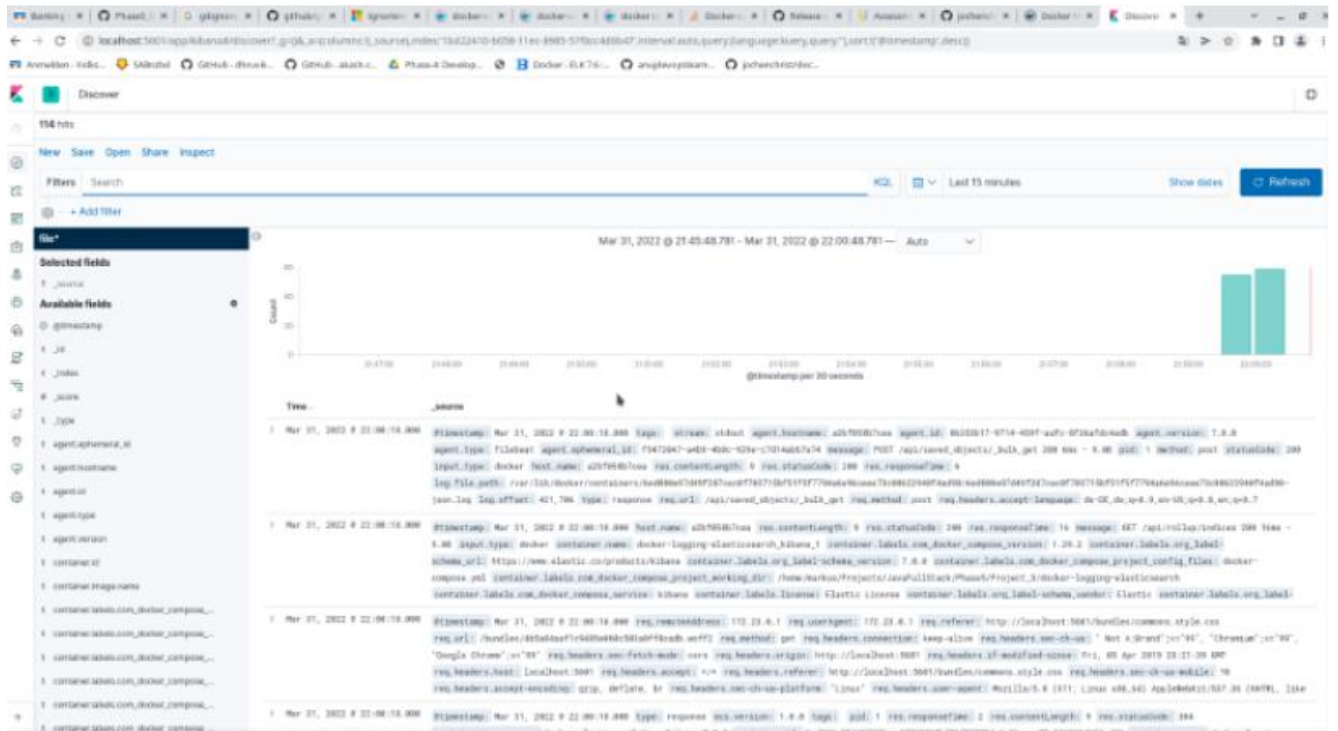


```

markus@master: /opt/jdk$ sudo docker ps
(base) markus@Labor:~/Projects/JavaFullStack/Phase5/Project_3$ sudo docker ps
[sudo] Passwort für markus:
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS        PORTS
6ed800e97d49   docker.elastic.co/kibana/kibana:7.0.0   "/usr/local/bin/kiba..."   25 hours ago   Up 25 hours   0.0.0.0:5601-->5601/tcp, :::5601/tcp
5601/tcp       docker-logging-elasticsearch_kibana_1
a2bf058b7ce0   docker-logging-elasticsearch_filebeat   "/usr/local/bin/dock..."   25 hours ago   Up 25 hours
docker-logging-elasticsearch_filebeat_1
c41dddec5925   docker.elastic.co/elasticsearch/elasticsearch:7.0.0   "/usr/local/bin/dock..."   25 hours ago   Up 25 hours   0.0.0.0:9200-->9200/tcp, :::9200/tcp
9200/tcp       elasticsearch
e844538aa1c1   docker-logging-elasticsearch_demo        "java -Djava.securit..."   25 hours ago   Up 25 hours   0.0.0.0:8080-->8080/tcp, :::8080/tcp
8080/tcp       docker-logging-elasticsearch_demo_1
(base) markus@Labor:~/Projects/JavaFullStack/Phase5/Project_3$

```



Create index pattern

★ file*

Time Filter field name: @timestamp

This page lists every field in the **file*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#).

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	
@version	number		●	●	
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				

Docker Compose :

version: '3.2'

services:

demo:

run `./mvnw clean package` before

build: ./demo

ports:

- 8080:8080

filebeat:
build: ./filebeat
volumes:
- /var/lib/docker/containers:/var/lib/docker/containers:ro
- /var/run/docker.sock:/var/run/docker.sock
networks:
- es
depends_on:
- elasticsearch
kibana:
image: docker.elastic.co/kibana/kibana:7.0.0
ports:
- 5601:5601
environment:
ELASTICSEARCH_URL: http://elasticsearch:9200
networks:
- es
depends_on:
- elasticsearch
elasticsearch:
image: docker.elastic.co/elasticsearch/elasticsearch:7.0.0
container_name: elasticsearch
environment:
- cluster.name=docker-cluster

```
- "ES_JAVA_OPTS=-Xms512m -Xmx512m"
- "network.host=0.0.0.0"
- "discovery.zen.minimum_master_nodes=1"
- "discovery.type=single-node"
```

ulimits:

memlock:

soft: -1

hard: -1

volumes:

```
- esdata:/usr/share/elasticsearch/data
```

ports:

```
- 9200:9200
```

networks:

```
- es
```

volumes:

esdata:

driver: local

networks:

es:

FileBeat

```
FROM docker.elastic.co/beats/filebeat:7.0.0
```

```
COPY filebeat.yml /usr/share/filebeat/filebeat.yml
```

```
# must run as root to access /var/lib/docker and /var/run/docker.
```

```
sock USER root
```

```
RUN chown root /usr/share/filebeat/filebeat.yml
```

```
# dont run with ==e, to dis
```

```
able output to stderr CMD ["""]
```

filebeat.yml

filebeat.inputs:

```
- type: docker
```

```
containers.ids: '*'
```

```
json.message_key: message
```

```
json.keys_under_root: true
```

```
json.add_error_key: true
```

```
json.overwrite_keys: true
```

processors:

```
- add_docker_metadata: ~
```

output.elasticsearch:

```
hosts: ["elasticsearch:9200"]
```

```
logging.to_files: true
```

```
logging.to_syslog: false
```

DemoApplication :

```
package com.example.demo;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
@SpringBootApplication
@RestController
public class DemoApplication {
private static final Logger logger =
LoggerFactory.getLogger(DemoApplication.class);
public static void main(String[] args) {
SpringApplication.run(DemoApplication.class, args);
}
@GetMapping("/")
public String hello() {
logger.info("Hello World");
logger.error("Ooops, there was an error", new
RuntimeException("I am a runtime exception"));
return "Hello World";
}
}
```