\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Containerized ELK Stack\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ELK stack can be containerized and used on demand without provisioning additional infrastructure. In this project I have completed all the implementation and steps which can help us to containerize Elasticsearch and Kibana on RHEL 8 EC2 instances.

Pre-requisites are,

As a prerequisite, docker should be installed and running on remote machine, also the host should be updated. Also create one application, which will keep on generating logs. Refer the below link,

https://betterstack.com/community/guides/logging/logstash-explained/#getting-started-with-logstash

Use below commands for doing basic config and downloading docker.

yum update -y && curl -fsSL https://get.docker.com -o get-docker.sh && sudo sh get-docker.sh && systemctl start docker && yum install net-tools -y && sudo hostnamectl set-hostname ELK-Stack && sudo timedatectl set-timezone Asia/Calcutta

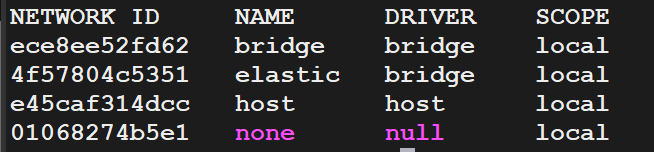
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Elasticsearch\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Elasticsearch is a distributed, open-source search and analytics engine used for a variety of applications, including log analysis, full-text search, security intelligence, and business analytics. It excels at handling large volumes of data with near real-time search capabilities.

Steps to install elasticsearch are as follows,

1. Create a new docker network.

docker network create elastic



1. Pull Elasticsearch container by below command,

docker pull docker.elastic.co/elasticsearch/elasticsearch:9.0.0



1. Before production container creation, run below command and modify the file /etc/sysctl.conf

sudo sysctl -w vm.max\_map\_count=262145

1. Run elastic container in background.

docker run --name es01 --net elastic -p 9200:9200 -m 5GB --restart=always -v elastic\_data:/usr/share/elasticsearch/data/ -it 6cec5391a4c7

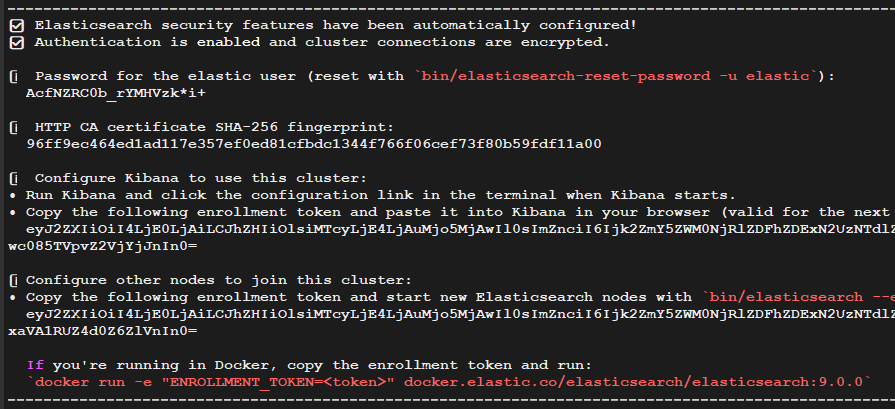
-v /root/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml

Note 🡪 If that container stops by itself bring it up manually by docker start <containerid> &

Make sure that it is always running. Also, the second mount -v is for copying the .yml file.

1. Check the logs and copy paste to your note pad,

docker logs <containerid> (if required)



Validate if there are any errors in container logs,

docker logs <container\_id> | grep "ERROR"

