**SETUP OF ELK FOR DOCKER**

**E**lasticSearch: this is where the data gets stored.

**L**ogstash: the program responsible for transforming logs to a format that is suitable for being stored in the ElasticSearch database.

**K**ibana: where you can communicate with the Elasticsearch API, run complex queries and visualize them to get more insight into the data. You can also use Kibana to set and send alerts when a threshold is crossed

**For the setup of ELK on your server need**

* Docker and docker compose Installation
* Open require port
* Create a ELK folder on your server and inside this folder crate all the following files

**Step 1]** Creating a file *main.py* , In which include code for application. To build python web application using a Flask and Docker.

* vi main.py

import redis

from flask import Flask

app = Flask(\_\_name\_\_)

redis = redis.Redis(host='redis', port=6379, db=0)

@app.route('/')

def hello\_world():

return 'Hello, World!'

@app.route('/visitor')

def visitor():

redis.incr('visitor')

visitor\_num = redis.get('visitor').decode("utf-8")

return "Visitor: %s" % (visitor\_num)

@app.route('/visitor/reset')

def reset\_visitor():

redis.set('visitor', 0)

visitor\_num = redis.get('visitor').decode("utf-8")

return "Visitor is reset to %s" % (visitor\_num)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0')

**Step 2]** Create a *requirements.txt*  , In which available all the dependencies are needed.

* vi requirement.txt

Flask==1.1.2

redis==3.4.1

gunicorn>=19,<20

**Step 3]** Create a *Dockerfile* , 🡪 vi Dockerfile

FROM python:3.7-alpine

RUN mkdir /app

WORKDIR /app

ADD requirements.txt /app

ADD main.py /app

RUN pip3 install -r requirements.txt

CMD ["gunicorn", "-w 4", "-b", "0.0.0.0:8000", "main:app"]

**Step 4]** Configuring Logstash to accept message with Gelf format :-

Creating a flie logstash.conf . In logstash.conf, store our Logstash configuration to accept message with Gelf format through port 12201via UDP connection. It’s allow Docker services to send messages to Elasticsearch with port 9200.

* vi logstash.conf

input {

gelf {

port => 12201

}

}

output {

elasticsearch {

hosts => ["elasticsearch:9200"]

index => "logstash-%{+YYYY-MM-dd}"

}

}

Within hosts , we specify **elasticsearch** as a hostname instead **localhost**. Because Docker will recognize **elasticsearch** instead **localhost** while they are communicating each other. On the index key, we specify index with prefix logstash- and suffixed by date string that will change every day.

We will use Gelf driver for Docker service logging. And it’s messages will be sent to Elasticsearch through Logstash. You need to specify gelf-address and tag for the options . We will use UDP protocol and port 12201 to send messages from Docker services. Than it will be distinguished based on the tag that we specified for each Docker service. Every Docker container messages will be sent to Logstash via this configuration.

**Step 5]** Containerise the ELK by using a Docker compose

version: '3'  
services:  
 app:  
 build: .  
 volumes:  
 - .:/app  
 ports:  
 - "8000:8000"  
 links:  
 - redis:redis  
 depends\_on:  
 - redis  
 logging:  
 driver: gelf  
 options:  
 gelf-address: "udp://localhost:12201"  
 tag: "demo2\_app"  
 redis:  
 image: "redis:alpine"  
 expose:  
 - "6379"  
 logging:  
 driver: gelf  
 options:  
 gelf-address: "udp://localhost:12201"  
 tag: "demo2\_redis"  
 elasticsearch:  
 image: docker.elastic.co/elasticsearch/elasticsearch:7.6.2  
 environment:  
 - discovery.type=single-node  
 ports:  
 - 9200:9200  
 kibana:  
 image: docker.elastic.co/kibana/kibana:7.6.2  
 ports:  
 - 5601:5601  
 depends\_on:  
 - elasticsearch  
 - logstash  
 logging:  
 driver: gelf  
 options:  
 gelf-address: "udp://localhost:12201"  
 tag: "demo2\_kibana"  
 logstash:  
 image: docker.elastic.co/logstash/logstash:7.6.2  
 links:  
 - elasticsearch  
 volumes:  
 - .:/etc/logstash  
 command: logstash -f /etc/logstash/logstash.conf  
 ports:  
 - 12201:12201/udp  
 depends\_on:  
 - elasticsearch

