Name: Venkata Lahari Goruputi

Role: DevOps Intern

Topic: ELK stack set up and EC2 Instance Monitoring in ELK

Creating ELK stack using Docker and Monitoring an EC2 Instance using ELK:

What is ELK:

- ELK is an acronym that stands for **Elasticsearch**, **Logstash**, and **Kibana**. These are three open-source projects that are often used together to collect, process, and visualize data in real-time. Together, they form a powerful stack for managing and analyzing log data, often referred to as the ELK Stack.

Elasticsearch:

- Elasticsearch is a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data so you can discover the expected and uncover the unexpected.
- Elasticsearch is a Search and Analytics Engine.
- It is helpful in Full-text search, real-time search and analytics, scalability, and high availability.

Logstash:

- Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch.
- It is useful for Data collection, parsing, enrichment, and transformation, support for various input/output plugins.

Kibana:

- Kibana is a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It provides powerful and easy-to-use features like histograms, line graphs, pie charts, and maps.
- It is the Data Visualization tool.
- Interactive charts, dashboards, data exploration tools, and the ability to search and view data stored in Elasticsearch indices.

How They Work Together:

Data Ingestion with Logstash:

- Logstash collects and processes data from various sources (e.g., logs, metrics, web applications) and sends the processed data to Elasticsearch.

Data Storage and Search with Elasticsearch:

- Elasticsearch stores the data indexed by Logstash and allows for fast search and analysis. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface.

Data Visualization with Kibana:

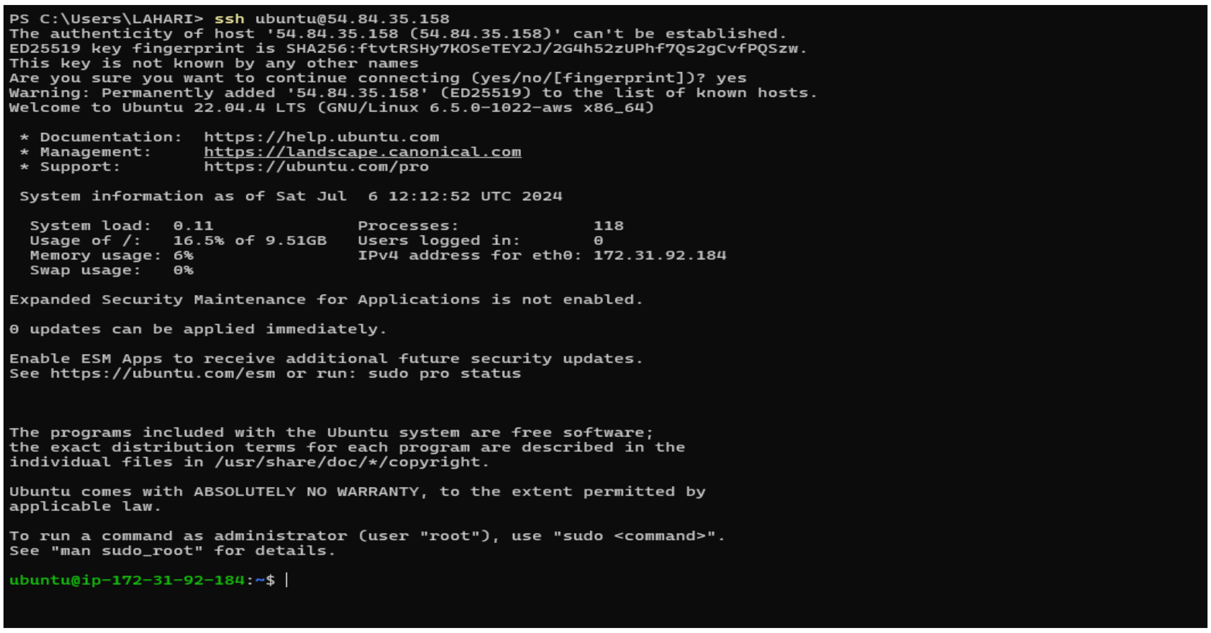
- Kibana connects to Elasticsearch and provides a web interface to visualize the data. Users can create dashboards and perform complex queries to gain insights from the data stored in Elasticsearch.
- Elasticsearch would index and store these logs.
- Logstash would collect logs from different servers.
- Kibana would visualize the log data, allowing system administrators to create dashboards that display server health, performance metrics, and error logs.

✓ Set Up Process:

- As part of ELK set up, we need to create an EC2 instance and install **docker** in it as we are setting up ELK via docker.

  Docker Install Script

- I have created an **ubuntu-22.04** with size **t2.large** EC2 instance in AWS.
- Log into it using the command **[ssh ubuntu@<Public-Ip>].**
- After log in to the instance we need to install docker.

