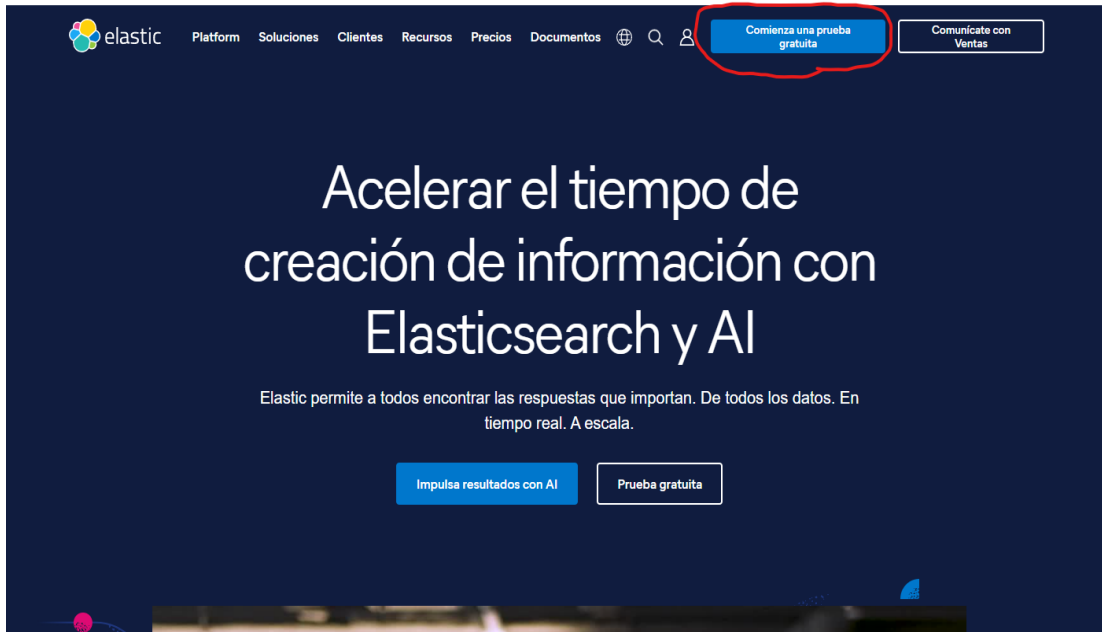
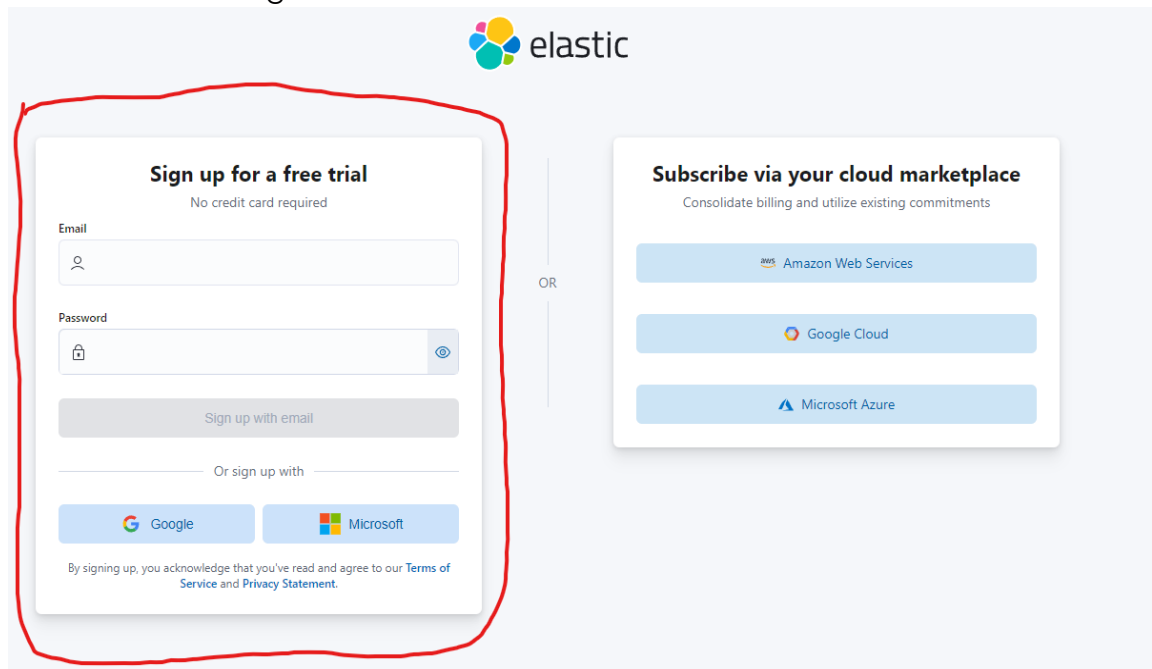


## Documentación técnica de practica elastic

1. Como primer paso es necesario entrar al sitio oficial y crear una cuenta de prueba gratuita <https://www.elastic.co/es/>



2. Posterior a eso aparecerá el panel de registro con dos opciones una creando una cuenta tradicional o bajando la app directamente a tu cloud en este caso se registrara de manera tradicional.



3. Para el tercer paso tendrás que crear tu cuenta y crear el servidor con la nube de tu preferencia en este caso se utilizara aws y se pondrá los parámetros proveniente descrita en la práctica.