**Step 6]** To apply docker compose RUN command

* docker compose up -d
* docker ps (For check the running containers)

Step 7] To check the kibana and elasticsearch are running or not

Please check at browser like

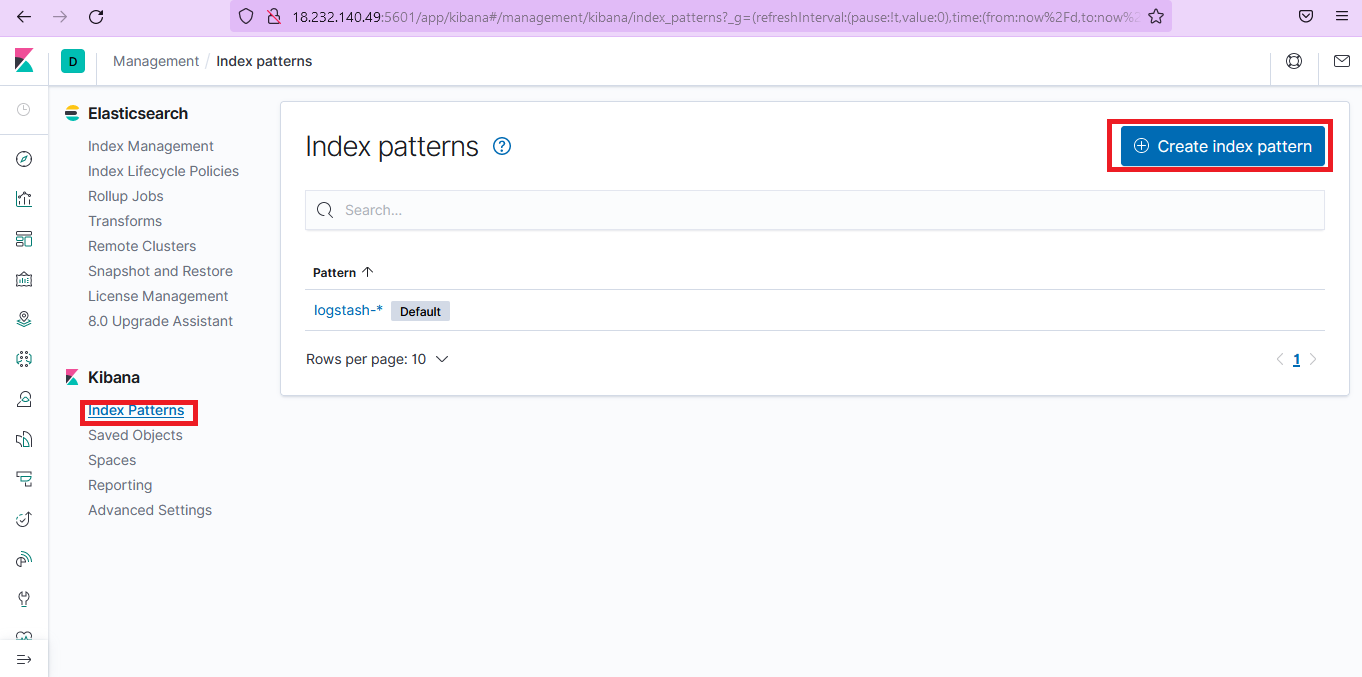
For kibana - ip@address of server:5601