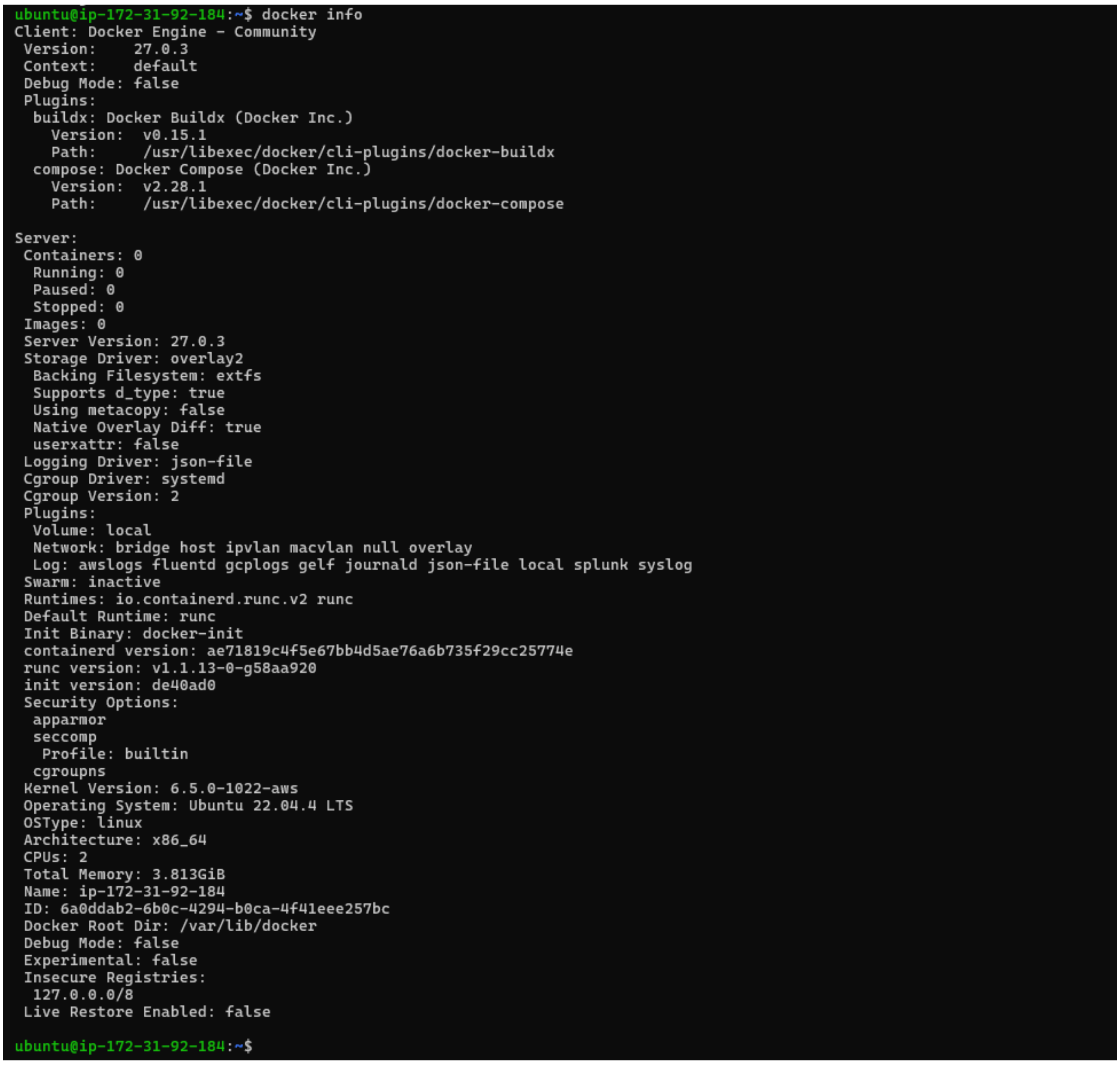


```
PS C:\Users\LAHARI> ssh ubuntu@54.84.35.158
The authenticity of host '54.84.35.158 (54.84.35.158)' can't be established.
ED25519 key fingerprint is SHA256:ftvtRSHy7KOSeTEY2J/2G4h52zUPhf7Qs2gCvfPQSzw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.84.35.158' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Jul  6 12:12:52 UTC 2024

  System load:  0.11              Processes:             118
  Usage of /:   16.5% of 9.51GB   Users logged in:       0
  Memory usage: 6%                IPv4 address for eth0: 172.31.92.184
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-172-31-92-184:~$ |
```

Steps to install docker:

```
curl -fsSL https://get.docker.com -o install-docker.sh

sh install-docker.sh

sudo usermod -aG docker <username-of-the-server>
```

- After doing the above steps logout of the server and login again and check for docker using **docker info** command, if this returns fine then docker is up and running.



```
ubuntu@ip-172-31-92-184:~$ docker info
Client: Docker Engine - Community
 Version:    27.0.3
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.15.1
    Path:     /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.28.1
    Path:     /usr/libexec/docker/cli-plugins/docker-compose

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 27.0.3
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: ae71819c4f5e67bb4d5ae76a6b735f29cc25774e
 runc version: v1.1.13-0-g58aa920
 init version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: builtin
  cgroupns
 Kernel Version: 6.5.0-1022-aws
 Operating System: Ubuntu 22.04.4 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 3.813GiB
 Name: ip-172-31-92-184
 ID: 6a0ddab2-6b0c-4294-b0ca-4f41eee257bc
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

ubuntu@ip-172-31-92-184:~$
```

✓ ELK stack set up:

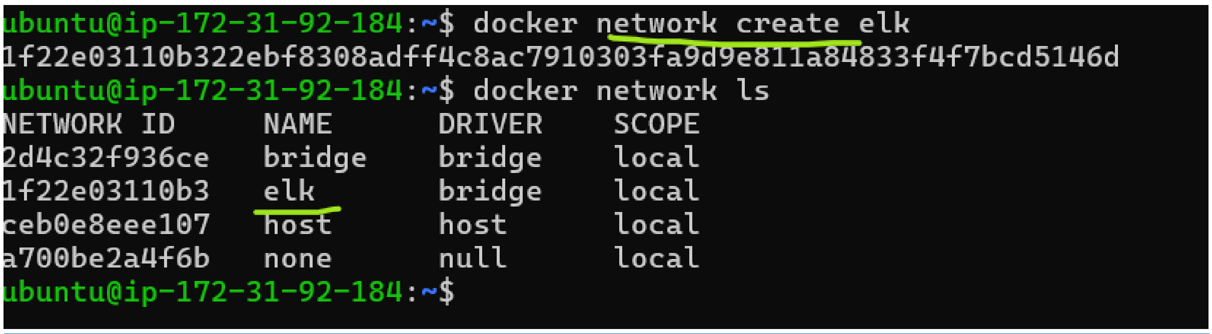
- ➢ Create a new docker network to deploy all the ELK containers in one network so that every container in that network can communicate with each other using the container names also.

Create network in docker:

Command:

[docker network create elk]

- ➢ The above command will create a network named **elk**. we will deploy all the ELK containers in this network only.

