Introducción al Stack Elastic

Para ejecutar las demos se debe clonar el proyecto https://github.com/immavalls/viu-elk-ml-talk

```
git clone git@github.com:immavalls/viu-elk-ml-talk.git
```

E instalar los prerequisitos documentados. Estas demos requieren **docker** y **docker-compose**.

Demo 1

Setup del entorno

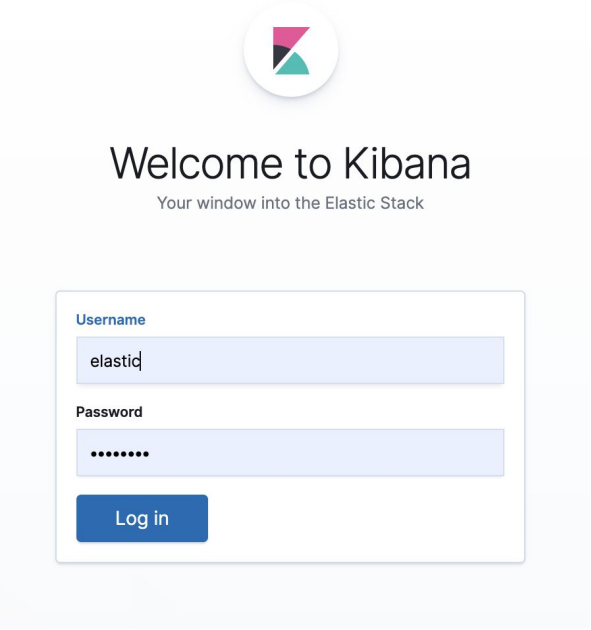
Para ello, situados en el raíz del proyecto **viu-elk-ml-talk**, ejecutar:

```
docker-compose up -d
```

Comprobar que ha arrancado correctamente:

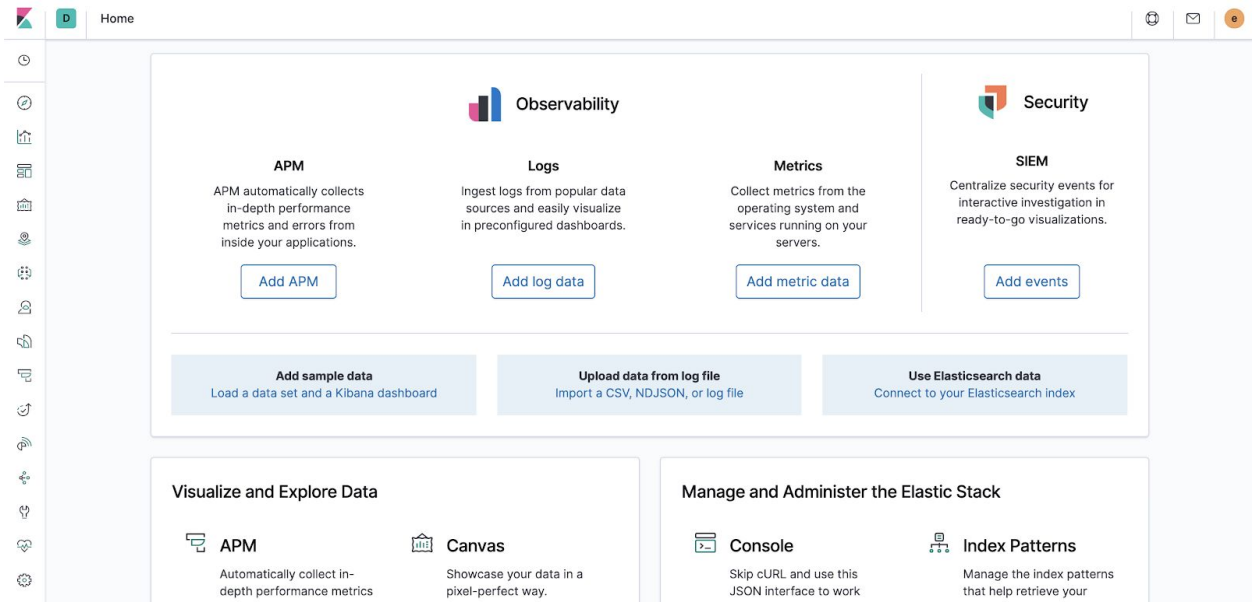
http://localhost:5601/

Si Elasticsearch y Kibana han arrancado correctamente, veremos el pantalla de Login de Kibana.