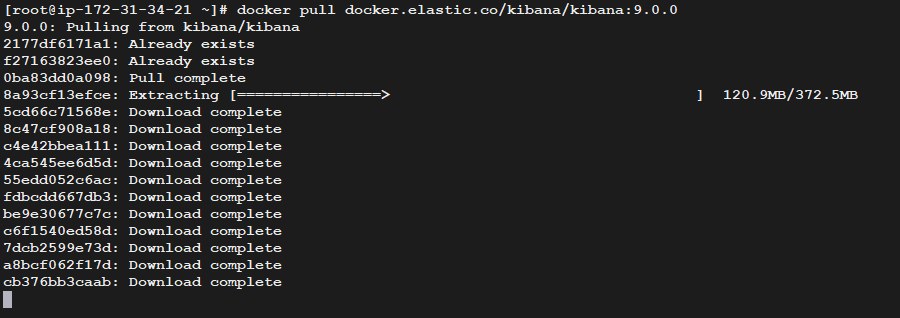
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Kibana\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Kibana is an open-source analytics and visualization platform designed to work with Elasticsearch. You use Kibana to search, view, and interact with data stored in Elasticsearch indices. You can easily perform advanced data analysis and visualize your data in a variety of charts, tables, and maps.

Steps to install Kibana are as follows,

1. Pull Kibana official image,

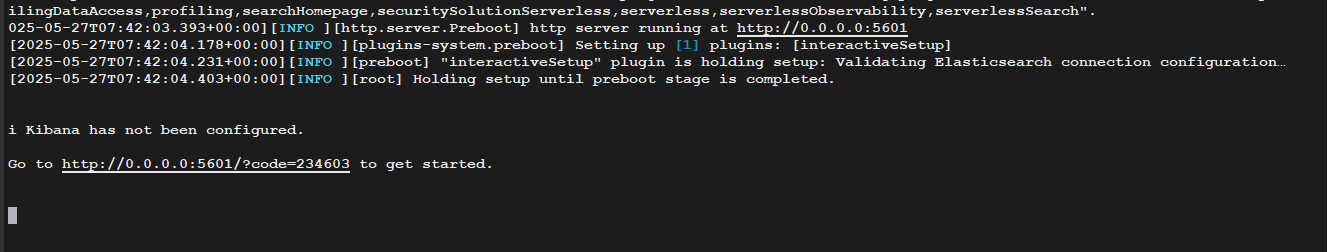
docker pull docker.elastic.co/kibana/kibana:9.0.0



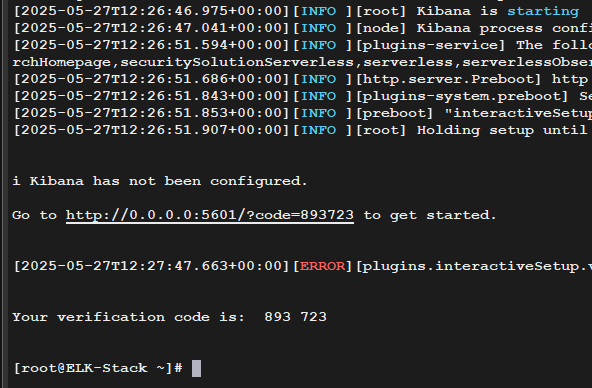
1. Run the image by below command.

docker run --name kib01 --net elastic -p 5601:5601 -m 2GB --restart=always -v kibana\_data:/usr/share/kibana/data/ -it af182398db7c

