

Documentación para la práctica Elasticsearch.



Banti Serna Brandon Aldair

11/Octubre/2021

Crear cuenta en Elastic.

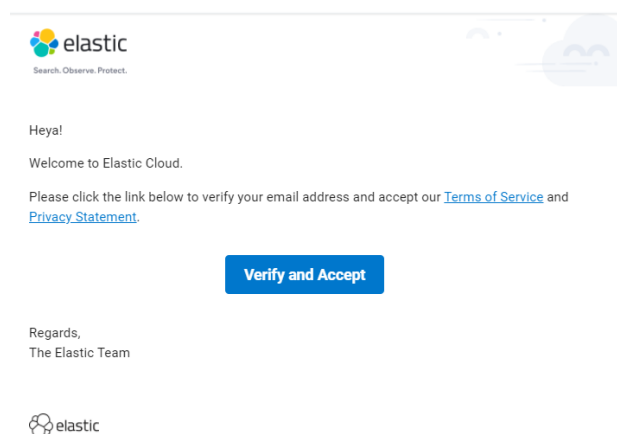
1._ Se inició en la página oficial de Elastic (<https://www.elastic.co/>)



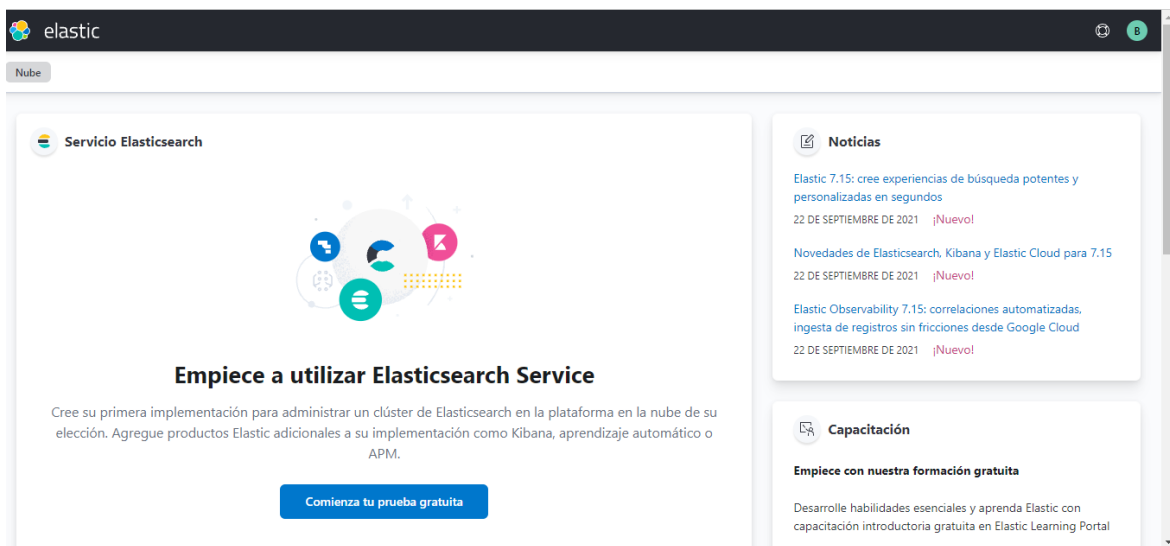
2._ Se creó una cuenta en Elastic Cloud trial.

A screenshot of the 'Start your free Elastic Cloud trial' form. It includes fields for 'Email' (filled with 'banser.100@gmail.com') and 'Password'. Below these is a 'Start free trial' button. There is also an option to 'Or sign up with' Google or Microsoft. At the bottom, a disclaimer states: 'By signing up, you acknowledge that you've read and agree to our Terms of Service and Privacy Statement.'

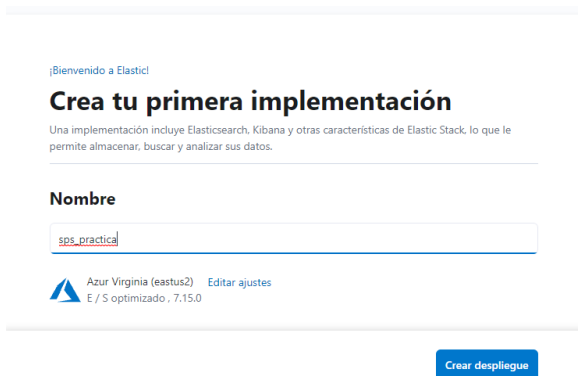
3._ Se validó el correo que fue ingresado.



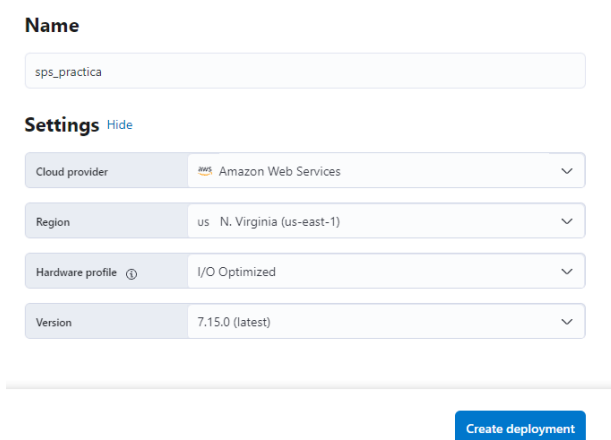
4. Después de que se validó el correo nos dirige a la página principal de Elastic.



5._ Requerimos iniciar con nuestra primera implementación, la creación del deployment.



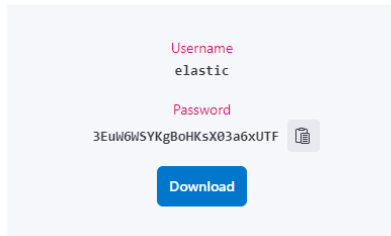
6._ Se creó con las siguientes configuraciones: (Nombre del deployment: sps_practica, Plataforma: Amazon Web Service Region: US East (N. Virginia), Elastic Stack versión: Mas reciente y Optimize your deployment: I/O Optimized.



7._ En el momento de la creación del deployments. Se observan las credenciales requeridas para el clouster.

Save the deployment credentials

These root credentials are shown only once.
They provide super user access to your deployment. Keep them safe.

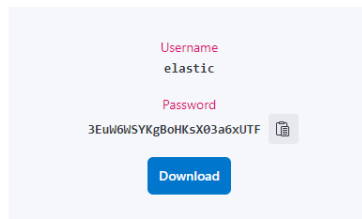


8._ En este proceso tarda unos minutos, posterior a ellos nos aparece una leyenda que el deployment está listo y un botón para continuar.