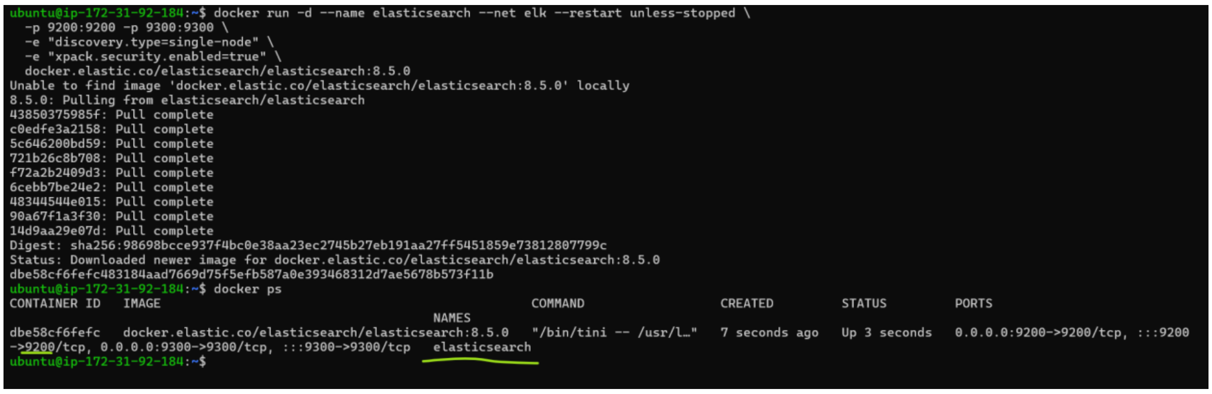


```
ubuntu@ip-172-31-92-184:~$ docker network create elk
1f22e03110b322ebf8308adff4c8ac7910303fa9d9e811a84833f4f7bcd5146d
ubuntu@ip-172-31-92-184:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
2d4c32f936ce   bridge    bridge    local
1f22e03110b3   elk       bridge    local
ceb0e8eee107   host      host      local
a700be2a4f6b   none      null      local
ubuntu@ip-172-31-92-184:~$
```

Installing ElasticSearch:

```
docker run -d --name elasticsearch --net elk --restart unless-stopped \
 -p 9200:9200 -p 9300:9300 \
 -e "discovery.type=single-node" \
 -e "xpack.security.enabled=true" \
 docker.elastic.co/elasticsearch/elasticsearch:8.5.0
```

- The above command will deploy elasticsearch container on port **9200** and **9300**.
- Then we need to generate the passwords for all the users which will be required for elastic search.
- Give 2 minutes of time after running the above command so that elastic search will start run.
- Then use the below command to auto generate the user passwords.



```
ubuntu@ip-172-31-92-184:~$ docker run -d --name elasticsearch --net elk --restart unless-stopped \
  -p 9200:9200 -p 9300:9300 \
  -e "discovery.type=single-node" \
  -e "xpack.security.enabled=true" \
  docker.elastic.co/elasticsearch/elasticsearch:8.5.0
Unable to find image 'docker.elastic.co/elasticsearch/elasticsearch:8.5.0' locally
8.5.0: Pulling from elasticsearch/elasticsearch
43850375985f: Pull complete
c0edfe3a2158: Pull complete
5c646200bd59: Pull complete
721b26c8b708: Pull complete
f72a2b2409d3: Pull complete
6cebb7be24e2: Pull complete
48344544e015: Pull complete
90a67f1a3f30: Pull complete
14d9aa29e07d: Pull complete
Digest: sha256:98698bcce937f4bc0e38aa23ec2745b27eb191aa27ff5451859e73812807799c
Status: Downloaded newer image for docker.elastic.co/elasticsearch/elasticsearch:8.5.0
dbe58cf6fefc483184aad7669d75f5efb587a0e393468312d7ae5678b573f11b
ubuntu@ip-172-31-92-184:~$ docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED         STATUS         PORTS
                                                     NAMES
dbe58cf6fefc   docker.elastic.co/elasticsearch/elasticsearch:8.5.0   "/bin/tini -- /usr/l…"   7 seconds ago   Up 3 seconds   0.0.0.0:9200->9200/tcp, :::9200
->9200/tcp, 0.0.0.0:9300->9300/tcp, :::9300->9300/tcp   elasticsearch
ubuntu@ip-172-31-92-184:~$
```

Command:

[docker exec -it elasticsearch bin/elasticsearch-setup-passwords auto]

- The above command will auto generate the passwords for the users which looks like as below.

- **<u>Changed password for user apm_system</u>**

  PASSWORD apm_system = 2jJwtege0YwrBH0Zpirv

- **<u>Changed password for user kibana_system</u>**

  PASSWORD kibana_system = BGRWL9tNfpFhCvGJglRD

- **<u>Changed password for user kibana</u>**

  PASSWORD kibana = BGRWL9tNfpFhCvGJglRD

- **<u>Changed password for user logstash_system</u>**

  PASSWORD logstash_system = mdLdcNvEo2qWneSMbJ5t

- **<u>Changed password for user beats_system</u>**

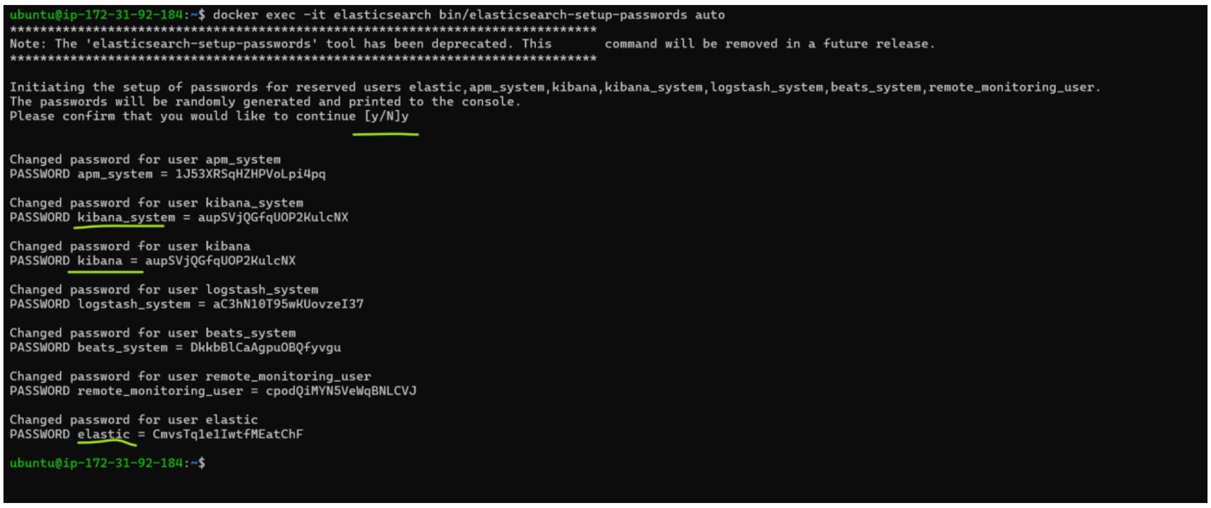
  PASSWORD beats_system = Qpkjit9RXFX5ZeZRJ1UB

