sources such as logs, metrics, network packets, and audit data. Once the data is collected, Beats can transform and filter the data before sending it to Logstash or Elasticsearch. This ensures that only relevant data is sent to the ELK stack for further analysis.

Logstash, which is a part of the ELK stack, can be used to process and filter the data collected by Beats. Logstash provides a set of plugins that can be used to parse, filter, and transform the data before sending it to Elasticsearch for indexing. This enables more efficient data analysis and indexing, as Logstash can handle complex data transformations and enrichments.

Once the data is processed and indexed by Elasticsearch, Kibana can be used to visualize and analyze the data. Kibana provides a user-friendly interface to create dashboards, visualizations, and reports based on data stored in Elasticsearch. This allows users to gain insights and monitor system performance in real-time.

Overall, ELK stack and Beats work together to provide a complete data analysis solution that can be used for various use cases such as log analysis, system monitoring, network analysis, and security analysis. Beats provide an efficient and lightweight way to collect data, while ELK stack provides a scalable, real-time search and analytics platform for processing, indexing, and visualizing the data.



Install and Configure ELK Stack on Ubuntu

Step 1: Check the OS version by using the below command

```
ramz@ramz:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.1 LTS
Release:        22.04
Codename:       jammy
```

Step 2: Install the dependency Java environment packages by using the below command
apt install default-jdk default-jre -y

```
ramz@ramz:~$ sudo apt install default-jdk default-jre -y
[sudo] password for ramz:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
default-jdk is already the newest version (2:1.11-72build2).
default-jre is already the newest version (2:1.11-72build2).
The following packages were automatically installed and are no longer required:
  g++-11 libflashrom1 libftdi1-2 libllvm13 libstdc++-11-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
```

Step 3: Check the Installed Java Version by using the below command
javac -version

```
ramz@ramz:~$ javac -version
javac 11.0.18
ramz@ramz:~$
```

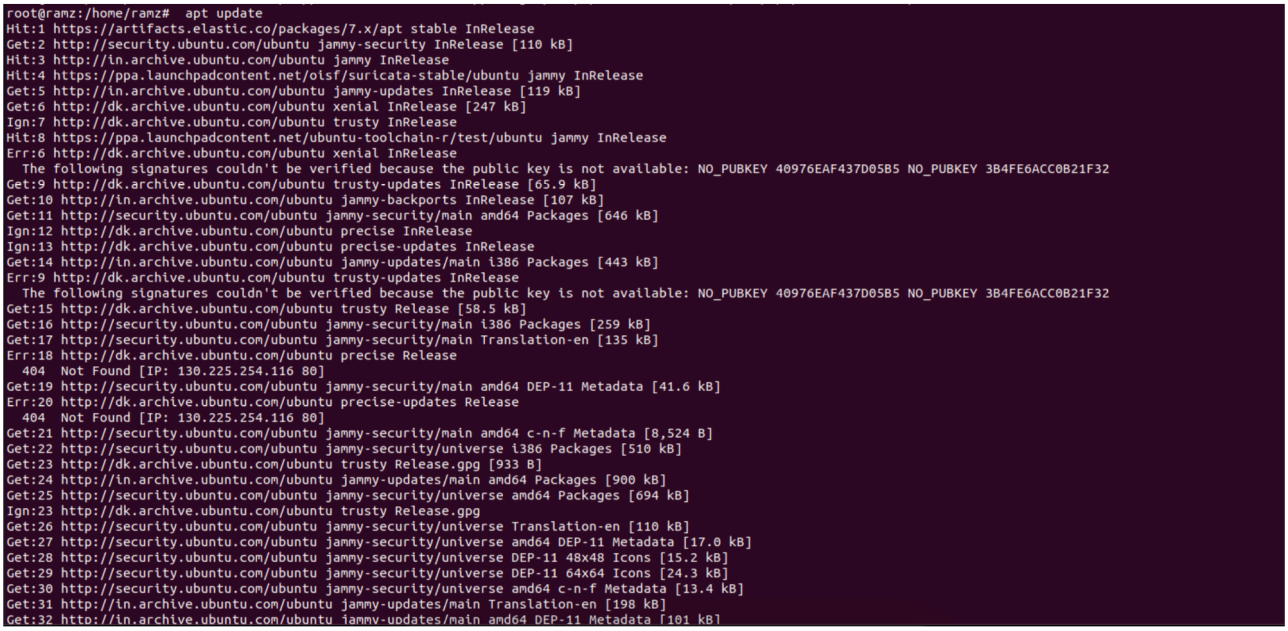
Step 4: Add the elasticsearch APT repository key by using the below command (run these commands in root privilege).
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -

```
root@ramz:/home/ramz# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
curl: (56) OpenSSL SSL_read: error:0A000126:SSL routines::unexpected eof while reading, errno 0
gpg: no valid OpenPGP data found.
root@ramz:/home/ramz#
```

