

Implementación del stack ELK

1) Creación del archivo de configuración de Logstash

Vamos a generar un archivo de configuración **dh-spring-elk-conf.conf** dentro de la carpeta conf de Logstash con las configuraciones vistas de input, filter y output.

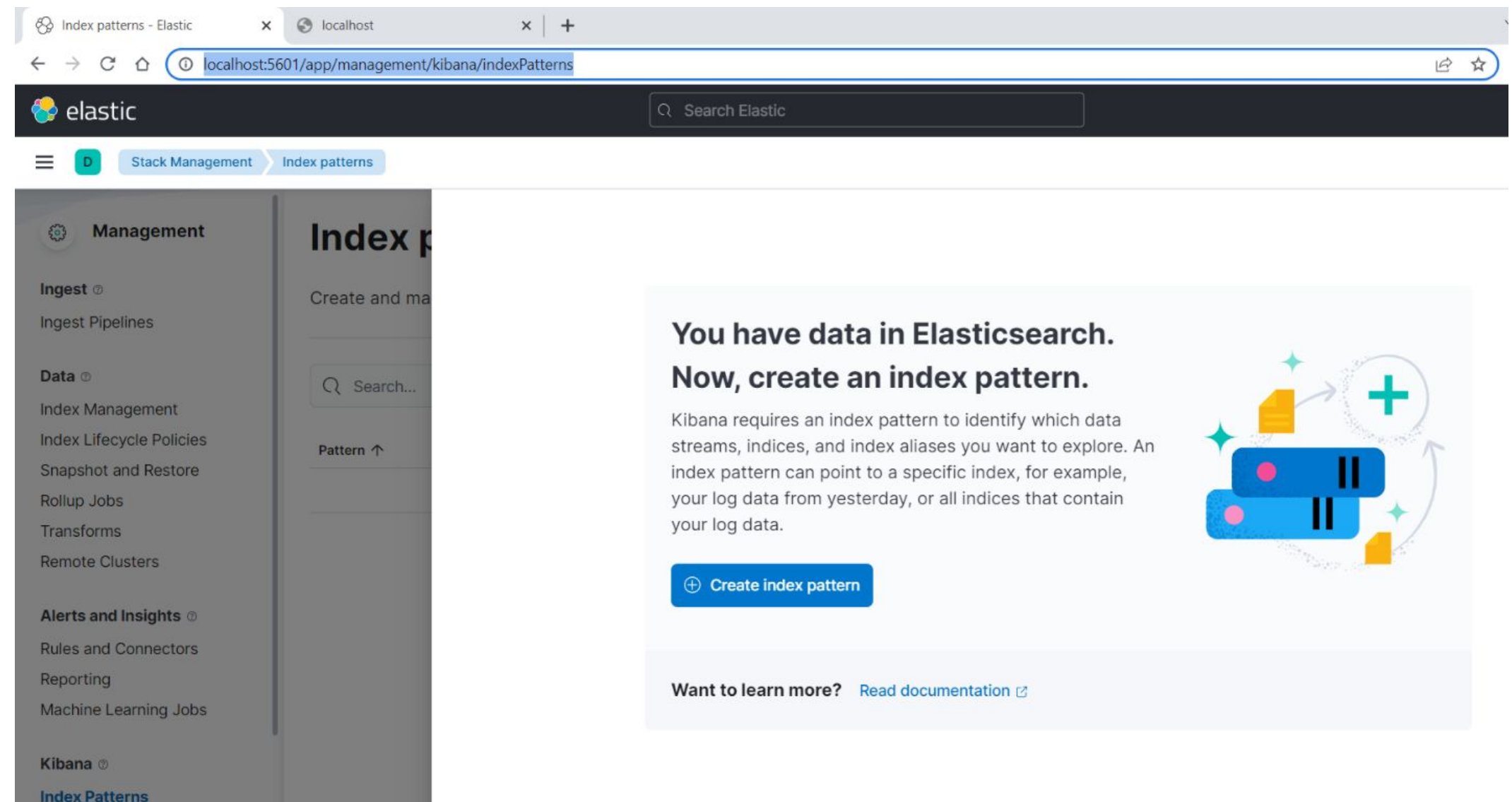
Este archivo será el usado en la inicialización de Logstash: **logstash -f ../config/dh-spring-elk-conf.conf**

2) Creación de índice en Kibana

Vamos a generar el **índice** en Kibana basados en el patrón de logueo que tiene Logstash al tomar la información de nuestro log de aplicación y ponerlo en Elastic. Este comienza con **logstash-***.

Entramos a la siguiente URL

<http://localhost:5601/app/management/kibana/indexPatterns> y seleccionamos la opción de creación de índice.



elastic

Search Elastic

Stack Management

Index patterns

Management

Ingest

Data

Ingest Pipelines

Index Management

Index Lifecycle Policies

Snapshot and Restore

Rollup Jobs

Transforms

Create and manage index patterns

Search...