- **<u>Changed password for user remote_monitoring_user</u>**

  PASSWORD remote_monitoring_user = X4iu8glhWivgLhkwe1Ly

- **<u>Changed password for user elastic</u>**

  PASSWORD elastic = 5Tm0Vr3n950vghgzNY06



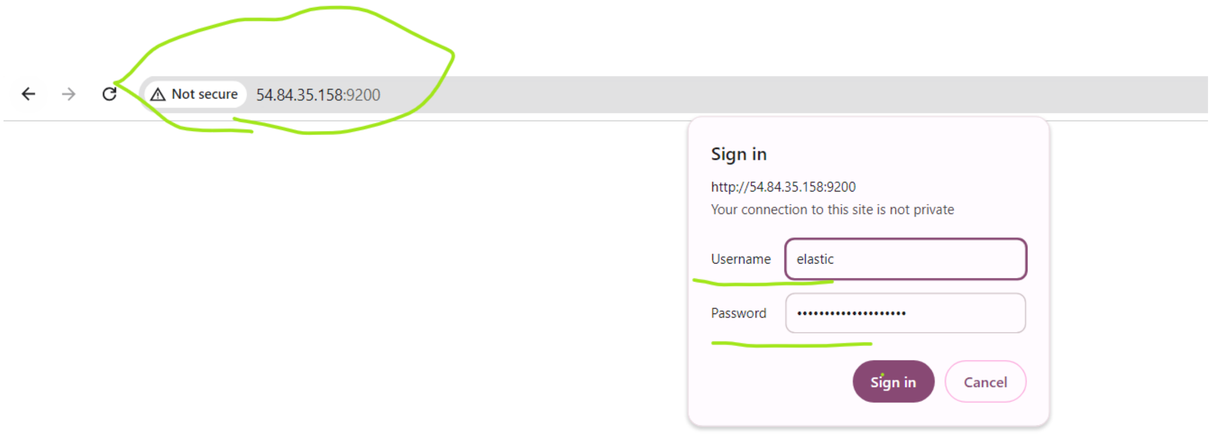
```
ubuntu@ip-172-31-92-184:~$ docker exec -it elasticsearch bin/elasticsearch-setup-passwords auto
******************************************************************************
Note: The 'elasticsearch-setup-passwords' tool has been deprecated. This       command will be removed in a future release.
******************************************************************************

Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.
The passwords will be randomly generated and printed to the console.
Please confirm that you would like to continue [y/N]y


Changed password for user apm_system
PASSWORD apm_system = 1J53XRSqHZHPVoLpi4pq

Changed password for user kibana_system
PASSWORD kibana_system = aupSVjQGfqUOP2KulcNX

Changed password for user kibana
PASSWORD kibana = aupSVjQGfqUOP2KulcNX

Changed password for user logstash_system
PASSWORD logstash_system = aC3hN10T95wKUovzeI37

Changed password for user beats_system
PASSWORD beats_system = DkkbBlCaAgpuOBQfyvgu

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = cpodQiMYN5VeWqBNLCVJ

Changed password for user elastic
PASSWORD elastic = CmvsTq1e1IwtfMEatChF

ubuntu@ip-172-31-92-184:~$
```

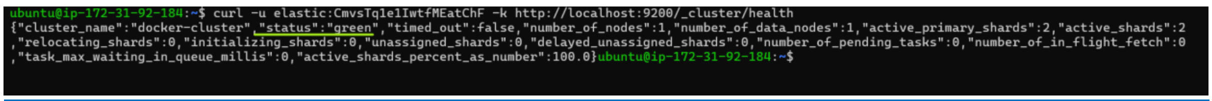
- Then we can access elasticsearch on port **9200** using the **Public-IP** of the server where elasticsearch container is running and then login using **elastic** user and **elastic_user** password.
- We can also check the cluster status using the below command.

Command:

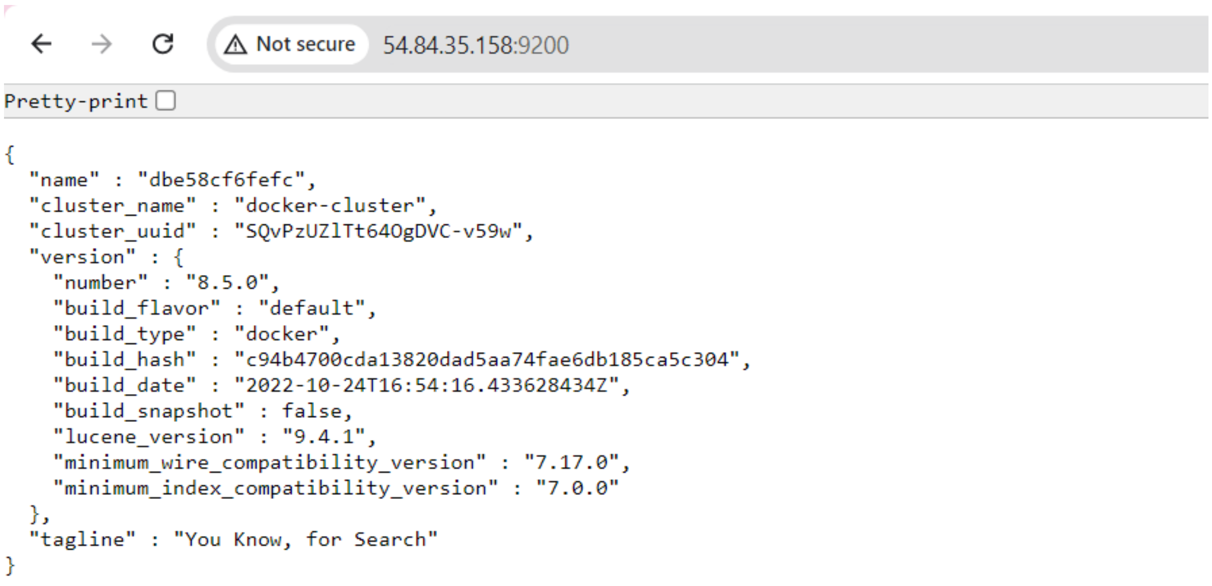
[`curl -u elastic:elastic_user_password -k http://localhost:9200/_cluster/health`]