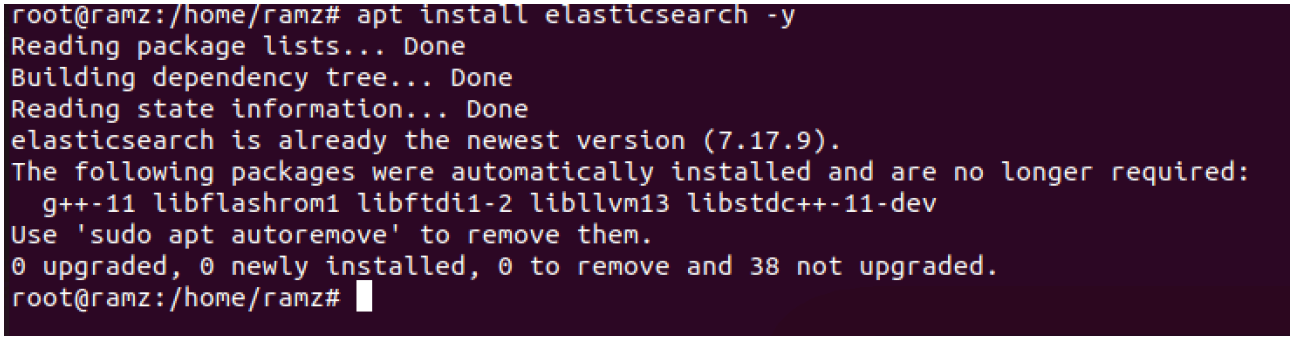
Step 5: Add the Elastic Search to the APT source List by using the below command
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/apt/sources.list.d/elastic-7.x.list

```
root@ramz:/home/ramz# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/apt/sources.list.d/elastic-7.x.list
root@ramz:/home/ramz#
```

Step 6: Update the APT source list by using the below command
apt update



Step 7: Install the Elastic Search by using the below command
apt install elasticsearch -y



Step 8: Configure the elastic search by using the below command
vim /etc/elasticsearch/elasticsearch.yml
Change the network.host and http.port as per the screenshot

```
# ---------------------------------- Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- 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: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
                                                                    69,1          56%
```

Step 9: Configure the JVM heap memory by using the below command
vim /etc/elasticsearch/jvm.options

```
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## and the min and max should be set to the same value. For
## example, to set the heap to 4 GB, create a new file in the
## jvm.options.d directory containing these lines:
##
## -Xms4g
## -Xmx4g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/heap-size.html
## for more information
##
################################################################


################################################################
## Expert settings
################################################################
##
## All settings below here are considered expert settings. Do
## not adjust them unless you understand what you are doing. Do
## not edit them in this file; instead, create a new file in the
## jvm.options.d directory containing your adjustments.
##
################################################################

-Xms512m
-Xmx512m

## GC configuration
8-13:-XX:+UseConcMarkSweepGC
8-13:-XX:CMSInitiatingOccupancyFraction=75
8-13:-XX:+UseCMSInitiatingOccupancyOnly

## G1GC Configuration
# NOTE: G1 GC is only supported on JDK version 10 or later
# to use G1GC, uncomment the next two lines and update the version on the
# following three lines to your version of the JDK
# 10-13:-XX:-UseConcMarkSweepGC
# 10-13:-XX:-UseCMSInitiatingOccupancyOnly
14-:-XX:+UseG1GC

## JVM temporary directory
                                                                    62,2
```

Step 10: Restart the Elastic Search by using the below command
systemctl restart elasticsearch

Step 11: Enable the Elastic Search to start on boot by using the below command

```
root@ramz:/home/ramz# systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
root@ramz:/home/ramz#
```

Step 12: Ping the Elastic Search to verify installation by using the below command

```
root@ramz:/home/ramz# curl -X GET "localhost:9200"
{
  "name" : "ramz",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "ly8j6Lf7SSCPaT-bs-876w",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef48222227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "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"
}
root@ramz:/home/ramz#
```

Step 13: Install the Logstash by using the below command

```
root@ramz:/home/ramz# apt install logstash -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
logstash is already the newest version (1:7.17.9-1).
The following packages were automatically installed and are no longer required:
  g++-11 libflashrom1 libftdi1-2 libllvm13 libstdc++-11-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 36 not upgraded.
root@ramz:/home/ramz#
```

Step 14: Start the Logstash Service by using the below command
systemctl start logstash

Step 15: Enable the Logstash Service to start on boot by using the below command
systemctl enable logstash