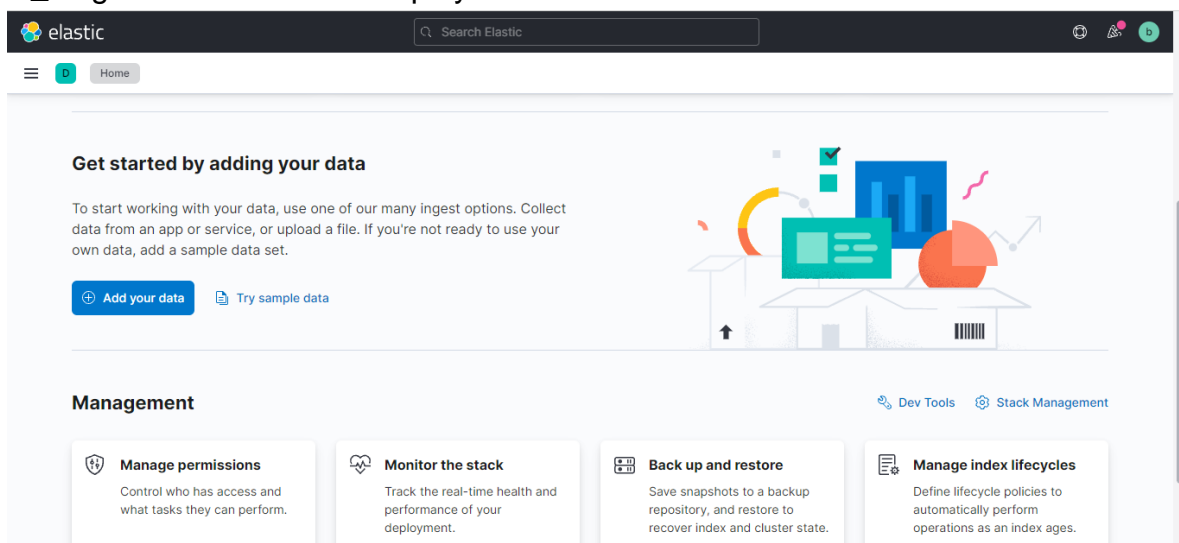


Save the deployment credentials

These root credentials are shown only once.
They provide super user access to your deployment. Keep them safe.



9._ Llegamos al Home del deployment creado.



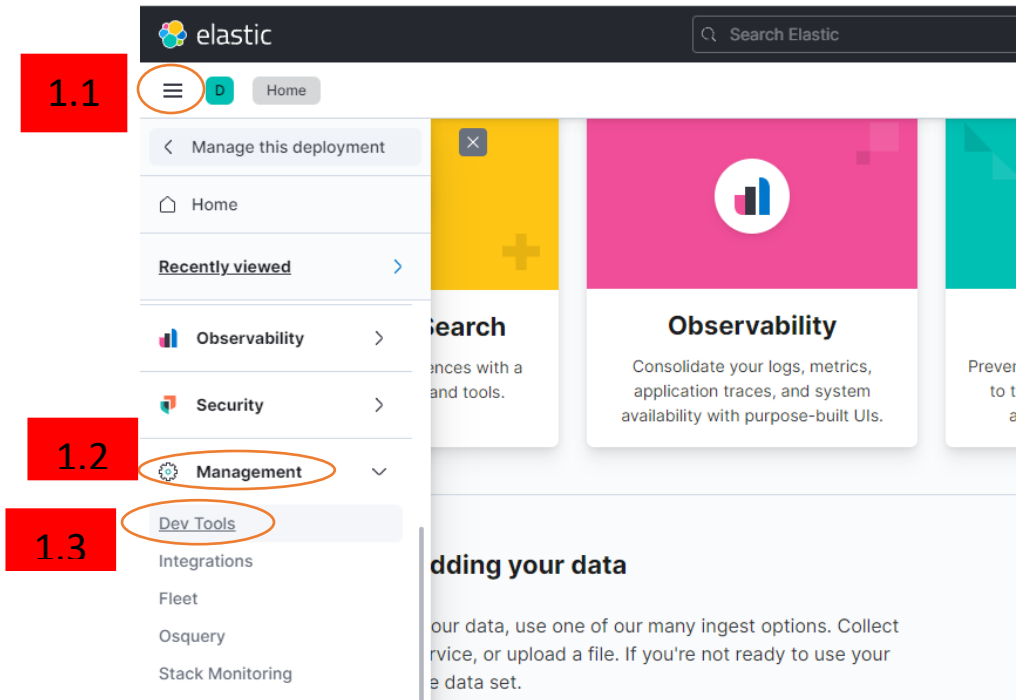
Creación de un índice.

1._ Lo primero es abrir Dev Tools (herramientas para desarrollador)

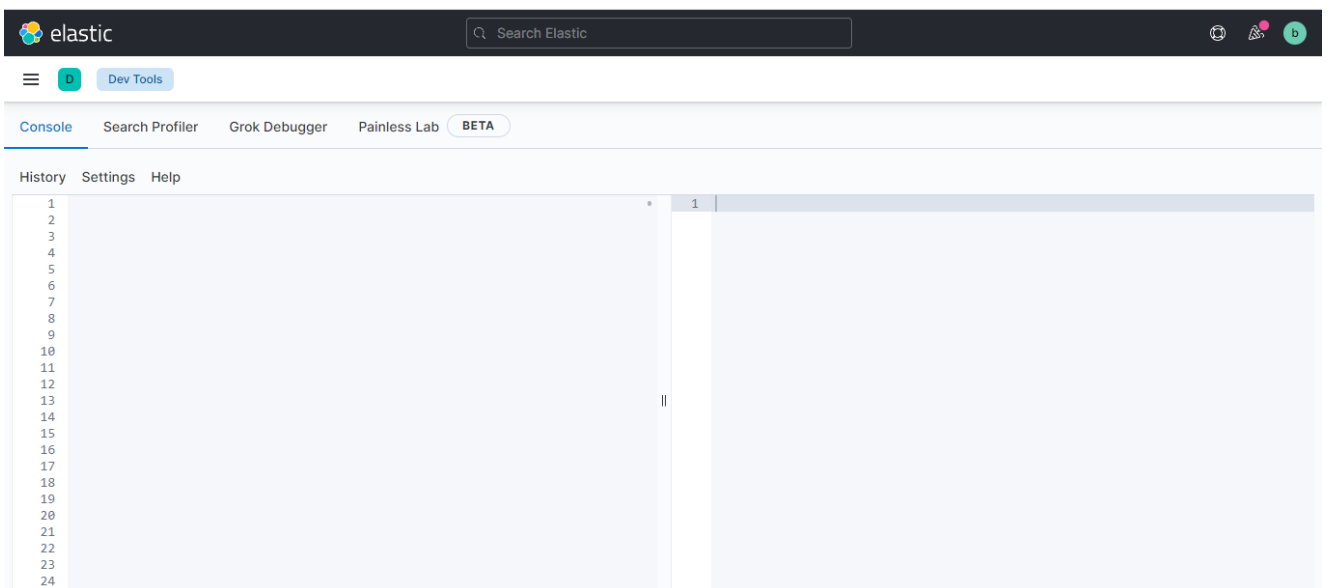
1.1._ Abrir el menú.