## Create your first deployment

A deployment includes Elasticsearch, Kibana, and other Elastic Stack features, allowing you to store, search, and analyze your data.

---

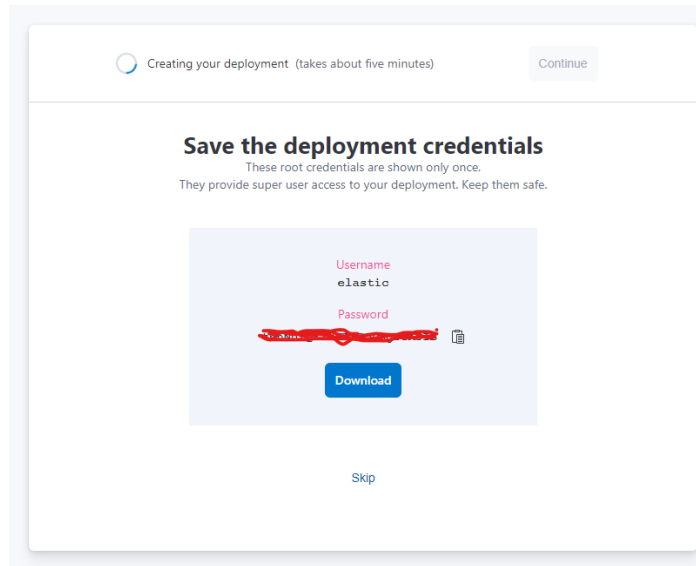
### Name

### Settings [Hide](#)

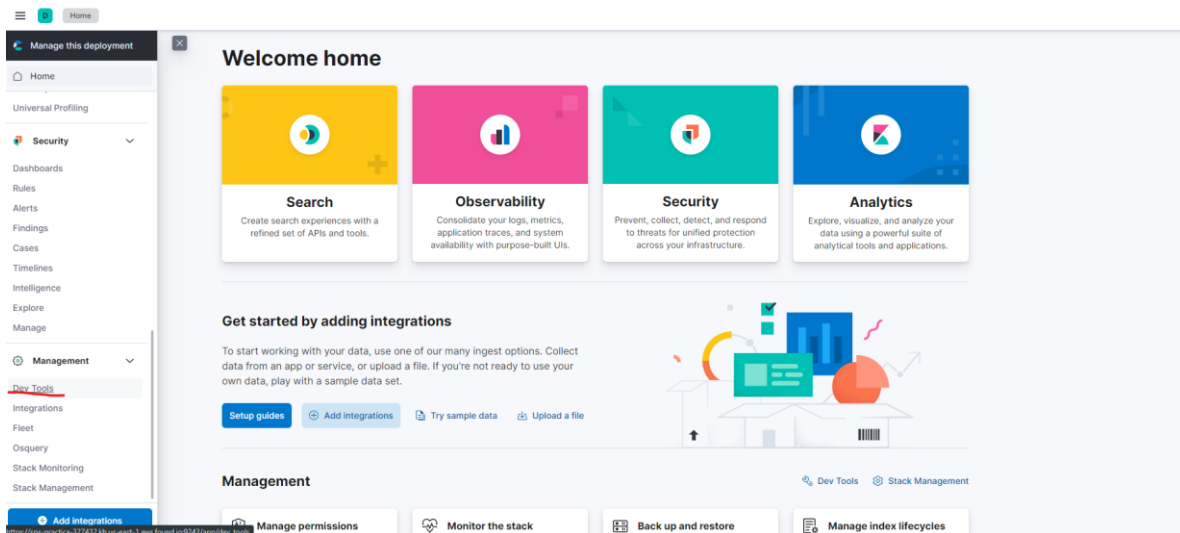
Cloud provider	 Amazon Web Services	▼
Region	us N. Virginia (us-east-1)	▼
Hardware profile ⓘ	Storage optimized	▼
Version ⓘ	8.12.0 (latest)	▼

Create deployment

- Posteriormente se creará esto puede tardar un tiempo al mismo tiempo se te dará los accesos para entrar.



- Una vez tengamos los accesos para la plataforma , en la barra de lado izquierdo buscaremos las herramientas **dev tolos**



6. En estos siguientes paso comenzara la actividad como tal dentro de dev tools para eso primero se tiene que crear el índice , para esto se utilizo la

```
1 # Click the Variables button, above, to create your own variables.  
2 PUT /log_consultas/  
3
```

instrucción **put** y el nombre del índice en este caso log consultas para mas información consultar el url

<https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html>

7. Posterior a eso al índice hice la prueba con la consulta \_doc para agregar un nuevo documento al índice en este caso el de la practica. Y poder extraer los datos en el mape.

