ELK Installation

Prerequisite

We need to use a ubuntu 20.04 ami of t2.medium .

1. ***sudo apt update***
2. ***sudo apt install openjdk-8-jdk***
3. ***sudo apt-get install -y nginx***
4. ***sudo systemctl enable nginx***

Install Elastic Search

1. ***wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.2.0-amd64.deb***
2. ***sudo dpkg -i elasticsearch-7.2.0-amd64.deb***

Install kibana

1. ***sudo wget https://artifacts.elastic.co/downloads/kibana/kibana-7.2.0-amd64.deb***
2. ***sudo dpkg -i kibana-7.2.0-amd64.deb***

Install Logstash

1. ***sudo wget https://artifacts.elastic.co/downloads/logstash/logstash-7.2.0.deb***
2. ***sudo dpkg -i logstash-7.2.0.deb***

Install Dependencies

1. ***sudo apt-get install -y apt-transport-https***

Install FileBeat

1. ***wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.2.0-amd64.deb***
2. ***sudo dpkg -i filebeat-7.2.0-amd64.deb***

Modify elasticsearch yaml file

1. ***sudo vi /etc/elasticsearch/elasticsearch.yml***
2. Make below changes in this file

cluster.name: my-application

node.name: node-1

http.port: 9200

network.host: localhost

```
cluster.name: my-application
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ---------------------------------- 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 ----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: localhost
#
# Set a custom port for HTTP:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
```

3. ***sudo systemctl start elasticsearch***

Modify Kibana yaml file

1. ***sudo vi /etc/kibana/kibana.yml***
2. Make below changes in the file

server.port: 5601

server.host: "localhost"

```
# 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

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

3. ***sudo systemctl start kibana***

4. ***sudo apt-get install -y apache2-utils***

5. ***sudo htpasswd -c /etc/nginx/htpasswd.users kibadmin***

```
ubuntu@ip-172-31-89-230:~$ sudo htpasswd -c /etc/nginx/htpasswd.users kibadmin
New password:
Re-type new password:
Adding password for user kibadmin
ubuntu@ip-172-31-89-230:~$
```

6. ***sudo vi /etc/nginx/sites-available/default***

```
server {
    listen 80;
    server_name 3.108.42.168;
    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;
    location / {
        proxy_pass http://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;
    }
}
```

Note: when we execute the above command, we need to add the above content to the bottom of the file.

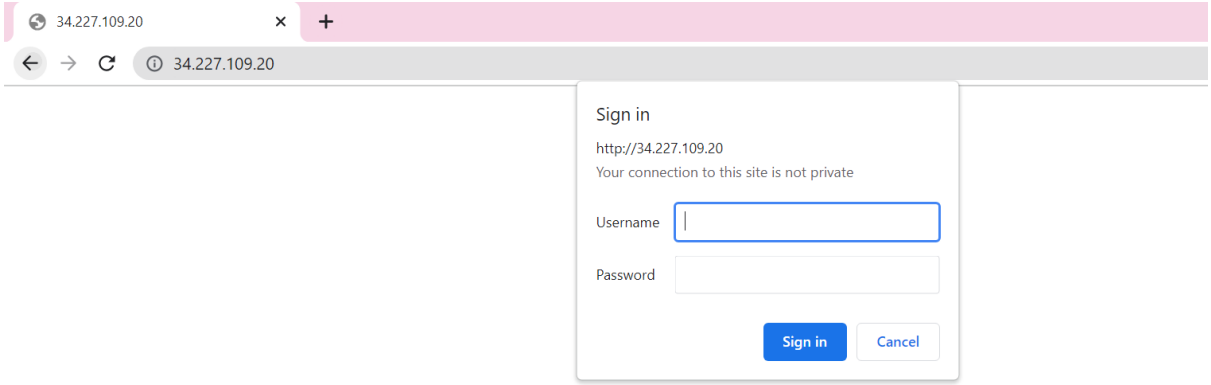
```
  server {
    listen 80;

    server_name 34.227.109.20;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://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;
    }
}
```

Public Ip of the machine

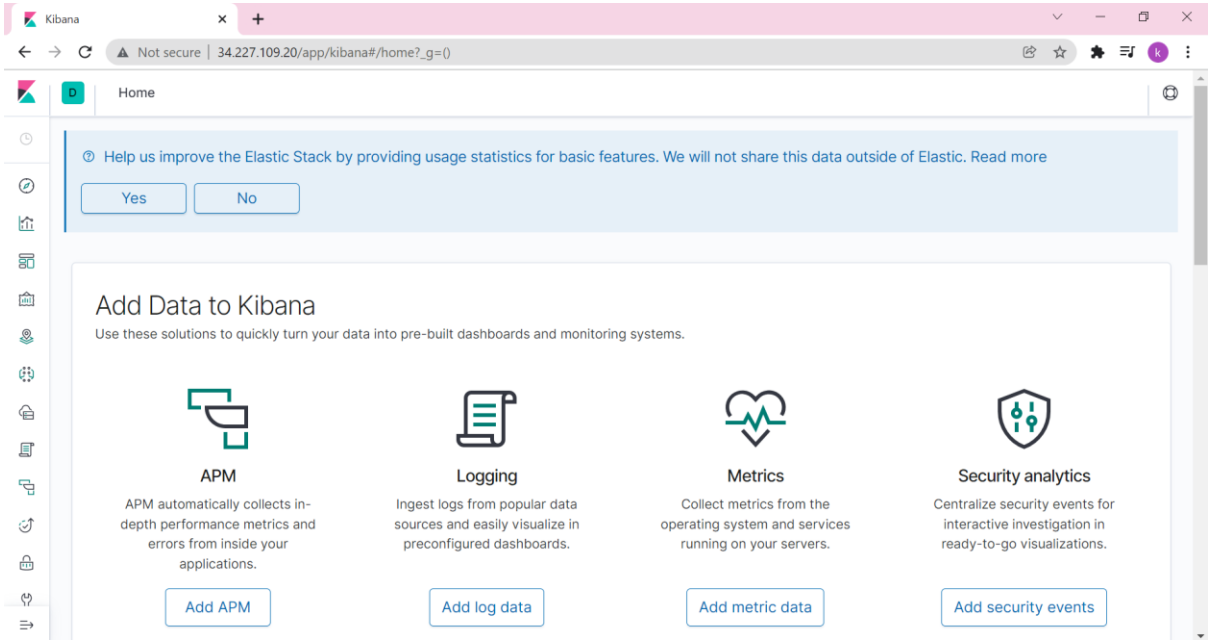
7. ***sudo systemctl restart nginx***

Accessing Kibana Dashboard

Copy the Ip of the machine and paste the Ip on the Browser.



Enter the username and password that you have configured for Kibana.



Therefore, we have successfully installed ELK.