Pattern

Create index pattern

Name

logstash-*

Use an asterisk (*) to match multiple characters. Spaces and the characters , / ? , " , < , > , | are not allowed.

Timestamp field

@timestamp

Select a timestamp field for use with the global time filter.

Show advanced settings

✓ Your index pattern matches 1 source.

logstash-2022.02.14-000001

Index

Rows per page: 10

logstash-*

Time field: '@timestamp'

View and edit fields in logstash-*. Field attributes, such as type and searchability, are based on field mappings in Elasticsearch.

Fields (11)

Scripted fields (0)

Field filters (0)

Search

All field types

Add field

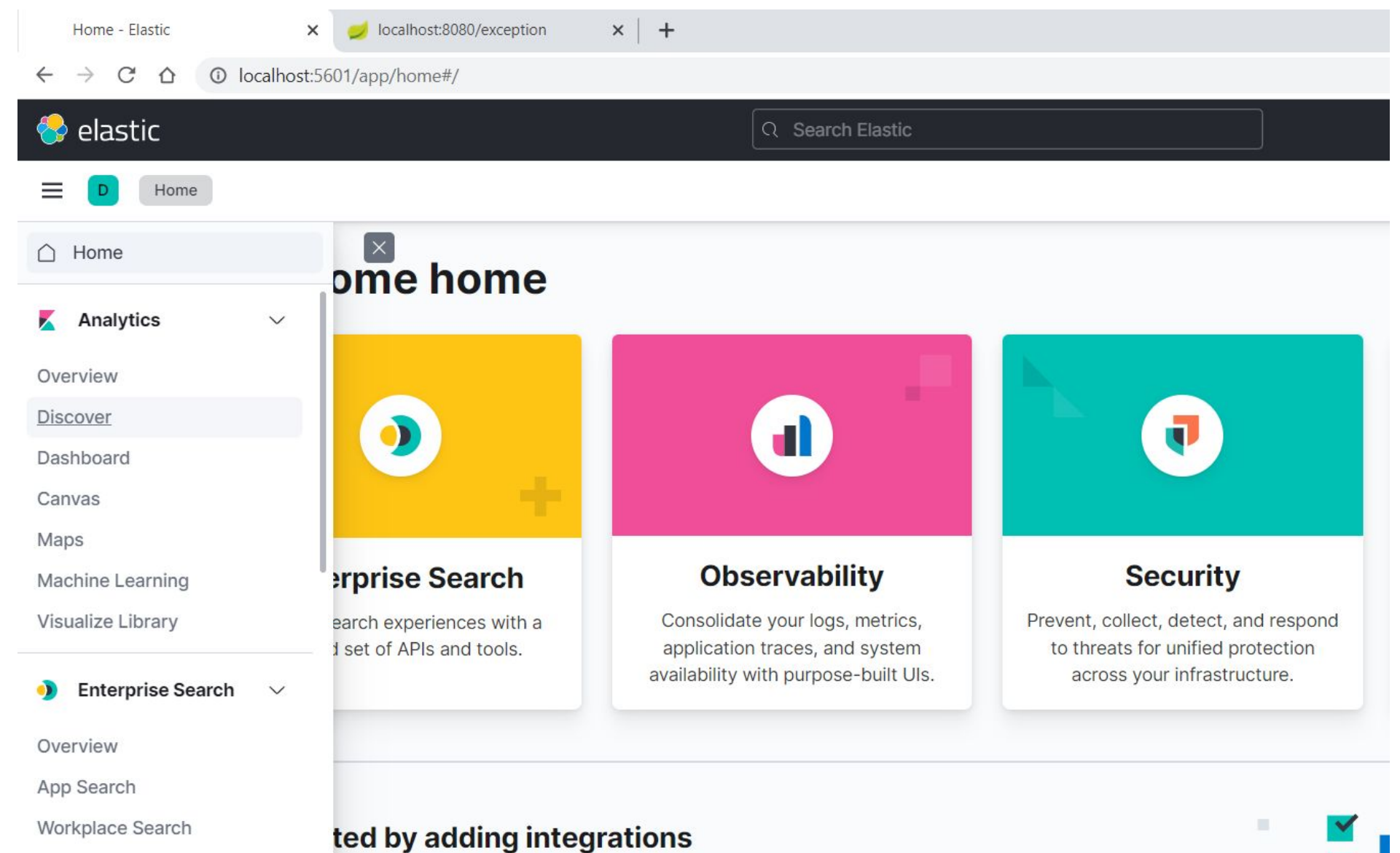
Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date				
@version	keyword				
id	id				

Luego de confirmar la creación, veremos la pantalla de éxito con los *fields* indexados de nuestro índice.

3) Visualización de logs centralizados

En este tercer paso realizamos interacciones con nuestra aplicación que genera los logs en nuestro archivo analizado por Logstash, para así alimentar nuestra base de datos de Elasticsearch.