Step 16: Check the status of the Logstash Service by using the below command

```
root@ramz:/home/ramz# systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-02 14:17:19 IST; 16s ago
   Main PID: 3283 (java)
      Tasks: 15 (limit: 7022)
     Memory: 350.3M
        CPU: 26.242s
     CGroup: /system.slice/logstash.service
             └─3283 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -Djava.awt.>

Mar 02 14:17:19 ramz systemd[1]: Started logstash.
Mar 02 14:17:19 ramz logstash[3283]: Using bundled JDK: /usr/share/logstash/jdk
Mar 02 14:17:19 ramz logstash[3283]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
lines 1-13/13 (END)
```

Step 17: Install the Kibana by using the below command

```
root@ramz:/home/ramz# apt install kibana -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kibana is already the newest version (7.17.9).
The following packages were automatically installed and are no longer required:
  g++-11 libflashrom1 libftdi1-2 libllvm13 libstdc++-11-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 36 not upgraded.
root@ramz:/home/ramz#
```

Step 18: Configure kibana in the following file by using the below command
vim /etc/kibana/kibana.yml

```
# 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: "your-hostname"

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

Step 19: Start the kibana Service by using the below command
systemctl start kibana

Step 20: Enable the kibana Service by using the below command
systemctl enable kibana

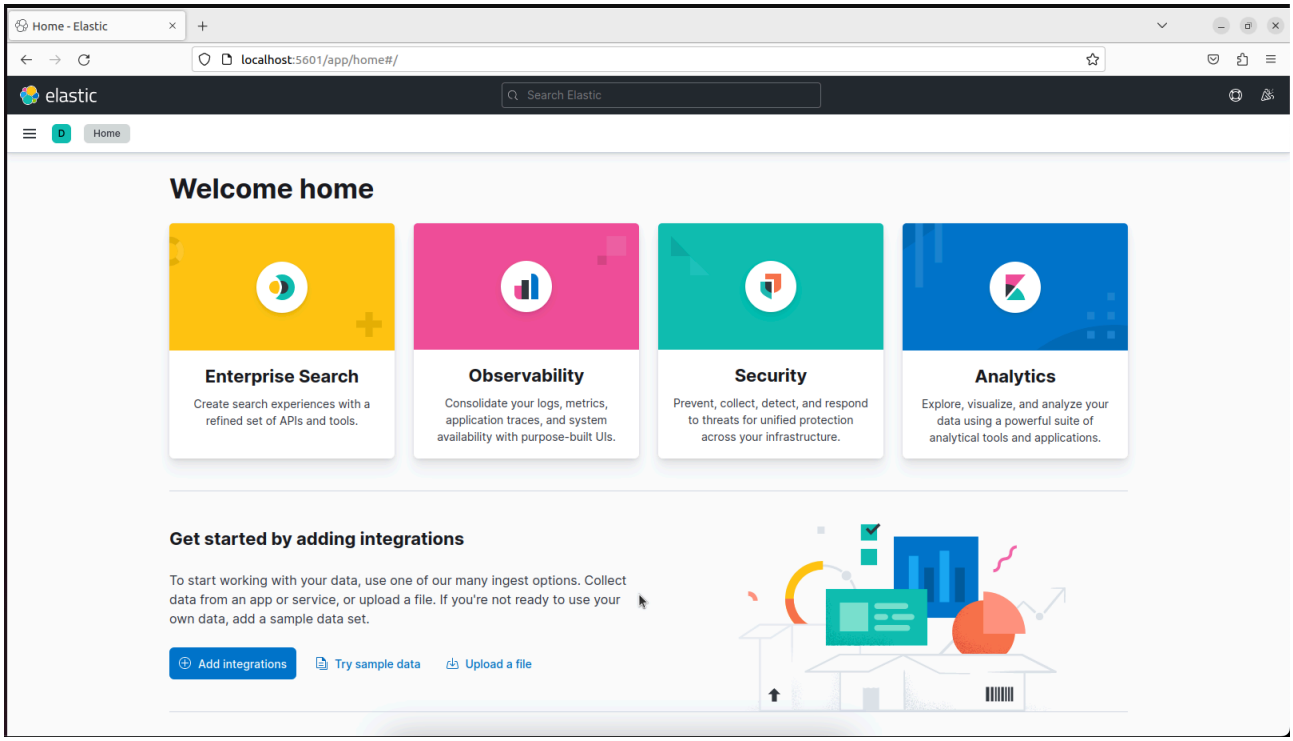
Step 21: Check the status of the kibana service by using the below command

```
root@ramz:/home/ramz# systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-02 19:40:42 IST; 5h 15min left
       Docs: https://www.elastic.co
   Main PID: 763 (node)
      Tasks: 11 (limit: 7022)
     Memory: 595.4M
        CPU: 52.578s
     CGroup: /system.slice/kibana.service
             └─763 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest=/var/log/kibana/kibana.log --pid.file=/run/kibana/kibana.pid "--

Mar 02 19:40:42 ramz systemd[1]: Started Kibana.