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

Para esto se utilizó instrucción **POST \_doc**:

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

8. En este paso se crea el mapping es importante entender que el mapping es la estructura que tiene el documento a esto hago referencia a los tipos de datos de cada contenido del documento. En este caso se utilizó la instrucción **\_mapping**

```
3 GET log_consultas/_mapping
4
```

Esta instrucción nos devolverá la estructura de cada campo del documento

```
{
  "log_consultas": {
    "mappings": {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "administrador": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "consultas_realizadas": {
          "type": "long"
        },
        "estado_consulta": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        }
      }
    }
  },
  "servicio": {
    "type": "text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    }
  }
}
```

A esta información se refiere como el tipo de dato, el nombre de la llave del archivo json y bueno el tipo keyword sirve para detectar email, nombres, códigos postales es como un identificador automático, si se desea consultar más información sobre el keyword se puede consultar

[:https://www.elastic.co/guide/en/elasticsearch/reference/current/keyword.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/keyword.html)

- Una vez que tenemos el mapeo se comienza a crear el template o la plantilla esto para que sirve para que podamos vincular una gran cantidad de json y ya tenga una guía para que de manera automática cada valor se asigne de manera correcta al documento , en esta instrucción utilice la instrucción **\_index\_template**

```
PUT _index_template/log_consultas_template
```

```
{
  "index_patterns": ["log_consultas_*"],
  "priority": 1,
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 0
    },
    "mappings": {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "administrador": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "consultas_realizadas": {
          "type": "long"
        },
        "estado_consulta": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "servicio": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        }
      }
    }
  }
}
```

Para crear lo fue necesario tener el mapeo anterior ya que se tiene que poner dentro de la instrucción dentro de **mappings**

Dentro de **settings** : Aquí se configuran las configuraciones del índice, como el número de fragmentos (number\_of\_shards) y el número de réplicas (number\_of\_replicas). En este caso, se ha configurado un solo fragmento y ninguna réplica ("number\_of\_shards": 1, "number\_of\_replicas": 0).

10. Ahora una vez creada la plantilla se anexa toda la data en este caso se opto por usar la instrucción **\_bulk** las cual nos permite agregar información con el template previamente creado literal se copio y pego el archivo json pero esto se puede automatizar con python

```
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}
{"index":{"_index":"log_consultas","_id":5}}
{"@timestamp":"2010-05-15T18:36:57","estado_consulta":"consumo","servicio":"consulta","administrador":"Carlos Lara","consultas_realizadas":5}
{"index":{"_index":"log_consultas","_id":6}}
{"@timestamp":"2010-05-15T11:21:05","estado_consulta":"informativo","servicio":"borrado","administrador":"Juan Carlos","consultas_realizadas":50}
{"index":{"_index":"log_consultas","_id":7}}
{"@timestamp":"2010-05-15T18:37:14","estado_consulta":"error","servicio":"modificacion","administrador":"Juan Carlos","consultas_realizadas":32}
{"index":{"_index":"log_consultas","_id":8}}
{"@timestamp":"2010-05-15T02:32:08","estado_consulta":"error","servicio":"modificacion","administrador":"Juan Lara","consultas_realizadas":27}
{"index":{"_index":"log_consultas","_id":9}}
{"@timestamp":"2010-05-15T09:02:41","estado_consulta":"consumo","servicio":"modificacion","administrador":"Juan Lara","consultas_realizadas":23}
{"index":{"_index":"log_consultas","_id":10}}
{"@timestamp":"2010-05-15T00:27:26","estado_consulta":"error","servicio":"consulta","administrador":"Carlos Lara","consultas_realizadas":53}
{"index":{"_index":"log_consultas","_id":11}}
{"@timestamp":"2010-05-15T11:57:20","estado_consulta":"consumo","servicio":"modificacion","administrador":"Juan
```

## 11. En este punto se comienza el ejercicio con las consultas

12. 1. Obtener el número de registros con estado\_consulta igual a error y consumo. Como bien lo comenta se utiliza la instrucción **\_search** y la

instrucción query la cual permite realizar las consultas

```
POST /log_consultas/_search
{
  "query": {
    "bool": {
      "should": [
        { "match": { "estado_consulta.keyword": "error" } },
        { "match": { "estado_consulta.keyword": "consumo" } }
      ]
    }
  }
}
```

Para esto se crea la llave query y dentro de eso se pone bool que es una consulta booleana y dentro should que es un operador OR y match para que coincida con el string que se ingrese



Salida:

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 183,
      "relation": "eq"
    },
    "max_score": 1.3440117,
    "hits": [
      {
        "_index": "log_consultas",
        "_id": "4",
        "_score": 1.3440117,
        "_source": {
          "@timestamp": "2010-05-15T22:33:34",
          "estado_consulta": "error",
          "servicio": "modificacion",
          "administrador": "Juan Carlos",
          "consultas_realizadas": 65
        }
      },
      {
        "_index": "log_consultas",
        "_id": "7",
        "_score": 1.3440117,
        "_source": {
          "@timestamp": "2010-05-15T18:37:14",
          "estado_consulta": "error",
          "servicio": "modificacion",
          "administrador": "Juan Carlos",
          "consultas_realizadas": 32
        }
      }
    ]
  }
}
```

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

Al igual que la instrucción pasada se utilizo lo mismo para traer todos los datos de juan lara

```
POST /log_consultas/_search
{
  "query": {
    "bool": {
      "should": [
        { "match": { "administrador.keyword": "Juan Lara" } }
      ]
    }
  }
}
```

Salida:

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 183,
      "relation": "eq"
    },
    "max_score": 1.3440117,
    "hits": [
      {
        "_index": "log_consultas",
        "_id": "4",
        "_score": 1.3440117,
        "_source": {
          "@timestamp": "2010-05-15T22:33:34",
          "estado_consulta": "error",
          "servicio": "modificacion",
          "administrador": "Juan Carlos",
          "consultas_realizadas": 65
        }
      }
    ]
  }
}
```

14. 3. Obtener el número de registros con estado\_consulta igual a informativo y servicio igual a borrado  
Consulta extra
15. 4. Obtener la suma de los valores en consultas realizadas con estado\_consulta igual a error

En esta consulta se realiza lo anterior pero con la llave aggs y sum que lo que

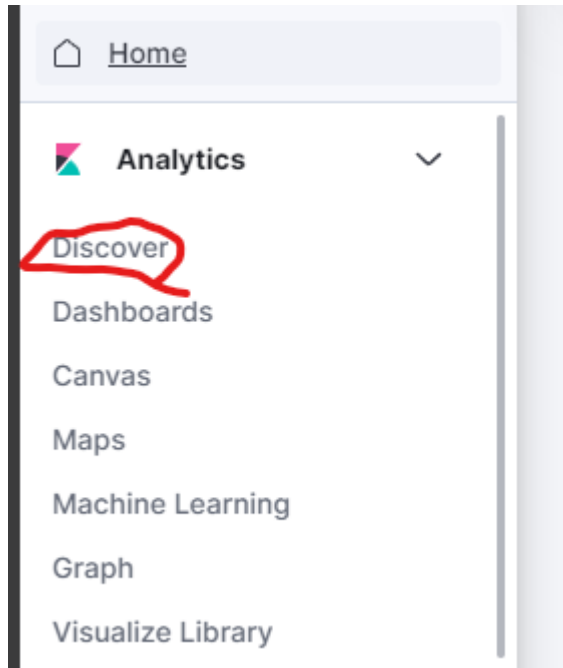
```
POST /log_consultas/_search
{
  "query": {
    "bool": {
      "must": [
        { "term": { "estado_consulta.keyword": "error" } }
      ]
    }
  },
  "aggs": {
    "Total_error": {
      "sum": {
        "field": "consultas_realizadas"
      }
    }
  }
}
```

hace es crear un acumulador y una variable donde sumara todas las consultas realizadas de error.

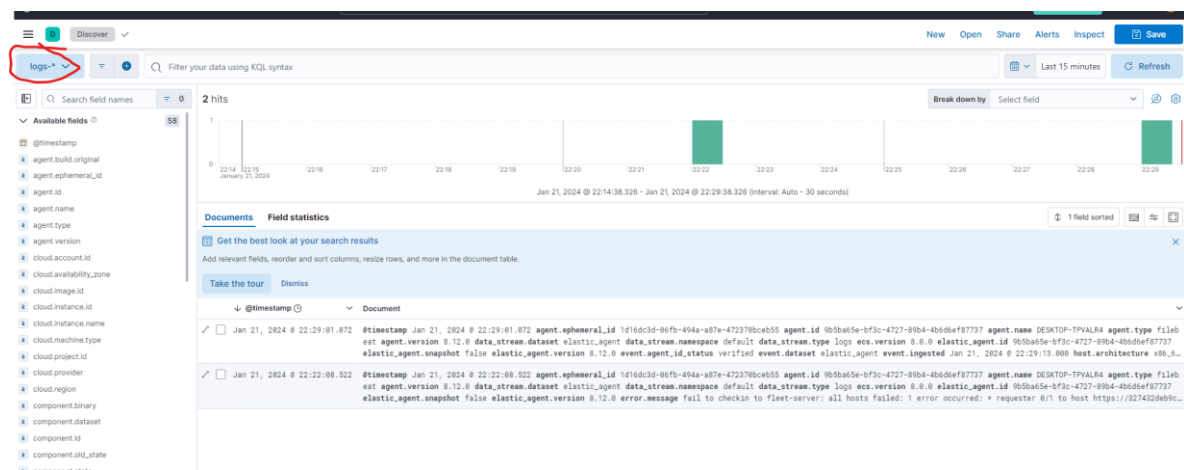
```
    ],
  },
  "aggregations": {
    "Total_error": {
      "value": 2865
    }
  }
}
```

## Realizar un tablero para visualizar información de empleados -

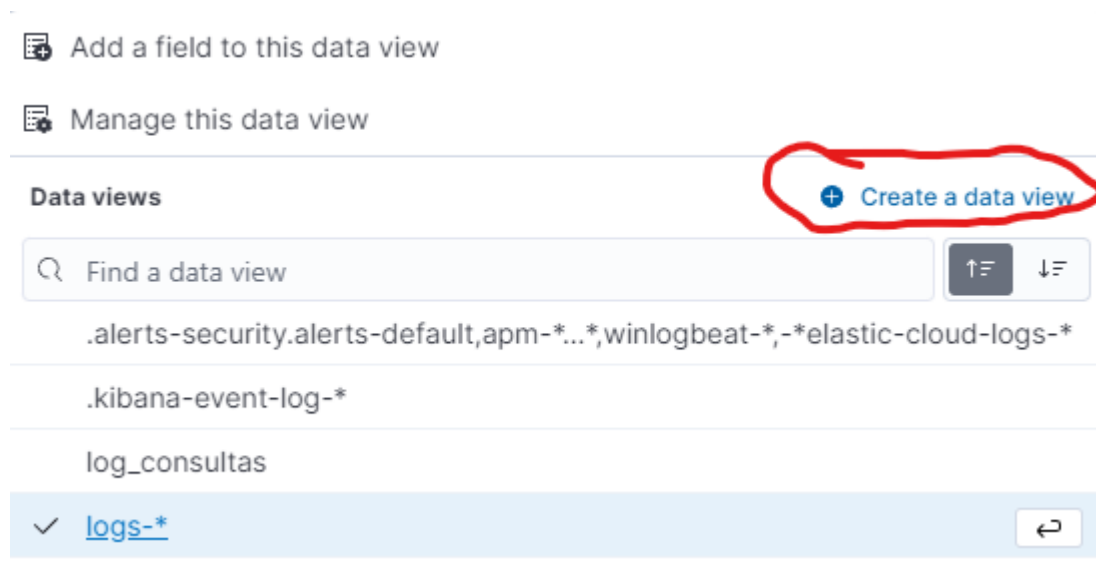
En este punto empezamos con la interpretación y los tableros para esto es necesario ir a dashboard



Posterior a eso no abrirá el siguiente panel pero esta no es nuestra colección seleccionamos donde dice logs



Posterior a eso creamos un nuevo data view



Una vez de eso se llenan los campos y le indicamos el index que queremos utilizar

The screenshot shows the 'Create data view' form in Kibana. The form has three main sections: 'Name', 'Index pattern', and 'Timestamp field'. The 'Name' field contains the text 'practica'. The 'Index pattern' field contains the text 'log\_consultas'. The 'Timestamp field' contains the text '@timestamp'. Below the 'Timestamp field' is a note: 'Select a timestamp field for use with the global time filter.' and a link 'Show advanced settings'. On the right side of the form, there is a message: '✓ Your index pattern matches 1 source.' Below this message is a table with two tabs: 'All sources' and 'Matching sources'. The 'Matching sources' tab is active, showing a table with one row: 'log\_consultas' and one column: 'Index'. Below the table is a dropdown menu labeled 'Rows per page: 10'. At the bottom of the form, there are three buttons: 'Close', 'Use without saving', and 'Save data view to Kibana'.

Ahora seleccionamos el data view

The screenshot shows the Kibana interface with the 'Discover' tab selected. A dropdown menu is open, showing a list of data views. The 'log\_consultas' view is highlighted with a red circle. The background shows a table of log data with columns for @timestamp and Document.

**Data views**

- Find a data view
- .alerts-security.alerts-default,apm-\*,winlogbeat-\*,elastic-cloud-logs-\*
- .kibana-event-log-\*
- log\_consultas**
- logs-\*
- metrics-\*

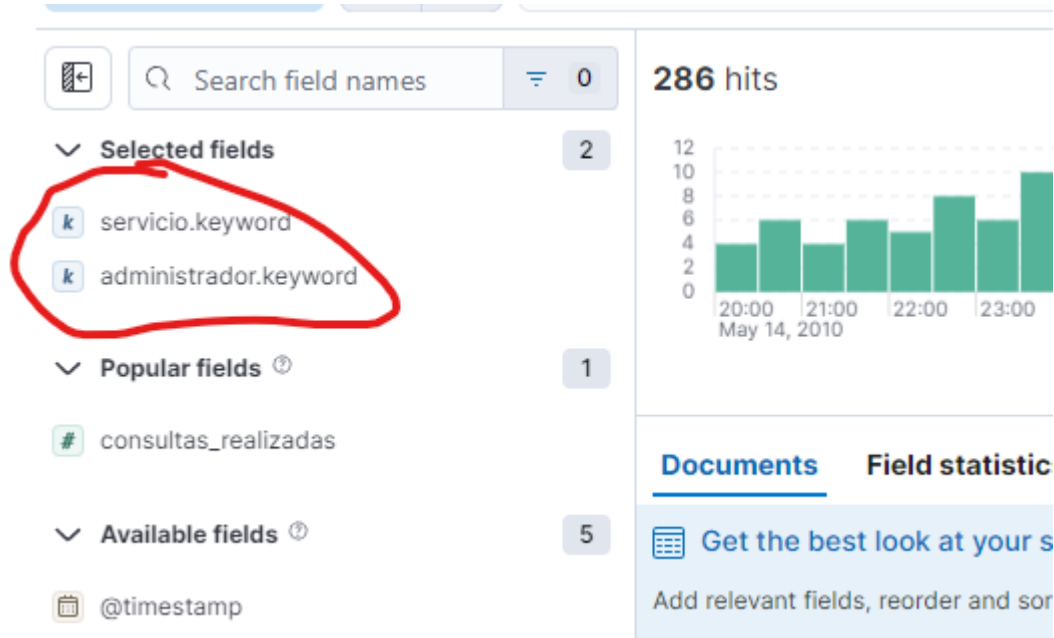
**Try ES|QL** Technical preview

	@timestamp	Document
<input type="checkbox"/>	Jan 21, 2024 @ 22:29:01.072	@timestamp Jan 21, 2024 @ 22:29:01.072 eat agent.version 8.12 elastic_agent.snapshot
<input type="checkbox"/>	Jan 21, 2024 @ 22:22:08.522	@timestamp Jan 21, 2024 @ 22:22:08.522 eat agent.version 8.12 elastic_agent.snapshot

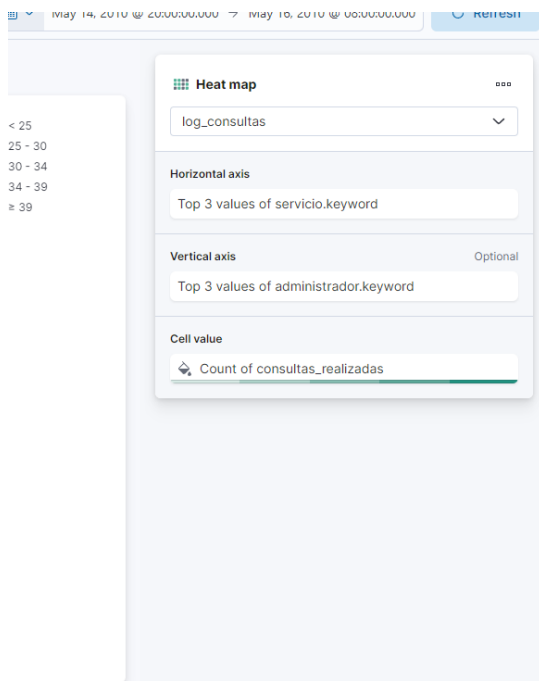
Una vez ya configurado comenzamos con la otra practica

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

Para crear este grafico se necesita agregar previamente los dos valores a utilizar y seleccionarlos que es servicio y administrador



Posteriormente a esto se configura los valores con los cuales se quiere graficar en este caso servicios y administradores



Marco Antonio Mena Landa

Posterior a eso configuramos el color verde



Lo guardamos para ver lo dentro del dashboard.

### Save Lens visualization

**Title**  
heat map

**Description** Optional

**Add to dashboard**

- ☐ Existing  
Select dashboard
- ☒ New
- ☐ None

☐ Add to library ⓘ

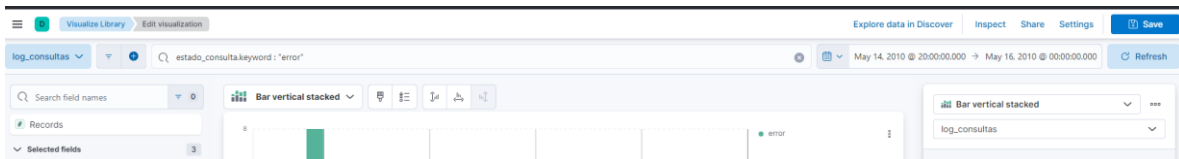
**Buttons:** Cancel, Save and go to Dashboard

## 2. Vista de Barras, donde se grafique el número de registros con estado\_consulta igual a error a través del tiempo.

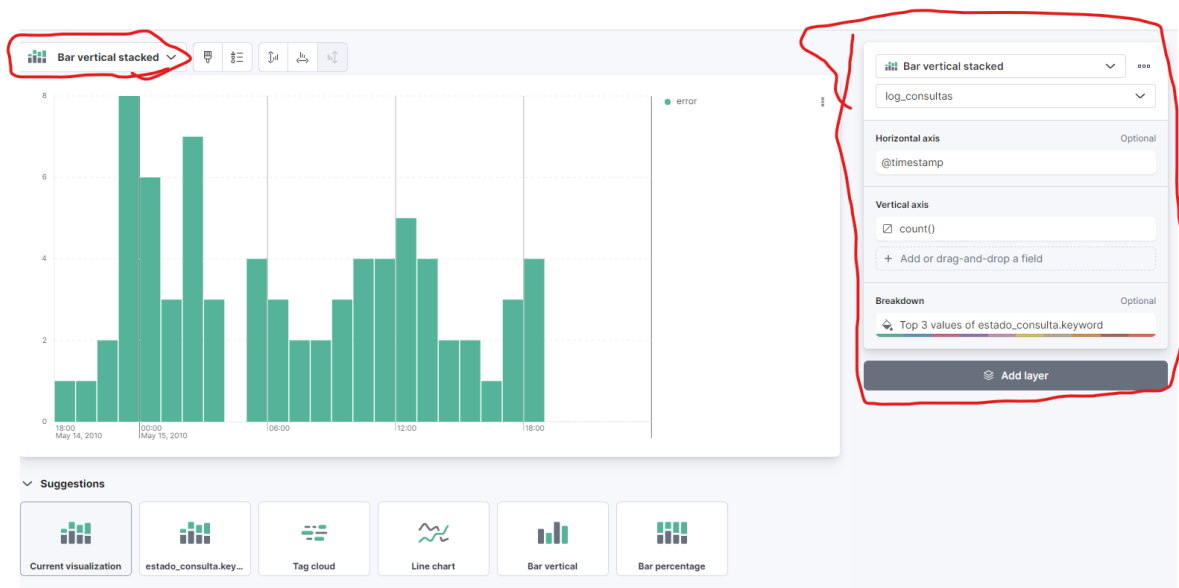
Es importante que al momento de ingresar la consulta tenemos que poner los valores de la consulta y el periodo de tiempo es **muy importante** según el json que



se mando en este caso se utilizo la consulta kql **estado\_consulta.keyword : "error"** y **periodo de tiempo del 14 mayo del 2010 al 16 de mayo del 2010**



Posterior a esto le damos refresh y configuramos los valores del grafico para poder dar continuidad para generar el grafico, en este caso se selecciona barras y en el **campo horizontal** ponemos timestamp y vertical que cuente las incidencias **count()**



En este caso ya se genera el grafico.

Para agregarlo al dashboard le ponemos un titulo y le damos nuevo y lo guardamos con save and go to dashboard.