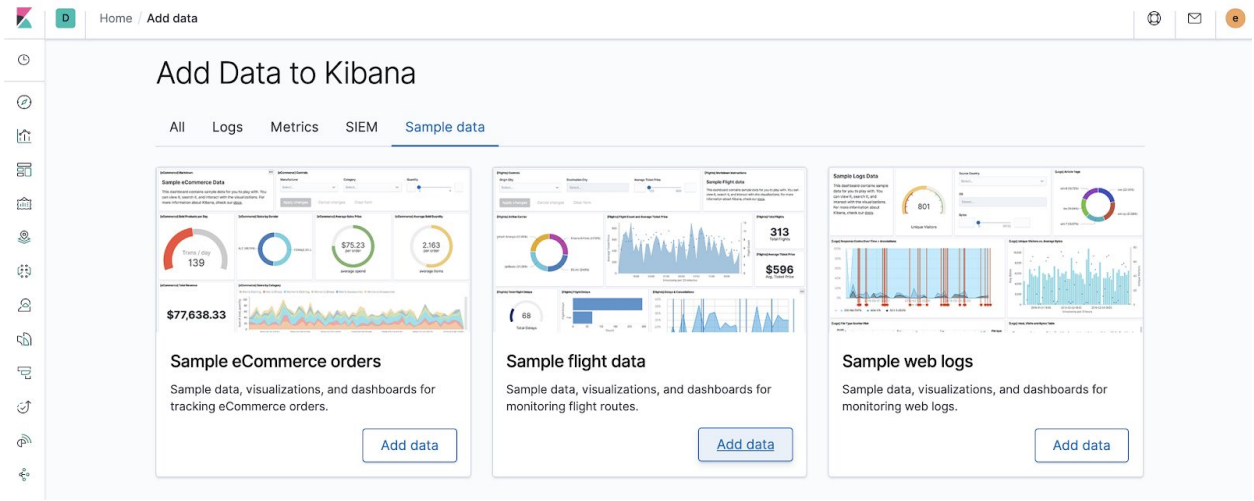
Entrar con:

Usuario: elastic
Contraseña: changeme



Vamos a cargar datos de prueba que proporciona Kibana, para facilitar la exploración del Stack.

En la parte central de la pantalla Home, Pulsar en “Add sample data!, “Load a data set and a Kibana dashboard”.



Pulsar “Add data” en **Sample flight data** y en **Sample web logs**.

En la parte inferior izquierda, en el menu, pulsar en la flecha  “Expand”. Aquí podemos ver todas las opciones de menú de Kibana. Escogemos  , “Dev Tools”.

Vamos a preparar los datos de vuelos para poder hacer búsquedas de texto libre. Para ello, ejecutar lo siguiente.

En primer lugar, añadiremos un campo de tipo text, necesarios para realizar las búsquedas.[1]

```
POST kibana_sample_data_flights/_mappings
{
  "properties": {
    "origin": {
      "type": "keyword",
      "fields": {
        "text": {
          "type": "text"
        }
      }
    }
  }
}
```

I forzaremos un update de todos los documentos del índice para que este campos se actualice[2]
.

```
POST kibana_sample_data_flights/_update_by_query
{
  "script": {
    "source": """
      ctx._source.origin = ctx._source.Origin;
    """,
    "lang": "painless"
  }
}
```