root@ramz:/home/ramz#
```

Step 22: Ping the http://localhost:5601 in browser to view the Dashboard of the kibana as show in the below image



Install and Configure ELK Stack on AWS EC2 Instance

As we seen before, following similar steps we can install ELK stack on cloud, the steps are as follows:

Step 1: Deploy an AWS instance of **elasticsearch** using Ubuntu choose t2.medium as Instance type.

Step 2: Connect to the above deployed instance using SSH.

Step 3: Enter sudo command and enter the below command.

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-153:~$ sudo su
root@ip-172-31-92-153:/home/ubuntu# hostnamectl set-hostname elastic
root@ip-172-31-92-153:/home/ubuntu#
```

Step 4: Update the instance using apt-get update.

```
root@ip-172-31-92-153:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [943 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [204 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.6 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [679 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [106 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [584 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [881 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [173 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [9652 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [3260 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]
```

Step 5: Install JDK using apt install default-jdk default-jre -y

```
root@ip-172-31-92-153:/home/ubuntu# apt install default-jdk default-jre -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service
  default-jdk-headless default-jre-headless fontconfig-config fonts-dejavu-core fonts-dejavu-extra
  gsettings-desktop-schemas java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
  libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2
  libllvm15 libpciaccess0 libpcsclite1 libpthread-stubs0-dev libsensors-config libsensors5 libsm-dev libsm6
  libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0
```

Step 6: Enter the command
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add - and sudo apt-get install apt-transport-https

```
root@ip-172-31-92-153:/home/ubuntu# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@ip-172-31-92-153:/home/ubuntu# sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 47 not upgraded.
Need to get 1506 B of archives.
After this operation, 169 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.8 [1506 B]
Fetched 1506 B in 0s (93.2 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 65994 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.8_all.deb ...
Unpacking apt-transport-https (2.4.8) ...
Setting up apt-transport-https (2.4.8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-92-153:/home/ubuntu#
```

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

```
root@ip-172-31-92-153:/home/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
root@ip-172-31-92-153:/home/ubuntu#
```

Step 8: Now execute apt-get update -y

```
root@ip-172-31-92-153:/home/ubuntu# apt-get update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:5 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Get:6 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [109 kB]
Fetched 348 kB in 1s (574 kB/s)
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@ip-172-31-92-153:/home/ubuntu#
```

Step 9: Now install elasticserch using the command: apt-get install elastic search

```
root@ip-172-31-92-153:/home/ubuntu# apt-get install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 47 not upgraded.
Need to get 315 MB of archives.
After this operation, 527 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.9 [315 MB]
80% [1 elasticsearch 315 MB/315 MB 100%]                                    52.0 MB/s 0s
```

Step 10: Make configuration changes in elasticsearch.yml file using:
nano /etc/elasticsearch/elasticsearch.yml and make changes as shown below. The network address is the Private IPv4 address of the instance. Save the file.

```
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 172.31.92.153
#
# 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 ----------------------------------
#
discovery.type: single-node
# 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"]
#
```

Step 11: Start the Elasticsearch using: systemctl start elasticsearch, and check the status using systemctl status elasticsearch.

```
[root@ip-172-31-92-153:/home/ubuntu#  systemctl start elasticsearch
[root@ip-172-31-92-153:/home/ubuntu# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-09 11:20:06 UTC; 1min 3s ago
       Docs: https://www.elastic.co
   Main PID: 5298 (java)
      Tasks: 58 (limit: 4689)
     Memory: 2.3G
        CPU: 45.087s
     CGroup: /system.slice/elasticsearch.service
             ├─5298 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.>
             └─5488 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Mar 09 11:19:47 elastic systemd[1]: Starting Elasticsearch...
Mar 09 11:20:06 elastic systemd[1]: Started Elasticsearch.
lines 1-14/14 (END)
```

Step 12: Check if the cluster is started using the command curl http://<IP>:9200

```
[root@ip-172-31-92-153:/home/ubuntu# curl http://172.31.92.153:9200
{
  "name" : "elastic",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "VbGsIFgNRv2FU5QQkMVUYA",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef48222227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "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"
}
root@ip-172-31-92-153:/home/ubuntu#
```

Step 13: Now deploy another instance named **logstashkibana** in Ubuntu using t2.medium.
Step 14: Connect to the above deployed instance using SSH.
Step 15: Update the instance using apt-get update.

```
ubuntu@ip-172-31-83-221:~$ sudo su
root@ip-172-31-83-221:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [687 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [141 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [8832 B]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [637 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [99.7 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [701 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [112 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [13.6 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4960 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [996 B]
```