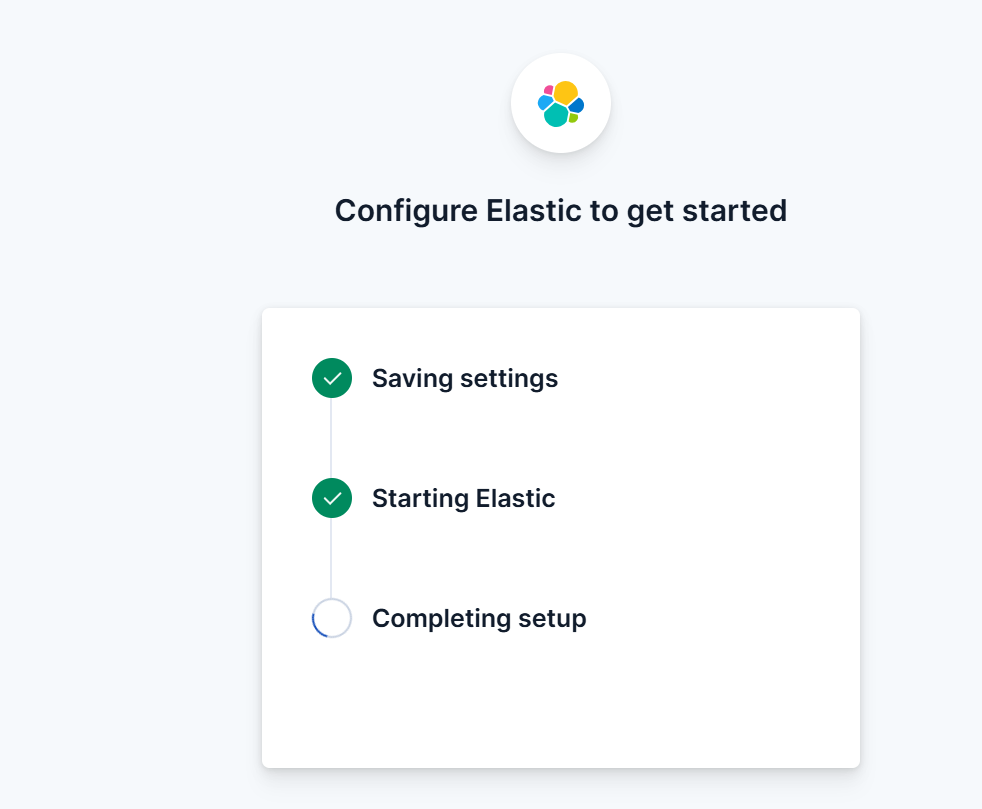
1. Inspect the logs of the kibana container and copy the URL to access kibana in browser window.



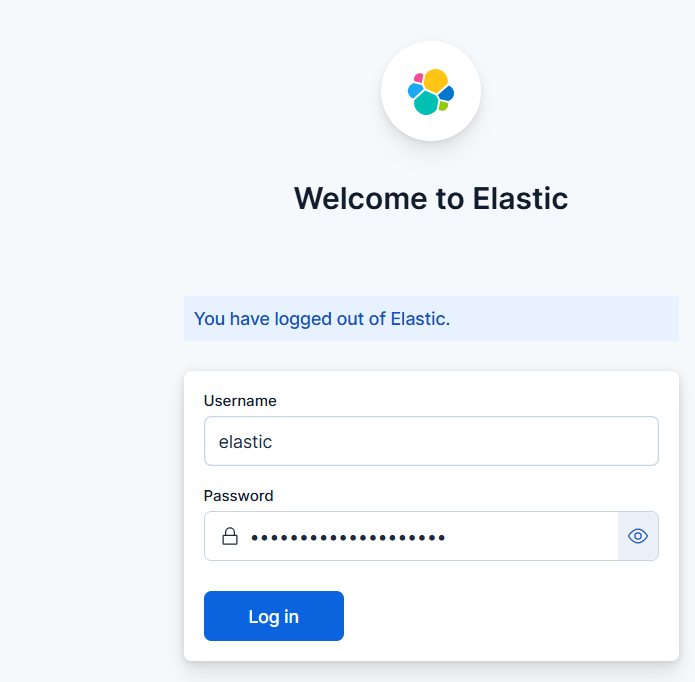
NOTE 🡪 After this step if Ui asks you a 6-digit passcode, then use docker logs commands, it will show up.



