

Para realizar la creación de un índice se debe usar la palabra “POST” seguido de el nombre que se le asignara a este, en este ejemplo el nombre del índice es “log\_consultas”:

Nombre del  
Índice

```
POST log_consultas/_doc/
{
  "timestamp": "2010-05-15T22:00:54",
  "estado_consulta": "consumo",
  "servicio": "consulta",
  "administrador": "Juan Carlos",
  "consultas_realizadas": 52
}
```

Al realizar la ejecución de este bloque en nuestra consola tendremos una salida que contiene detalles generales de la creación de nuestro índice y también el status de “creado”:

```
1 {
2   "_index" : "log_consultas",
3   "_type" : "_doc",
4   "_id" : "zWYtP3wB46yHbDBVG7LI",
5   "_version" : 1,
6   "result" : "created",
7   "_shards" : {
8     "total" : 2,
9     "successful" : 1,
10    "failed" : 0
11  },
12   "_seq_no" : 0,
13   "_primary_term" : 1
14 }
15
```

Nombre del  
Índice

Índice creado

Para obtener el mapping de nuestro índice anterior, se hizo uso de un “GET” con el nombre de nuestro índice y “\_mapping” como se muestra en la siguiente imagen:

```
10 GET /log_consultas/_mapping
11
```

Lo cual en consola nos muestra lo siguiente:

```

"log_consultas" : {
  "mappings" : {
    "_meta" : {
      "created_by" : "file-data-visualizer"
    },
    "properties" : {
      "@timestamp" : {
        "type" : "date",
        "format" : "iso8601"
      },
      "administrador" : {
        "type" : "keyword"
      },
      "consultas_realizadas" : {
        "type" : "long"
      },
      "estado_consulta" : {
        "type" : "keyword"
      },
      "index" : {
        "properties" : {
          "_id" : {
            "type" : "long"
          },
          "_index" : {
            "type" : "text",

```

Esto nos da los datos que almacena el índice así como su tipo de dato.

Para realizar el template hacemos uso de este mapping para la creación de el para esto hacemos uso de un “PUT” seguido de “\_index\_template” y después se le asigna un nombre al template mismo donde nuestro índice de patron es “log\_consultas\*”:

```

12 PUT _index_template/consultas_template
13 {
14   "index_patterns": [
15     "log_consultas*"
16   ],
17   "template": {
18     "settings": {
19       "number_of_shards": 1
20     },
21     "mappings": {
22       "properties": {
23         "@timestamp": {
24           "type": "date"
25         },
26         "administrador": {
27           "type": "text",
28           "fields": {
29             "keyword": {
30               "type": "keyword",
31               "ignore_above": 256
32             }
33           }
34         },
35         "consultas_realizadas": {
36           "type": "long"

```

Nombre del  
template

Índice del  
patrón


Al ejecutar este bloque nos lanzara en consola el siguiente mensaje que indica que el template fue creado correctamente:

```
1 {  
2   "acknowledged" : true  
3 }  
4
```

Para realizar la carga de datos de el “json” proporcionado, se hizo uso de la interfaz proporcionada por la plataforma de elastic de la siguiente manera:

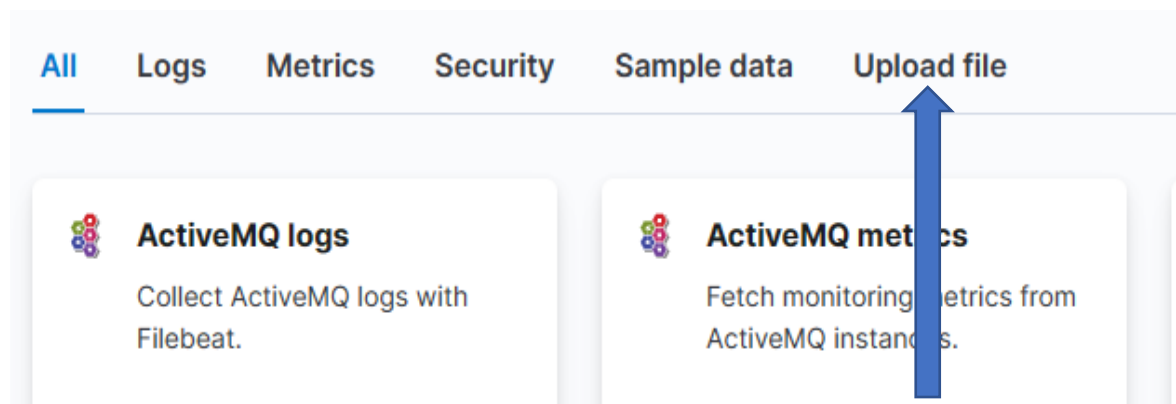
## Get started by adding your data

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

 Add your data



Al presionar en ese botón, nos dirigira al siguiente menú donde tenemos que seleccionar la opción de “upload file”



Nos dirigira a la siguiente pestaña donde podremos seleccionar nuestro archivo

## Visualize data from a log file

Upload your file, analyze its data, and optionally import the data into an Elasticsearch index.

The following file formats are supported:

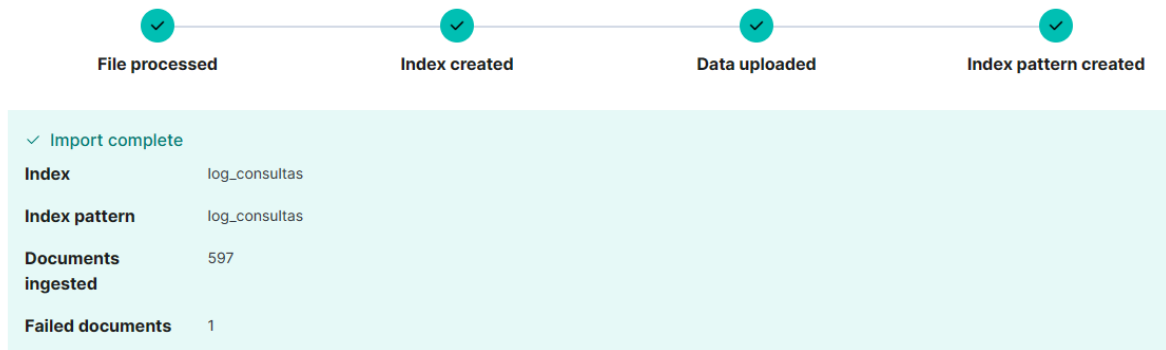
- Delimited text files, such as CSV and TSV
- Newline-delimited JSON
- Log files with a common format for the timestamp

You can upload files up to 100 MB.



Select or drag and drop a file

Al seleccionar nuestro archivo se procesará y quedara añadido a el archivo en el index que seleccionemos en este caso “log\_consultas”



Para obtener los números de registros se usa un “GET” seguido del nombre del índice y el API search un signo y la consulta que se quiere realizar.

Consulta de “error”:

```
61 GET /log_consultas/_search?q=error
62
```

Salida en consola de la consulta:

```
1 {
2   "took" : 2,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 78,
13      "relation" : "eq"
14    },
15    "max_score" : 1.3406837,
16    "hits" : [
17
```



La cantidad de valores con “error”

Consulta de “consumo”:

```
63 GET /log_consultas/_search?q=consumo
64
```

Salida en consola de la consulta:

```
1 {
2   "took" : 2,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 105,
13      "relation" : "eq"
14    },
15    "max_score" : 1.0450715,
16    "hits" : [
```



La cantidad de valores con “consumo”

Consulta de administrador "Juan Lara":

```
65 GET /log_consultas/_search?q=Juan Lara
66
```

Salida en consola de la consulta:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 98,
      "relation" : "eq"
    },
    "max_score" : 1.1137259,
    "hits" : [
```



La cantidad de valores con "Juan Lara"

Consulta de administrador "Informativo":