Step 16: Install JDK using apt install default-jdk default-jre -y

```
root@ip-172-31-83-221:/home/ubuntu# apt install default-jdk default-jre -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service
  default-jdk-headless default-jre-headless fontconfig-config fonts-dejavu-core fonts-dejavu-extra
  gsettings-desktop-schemas java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
  libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2
  libllvm15 libpciaccess0 libpcsclite1 libpthread-stubs0-dev libsensors-config libsensors5 libsm-dev libsm6
  libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0
  libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3
  libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt-dev
  libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless session-migration x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
```

Step 17: Repeat Step 6,7,8
Step 18: Install Kibana using: apt-get install kibana
Step 19: Configure kibana using: vim /etc/kibana/kibana.yml and make changes as shown below.

```
# 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 value
s.
# 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: "172.31.83.221"

# 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: "your-hostname"

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://172.31.92.153:9200"]
```

The server.host is the private IP of **logstashkibana** instance and elasticsearch.hosts is the private IP of **elasticsearch** instance.

Step 20: Start Kibana using systemctl start kibana and check the status using systemctl status kibana

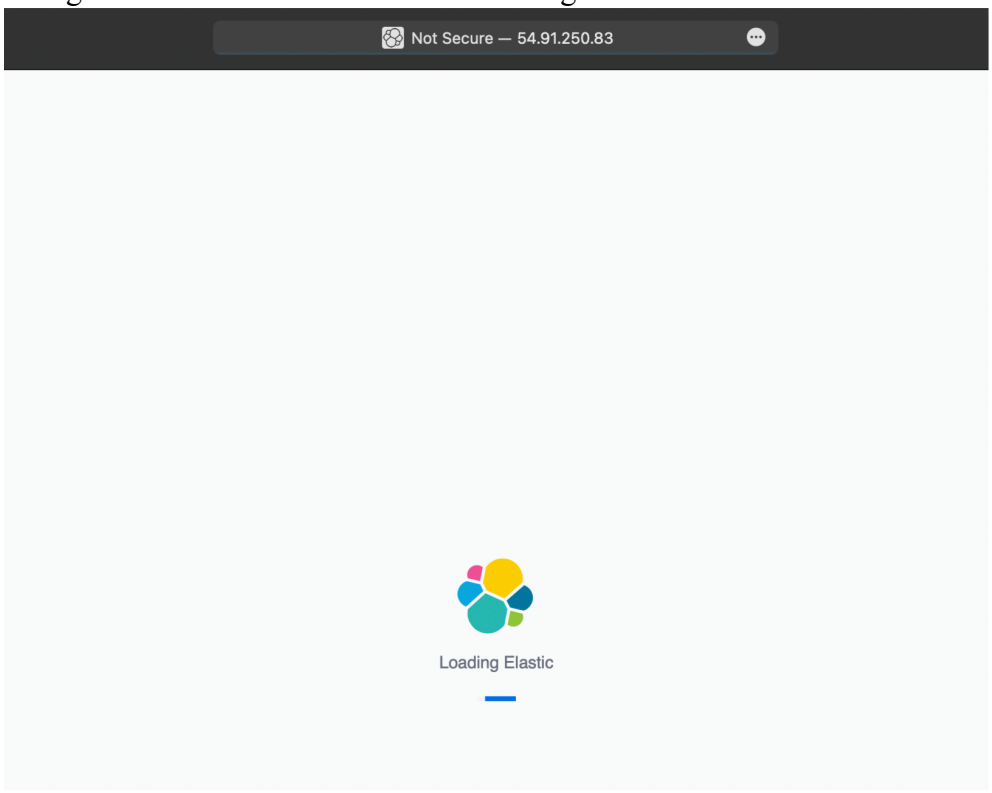
```
root@ip-172-31-83-221:/home/ubuntu# systemctl start kibana
root@ip-172-31-83-221:/home/ubuntu# systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-09 13:59:15 UTC; 9s ago
       Docs: https://www.elastic.co
   Main PID: 5141 (node)
      Tasks: 11 (limit: 4689)
     Memory: 219.5M
        CPU: 10.477s
     CGroup: /system.slice/kibana.service
             └─5141 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest=/var>

Mar 09 13:59:15 ip-172-31-83-221 systemd[1]: Started Kibana.
lines 1-12/12 (END)
root@ip-172-31-83-221:/home/ubuntu#
```