[1] https://www.elastic.co/blog/strings-are-dead-long-live-strings
[2] https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-update-by-query.html

CRUD

- Crear un documento con identificador 1.

```
POST kibana_sample_data_flights/_doc/1
{
  "FlightNum": "652J760",
  "Origin": "Amsterdam Airport Schiphol",
  "Dest": "Stockholm-Arlanda Airport",
  "FlightDelayMin": 0,
  "Cancelled" : false,
  "timestamp" : "2020-05-01T05:21:34"
}
```

- Leer un documento

```
GET kibana_sample_data_flights/_doc/1
```

- Actualizar un documento

```
POST kibana_sample_data_flights/_update/1
{
  "doc": {
    "FlightDelayMin": 260
  }
}
```

- Borrar un documento

```
DELETE kibana_sample_data_flights/_doc/1
```

Queries

Podemos realizar distintas queries, usando Query DSL[3].

- ¿Vuelos con origen en Amsterdam?

```
GET kibana_sample_data_flights/_search
{
  "query": {
    "match": {
      "origin.text": "amsterdam"
    }
  }
}
```

- ¿Vuelos retrasados 60 minutos o más?

```
GET kibana_sample_data_flights/_search?filter_path=hits.hits._source
{
  "size": 10,
  "_source": ["FlightDelayMin","FlightNum"],
  "query": {
    "range": {
      "FlightDelayMin": {
        "gte": 60
      }
    }
  }
}
```

[3] https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

- ¿Vuelos con destino Amsterdam retrasados 60 minutos o más?

```
GET kibana_sample_data_flights/_search
{
  "size": 5,
  "_source": [
    "FlightDelayMin",
    "FlightNum"
  ],
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "origin.text": "amsterdam"
          }
        }
      ],
      "filter": [
        {
          "range": {
            "FlightDelayMin": {
              "gte": 60
            }
          }
        }
      ]
    }
  }
}
```

Agregaciones

- ¿Cuáles son los tope 3 aeropuertos origen?

```
GET kibana_sample_data_flights/_search
{
  "size": 0,
  "aggregations": {
    "top_aeropuertos_origen": {
      "terms": {
        "field": "origin",
        "size": 3
      }
    }
  }
}
```

- ¿Qué retraso tienen en media los vuelos en los top 2 aeropuertos origen?

```
GET kibana_sample_data_flights/_search
{
  "size": 0,
  "aggregations": {
    "top_aeropuertos_origen": {
      "terms": {
        "field": "origin",
        "size": 3
      },
      "aggregations": {
        "media_retraso": {
          "avg": {
            "field": "FlightDelayMin"
          }
        }
      }
    }
  }
}
```