For elastiseach – ip@address of server:9201

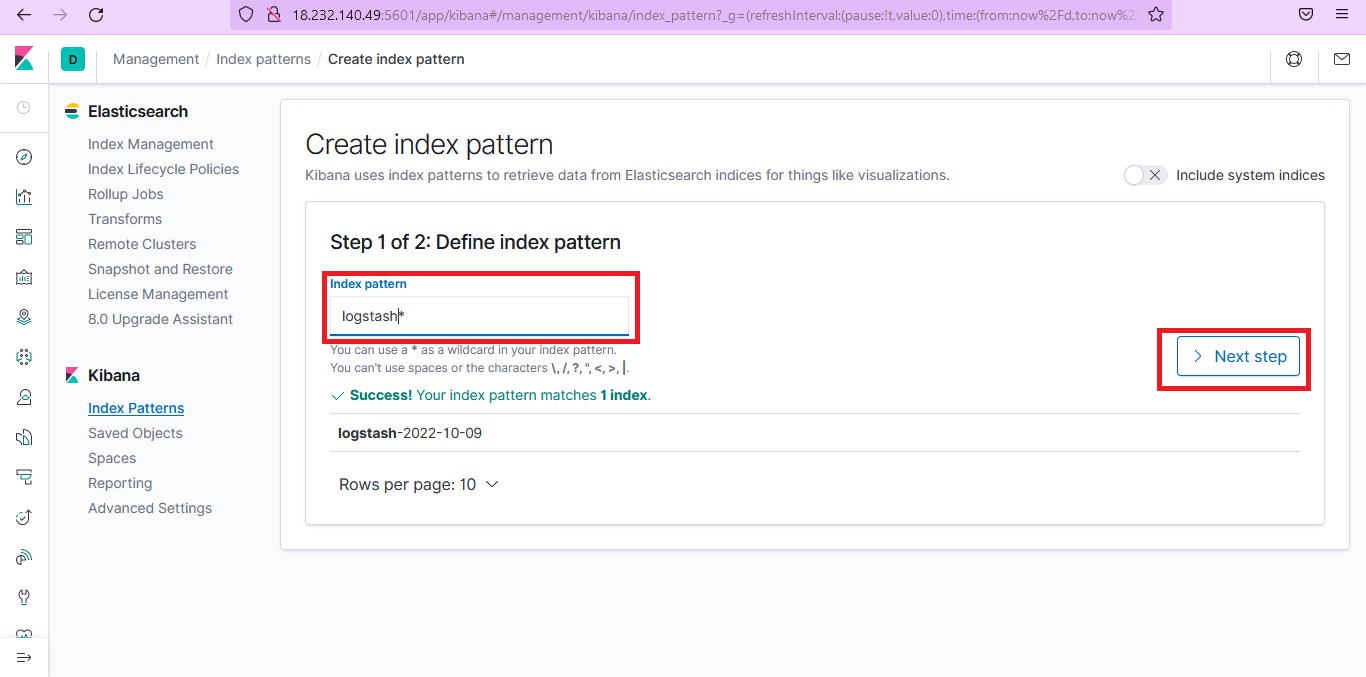
Hit at browser.

**Step 8]** When kibana running please do following configuration for setup kibana dashboard

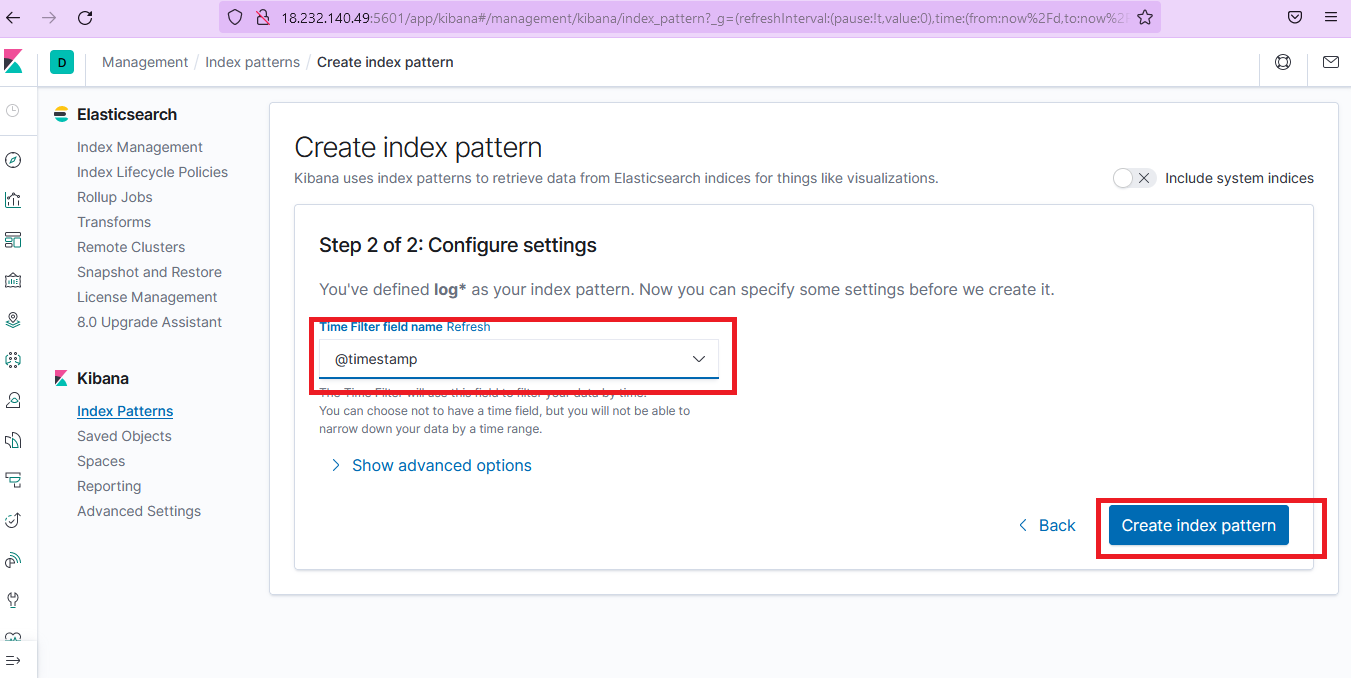
* Go to the management
* In management , click at Index pattern
* Create Index Pattern