- If the above command returns the status as `green` then the elasticsearch is running fine.





```
ubuntu@ip-172-31-92-184:~$ curl -u elastic:CmvsTq1e1IwtfMEatChF -k http://localhost:9200/_cluster/health
{"cluster_name":"docker-cluster","status":"green","timed_out":false,"number_of_nodes":1,"number_of_data_nodes":1,"active_primary_shards":2,"active_shards":2
,"relocating_shards":0,"initializing_shards":0,"unassigned_shards":0,"delayed_unassigned_shards":0,"number_of_pending_tasks":0,"number_of_in_flight_fetch":0
,"task_max_waiting_in_queue_millis":0,"active_shards_percent_as_number":100.0}ubuntu@ip-172-31-92-184:~$
```



```
{
  "name" : "dbe58cf6fefc",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "SQvPzUZlTt64OgDVC-v59w",
  "version" : {
    "number" : "8.5.0",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "c94b4700cda13820dad5aa74fae6db185ca5c304",
    "build_date" : "2022-10-24T16:54:16.433628434Z",
    "build_snapshot" : false,
    "lucene_version" : "9.4.1",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

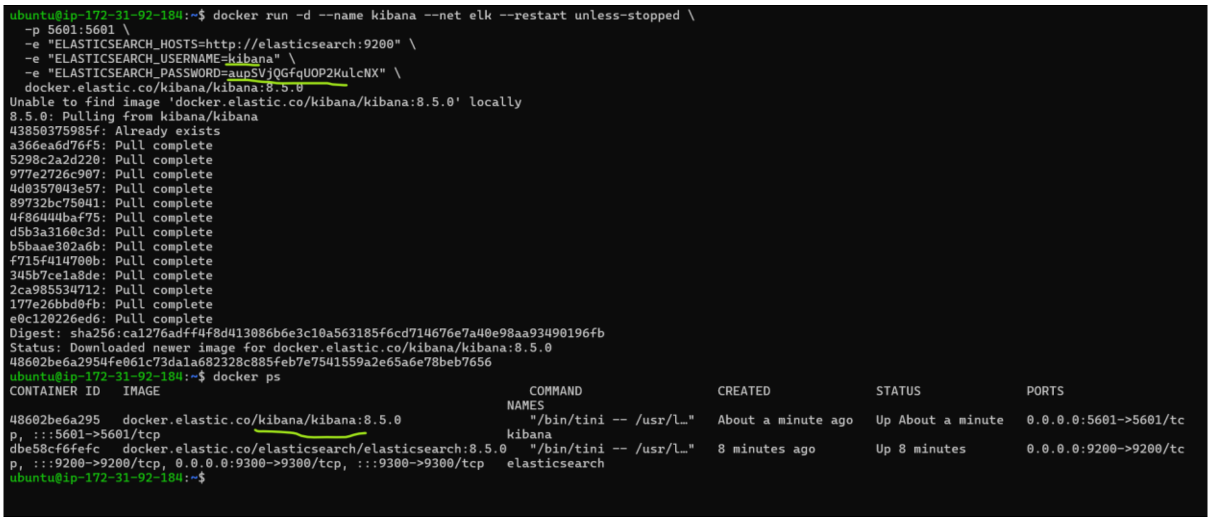
Installing Kibana:

Note:

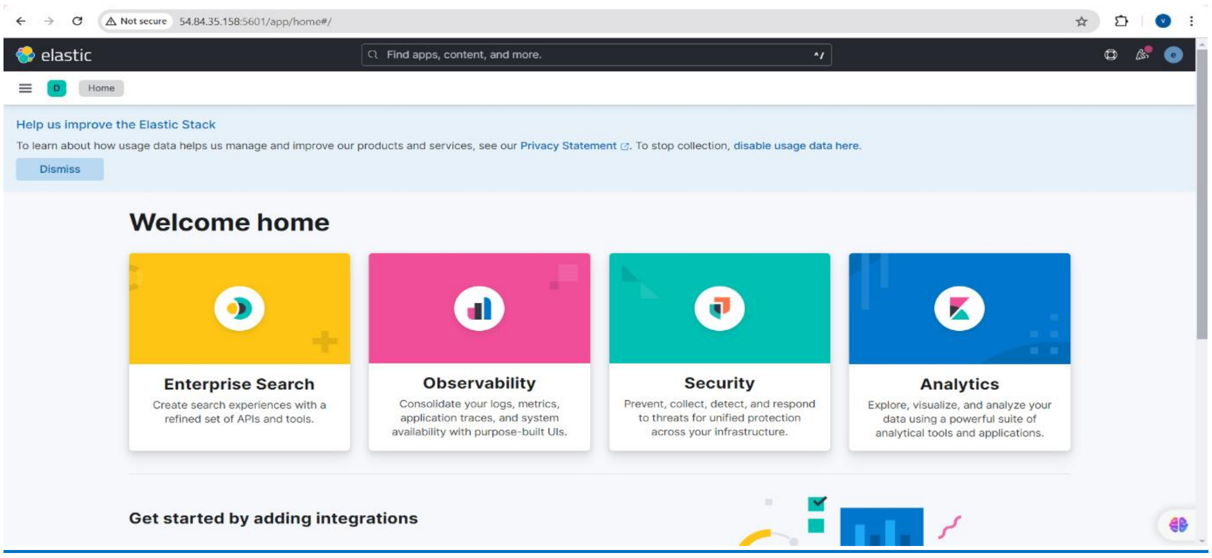
- In the below command replace the **kibana_password** with the kibana user password which is generated above.

```
docker run -d --name kibana --net elk --restart unless-stopped \
 -p 5601:5601 \
 -e "ELASTICSEARCH_HOSTS=http://elasticsearch:9200" \
 -e "ELASTICSEARCH_USERNAME=kibana" \
 -e "ELASTICSEARCH_PASSWORD=<kibana_password>" \
 docker.elastic.co/kibana/kibana:8.5.0
```

- The above command will deploy kibana container on port **5601.**
- Then we can access kibana on port **5601** using the Public-IP of the server then use the kibana username and password to login. Here, we must need to use kibana password only, otherwise we can not access the kibana server, it will stay in server ready state.
- After login we can able to see kibana dashboard. Here, When it is asking for credentials to login we need to pass elastic user credentials because we are accessing kibana as a elastic user.