Gist

https://gist.github.com/immavalls/b3d9b1b3985e75f86f64836632250baf

Demo 2 - Ingesta de documentos

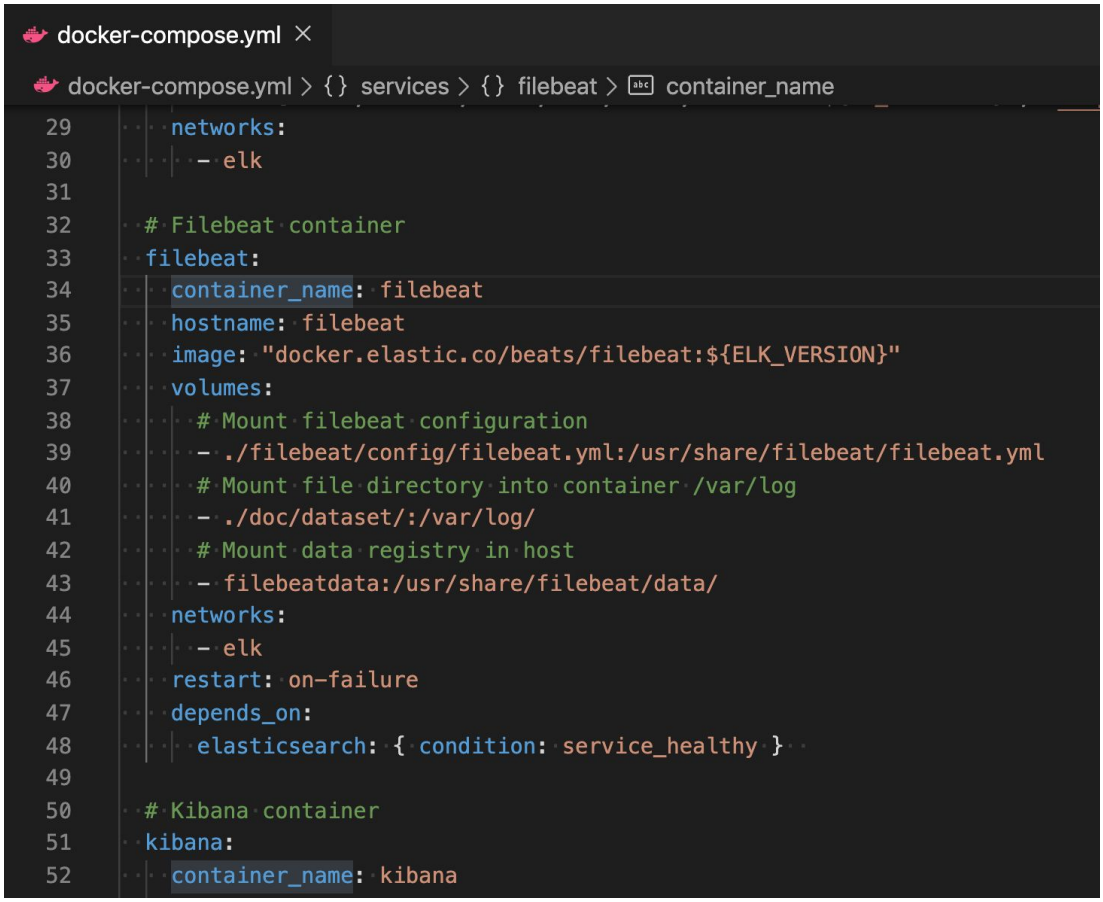
En esta demo, vamos a ingestar en Elasticsearch un documento en formato JSON. Para ello, usaremos Filebeat.

El documento en JSON que usaremos como ejemplo es el listado de bibliotecas de la Comunitat Valenciana en 2020: https://dadesobertes.gva.es/es/dataset/cul-dir-bibliotecas-2020

Dentro del proyecto encontraréis el documento en formato CSV (original) y JSON. El fichero que vamos a usar es el ubicado en `doc/datasets/bibliotecas-comunidad-valenciana-2020.ndjson`. Hemos convertido el CSV a JSON, para facilitar la ingesta.

Para ingestar este fichero, usaremos Filebeat. La pipeline de ingestar sería la siguiente, en la que Filebeat va a leer de un fichero en formato JSON (un documento por línea), y lo enviará a Elasticsearch:

Para arrancar Filebeat, editaremos el fichero docker-compose.yml que está en el raíz del proyecto, y descomentamos la parte donde se define el contenedor de Filebeat.



Ese contenedor está configurado para leer el fichero `doc/datasets/bibliotecas-comunidad-valenciana-2020.ndjson` y enviar los documentos JSON a Elasticsearch.