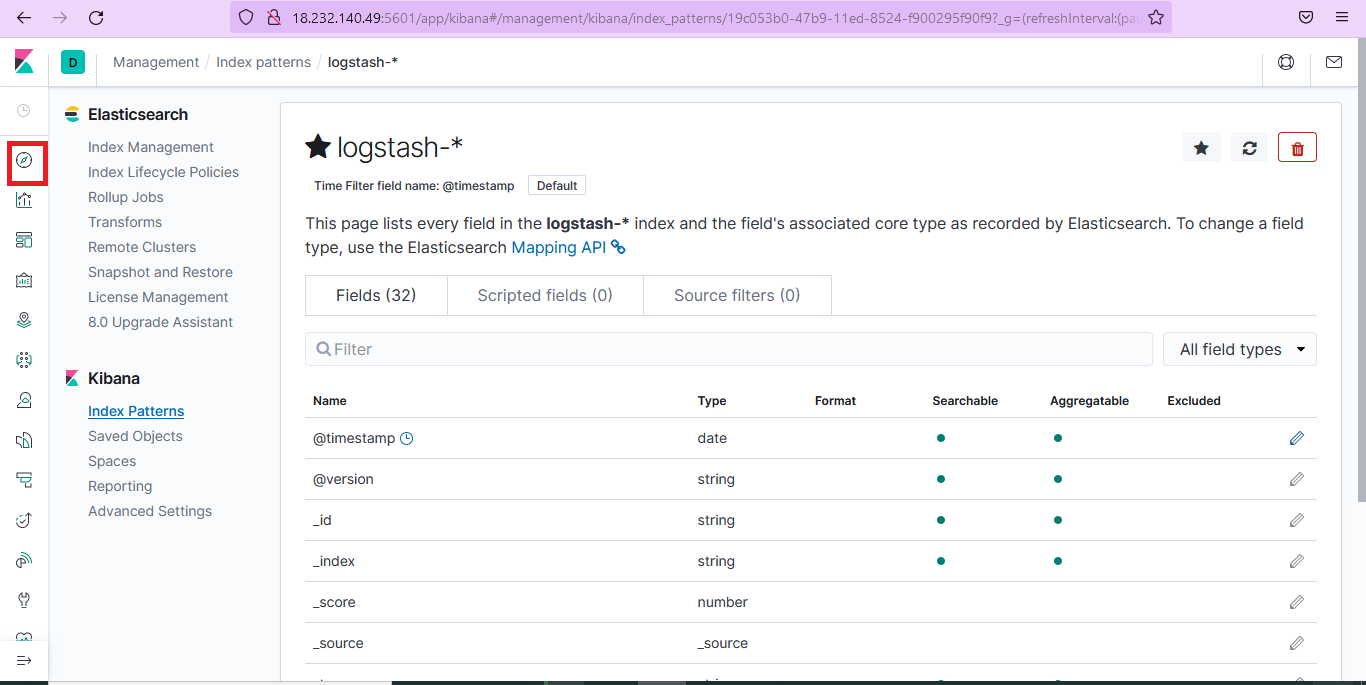
* AT Index patter
* Search logstash and click on a Next step



* Select @timestamp and click on Create index pattern



* Your index pattern now ready
* Then at the left hand side select Discover (mark by red box) for showing logs of your server in the form of graph



* Final out put