```
ubuntu@ip-172-31-92-184:~$ docker run -d --name kibana --net elk --restart unless-stopped \
  -p 5601:5601 \
  -e "ELASTICSEARCH_HOSTS=http://elasticsearch:9200" \
  -e "ELASTICSEARCH_USERNAME=kibana" \
  -e "ELASTICSEARCH_PASSWORD=aupSVjQGfqUOP2KulcNX" \
  docker.elastic.co/kibana/kibana:8.5.0
Unable to find image 'docker.elastic.co/kibana/kibana:8.5.0' locally
8.5.0: Pulling from kibana/kibana
43850375985f: Already exists
a366ea6d76f5: Pull complete
5298c2a2d220: Pull complete
977e2726c907: Pull complete
4d0357043e57: Pull complete
89732bc75041: Pull complete
4f86444baf75: Pull complete
d5b3a3160c3d: Pull complete
b5baae302a6b: Pull complete
f715f414700b: Pull complete
345b7ce1a8de: Pull complete
2ca985534712: Pull complete
177e26bbd0fb: Pull complete
e0c120226ed6: Pull complete
Digest: sha256:ca1276adff4f8d413086b6e3c10a563185f6cd714676e7a40e98aa93490196fb
Status: Downloaded newer image for docker.elastic.co/kibana/kibana:8.5.0
48602be6a2954fe061c73da1a682328c885feb7e7541559a2e65a6e78beb7656
ubuntu@ip-172-31-92-184:~$ docker ps
CONTAINER ID   IMAGE                                                  COMMAND                  CREATED              STATUS              PORTS
                                                                     NAMES
48602be6a295   docker.elastic.co/kibana/kibana:8.5.0                    "/bin/tini -- /usr/l…"   About a minute ago   Up About a minute   0.0.0.0:5601->5601/tc
p, :::5601->5601/tcp                                                 kibana
dbe58cf6fefc   docker.elastic.co/elasticsearch/elasticsearch:8.5.0    "/bin/tini -- /usr/l…"   8 minutes ago        Up 8 minutes        0.0.0.0:9200->9200/tc
p, :::9200->9200/tcp, 0.0.0.0:9300->9300/tcp, :::9300->9300/tcp   elasticsearch
ubuntu@ip-172-31-92-184:~$
```


Not secure 54.84.35.158:5601/app/home#/
elastic
Find apps, content, and more.
Home
Help us improve the Elastic Stack
To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.
Dismiss
Welcome home
Enterprise Search
Create search experiences with a refined set of APIs and tools.
Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.
Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.
Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.
Get started by adding integrations

Installing LogStash:

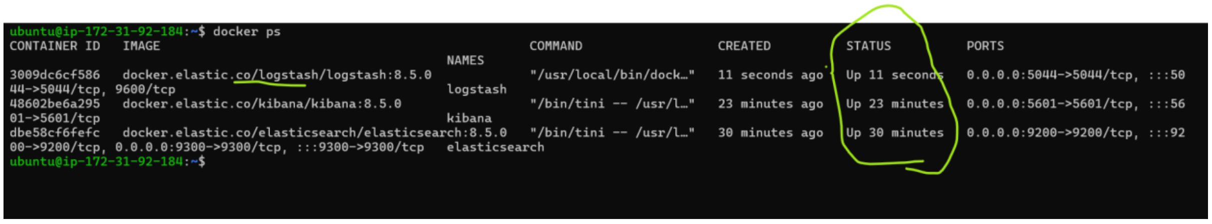
- create a file with name **logstash.conf** in the current directory and add the below content to it to let know LogStash about the elasticsearch.

```
input {
 beats {
   port => 5044
 }
}

filter {
 grok {
   match => { "message" => "%{SYSLOGLINE}" }
 }
 date {
   match => [ "timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
 }
}

output {
 elasticsearch {
   hosts => ["http://elasticsearch:9200"]
   user => "elastic"
   password => " CmvsTq1e1IwtfMEatChF "
   index => "syslog"
 }
 stdout { codec => rubydebug }
}
```

- To create the file use **sudo vi logstash.conf** command and add the above content and save the file using **:wq!** instruction.
- In the above replace the **elastic_user_password** with the **elastic** user password which is generated in the 1st step.
- Then run the below command to deploy LogStash container.

```
docker run -d --name logstash --net elk --restart unless-stopped \
 -v $(pwd)/logstash.conf:/usr/share/logstash/pipeline/logstash.conf \
 -p 5044:5044 \
 docker.elastic.co/logstash/logstash:8.5.0
```

- After deploying it check the containers status using **docker ps** (or) **docker container ls** command, if this command shows all three containers status as running then ELK set up done.



```
ubuntu@ip-172-31-92-184:~$ docker ps
CONTAINER ID   IMAGE                                                  COMMAND                  CREATED          STATUS          PORTS
                                                      NAMES
3009dc6cf586   docker.elastic.co/logstash/logstash:8.5.0              "/usr/local/bin/dock…"   11 seconds ago   Up 11 seconds   0.0.0.0:5044->5044/tcp, :::50
44->5044/tcp, 9600/tcp                                logstash
48602be6a295   docker.elastic.co/kibana/kibana:8.5.0                  "/bin/tini -- /usr/l…"   23 minutes ago   Up 23 minutes   0.0.0.0:5601->5601/tcp, :::56
01->5601/tcp                                          kibana
dbe58cf6fefc   docker.elastic.co/elasticsearch/elasticsearch:8.5.0    "/bin/tini -- /usr/l…"   30 minutes ago   Up 30 minutes   0.0.0.0:9200->9200/tcp, :::92
00->9200/tcp, 0.0.0.0:9300->9300/tcp, :::9300->9300/tcp   elasticsearch
ubuntu@ip-172-31-92-184:~$
```

Monitoring EC2 instance using ELK:

- To monitor an EC2 instance we need to install betas in the server which we wanted to monitor.

What are beats in ELK:

- Beats are lightweight data shippers designed to collect and send various types of operational data (logs, metrics, network data, etc.) from different machines to Logstash or Elasticsearch.
- Beats are part of the Elastic Stack, which includes Elasticsearch, Logstash, Kibana, and Beats.
- Each Beat is purpose-built to collect specific types of data, and they are often used to collect data from edge nodes and ship it to a central location for processing and analysis.

Commonly used beats:

- File Beat
- Metric Beat
- Heart Beat
- Packet Beat

Here I have taken **File Beat** and **Metric Beat** because,

- **File Beat** is a lightweight shipper for forwarding and centralizing log data. It monitors log files or locations you specify, collects log events, and forwards them to Elasticsearch or Logstash.
- **Metric Beat** collects metrics from the operating system and from services running on the server. It can monitor system-level metrics such as CPU, memory, and network usage, as well as application-level metrics.

- Here, in the server which I wanted to monitor, I'm installing the beats using docker, so I have created another **ubuntu-22.04** EC2 instance of size **t2.micro** and installed docker as above.

Installing File Beat:

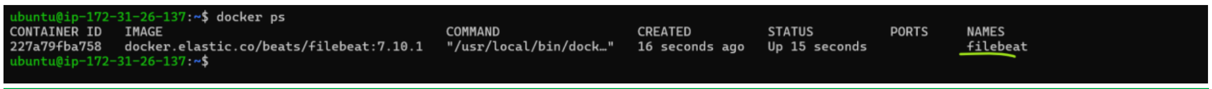
- Firstly, we need to create a **filebeat.yml** file and insert the below data, which is essential to let know file beat about the elasticsearch details.

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/*.log
    - /path/to/your/logs/*.log

output.elasticsearch:
  hosts: ["http://54.84.35.158:9200/ "]
  username: "elastic"
  password: "elastic_user_password"
```

