ELK on ubuntu:

Install ELK in this order => java -> nginx -> Elasticsearch -> Kibana -> logstash -> beats

Install java:

sudo apt update

apt install default-jdk

java -version

Installing nginx:

apt-get install nginx

nginx -v

systemctl status nginx

Installing ElasticSearch:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

apt-get install apt-transport-https

echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list

apt-get update && sudo apt-get install elasticsearch

systemctl status elasticsearch

Installing kibana:

Because you've already added the Elastic package source in the previous step, you can just install the remaining components of the Elastic Stack using apt:

apt install kibana

systemctl enable kibana

systemctl status kibana

Installing Logstash:

apt install logstash

systemctl status logstash

After downloading and installing in above order perform below steps:

Configure ElasticSearch:

Edit /etc/elasticsearch/elasaticsearch.yml

uncomment cluster.name, node.name

uncomment network.host and edit the IP with the Ip of server or local host

uncomment http.port

Now save the file and

enable elasticsearch:systemctl enable elasticsearch

start the elasticsearch: systemctl start elasticsearch

Configure Kibana:

Edit /etc/kibana/kibana.yml

uncomment server.port

uncomment server.host and edit the IP with the Ip of server or localhost

uncomment http.port

Now save the file and start the kibana: systemctl start kibana

In order to setup passowrd to access kibana:

Install apache utils: sudo apt-get install -y apache2-utils

Generate password:

htpasswd -c /etc/nginx/htpasswd.users kibadmin

enter password

Because Kibana is configured to only listen on localhost, we must set up a reverse proxy to allow external access to it.

We will use Nginx for this purpose, which should already be installed on your server.

Now configure nginx default file to connect nginx with kibana

Edit /etc/nginx/sites-available/default

update server_name with public IP address of kibana

auth_basic_user_file /etc/nginx/htpasswd.users

Now start nginx: systemctl start nginx

Access the UI using public IP on port 80 (traffic goes nginx -> kibana)

1. Ingest static apache access logs using logstash and visualize iin kibana:

Steps:

Download sample apache logs: wget https://logz.io/sample-data

Now go to /etc/logstash/conf.d

Create a file: vi apachelog.conf

we need to create a pipeline in this file, pipeline will input/source of data, filter and output (where to send)

```
input{
 file{
        path => "home/ubuntu/apache.log"
        start_position => "beginning"
        sincedb_path => "dev/null"  ?? since there in no db we will not link here
        }
        }

filter {
        grok {
                match => {"message" => "%{COMBINEDAPACHELOG}"}
                }
        date {
                match => ["timestamp", "dd/MMM/yyyy:HH:mm:ss Z"]
                }
        geoip {
                source => "clientip"
                        }
                }

output {
        elasticsearch {
                hosts => ["localhost:9200"]
                index => "petclinic-prd1"
```

```
        }
    }
```

Now start logstash: systemctl start logstash

Now access Kibana and create index pattern

To visualize data iin kibana you need ot create an index pattern first.

2. Collect static csv using logstash and analyze in kibana

download the csv file: curl -O https://raw.githubusercontent.com/PacktPublishing/Kibana-7-Quick-Start-Guidemaster/Chapter02/cimes_2001.csv

Go to folder: cd /etc/logstash/conf.d

Create another conf file: vi crimes.conf

```
input{
 file{
        path => "home/ubuntu/crimes.csv"
        start_position => "beginning"
        }
        }

filter {
        csv {
                columns => [
                                        "ID",
```

```
                                        "Case Number",
                                        "Date",
                                        "Block",
                                        "IUCR",
                                        "Primary Type",
                                        "Description",
                                        "Arrest"
                                        ]
                separator => ","
                }
        }

output {
        elasticsearch {
                action => "index"
                hosts => ["localhost:9200"]
                index => "crimes"
                }
        }
```

Save the file and restart logstash: systemctl restart logstash

Create index pattern and start visualizing the csv data in kibana.

3. Using filebeats to send realtime data directly to elasticsearch without using logstash

You do not have to wrote pipleing when using beats

list enabled and disabled modules: filebeat modules list

Enable nginx and system module: filebeat modules enable nginx and filebeat modules enable system

Configure these modules

go to firectory: cd /etc/filebeat/modules.d/

vi nginx.yml

update file with below:

var.paths: ["/var/log/nginx/access.log*"]

var.paths: ["/var/log/nginx/error.log*"]

Save the file

vi system.yml

update file with below:

var.paths: ["/var/log/syslog*"]

var.paths: ["/var/log/auth.log*"]

save the file

Start the filebeat: systemctl start filebeat

Check the status: systemctl status filebeat

The logs will be sent to the index that is created automatically when using filebeat.

Now create a index pattern and visualize in kibana.