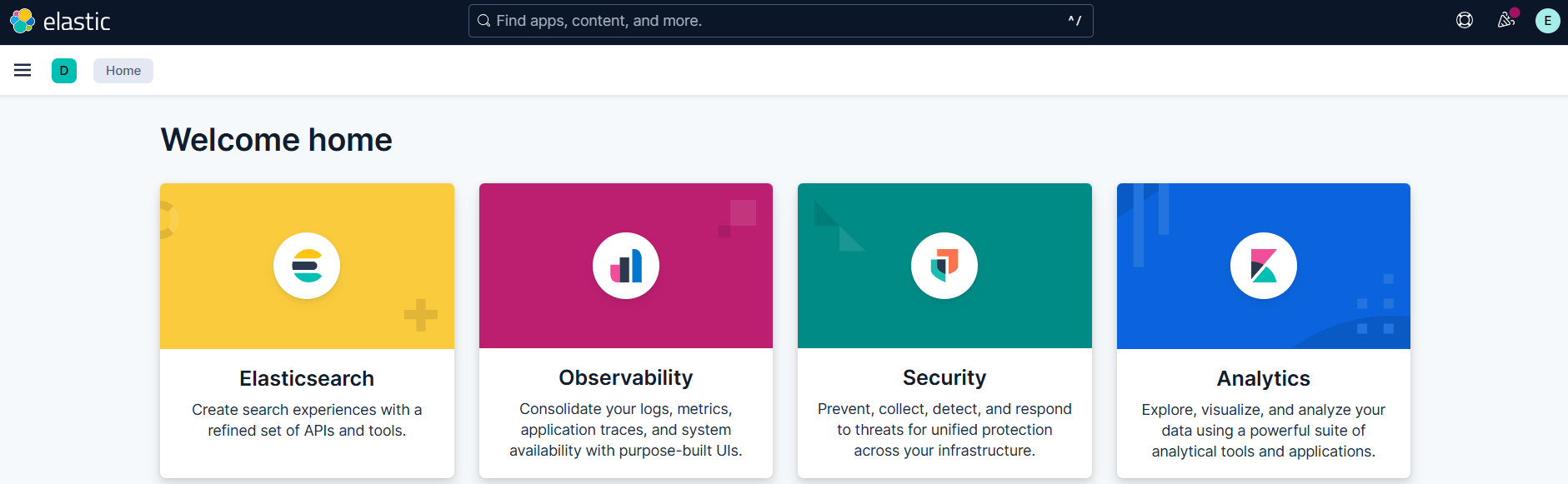
1. After accessing, now enter the token from step 5, point no 3. After entering token, you will see UI like below.