Arrancamos Filebeat con:

```
docker-compose up -d
```

Y podemos visualizar los logs para comprobar que está funcionando correctamente:

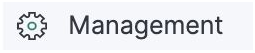
```
docker-compose logs -f filebeat
```

Para comprobar que la ingesta ha sido correcta ejecutando en *Kibana Dev Tools* la siguiente query:

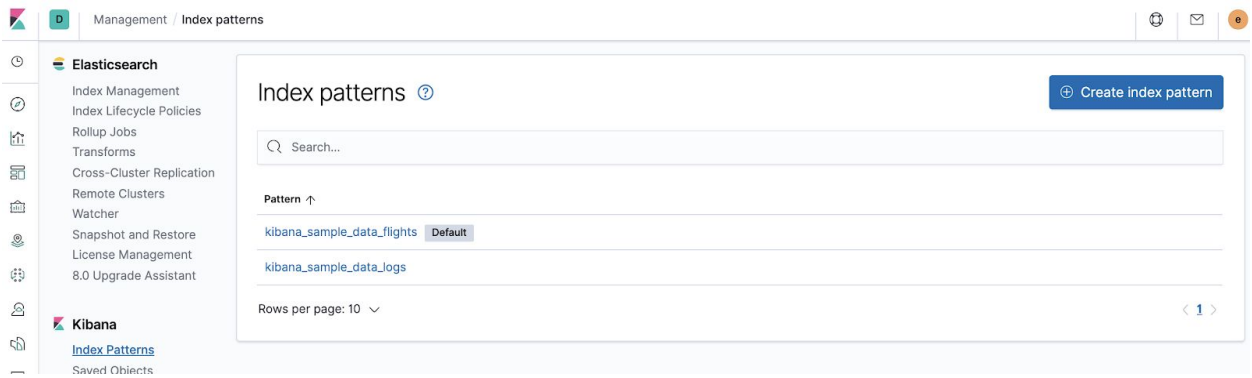
```
GET bibliotecas-valencia/_search?filter_path=hits.total,hits.hits._source.NOMBRE
```

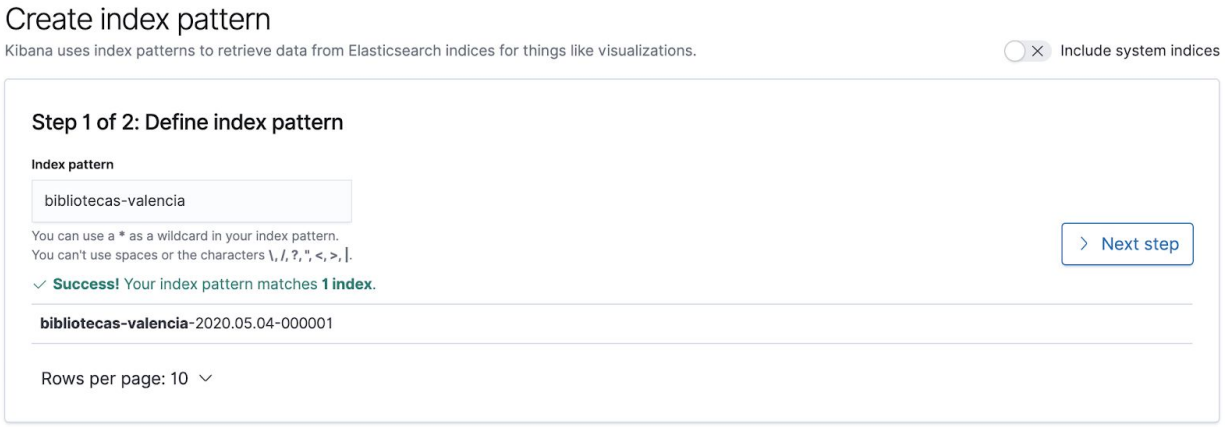
Que nos devolverá un total de 685 resultados, y al no definir size en la query, 10 de esos documentos:

```
{
  "hits" : {
    "total" : {
      "value" : 685,
      "relation" : "eq"
    },
    "hits" : [
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA SANTA CECILIA MISLATA"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA  MOIXENT"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "BIBLIOTECA PUBLICA MUNICIPAL  MONTSERRAT"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "OTRAS PÚBLICAS  MONTÁN"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA  MONTAVERNER"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA  MONTROI"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "BIBLIOTECA PUBLICA MUNICIPAL JUAN BAUTISTA MUÑOZ FERRANDIS MUSEROS"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "OTRAS PÚBLICAS  NOVETLÈ"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA SANT RAFEL ONTINYENT"
        }
      },
      {
        "_source" : {
          "NOMBRE" : "AGENCIA DE LECTURA  ORBA"
        }
      }
    ]
  }
}
```