Al finalizar, podemos visualizar las entradas en Kibana dirigiéndonos a la opción **Analytics** y seleccionando **Discover**.



Discover - Elastic

+

localhost:5601/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now%2Fd,to:now%2Fd))&_a=(columns:!(),filters:!(),index:'67501780-8db2-11ec-ae63-c9d4d9...)

elastic

Search Elastic

Discover

OptionsNewOpenShareInspect

SearchKQLTodayShow dates

+ Add filter

logstash-*

Search field names

Filter by type0

Available fields11

_id

_index

_score

_type

@timestamp

@version

host

message

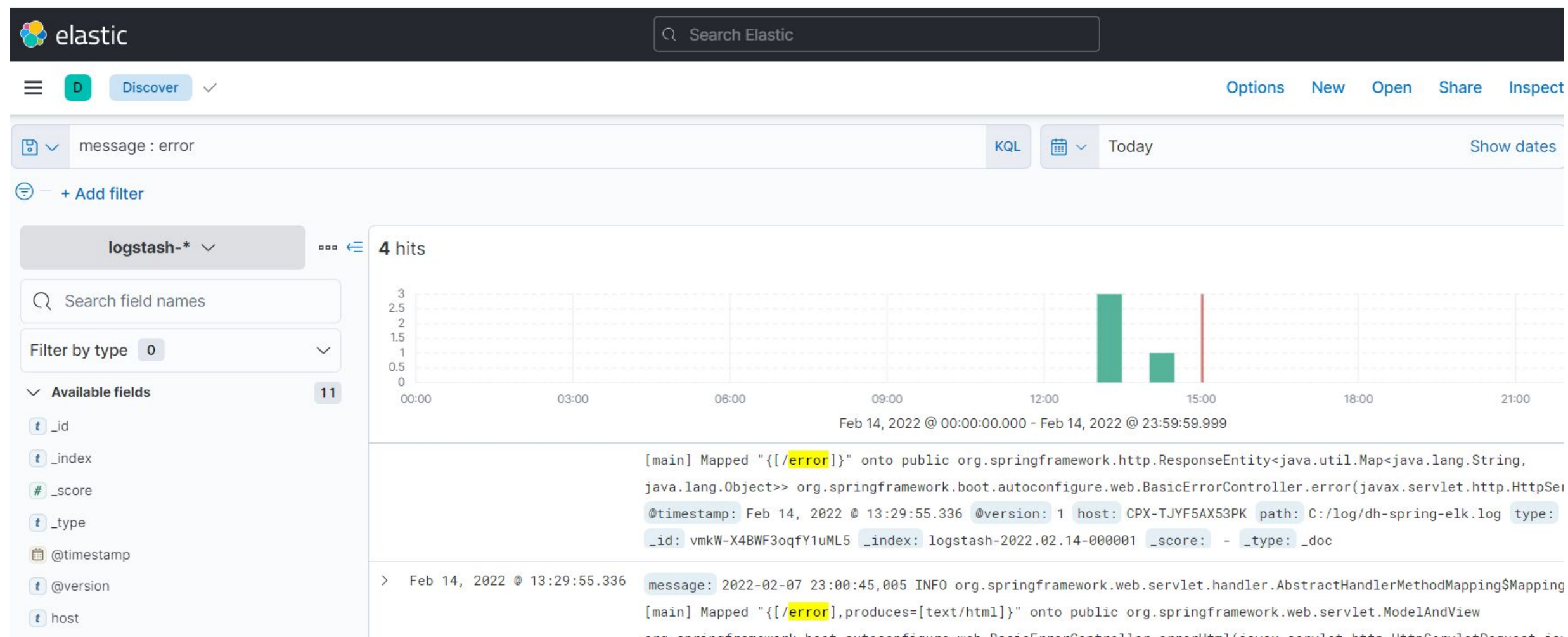
48 hits

Feb 14, 2022 @ 00:00:00.000 - Feb 14, 2022 @ 23:59:59.999

Time	Document
> Feb 14, 2022 @ 14:25:45.153	@timestamp: Feb 14, 2022 @ 14:25:45.153 @version: 1 host: CPX-TJYF5AX53PK message: 2022-02-14 14:25:44,602 ERROR com.example.consumerservice.ELKService [http-nio-8080-exec-2] test exception java.lang.IllegalArgumentException: e generada en path /exception at com.example.consumerservice.ELKService.exception(ELKServiceApplication.java:40) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at
> Feb 14, 2022 @ 14:25:45.101	@timestamp: Feb 14, 2022 @ 14:25:45.101 @version: 1 host: CPX-TJYF5AX53PK message: 2022-02-14 14:24:56,266 INFO

4) Consultas por queries KQL

Por último, Kibana nos permite realizar consultas en los datos indexados en Elasticsearch mediante la sintaxis del framework KQL. Podemos, por ejemplo, consultar todos los errores donde el mensaje que se loguea sea de tipo error:



¡Muchas gracias!