Note:

- Replace the elasticsearch host and elastic user password and save the file.
- To create and open the file with an editor use **sudo vi filebeat.yml** command and add the above content and then save it using **:wq!** instruction.
- Then use the below command which will deploy a file beat container.
- The above command will add the created **filebeat.yml** file to the container as a volume for the container.

```
docker run -d --name=filebeat \
  --user=root \
  --volume="$(pwd)/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro" \
  --volume="/var/log:/var/log:ro" \
  --volume="/path/to/your/logs:/path/to/your/logs:ro" \
  docker.elastic.co/beats/filebeat:7.10.1 filebeat -e -strict.perms=false
```



```
ubuntu@ip-172-31-26-137:~$ docker ps
CONTAINER ID   IMAGE                                     COMMAND                  CREATED          STATUS          PORTS     NAMES
227a79fba758   docker.elastic.co/beats/filebeat:7.10.1   "/usr/local/bin/dock…"   16 seconds ago   Up 15 seconds             filebeat
ubuntu@ip-172-31-26-137:~$
```

Installing Metric Beat:

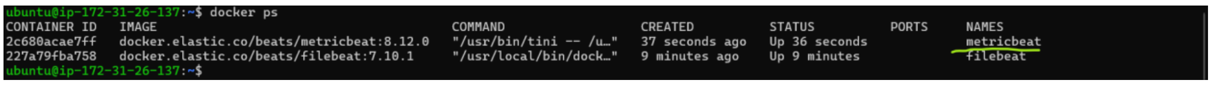
- Firstly, we need to create a **metricbeat.yml** file and insert the below data, which is essential to let know filebeat about the elasticsearch details.

Note:

- Replace the **elasticsearch host** and **elastic user password** and save the file.
- To create and open the file with an editor use **sudo vi metricbeat.yml** command.
- Add the below content and then save it using **:wq** instruction in **vi** editor.
- If you are using **nano** editor then use **sudo nano metricbeat.yml** command to create and open the file and save the file using **Ctrl+X** and **Y.**
- Add the below content in the **metricbeat.yml** file.

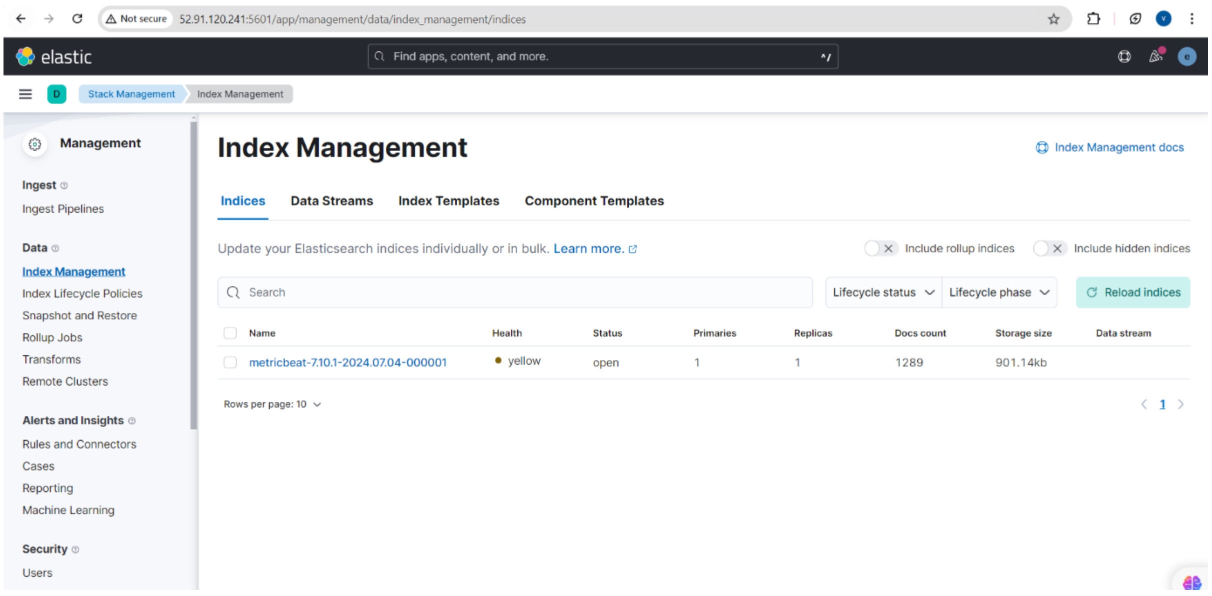
```
metricbeat.modules:
- module: system
  metricsets:
    - cpu
    - load
    - memory
    - network
    - process
    - process_summary
    - socket_summary
    - uptime
  enabled: true
  period: 10s
  processes: ['.*']

output.elasticsearch:
  hosts: ["http://54.84.35.158:9200/ "]
  username: "elastic"
  password: " CmvsTq1e1IwtfMEatChF "
```

```
docker run -d --name=metricbeat \
 --user=root \
 --volume="$(pwd)/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml:ro" \
 --volume="/sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro" \
 --volume="/proc:/hostfs/proc:ro" \
 --volume="/:/hostfs:ro" \
 docker.elastic.co/beats/metricbeat:8.12.0 \
 -system.hostfs=/hostfs
```



```
ubuntu@ip-172-31-26-137:~$ docker ps
CONTAINER ID   IMAGE                                       COMMAND                  CREATED          STATUS          PORTS     NAMES
2c680acae7ff   docker.elastic.co/beats/metricbeat:8.12.0   "/usr/bin/tini -- /u…"   37 seconds ago   Up 36 seconds             metricbeat
227a79fba758   docker.elastic.co/beats/filebeat:7.10.1     "/usr/local/bin/dock…"   9 minutes ago    Up 9 minutes              filebeat
ubuntu@ip-172-31-26-137:~$
```