Step 21: Check the log of Kibana to confirm it is running using tail -f /var/log/kibana/kibana.log

```
root@ip-172-31-83-221:/home/ubuntu# tail -f /var/log/kibana/kibana.log
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installing resources for index .alerts-observability.uptime.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installing resources for index .alerts-observability.logs.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installing resources for index .alerts-observability.metrics.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installing resources for index .alerts-observability.apm.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installed resources for index .alerts-observability.logs.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installed resources for index .alerts-observability.uptime.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installed resources for index .alerts-observability.apm.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","ruleRegistry"],"pid":5170,"message":
"Installed resources for index .alerts-observability.metrics.alerts"}
{"type":"log","@timestamp":"2023-03-09T14:04:57+00:00","tags":["info","plugins","reporting","chromium"],"pid":5170,"m
essage":"Browser executable: /usr/share/kibana/x-pack/plugins/reporting/chromium/headless_shell-linux_x64/headless_sh
ell"}
{"type":"log","@timestamp":"2023-03-09T14:05:00+00:00","tags":["info","status"],"pid":5170,"message":"Kibana is now a
vailable (was degraded)"}
{"type":"response","@timestamp":"2023-03-09T14:05:12+00:00","tags":[],"pid":5170,"method":"get","statusCode":302,"req
":{"url":"/","method":"get","headers":{"host":"54.91.250.83:5601","upgrade-insecure-requests":"1","accept":"text/html
,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15","accept-language":"en-IN,en-GB;q=0.9,en;q=0.8
","accept-encoding":"gzip, deflate","connection":"keep-alive"},"remoteAddress":"115.244.41.200","userAgent":"Mozilla/
5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15"},"res"
:{"statusCode":302,"responseTime":22},"message":"GET / 302 22ms"}
{"type":"response","@timestamp":"2023-03-09T14:05:12+00:00","tags":[],"pid":5170,"method":"get","statusCode":302,"req
":{"url":"/spaces/enter","method":"get","headers":{"host":"54.91.250.83:5601","upgrade-insecure-requests":"1","accept
":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac O
S X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15","accept-language":"en-IN,en-GB;q=
0.9,en;q=0.8","accept-encoding":"gzip, deflate","connection":"keep-alive"},"remoteAddress":"115.244.41.200","userAgen
t":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.
1.15"},"res":{"statusCode":302,"responseTime":10},"message":"GET /spaces/enter 302 10ms"}
{"type":"response","@timestamp":"2023-03-09T14:05:12+00:00","tags":[],"pid":5170,"method":"get","statusCode":200,"req
":{"url":"/app/home","method":"get","headers":{"host":"54.91.250.83:5601","upgrade-insecure-requests":"1","accept":"t
ext/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15","accept-language":"en-IN,en-GB;q=0.9,
en;q=0.8","accept-encoding":"gzip, deflate","connection":"keep-alive"},"remoteAddress":"115.244.41.200","userAgent":"
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15
```

Step 21: Ping the kibana instance in a new tab using <PUBLIC IP>:5601



Step 22: Install Logstash using apt-get install logstash

```
root@ip-172-31-83-221:/home/ubuntu# apt-get install logstash
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 47 not upgraded.
Need to get 367 MB of archives.
After this operation, 628 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 logstash amd64 1:7.17.9-1 [367 MB]
Fetched 367 MB in 29s (12.5 MB/s)
Selecting previously unselected package logstash.
(Reading database ... 111128 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a7.17.9-1_amd64.deb ...
Unpacking logstash (1:7.17.9-1) ...
```

Step 23: Change directory to **/etc/logstash/conf.d/** and type vim apache.confg, and add the data as shown below and save it. The IP in hosts is private IP of **elasticsearch** instance.

```
input{
        beats{
                port => "5044"
        }
}
filter{
        grok{
                match => { "message" => "%{COMBINEDAPACHELOG}"}
        }
}
output{
        elasticsearch{
                hosts =>["http://172.31.92.153:9200"]
                index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        }
        stdout{
                codec => rubydebug
        }
}
~
```

Step 24: Start and check status of Logstash using systemctl start logstash , systemctl status logstash

```
[root@ip-172-31-83-221:/etc/logstash/conf.d# systemctl start logstash
[root@ip-172-31-83-221:/etc/logstash/conf.d# systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-09 14:27:47 UTC; 7s ago
   Main PID: 5495 (java)
      Tasks: 15 (limit: 4689)
     Memory: 326.2M
        CPU: 14.603s
     CGroup: /system.slice/logstash.service
             └─5495 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupanc>

Mar 09 14:27:47 ip-172-31-83-221 systemd[1]: Started logstash.
Mar 09 14:27:47 ip-172-31-83-221 logstash[5495]: Using bundled JDK: /usr/share/logstash/jdk
Mar 09 14:27:47 ip-172-31-83-221 logstash[5495]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was dep>
lines 1-13/13 (END)
```