**Save Lens visualization**

Title

Edit visualization

Description Optional

Add to dashboard

☒ Existing

Select dashboard

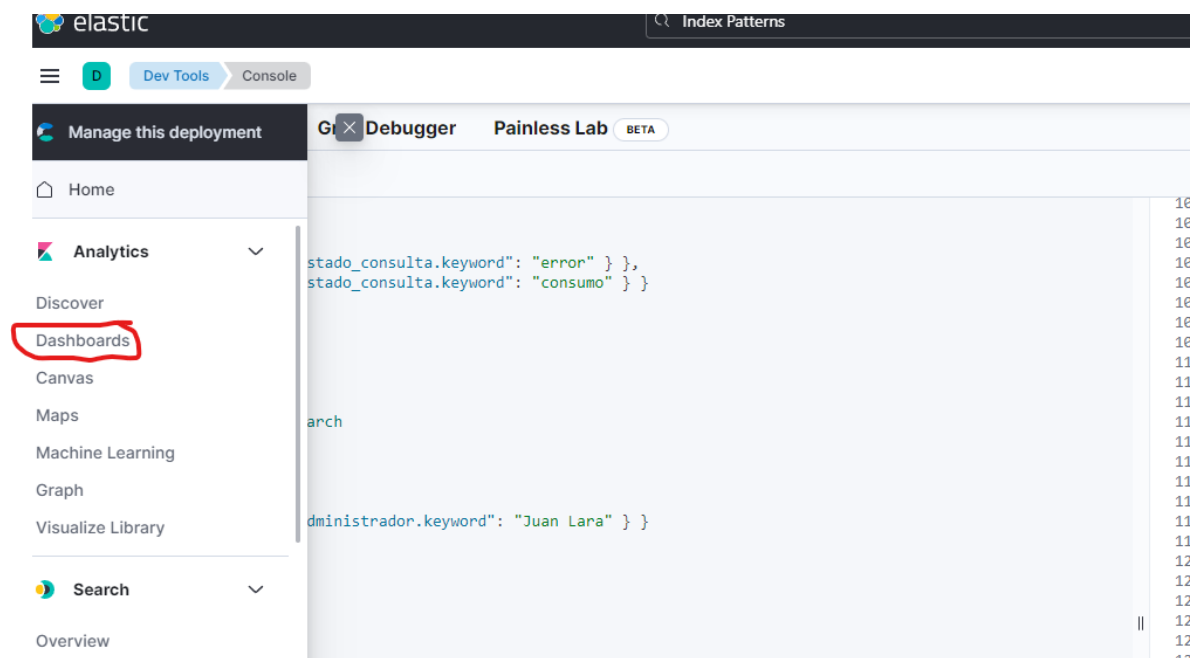
☒ New

☐ None

☐ Add to library ⓘ

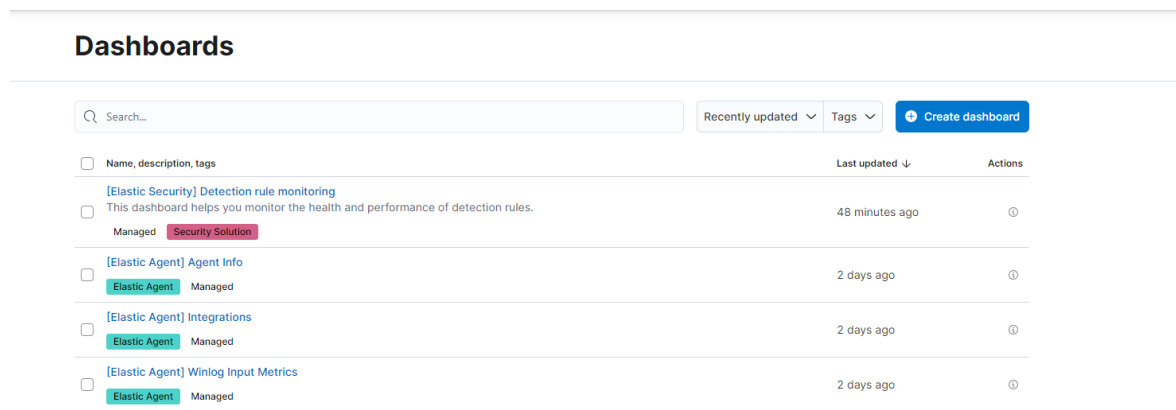
Cancel Save and go to Dashboard

Para ver los en el dashboard seleccionamos de lado izquierdo Dashboards le damos clic

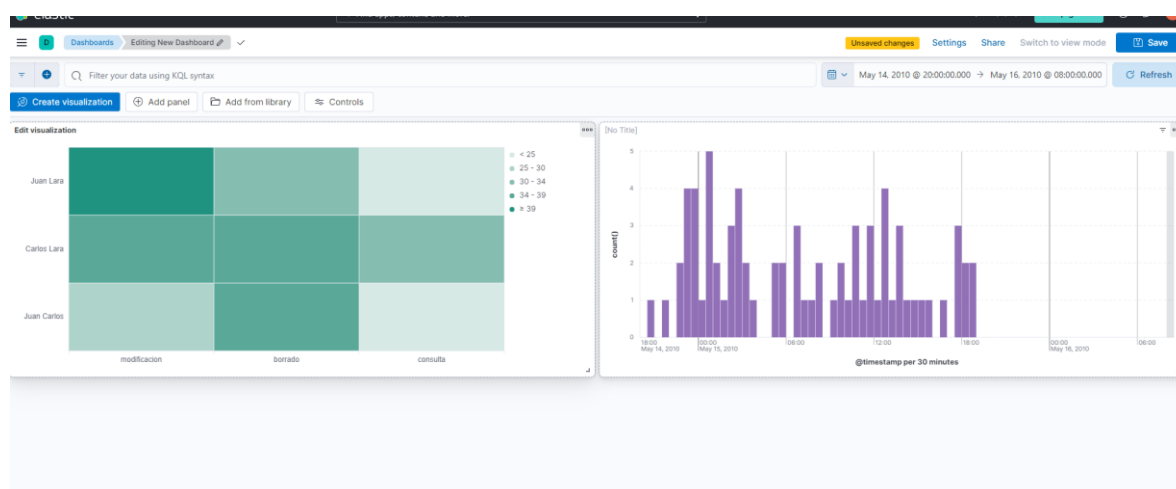


Marco Antonio Mena Landa

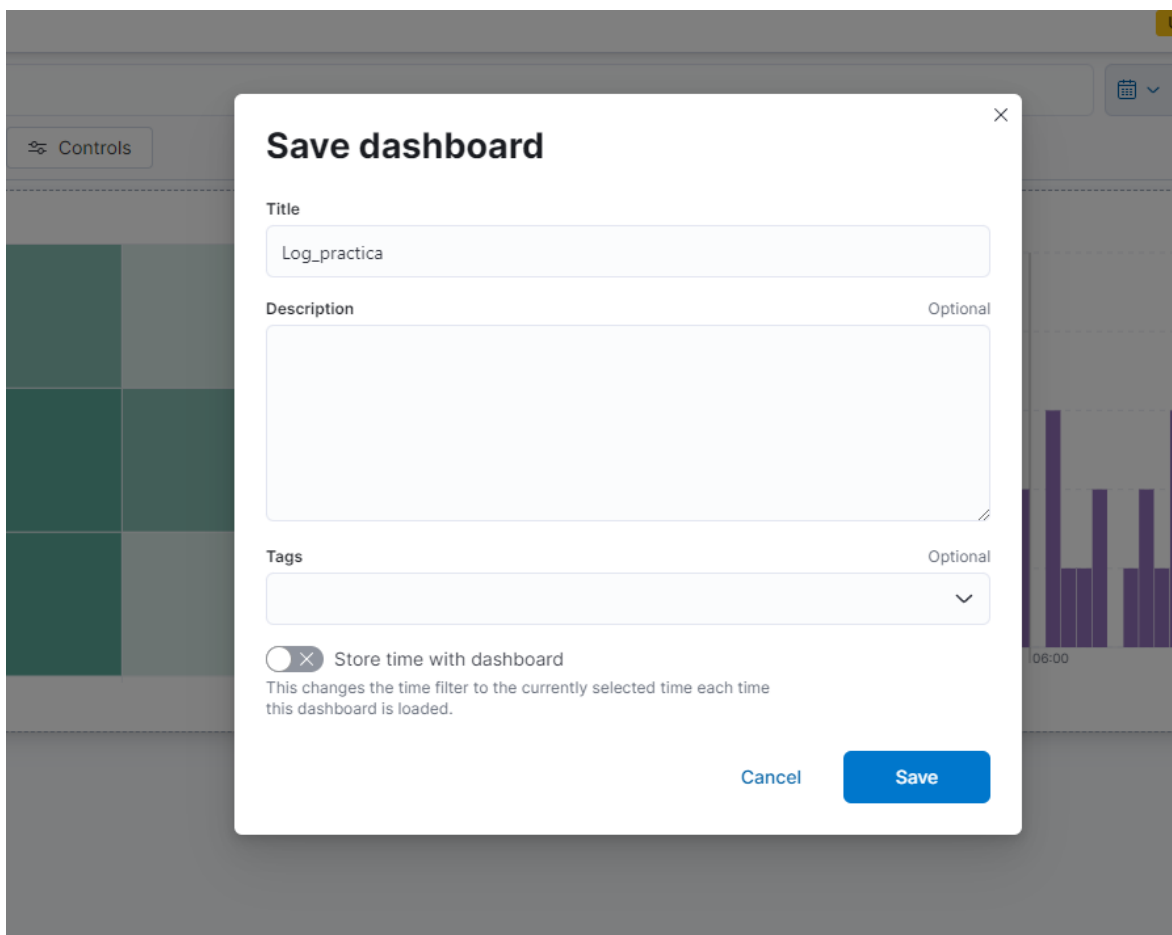
Le damos en crear dashboard



Y agregamos los paneles anteriormente realizados.



Una vez terminado le damos en guardar dashboard



The image shows a 'Save dashboard' modal dialog box. It has a title bar with a close button (X). The form contains the following fields:

- Title:** A text input field containing 'Log\_practica'.
- Description:** A large text area with a placeholder. It is labeled 'Optional'.
- Tags:** A dropdown menu with a downward arrow. It is labeled 'Optional'.
- Store time with dashboard:** A toggle switch that is currently turned on. Below it, a note reads: 'This changes the time filter to the currently selected time each time this dashboard is loaded.'

At the bottom right of the modal are two buttons: 'Cancel' and 'Save'.

Listo se guarda exitosamente

Dashboards			
<input type="text" value="Search..."/>		Recently updated ▾	Tags ▾ <button>Create dashboard</button>
<input type="checkbox"/>	Name, description, tags	Last updated ▾	Actions
<input type="checkbox"/>	Log_practica	7 seconds ago	