1.2._ Buscar el apartado de Management.

1.3._ Abrir Dev Tools.



2._ Nos aparecerá una consola con la cual trabajaremos.



3._ Continuamos con la revisión de la documentación correspondiente. Se realizó una investigación más a fondo para terminar de entender los conceptos básicos.

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index.html#docs-index>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index.html>
- <https://www.youtube.com/watch?v=uTg7w0prP6A>
- <https://www.youtube.com/watch?v=H-l5m3u-3oA>
- <https://www.youtube.com/watch?v=kPJmrt1zT8c>
- <https://www.youtube.com/watch?v=YTCQHFch8pc>

4._ Se realizaron peticiones REST para ir interactuando con la consola.

```
1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
7
8 GET /
9
10 GET _cat/indices
```

```
1 green open .ent-search-actastic-oauth_access_tokens-refresh_token-unique
   -constraint
   -dt0aQ5ucF8319oOM_A 1 1 0 0 416b 208b CH
2 green open .ent-search-actastic-workplace_search_accounts_v16
   -KTDf8pxkZA 1 1 1 0 12.1kb 6kb Qllzax3eR0u
3 green open .ent-search-actastic-app_search_crawler_content_metadata
   -content_hash-engine_oid-unique-constraint
   19uISCrFQ-uoQVn_eG-Rw 1 1 0 0 416b 208b
4 green open apm-7.15.0-profile-000001
   --iMKPPqNA 1 1 0 0 416b 208b J8Yx88tGRB5j
5 green open .ent-search-db-lock-20200304
   _BhD_yPaTeeHN1kT62komQ 1 1 0 4 398.5kb 81.5kb
6 green open .ent-search-workplace-search-analytics-ecs-ilm-logs-production
   -2021.10.07-000001
   aZC7mjPQDKpd07zV3A_6Q 1 1 0 0 416b 208b
7 green open .ent-search-actastic-oauth_access_tokens-token-unique-constraint
   -qRE8g 1 1 1 0 7.3kb 3.6kb kJzIip1KTqOuFxy
8 green open .kibana-event-log-7.15.0-000001
   NqITjZ0uTueSv36T027jw 1 1 1 0 12.1kb 6kb
9 green open .ent-search-actastic-workplace search search groups v4-name-unique
```

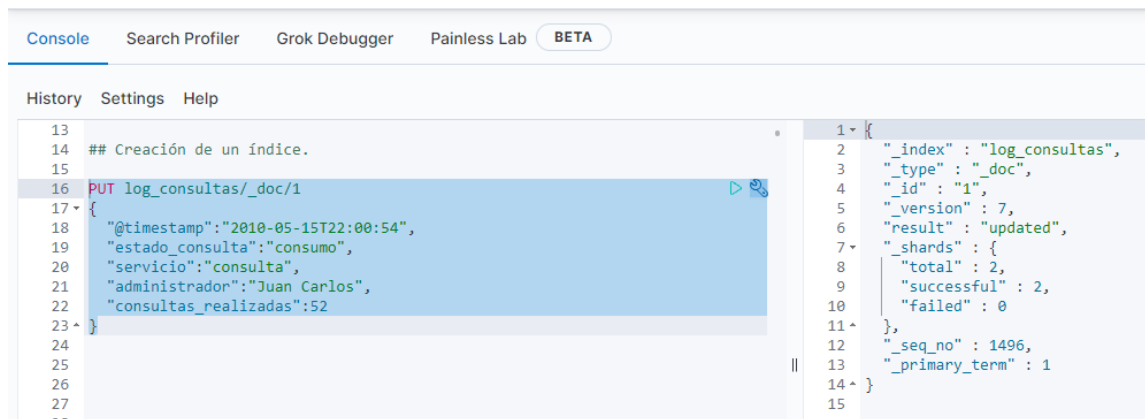
5._ Se escribió el siguiente código para la creación del índice.

```
PUT log_consultas/_doc/1
{
  "@timestamp": "2010-05-15T22:00:54",
  "estado_consulta": "consumo",
  "servicio": "consulta",
  "administrador": "Juan Carlos",
  "consultas_realizadas": 52
}
```

Donde

- PUT es el método HTTP, que se utiliza para crear o actualizar.
- log_consultas, es el nombre que recibe el índice.
- _doc, se refiere al tipo de documento.
- El número 1, corresponde al id del documento.
- {...}, corresponde al cuerpo del documento.

Generando el resultado siguiente.



The screenshot shows a console window with a top navigation bar containing 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and a 'BETA' button. Below the navigation bar are tabs for 'History', 'Settings', and 'Help'. The main area displays a REST client interface with a PUT request on the left and its JSON response on the right. The request is to create a document in the 'log_consultas' index. The response shows the document was successfully updated.

```
13
14 ## Creación de un índice.
15
16 PUT log_consultas/_doc/1
17 {
18   "@timestamp": "2010-05-15T22:00:54",
19   "estado_consulta": "consumo",
20   "servicio": "consulta",
21   "administrador": "Juan Carlos",
22   "consultas_realizadas": 52
23 }
24
25
26
27
28
29
30
```

```
1 {
2   "_index": "log_consultas",
3   "_type": "_doc",
4   "_id": "1",
5   "_version": 7,
6   "result": "updated",
7   "_shards": {
8     "total": 2,
9     "successful": 2,
10    "failed": 0
11  },
12   "_seq_no": 1496,
13   "_primary_term": 1
14 }
```

Obtén el mapping del índice log_consultas.

1._ Se revisó la documentación correspondiente.

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/explicit-mapping.html>

2._ Para consultar el mapping correspondiente al índice log_consultas, se escribió el código: GET log_consultas/_mapping



The screenshot shows a console window with a REST client interface. The left pane shows a GET request for the mapping of the 'log_consultas' index. The right pane shows the resulting JSON mapping, which defines the data types for various fields in the index.

```
1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
7
8 ## Revisamos la información del Cluster.
9 GET /
10
11 ## Revisamos el catalogo de índices.
12 GET _cat/indices
13
14 ## Creación de un índice.
15
16 PUT log_consultas/_doc/1
17 {
18   "@timestamp": "2010-05-15T22:00:54",
19   "estado_consulta": "consumo",
20   "servicio": "consulta",
21   "administrador": "Juan Carlos",
22   "consultas_realizadas": 52
23 }
24
25 ## Obtenemos el mapping del índice creado.
26
27 GET /log_consultas/_mapping
28
29
30
```

```
1 {
2   "log_consultas": {
3     "mappings": {
4       "properties": {
5         "@timestamp": {
6           "type": "date"
7         },
8         "administrador": {
9           "type": "text",
10          "fields": {
11            "keyword": {
12              "type": "keyword",
13              "ignore_above": 256
14            }
15          }
16        },
17        "consultas_realizadas": {
18          "type": "long"
19        },
20        "estado_consulta": {
21          "type": "text",
22          "fields": {
23            "keyword": {
24              "type": "keyword",
25              "ignore_above": 256
26            }
27          }
28        },
29        "servicio": {
30          "type": "text",
31          "fields": {
32            "keyword": {
33              "type": "keyword",
34              "ignore_above": 256
35            }
36          }
37        }
38      }
39    }
40  }
```

Donde:

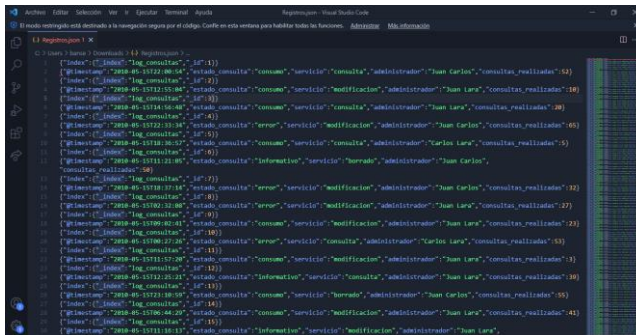
- GET es el método HTTP, que se utiliza para obtener información solicitada.
- log_consultas, es el nombre que recibe el índice.
- _mapping, la cual es una Api.

Genera un template a partir de este índice creado.

1._ Se investigó en la documentación siguiente:

- <https://www.youtube.com/watch?v=eeZCHq0uGss>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/index-templates.html#index-templates>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates-v1.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/number.html>

2._ Se analizó el JSON registros.



3._ Se necesitó de a Api Template para escribir el siguiente código

```
PUT _template/template_1
{
  "index_patterns": ["log_consultas*"],
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "@timestamp": {
        "type": "date"
      },
      "estado_consulta": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256
          }
        }
      },
      "servicio": {
```


Donde:

- PUT es el método HTTP, que se utiliza para crear o actualizar.
- _template, corresponde a la Api solicitada.
- template_1, es el nombre que recibe el template.
- index_patterns, se le asigna el valor "log_consultas*" para trabajar con los datos de Registros.json.
- settings, Define la configuración que se aplicara.
- mapping, se definen las propiedades, tipos y analizadores.

Para un resultado exitoso.

```
1 1  #! Legacy index templates are deprecated in favor of composable templates.
2 2  {
3 3    "acknowledged" : true
4 4  }
5 5  |
```

Carga de datos a índice con la Api Bulk.

1._ Se analizó la siguiente documentación:

- <https://www.youtube.com/watch?v=3s4UIA8dieo>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html>

2._ Para la carga de los datos es necesario abrir el archivo Registros.json, para copiar la información y pegarla la información después de POST _bulk

The screenshot shows the Elastic Dev Tools interface. The left pane displays a JSON array of documents being indexed via the POST _bulk API. The documents include fields like @timestamp, _index, _id, estado_consulta, servicio, and consultas_realizadas. The right pane shows the response from the API, indicating that 200 documents were successfully indexed. The response includes details like 'took' (50ms), 'errors' (false), and 'items' (an array of document summaries).

```
75 ## Carga de datos con la Api Bulk.
76
77 POST _bulk
78 {
79   "index": {"_index": "log_consultas", "_id": 1}
80   {
81     "@timestamp": "2010-05-15T22:00:54", "estado_consulta": "consumo", "servicio": "consulta", "administrador": "Juan Carlos", "consultas_realizadas": 52
82   }
83   "index": {"_index": "log_consultas", "_id": 2}
84   {
85     "@timestamp": "2010-05-15T12:55:04", "estado_consulta": "consumo", "servicio": "modificacion", "administrador": "Juan Lara", "consultas_realizadas": 10
86   }
87   "index": {"_index": "log_consultas", "_id": 3}
88   {
89     "@timestamp": "2010-05-15T14:56:48", "estado_consulta": "consumo", "servicio": "consulta", "administrador": "Juan Lara", "consultas_realizadas": 20
90   }
91   "index": {"_index": "log_consultas", "_id": 4}
92   {
93     "@timestamp": "2010-05-15T22:33:34", "estado_consulta": "error", "servicio": "modificacion", "administrador": "Juan Carlos", "consultas_realizadas": 65
94   }
95   "index": {"_index": "log_consultas", "_id": 5}
96   {
97     "@timestamp": "2010-05-15T18:36:57", "estado_consulta": "consumo", "servicio": "consulta", "administrador": "Carlos Lara", "consultas_realizadas": 5
98   }
99   "index": {"_index": "log_consultas", "_id": 6}
100  {
101    "@timestamp": "2010-05-15T11:21:05", "estado_consulta": "Informativo", "servicio": "borrado", "administrador": "Juan Carlos", "consultas_realizadas": 50
102  }
103  "index": {"_index": "log_consultas", "_id": 7}
104  {
105    "@timestamp": "2010-05-15T18:37:14", "estado_consulta": "error", "servicio": "modificacion", "administrador": "Juan Carlos", "consultas_realizadas": 32
106  }
107 }
```

```
1 {
2   "took" : 50,
3   "errors" : false,
4   "items" : [
5     {
6       "index" : {
7         "_index" : "log_consultas",
8         "_type" : "_doc",
9         "_id" : "1",
10        "version" : 2,
11        "result" : "updated",
12        "_shards" : {
13          "total" : 2,
14          "successful" : 2,
15          "failed" : 0
16        },
17        "_seq_no" : 1,
18        "_primary_term" : 1,
19        "status" : 200
20      }
21    },
22    {
23      "index" : {
24        "_index" : "log_consultas",
25        "_type" : "_doc",
```

Realizar búsquedas sobre el índice.

1._ Se revisó la documentación:

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-count.html>

2._ Consulta: Obtener el número de registros con **estado_consulta** igual a error y consumo.

2.1_ Se ejecutaron los siguientes códigos.

```
## Obtener el número de registros con estado_consulta igual a error y consumo.

GET log_consultas/_search
{
  "query":{
    "match": {
      "estado_consulta": " error consumo"
    }
  }
}
```

```
GET log_consultas/_count
{
  "query":{
    "match": {
      "estado_consulta": " error consumo"
    }
  }
}
```

Donde:

- GET es el método HTTP, que se utiliza para obtener información.
- log_consultas, corresponde al nombre del índice.
- _search, es el Api que se consume en la consulta de búsqueda.
- _count, es el Api que obtiene el número de coincidencias para una consulta de búsqueda.
- {...}, corresponde al cuerpo de la consulta.

3._ Consulta: Obtener el número de registros realizados por el **administrador** Juan Lara. Para esta consulta se ejecutaron los siguientes códigos.

```
## Obtener el número de registros realizados por el administrador Juan Lara

GET log_consultas/_search
{
  "query":{
    "match": {
      "administrador":{
        "query" : "Juan Lara",
        "operator":"and"
      }
    }
  }
}

GET log_consultas/_count
{
  "query":{
    "match": {
      "administrador":{
        "query" : "Juan Lara",
        "operator":"and"
      }
    }
  }
}
```

```
1 {
2   "took" : 12,
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" : 98,
13      "relation" : "eq"
14    },
15    "max_score" : 0.8232368,
16    "hits" : [
17      {
18        "_index" : "log_consultas",
19        "_type" : "_doc",
20        "_id" : "2",
21        "_score" : 0.8232368,
22        "_source" : {
23          "@timestamp" : "2010-05-15T12:55:04",
24          "estado_consulta" : "consumo",
25          "servicio" : "modificacion",
```

```
## Obtener el número de registros realizados por el administrador Juan Lara

GET log_consultas/_search
{
  "query":{
    "match": {
      "administrador":{
        "query" : "Juan Lara",
        "operator":"and"
      }
    }
  }
}

GET log_consultas/_count
{
  "query":{
    "match": {
      "administrador":{
        "query" : "Juan Lara",
        "operator":"and"
      }
    }
  }
}
```

```
1 {
2   "count" : 98,
3   "_shards" : {
4     "total" : 1,
5     "successful" : 1,
6     "skipped" : 0,
7     "failed" : 0
8   }
9 }
10
```

4._ Consulta: Obtener el número de registros con estado_consulta igual a informativo y servicio igual ha borrado.

```
3
4 ##Obtener el número de registros con estado_consulta igual a informativo y
  servicio igual a borrado.
5
6 GET log_consultas/_search
7 {
8   "query":{
9     "bool": {
10      "must": [
11        {
12          "match": {
13            "estado_consulta": "informativo"
14          }
15        },
16        {
17          "match": {
18            "servicio": "borrado"
19          }
20        }
21      ]
22    }
23  }
24 }
25
```

```
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" : 52,
13      "relation" : "eq"
14    },
15    "max_score" : 0.9373441,
16    "hits" : [
17      {
18        "_index" : "log_consultas",
19        "_type" : "_doc",
20        "_id" : "6",
21        "_score" : 0.9373441,
22        "_source" : {
23          "@timestamp" : "2010-05-15T11:21:05",
24          "estado_consulta" : "informativo",
25          "servicio" : "borrado",
```

```
1 GET log_consultas/_count
2 {
3   "query":{
4     "bool": {
5       "must": [
6         {
7           "match": {
8             "estado_consulta": "informativo"
9           }
10        },
11        {
12          "match": {
13            "servicio": "borrado"
14          }
15        }
16      ]
17    }
18  }
19 }
20
```

```
1 {
2   "count" : 52,
3   "_shards" : {
4     "total" : 1,
5     "successful" : 1,
6     "skipped" : 0,
7     "failed" : 0
8   }
9 }
10
```

5._ Consulta extra: Obtener la suma de los valores en consultas_realizadas con estado_consulta igual a error.

5.1._ Se consultó la documentación de aggregation para poder resolver el query.

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-metrics-sum-aggregation.html>

Para esta consulta se ejecutaron los siguientes códigos.

```
##Extra
##Obtener la suma de los valores en consultas_realizadas con estado_consulta
igual a error

POST log_consultas/_search?size=0
{
  "query":{
    "match": {
      "estado_consulta":"error"
    }
  },
  "aggs": {
    "Total consultas": { "sum": { "field": "consultas_realizadas" } }
  }
}
```

```
1 {
2   "took" : 1,
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" : null,
16    "hits" : [ ]
17  },
18  "aggregations" : {
19    "Total consultas" : {
20      "value" : 2865.0
21    }
22  }
23 }
```

Tablero para visualizar información de empleados

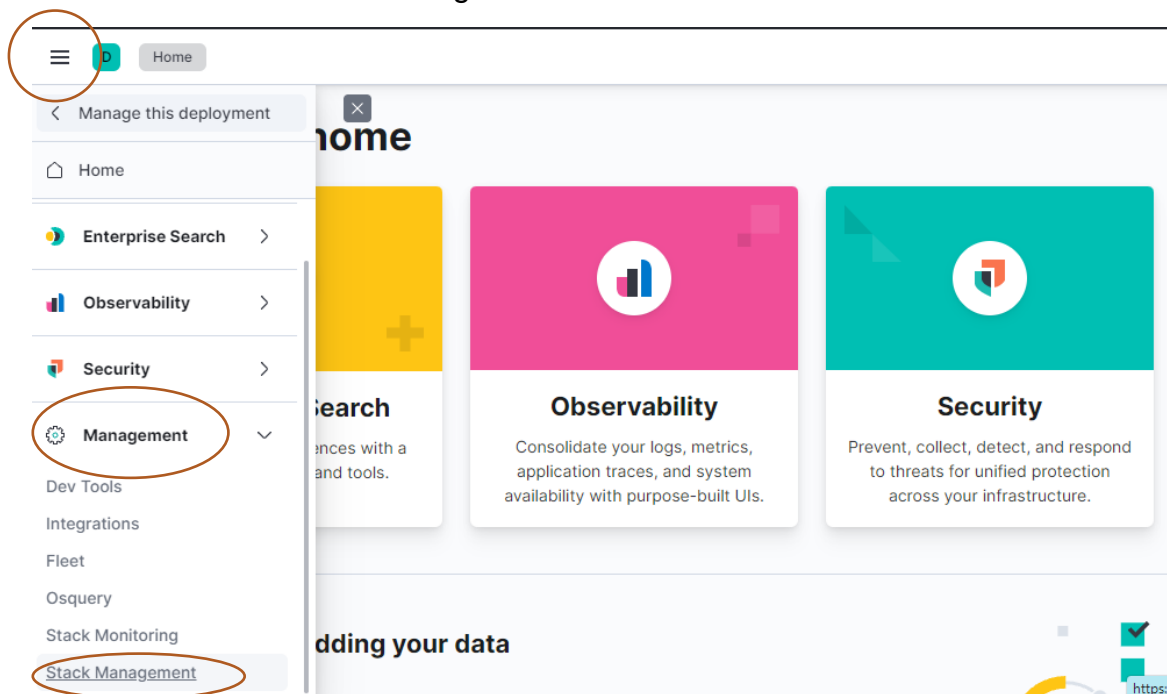
1._ Se realizó la creación de un Index-Patterns

1.1._ Regresar al apartado home.

1.2._ Abrir el menú.

1.3._ Buscar el apartado de Management.

1.4._ Abrir Stack Management.



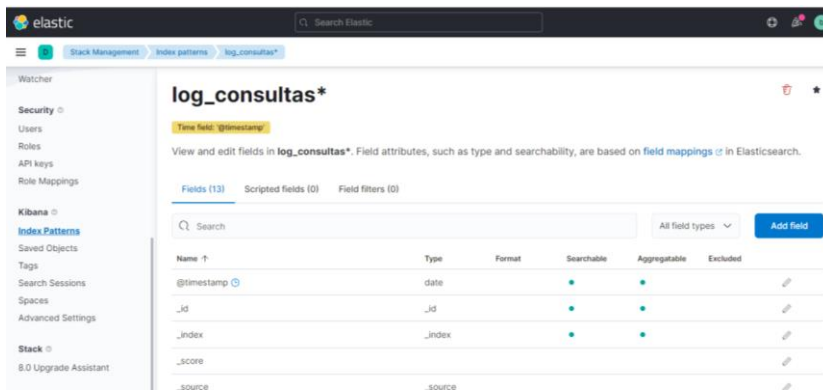
1.5._ Una vez dentro stack managment, buscando el apartado de Kibana, posteriormente ingresara a indemex patterns.

The screenshot shows the Elastic Stack Management interface. On the left, the 'Kibana' section is expanded, and 'Index Patterns' is highlighted. The main content area features a 'Welcome to Stack Management 7.15.0' message and a 'You have data in Elasticsearch. Now, create an index pattern.' prompt. A blue button labeled 'Create index pattern' is visible. Below this, there is a link to 'Read documentation'.

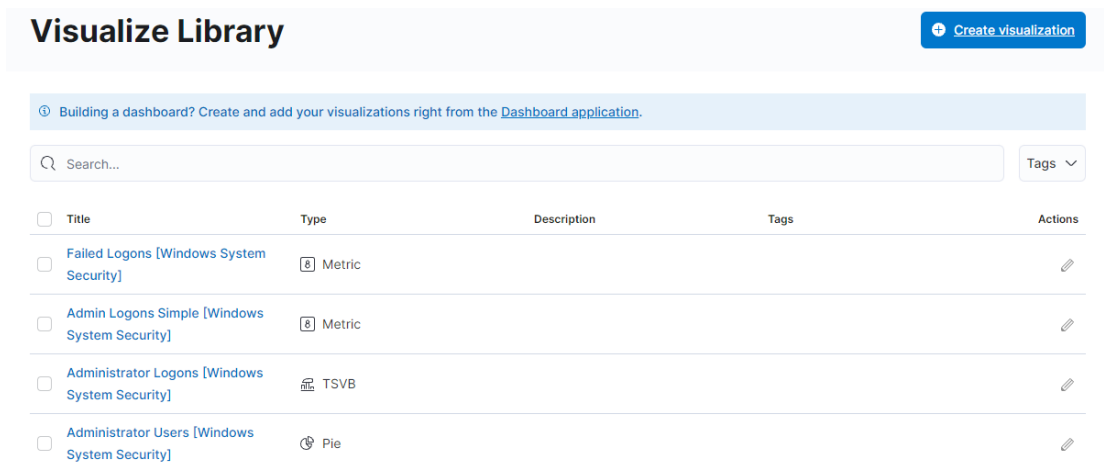
1.6._ Se ingresó el nombre y el Time Filter con @timestamp.

The 'Create index pattern' dialog box is shown. The 'Name' field contains 'log_consultas*'. Below it, a note states: 'Use an asterisk (*) to match multiple characters. Spaces and the characters , / , ? , * , < , > , | are not allowed.' The 'Timestamp field' dropdown is set to '@timestamp'. Below this, a note states: 'Select a timestamp field for use with the global time filter.' A link to 'Show advanced settings' is present. On the right, a confirmation message states: 'Your index pattern matches 1 source.' Below this, the 'log_consultas' index is listed. The 'Rows per page' is set to 50. At the bottom, there are 'Close' and 'Create index pattern' buttons.

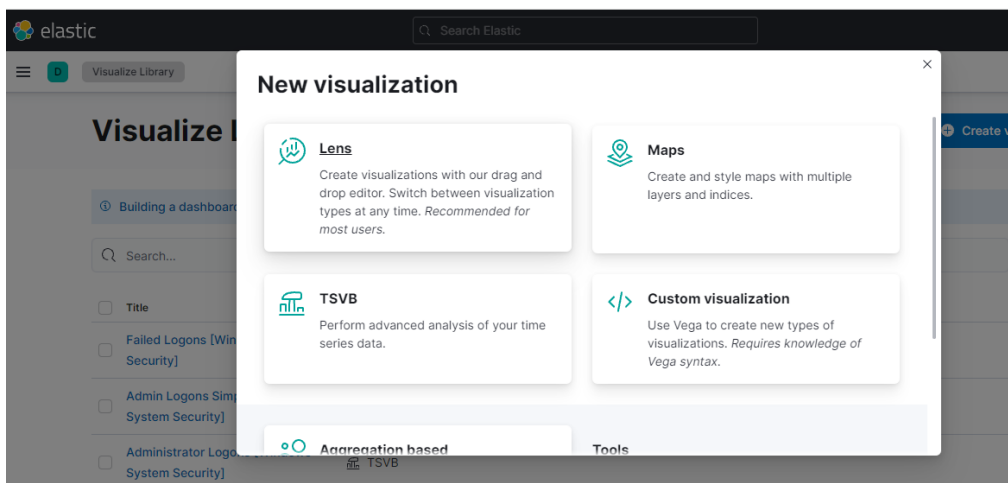
1.7._ Se obtiene como resultado.



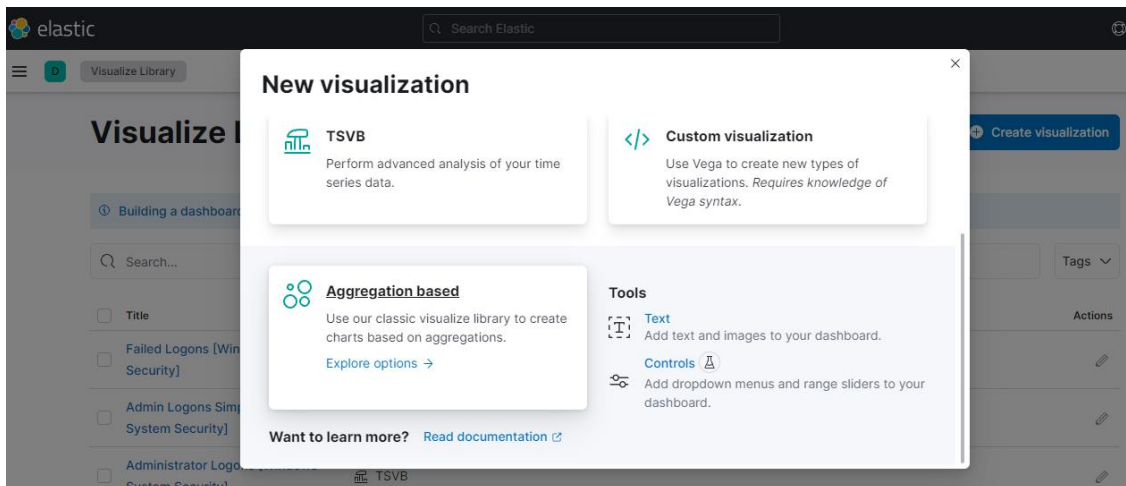
2._ Antes de este punto debemos generar un Visualize con la librería de Kibana.



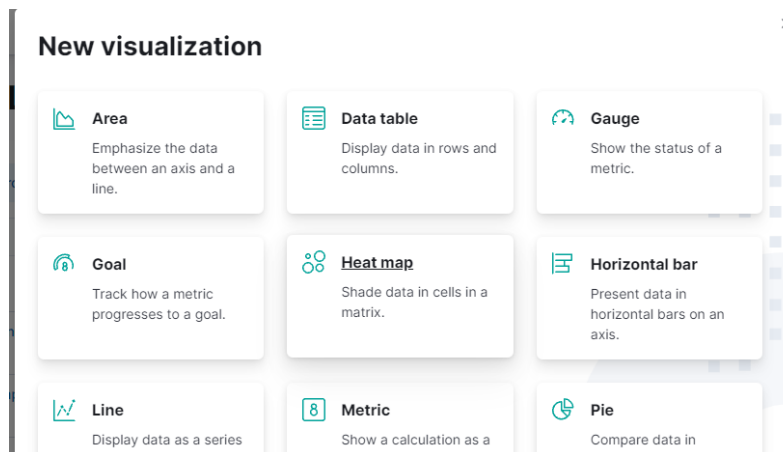
2.1_ Daremos click en el botón de create visualization.



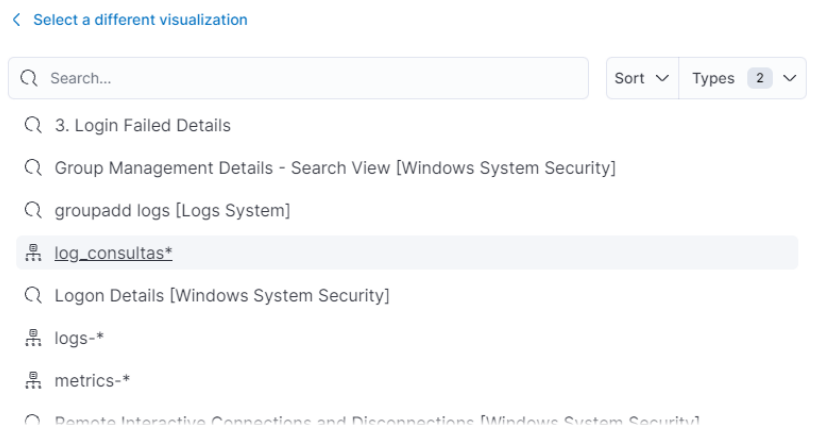
2.2._ Se bajara el scroll, para poder encontrar la opción de "Aggregation_bassed".



2.3._ Una vez dentro del menú elegiremos las gráficas Heat map.



2.4._ Seleccionares al índice para empezar a obtener las visualizaciones.

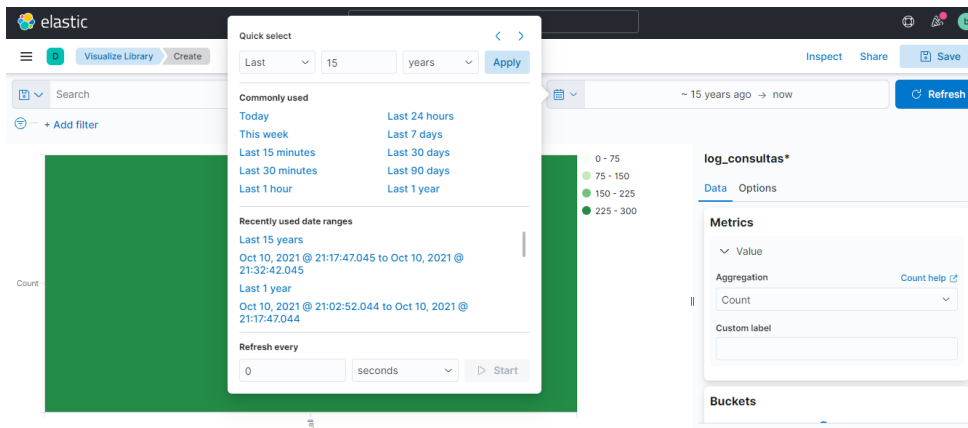


3._ Vista de heat map, donde mostraras el número de servicios realizados por administrador.

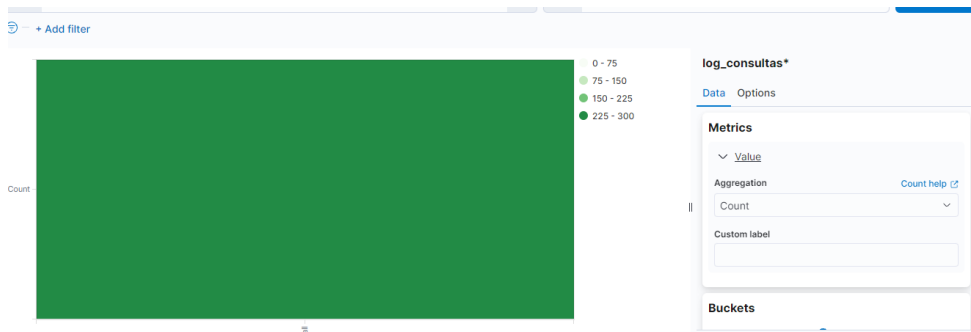
3.1._ Se analizó la documentación:

- <https://www.youtube.com/watch?v=MsqaYyzKi8I&t=278s>
- <https://www.elastic.co/guide/en/kibana/current/lens.html>
- <https://www.youtube.com/watch?v=ItBnnvP1UNI>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-date-format.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html>

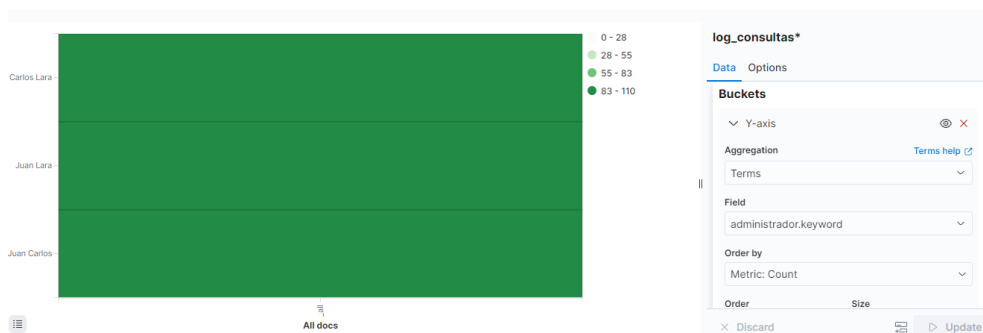
3.2._ Se modifica la fecha para obtener todos los datos que se subieron.



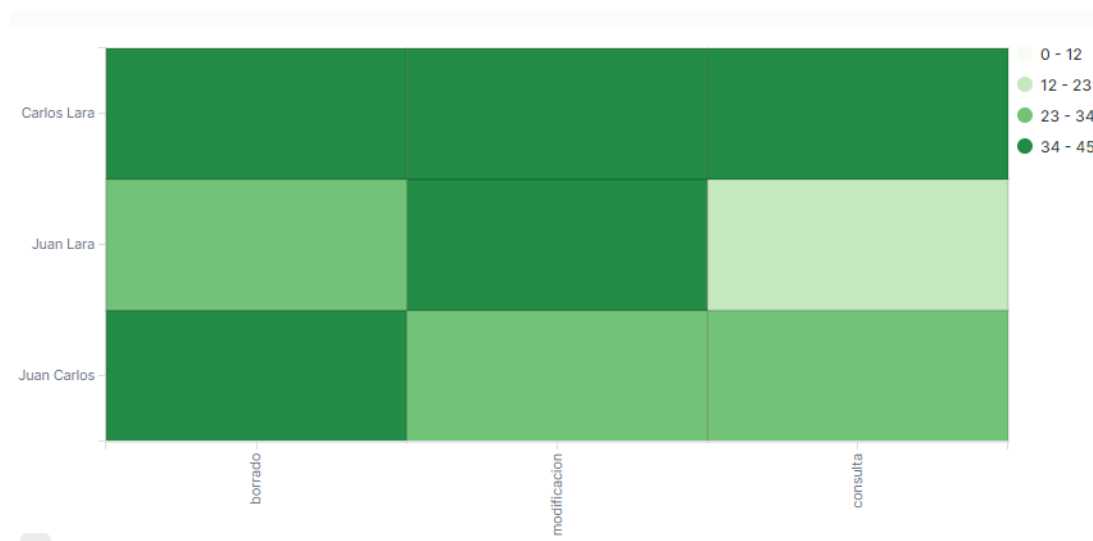
3.3._ Para el campo de las métricas se declara un Count.



3.4._ Se agregar los Buckets



Como resultado final queda de esta forma la visualización.



4._ Vista de Barras, donde se grafique el número de registros con estado_consulta igual a error a través del tiempo.

4.1._ Se analizó la documentación:

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-date-format.html>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html>

4.2._ Para el campo de las métricas se declara un Count.

Metrics

Y-axis

Aggregation: Sum Bucket [Sum Bucket help](#)

Bucket

Aggregation: Filters [Filters help](#)

Filter 1: estado_consulta.keyword : "error" [KQL](#)

[+ Add filter](#)

[Advanced](#)

4.3 Se agregar los Buckets

Buckets

✓ X-axis ⓘ ✕

Aggregation [Date Histogram help](#)

Date Histogram

Field

@timestamp

Minimum interval

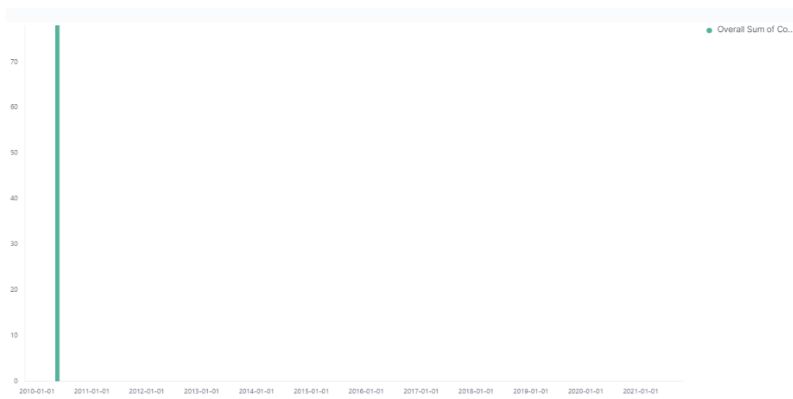
Minute ⓘ

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

☐ Drop partial buckets

Custom label

Como resultado final queda de esta forma la visualización.



Nota: Se colocó en el buckets que se dividiera por minutos el @timestamp, sin embargo se divide por días obteniendo una solo barra, ya que todos los datos corresponden a una fecha.

¡EXTRA!

Si te da tiempo genera un tablero con las 2 visualizaciones que acabas de crear. Para ello dirígete a la opción del menú Dashboard.