Step 25: Check the log to confirm if Logstash is running using tail -f /var/log/logstash/logstash-plain.log

```
[root@ip-172-31-83-221:/etc/logstash/conf.d# tail -f /var/log/logstash/logstash-plain.log
[2023-03-09T14:28:12,700][INFO ][logstash.outputs.elasticsearch][main] Config is not compliant with data streams. `da
ta_stream => auto` resolved to `false`
[2023-03-09T14:28:12,702][INFO ][logstash.outputs.elasticsearch][main] Config is not compliant with data streams. `da
ta_stream => auto` resolved to `false`
[2023-03-09T14:28:12,781][INFO ][logstash.outputs.elasticsearch][main] Using a default mapping template {:es_version=
>7, :ecs_compatibility=>:disabled}
[2023-03-09T14:28:12,841][INFO ][logstash.outputs.elasticsearch][main] Installing Elasticsearch template {:name=>"log
stash"}
[2023-03-09T14:28:13,079][INFO ][logstash.javapipeline    ][main] Starting pipeline {:pipeline_id=>"main", "pipeline.
workers"=>2, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>250, "pipeline.sources"
=>["/etc/logstash/conf.d/apache.conf"], :thread=>"#<Thread:0x2cf91f3a run>"}
[2023-03-09T14:28:13,999][INFO ][logstash.javapipeline    ][main] Pipeline Java execution initialization time {"secon
ds"=>0.92}
[2023-03-09T14:28:14,026][INFO ][logstash.inputs.beats    ][main] Starting input listener {:address=>"0.0.0.0:5044"}
[2023-03-09T14:28:14,059][INFO ][logstash.javapipeline    ][main] Pipeline started {"pipeline.id"=>"main"}
[2023-03-09T14:28:14,186][INFO ][org.logstash.beats.Server][main][c460b525846ea82d53d1ebeab2c032ef06a337f00c388b65770
fda7678c6fe86] Starting server on port: 5044
[2023-03-09T14:28:14,229][INFO ][logstash.agent           ] Pipelines running {:count=>1, :running_pipelines=>[:main]
, :non_running_pipelines=>[]}
```

Step 26: Launch another instance this will be our client instance. Connect to this instance with SSH

Step 27: Update the instance using apt-get update.

```
ubuntu@ip-172-31-23-15:~$ sudo su
root@ip-172-31-23-15:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [943 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [204 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.6 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [679 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [106 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [584 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [881 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [173 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.0 kB]
```

Step 28: Install apache using apt-get install apache2

```
root@ip-172-31-23-15:/home/ubuntu# apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 47 not upgraded.
Need to get 2138 kB of archives.
After this operation, 8505 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [1
08 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.
1 [92.6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ub
untu4.22.04.1 [11.3 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.2
2.04.1 [9168 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.3 [13
45 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.3 [165
 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.3 [
```

Step 29: Install filebeats using curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.6-amd64.deb

```
root@ip-172-31-23-15:/home/ubuntu# curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.6-a
md64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 33.6M  100 33.6M    0     0  31.8M      0  0:00:01  0:00:01 --:--:-- 31.8M
```

Step 30: Run the command dpkg -i filebeat-7.17.6-amd64.deb

```
root@ip-172-31-23-15:/home/ubuntu# dpkg -i filebeat-7.17.6-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 66763 files and directories currently installed.)
Preparing to unpack filebeat-7.17.6-amd64.deb ...
Unpacking filebeat (7.17.6) ...
Setting up filebeat (7.17.6) ...
root@ip-172-31-23-15:/home/ubuntu#
```

Step 31: Type vim /etc/filebeat/filebeat.yml and make changes as per below screenshots the IP address in hosts is private IP of **logstashkibana** instance

```
# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/apache2/access.log
    #- c:\programdata\elasticsearch\logs\*
```

```
# ---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost: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: ["172.31.83.221:5044"]

  # 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"
```

Step 32: Run filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["<elasticsearch IP>:9200"]'

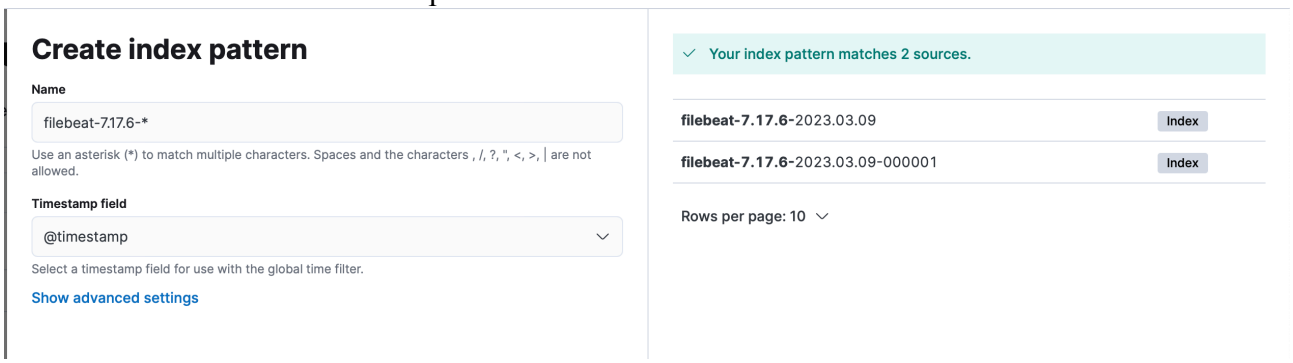
```
[root@ip-172-31-23-15:/home/ubuntu# filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elas]
ticsearch.hosts=["172.31.92.153:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
root@ip-172-31-23-15:/home/ubuntu#
```

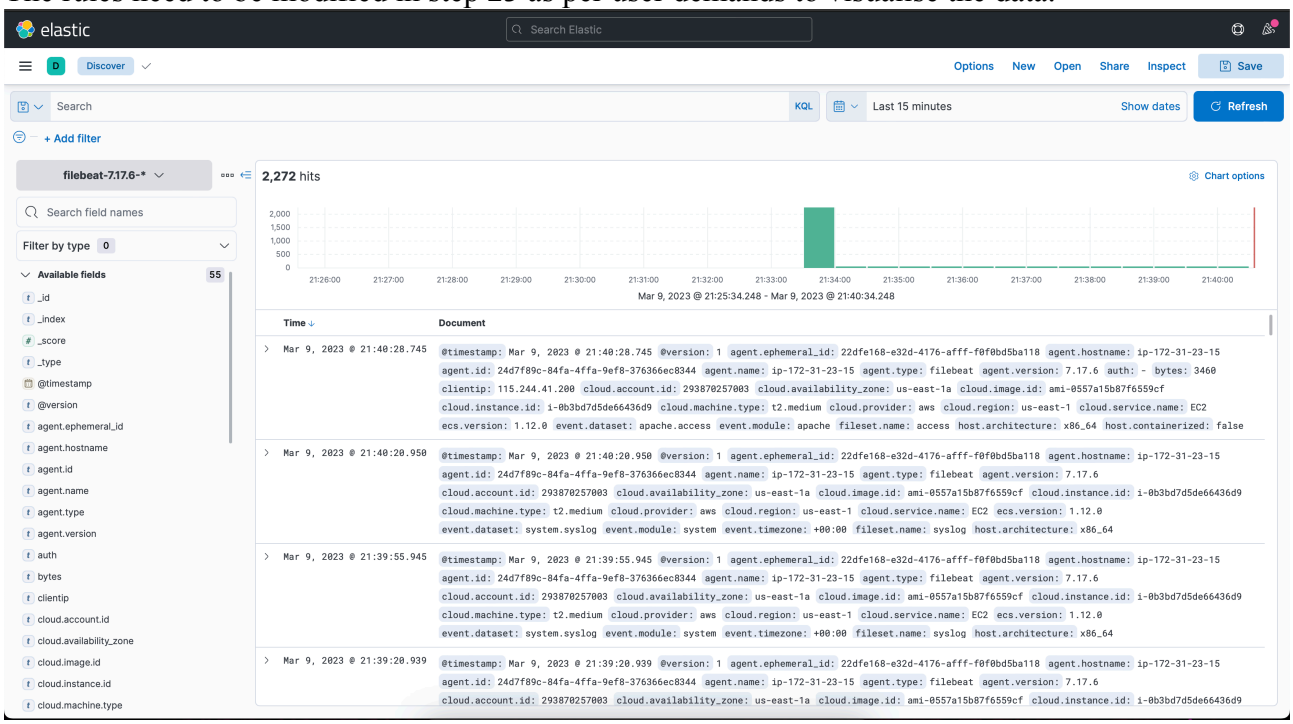
Step 33: Run the following command

```
[root@ip-172-31-23-15:/home/ubuntu# filebeat modules enable system
 Enabled system
[root@ip-172-31-23-15:/home/ubuntu# filebeat modules enable apache
 Enabled apache
[root@ip-172-31-23-15:/home/ubuntu# systemctl restart filebeat.service
[root@ip-172-31-23-15:/home/ubuntu# filebeat test output
 logstash: 172.31.83.221:5044...
   connection...
     parse host... OK
     dns lookup... OK
     addresses: 172.31.83.221
     dial up... OK
   TLS... WARN secure connection disabled
   talk to server... OK
 root@ip-172-31-23-15:/home/ubuntu#
```

Step 34: Now access the kibana instance in a new tab using <PUBLIC IP>:5601 Head towards Discover and create a new index pattern as shown below.



Once this index pattern is created go back to discover to visualise the data.
The rules need to be modified in step 23 as per user demands to visualise the data.



Similarly we can deploy other beats in the same instance, and monitor the network. We can see the above image where the logs can be accessed.

References

[1] https://aws.amazon.com/what-is/elk-stack/#:~:text=Often referred to as Elasticsearch,, security analytics, and more.

[2] https://www.elastic.co/what-is/elk-stack

[3] https://www.elastic.co/guide/index.html

[4] https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html