1. Use default password from line one step 5, and login as elastic user.

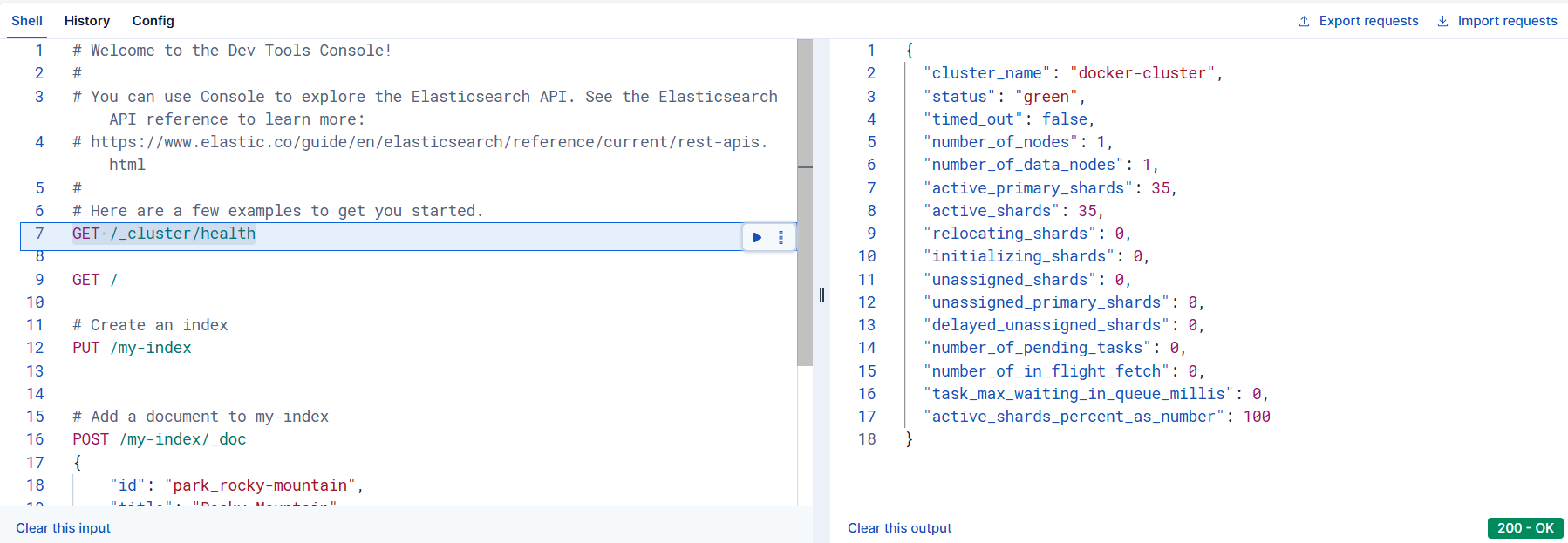


1. After that you can see login details,



1. When everything is done do verify elastic is interacting properly and it is responding.

Go to mgmt 🡪 Dev tools. The below command shows cluster health.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Logstash\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Logstash is an open-source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice. Use below commands for doing basic config and downloading docker.

Steps to install & configure Logstash are as follows.

1. Create a simple Dockerfile which uses the base image for logstash.

FROM docker.elastic.co/logstash/logstash:9.0.0

COPY logstash.conf /usr/share/logstash/pipeline/logstash.conf

EXPOSE 5044

1. Simple Logstash configuration file looks like below,

input {

beats {

port => 5044

}

}

filter {

if [message] =~ /^{.\*}$/ {

json {

source => "message"

}

}

mutate {

remove\_field => ["event", "message", "emailAddress"]

add\_field => { "env" => "development" }

}

}

output {

elasticsearch {

hosts => ["https://13.201.48.40:9200"]

index => "my-index"

ssl\_verification\_mode => "none"

user=> "elastic"

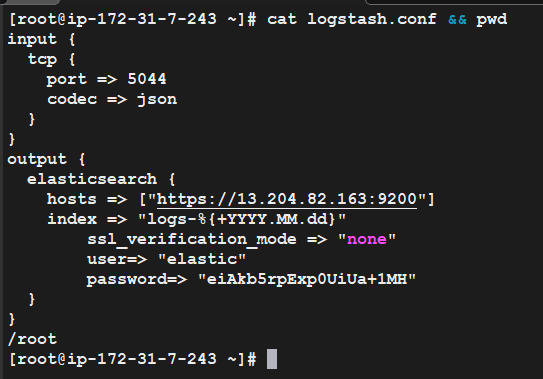
password=> "npKYTOu6D\*\*WYeKQ+9K-"

}

}

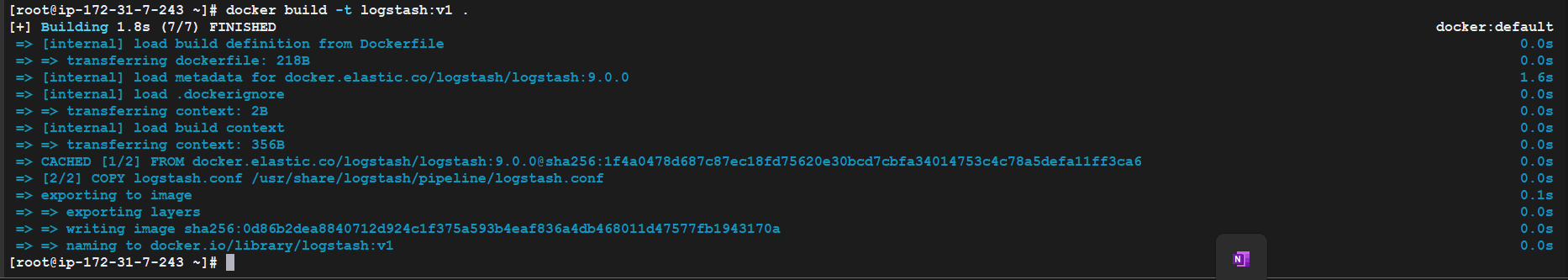
So, the Logstash config files will have 3 plugins, input, output & filter which are commonly used. These 3 components form a pipeline. In the above file, input is coming from beats, which running as a container on application server. We are using filter to convert it into. json document, “mutate” plugin is used to modify the sensitive fields from the log. Then later, these logs will be forwarded to elastic for storage.