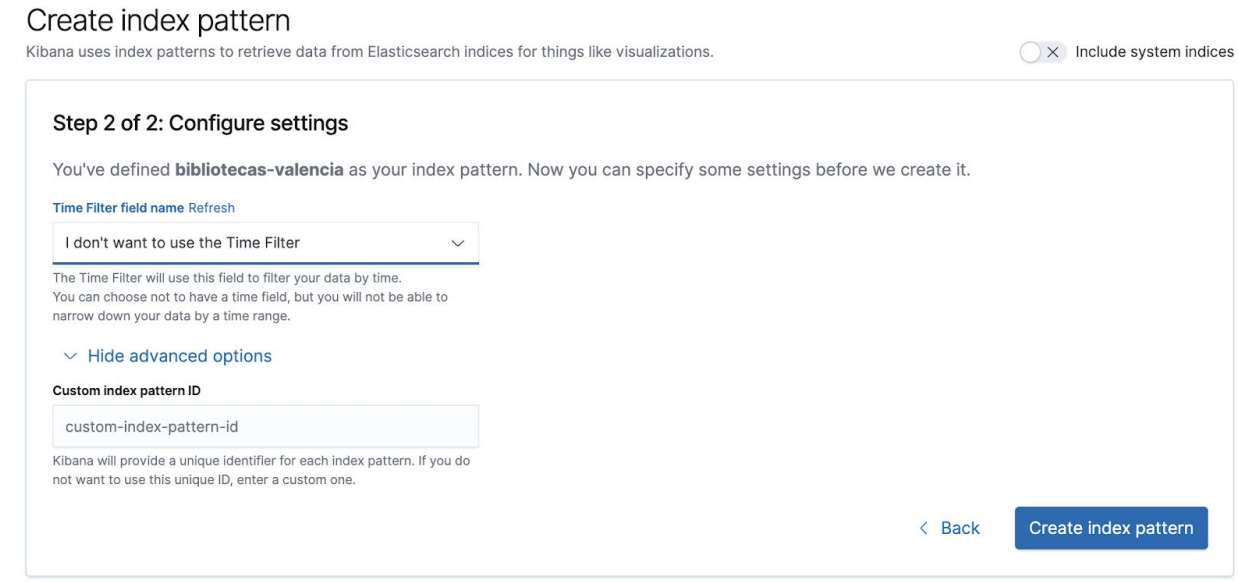
Finalmente, si queremos visualizar estos datos en Kibana, crearíamos un **Index Pattern**[4]. Para ello, pulsamos en el menú de Kibana, 