```
GET /log_consultas/_search?q=informativo
```

Salida en consola de la consulta:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 116,
      "relation" : "eq"
    },
    "max_score" : 0.94589114,
    "hits" : [
      {

```



La cantidad de valores con "Informativo"

Consulta de administrador “borrado”:


```
69 GET /log_consultas/_search?q=borrado|
70
```

Salida en consola de la consulta:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 105,
      "relation" : "eq"
    },
    "max_score" : 1.0450715,
    "hits" : [
```

La cantidad de  
valores con  
“Borrado”

Para crear un patron índice buscamos “Kibana/Index patterns” lo que nos dirigira a la siguiente pestaña y deberemos seleccionar la siguiente opción:

A blue rectangular button with rounded corners. On the left is a white plus sign inside a circle. To its right is the text "Create index pattern" in white.

Nos arroja la siguiente pestaña y llenaremos los campos con lo solicitado:

## Create index pattern

Name

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

Timestamp field



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

[Show advanced settings](#)

× Close

Create index pattern



En la siguiente parte nos dirigimos a la opción de “visualize” para encontrar las vistas disponibles y hacemos clic aquí:

## Visualize Library



+ Create visualization

Seleccionamos la opción de “explore options”:



### Aggregation based

Use our classic visualize library to create charts based on aggregations.

[Explore options →](#)

Y seleccionamos nuestro Index:



3. Login Failed Details



Group Management Details - Search View [Windows System Security]