1. Make sure that this logstash.conf file is present in same directory as of Dockerfile.



1. Build the docker image,

docker build -t logstash:V1 .

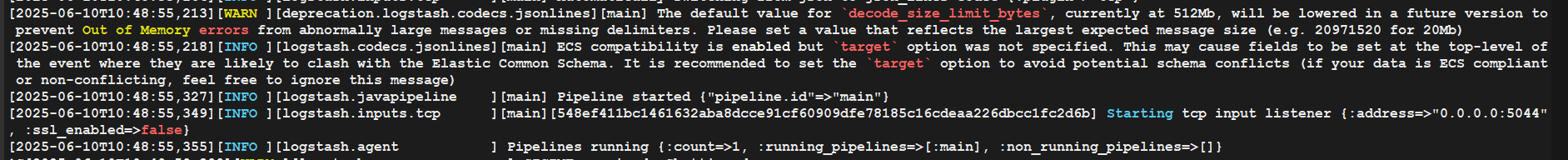


1. Run the container by below command. Container logs should reflect like below which should be good.

docker run --name ls01 --net elastic -p 5044:5044 -m 3GB --restart=always -v logstash\_data:/usr/share/logstash/ --restart=always -it 690404710b6b

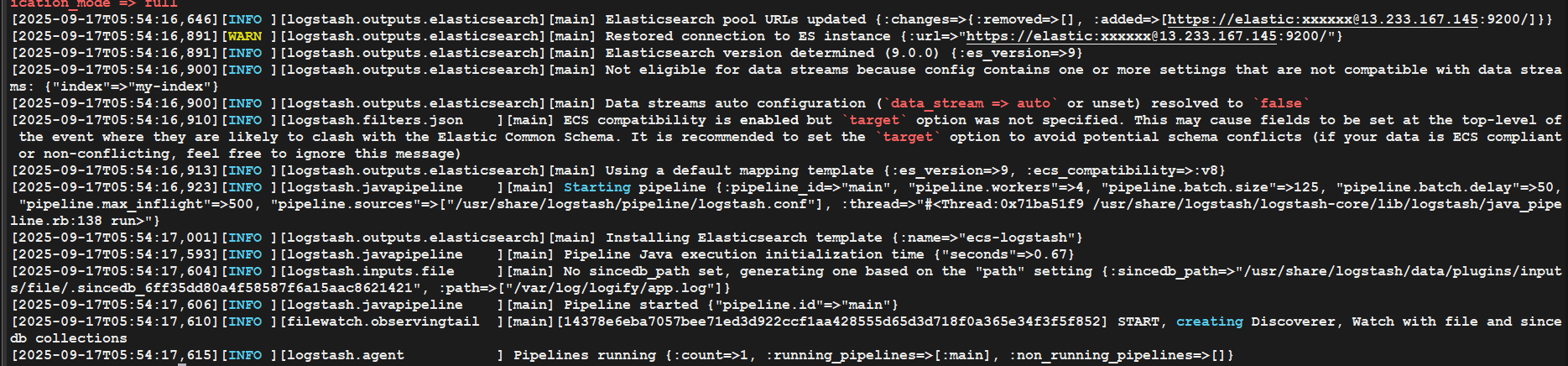
Note 🡪 --config.test\_and\_exit can be used to test the container without creating it on runtime.

docker run --rm -v /root/logstash.yml:/usr/share/logstash/config/logstash.yml f7fff9616c42



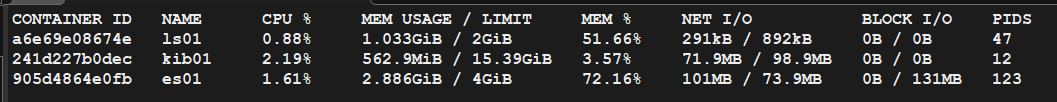


1. After this container is running check if it is processing by pipeline, it will not have any errors.



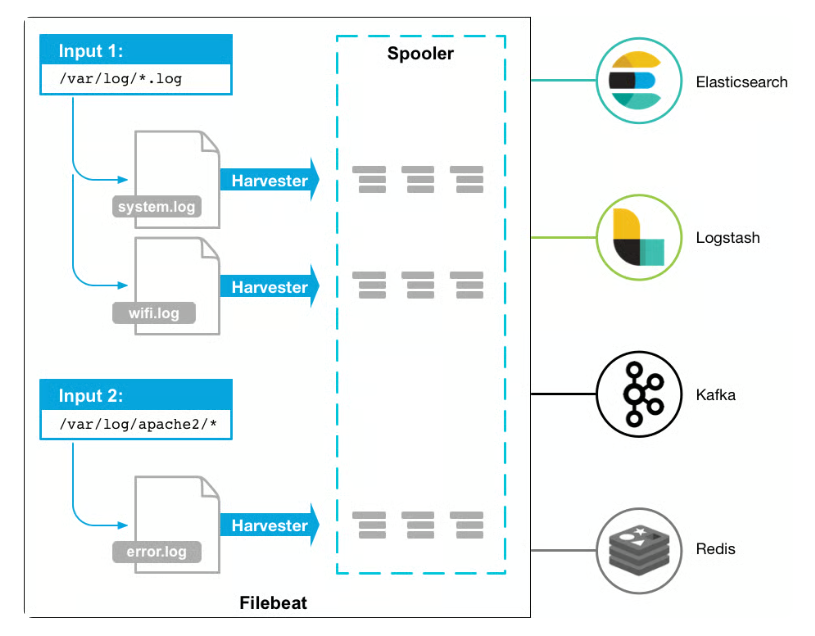
1. If you see no errors in above container logs then we are good to monitor the container stats,

docker stats



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Filebeat\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Filebeat is a lightweight, open-source data shipper from Elastic that monitors log files and sends them to destinations like Elasticsearch or Logstash for analysis. As a member of the Beats family, it's installed as an agent on servers, containers, or cloud environments to collect log events and forward them with a low memory footprint. Filebeat uses harvesters to read log files line-by-line and inputs to manage these harvesters and find log sources, remembering its place during interruptions to ensure reliable data transfer.



1. Download official filebeat image by below command.

docker pull elastic/filebeat:9.0.0

1. For testing filebeat configuration use the below commands,

docker run --rm -v /root/config/filebeat.yml:/usr/share/filebeat/filebeat.yml -it 30846bc6e549 filebeat test config

Above command should return as config OK on the terminal. Below is the simple configuration of filebeat.yml where it takes logs and send it to logstash. Make sure to enter correct IP address for logstash server.

filebeat.inputs:

- type: filestream

id: logify-logs-2025091537463

paths:

- /var/log/logify/app.log

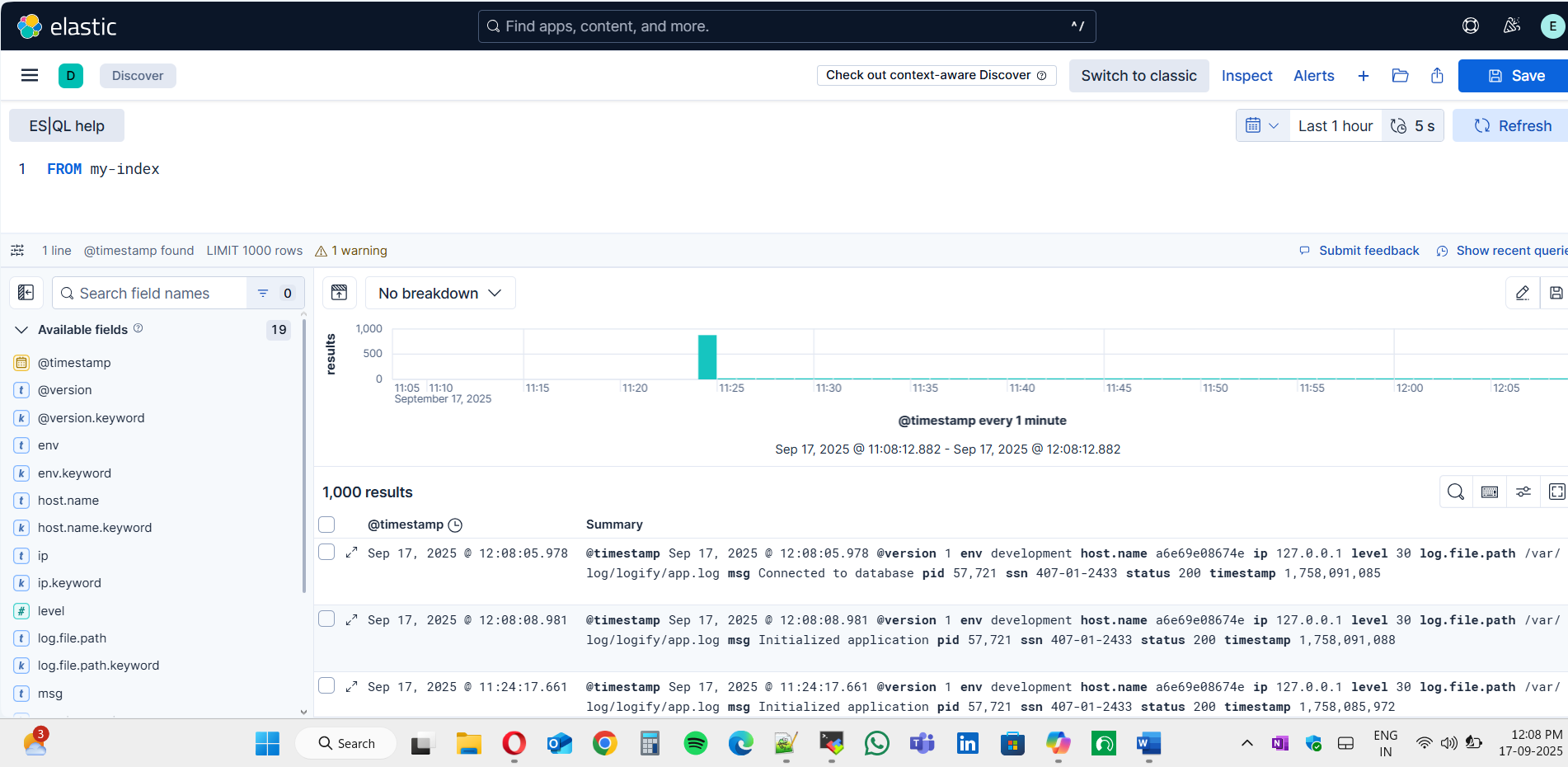
output.logstash:

hosts: ["13.201.48.40:5044"]

1. For running filebeat container, run the below command,

docker run --name fb01 -p 5044:5044 -m 2GB --restart=always -v filebeat\_data:/usr/share/filebeat/data/ -v /root/config/filebeat.yml:/usr/share/filebeat/filebeat.yml -v /var/log/logify:/var/log/logify -it 30846bc6e549

1. Monitor any error logs by docker logs command. There should not be any.
2. Check over kibana UI on visualization part,



Index management section shows like below. Docs & storage size should be changing which is a good sign.