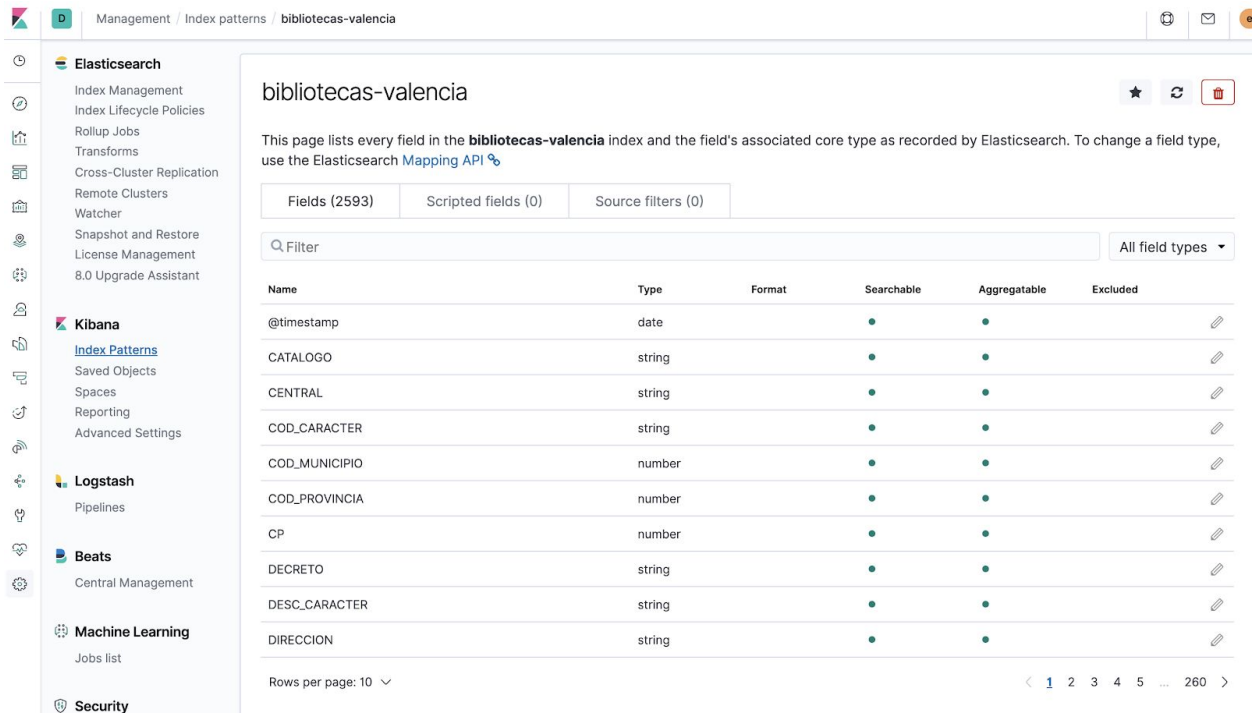
"*Management*"., y creamos un index pattern para *"bibliotecas-valencia"*, seleccionando en "*I don't want to use the Time Filter"* (estos documentos no tienen ningún timestamp).

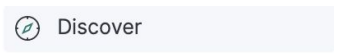




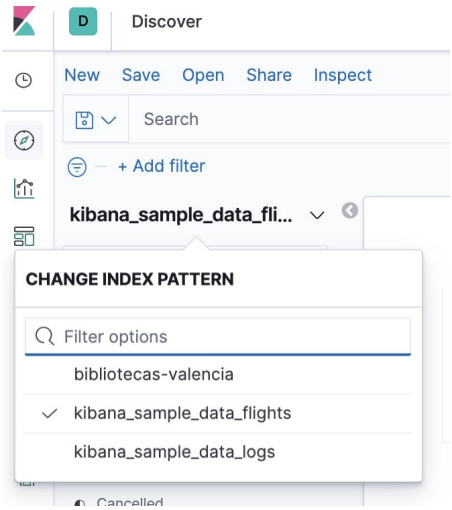
[4] https://www.elastic.co/guide/en/kibana/current/index-patterns.html





Una vez creado el patrón para el índice, escogemos Kibana 

en el menú de la izquierda. Seleccionaremos ese index pattern, y podremos ver los documentos creados.


Discover
New
Save
Open
Share
Inspect
Search
+ Add filter
kibana_sample_data_fli...
CHANGE INDEX PATTERN
Filter options
bibliotecas-valencia
kibana_sample_data_flights
kibana_sample_data_logs
Cancelled


Discover
New
Save
Open
Share
Inspect
Search
KQL
Refresh
+ Add filter
bibliotecas-valencia
685 hits
_source
Search field names
Filter by type
0
Selected fields
_source
Available fields
@timestamp
CATALOGO
CENTRAL
COD_CARACTER