groupadd logs [Logs System]



log\_consultas



Para crear el **Heat Map** debemos seleccionar en el eje X las siguientes especificaciones de términos de servicio:

▼ X-axis  = 

Aggregation [Terms help](#)

Terms ▼

Field

servicio ▼

Order by

Metric: Count ▼

Order

Descending ▼

Size

5

Y para el caso del eje Y el siguiente termino con el administrador:

Y-axis

Sub aggregation [Terms help](#)

Terms

Field

administrador

Order by

Metric: Count

Order

Descending

Size

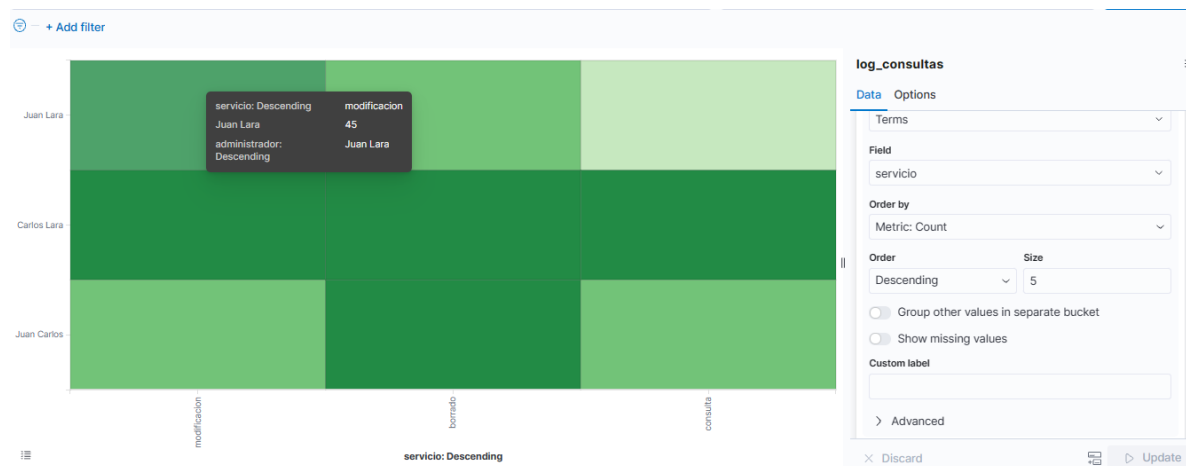
5

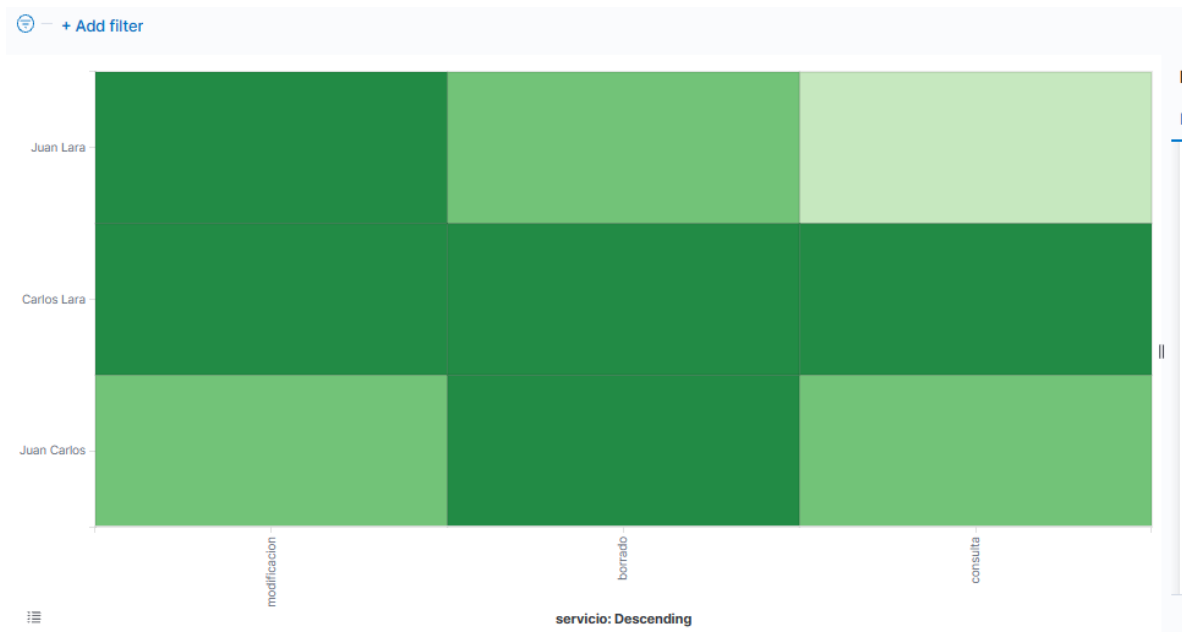
☐ Group other values in separate bucket

☐ Show missing values

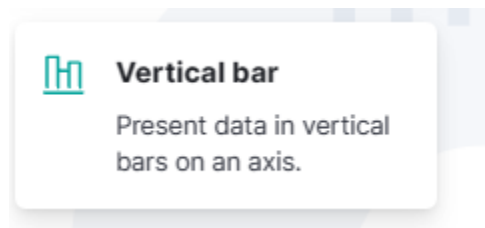
Custom label

Lo que al hacer clic en el botón de “update” nos arrojará el siguiente mapa:





Para crear la grafica de barras repetimos los pasos iniciales y seleccionamos la visualización en barras y nuevamente seleccionamos nuestro índice:



Determinamos las siguientes especificaciones para el eje x:

X-axis
👁 = ✖

Aggregation
Date Histogram help

Date Histogram

Field
@timestamp

Minimum interval
Auto

Select an option or create a custom value. Examples: 30s, 20m, 24h, 2d, 1w, 1M

☐ Drop partial buckets

Custom label

Y los siguientes parámetros para dividir las barras de esta:

Split series

Sub aggregation

Terms

Field

estado\_consulta

Order by

Metric: error

Order

Ascending

Size

5

☐ Group other values in separate bucket

☐ Show missing values

Custom label

Lo que nos arrojará las siguientes gráficas en las que podemos seleccionar lo que vemos y en el rango de tiempo en el que lo vemos:

