Sebastián Martínez Santos

Práctica Elasticsearch

Creación de un índice – Tiempo 2 horas

1. Selecciona tu consola de Dev Tools para realizar la siguiente actividad.
2. Crea un índice con el nombre de log_consultas a partir del siguiente JSON:

{
"@timestamp":"2010-05-15T22:00:54",
"estado_consulta":"consumo", "servicio":"consulta", "administrador":"Juan Carlos",
"consultas_realizadas":52
}

Una forma de crear un índice basado en los campos del JSON anterior es usando la API PUT con el cual podemos especificar la estructura y el tipo de variables (*mapping*) del índice.

```
PUT /log_consultas
{
  "mappings": {
    "properties": {
      "@timestamp" : {"type": "date"},
      "estado_consulta": { "type": "keyword" },
      "servicio" : {"type": "keyword"},
      "administrador":  {"type": "keyword"},
      "consultas_realizadas": {"type": "integer"}
    }
  }
}
```

La primera dificultad a la que me enfrenté fue entender la diferencia entre las variables *keyword* y las *string*. Las variables *keyword* se usan cuando estamos esperando palabras son bsuqeudas exactas, mientras que los *strings* sirven cuando queremos buscar partes determinadas del *string*. En este caso las variables *keyword* son la que necesitamos.

3. Obtén el mapping del índice anterior y genera un template a partir de este índice haciendo uso del API de plantillas (TEMPLATE). El patrón para el índice debe ser: "log_consultas*". Verífica los tipos de datos hagan sentido con la información almacenada.

Para obtener el mapping podemos correr el siguiente código.

```
GET /log_consultas/_mapping
```

Lo cual nos devuelve el mapeo que determinamos anteriormente:

```
{
  "log_consultas": {
    "mappings": {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "administrador": {
          "type": "keyword"
        },
        "consultas_realizadas": {
          "type": "integer"
        },
        "estado_consulta": {
          "type": "keyword"
        },
        "servicio": {
          "type": "keyword"
        }
      }
    }
  }
}
```

Ahora, usemos este output para crear un *template* para el índice.

```
PUT _index_template/template_1
{
  "index_patterns": ["log_consultas*"],
  "template": {
    "mappings":  {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "administrador": {
          "type": "keyword"
        },
        "consultas_realizadas": {
          "type": "integer"
        },
        "estado_consulta": {
          "type": "keyword"
        },
        "servicio": {
          "type": "keyword"
        }
      }
    }

  }
}
```

4. Una vez definido tú template cargarás una serie de documentos en tu índice utilizando el archivo que se encuentra en el escritorio: log_consultas.json. Para esto utiliza la API (BULK).

Haciendo uso de la API BULK podemos cargar múltiples documentos al índice en un solo comando. Se ve de la siguiente manera:

```
POST /log_consultas/_bulk
{"index":{"_index":"log_consultas","_id":1}}
{"@timestamp":"2010-05-15T22:00:54","estado_consulta":"consumo","servicio":"consulta"
  ,"administrador":"Juan Carlos","consultas_realizadas":52}
{"index":{"_index":"log_consultas","_id":2}}
{"@timestamp":"2010-05-15T12:55:04","estado_consulta":"consumo","servicio"
  :"modificacion","administrador":"Juan Lara","consultas_realizadas":10}
{"index":{"_index":"log_consultas","_id":3}}
{"@timestamp":"2010-05-15T14:56:48","estado_consulta":"consumo","servicio":"consulta"
  ,"administrador":"Juan Lara","consultas_realizadas":20}
{"index":{"_index":"log_consultas","_id":4}}
{"@timestamp":"2010-05-15T22:33:34","estado_consulta":"error","servicio":"modificacion"
  ,"administrador":"Juan Carlos","consultas_realizadas":65}
```

Finalmente, verifiquemos si ahora el índice ya contiene documentos. Esto se hace con el siguiente código

```
GET /_cat/indices/log_consultas
```

Lo que nos arroja

```
green open log_consultas
  4s9VobC5RNeI85RaetaSrw 1 1 299
  0 40.1kb 20.1kb
```

Lo que significa que efectivamente hay datos en el índice.

Realizar búsquedas sobre el índice -- Tiempo 2 horas

Ya habiendo docuementos en el índice “log_consultas” podemos realizar busquedas y consultas de la información almacenada.

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

Cuando queremos obtener la cantidad de archivos que cumplen alguna característica, se usa el commando “_count”.

Por otro lado, es neceario generar un *query* como el siguiente:

```
GET /log_consultas/_count
{
  "query": {
    "bool":{
      "should":[
      {
       "match":
        {
        "estado_consulta": "consumo"
        }

      },
        {
         "match": {
        "estado_consulta": "error"
        }
      }
      ]
    }
  }
}
```

El cual contiene un parámetor “bool” el cual de ser verdadero arrojará la cuenta de los documentos que cumplen con la condición lógica.

En este caso, nos interesan los documentos que tienen como estado de la consulta “consumo” o “error”. Por ende, es necesario usar el parámetro “should”.

El resultado nos indica que existen 182 documentos que tienen como estado de la consulta “consumo” o “error”.

2. Obtener el número de registros realizados por el administrador Juan Lara.

Debido a que solo hay una condición lógica, no es necesario usar el parámetro “bool” ni el “should”. Simplemente, usaremos el comando “match_phrase” para buscar el nombre “Juan Lara” dentro de la variable administrador. Es decir, nos queda el siguiente *query.*

```
GET /log_consultas/_count
{
  "query": {

      "match_phrase":
        {
        "administrador": "Juan Lara"
        }
  }
}
```

Cuyo output nos indica que existen 98 documentos generados por el administrador “Juan Lara”.

3. Obtener el número de registros con estado_consulta igual a informativo y servicio igual a borrado Consulta extra

Siguiendo la lógica del inciso 1 podemos generar el query necesario. La dierencia es que ahora buscamos la cantidad de archivos con el estado de la consulta igual a informativo y con el servicio igual a borrado. Cuando buscamos una intersección usamos el parámetro “must”.

```
GET /log_consultas/_count
{
  "query": {
    "bool":{
      "must":[
      {
       "match":
        {
        "estado_consulta": "informativo"
        }

      },
        {
          "match": {
         "servicio": "borrado"
        }
      }
      ]
    }
  }
}
```

Como resultado obtenemos que la cantidad de archivos con el estado de la consulta igual a informativo y con el servicio igual a borrado es 52 documentos.

4. Obtener la suma de los valores en consultas_realizadas con estado_consulta igual a error

A diferencia de las búsquedas anteriores, ahora queremos realizar una operación (suma) con los datos de los documentos que cumplen con cierta caracteristica.

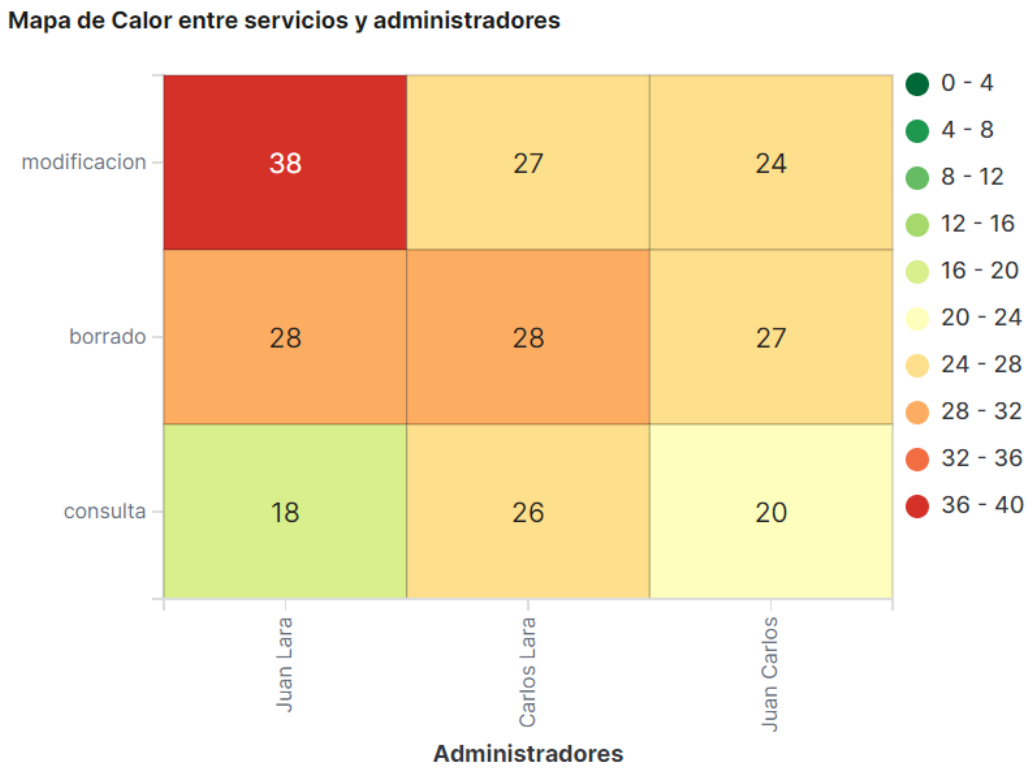
Para lograr esto se usan agregaciones (“aggs”) las cuales nos ayudan a realizar varias funciones como la suma. La estructura necesaria del query es la siguiente.

```
POST /log_consultas/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "match": { "estado_consulta": "error" }
      }
    }
  },
  "aggs": {
    "Suma de consultas realizadas con estado = error": { "sum": { "field": "consultas_realizadas" } }
  }
}
```

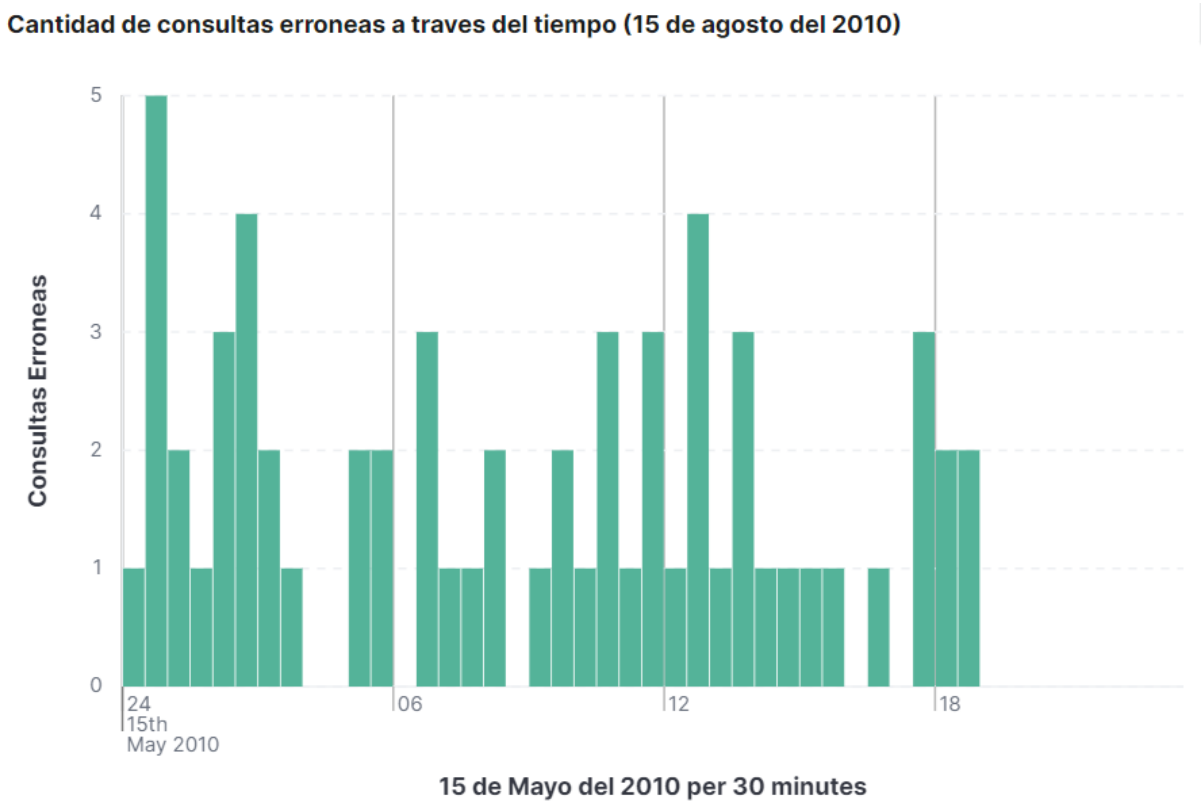
Obtenemos que suman 2865 consultas las que contienen un estado consulta igual a error.

Realizar un tablero para visualizar información de empleados -
- Tiempo 2 horas

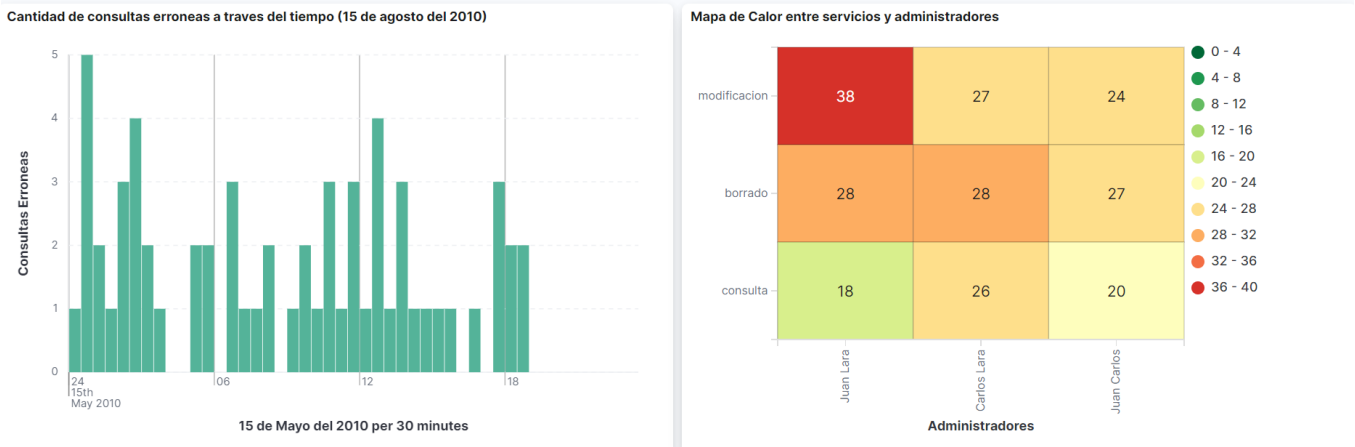
1. Vista de heat map, donde mostraras el número de servicios realizados por administrador.



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



Tablero con las dos visualizaciones:



Fuentes

- *Getting Started with Dev Tools in Elasticsearch*. (2019, 4 febrero). Maelfabien. Recuperado 20 de agosto de 2022, de https://maelfabien.github.io/bigdata/DevTools/#5-aggregations

- *cat indices API | search Guide [7.17]*. (2022). Elastic. https://www.elastic.co/guide/en/elasticsearch/reference/7.17/cat-indices.html

- *¿Qué es Elasticsearch?* (2022). Elastic. https://www.elastic.co/es/what-is/elasticsearch

- *Aggregations | search Guide [8.3]*. (2022). Elastic. https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations.html

- Shafir, O. (2022, 10 julio). *Elasticsearch Index – How to create Elasticsearch Index and what it*. Opster. https://opster.com/guides/elasticsearch/glossary/elasticsearch-index/

- *When to use the keyword type vs text datatype in Elasticsearch | ObjectRocket*. (2022). ObjectRocket. https://kb.objectrocket.com/elasticsearch/when-to-use-the-keyword-type-vs-text-datatype-in-elasticsearch

- *Kibana - Working With Heat Map*. (2022). Kibana. https://www.tutorialspoint.com/kibana/kibana_working_with_heat_map.htm