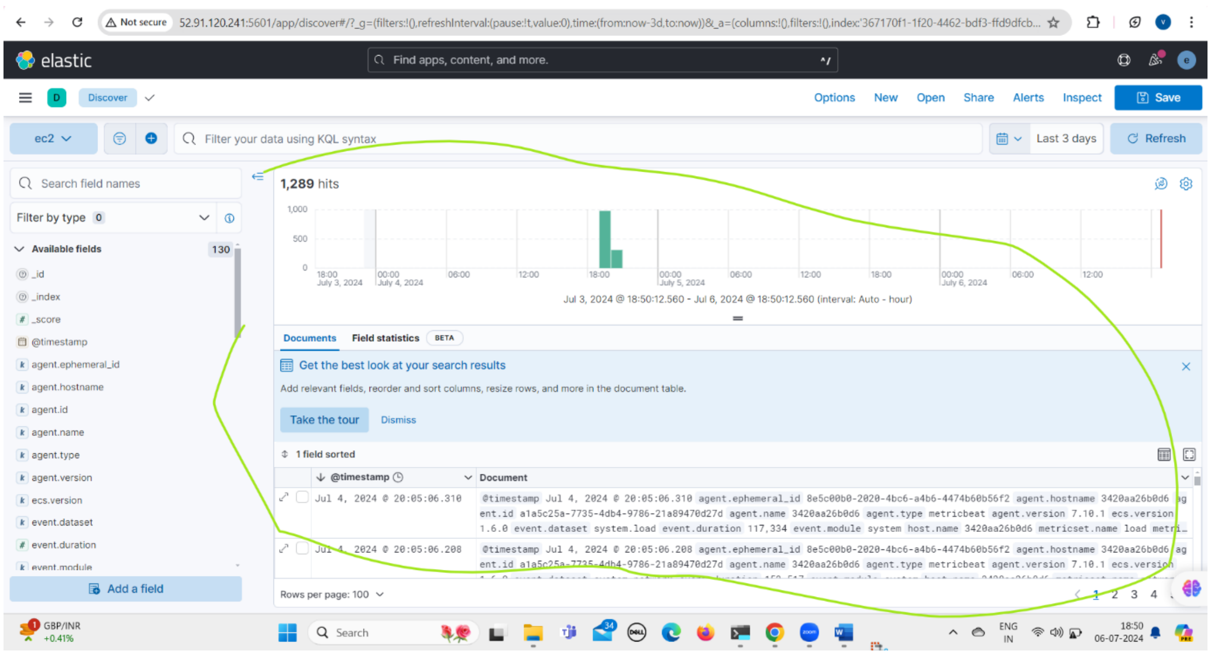
- ➢ Then use the above command which will deploy a filebeat container. This command will add the created **metricbeat.yml** file to the container as a volume for the container.
- ➢ Then login to kibana dashboard and to Configure Index Patterns > Go to Management > Index Patterns.
- ➢ Create index patterns for **filebeat-*** and **metricbeat-***.
- ➢ Navigate to Discover in Kibana to start exploring logs and metrics collected from your EC2 instance.
- ➢ You can also create visualizations and dashboards based on the data.



Index Management

Update your Elasticsearch indices individually or in bulk. Learn more.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
metricbeat-7.10.1-2024.07.04-000001	● yellow	open	1	1	1289	901.14kb	



52.91.120.241:5601/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-3d,to:now))&_a=(columns:!(),filters:!(),index:'367170f1-1f20-4462-bdf3-ffd9dfcb...
elastic
Discover
Options New Open Share Alerts Inspect Save
ec2
Filter your data using KQL syntax
Last 3 days
Refresh
1,289 hits
Jul 3, 2024 @ 18:50:12.560 - Jul 6, 2024 @ 18:50:12.560 (interval: Auto - hour)
Documents
Field statistics
Available fields 130
_id
_index
_score
@timestamp
agent.ephemeral_id
agent.hostname
agent.id
agent.name
agent.type
agent.version
ecs.version
event.dataset
event.duration
event.module
Add a field

Get the best look at your search results

Add relevant fields, reorder and sort columns, resize rows, and more in the document table.

1 field sorted

@timestamp	Document
Jul 4, 2024 @ 20:05:06.310	@timestamp Jul 4, 2024 @ 20:05:06.310 agent.ephemeral_id 8e5c00b0-2020-4bc6-a4b6-4474b60b56f2 agent.hostname 3420aa26b0d6 agent.id a1a5c25a-7735-4db4-9786-21a89470d27d agent.name 3420aa26b0d6 agent.type metricbeat agent.version 7.10.1 ecs.version 1.6.0 event.dataset system.load event.duration 117,334 event.module system host.name 3420aa26b0d6 metricset.name load metri…
Jul 4, 2024 @ 20:05:06.208	@timestamp Jul 4, 2024 @ 20:05:06.208 agent.ephemeral_id 8e5c00b0-2020-4bc6-a4b6-4474b60b56f2 agent.hostname 3420aa26b0d6 agent.id a1a5c25a-7735-4db4-9786-21a89470d27d agent.name 3420aa26b0d6 agent.type metricbeat agent.version 7.10.1 ecs.version

Rows per page: 100

- This is the process to setup ELK using docker and monitor an EC2 instance using beats.

Thanks,

Lahari G