Programa AWS Big Data Analytics Specialist

Informe
Arquitectura Final AWS

Integrantes:

Anthony Steve Brian Manosalva López
Julian Castiblanco P
Jennyfer Nereida Combariza Vanegas
Isel Paola Santos Rojas

Junio 2020

Introducción

Este proyecto constituye el informe final del curso AWS Big Data Analytics Specialist.

Comprender las emociones de las personas cada día se convierte en una de las herramientas más esenciales para las empresas, ya que les permite escuchar atentamente a sus clientes y adaptar sus productos y/o servicios para satisfacer sus necesidades.

En este breve proyecto vamos a explicar como hacer una arquitectura serverless para realizar análisis de sentimientos de tweets relacionados con el covid en Perú.

Para esa arquitectura serverless usaremos los siguientes servicios de AWS:

1. Kinesis Firehose: servicio de AWs que permite el procesar datos en tiempo real.
2. Lambda: servicio de AWS que ejecuta cómputo a demanda.
3. Comprehend: servicio de procesamiento de lenguaje natural.
4. Lambda: servicio de AWS que ejecuta cómputo a demanda.
5. Elastic Search: servicio de AWS facilita la implementación, la protección y la ejecución de Elasticsearch a escala de manera rentable
6. Kibana. Ventana al Elastic Stack. Kibana es una interfaz de usuario gratuita y abierta que te permite visualizar los datos de Elasticsearch

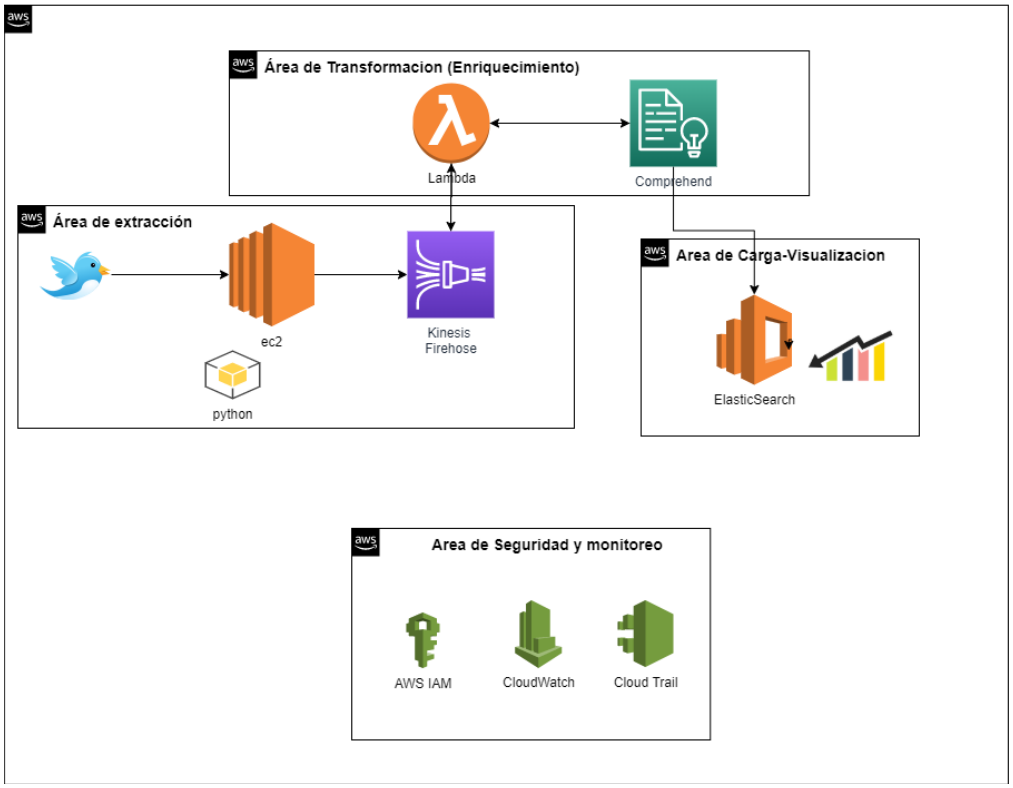
Para este análisis de sentimientos recopilamos los tweets con Kinesis Firehose, los procesamos con comprehend y visualizamos utilizando Elastic Search- Kibana.

1. Objetivo

Construir una arquitectura serverless que permita la visualización de análisis de sentimientos de tweets en tiempo real.

2. Arquitectura propuesta

Arquitectura propuesta
End to End



3. Alcance

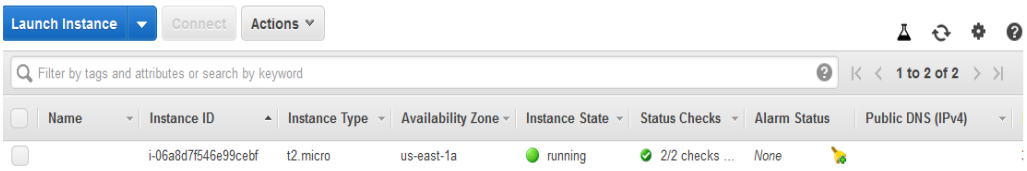
El presente documento describe los lineamientos llevados a cabo para la generación de la arquitectura propuesta desde la extracción de los tweets hasta la visualización de los resultados obtenidos del análisis de sentimientos.

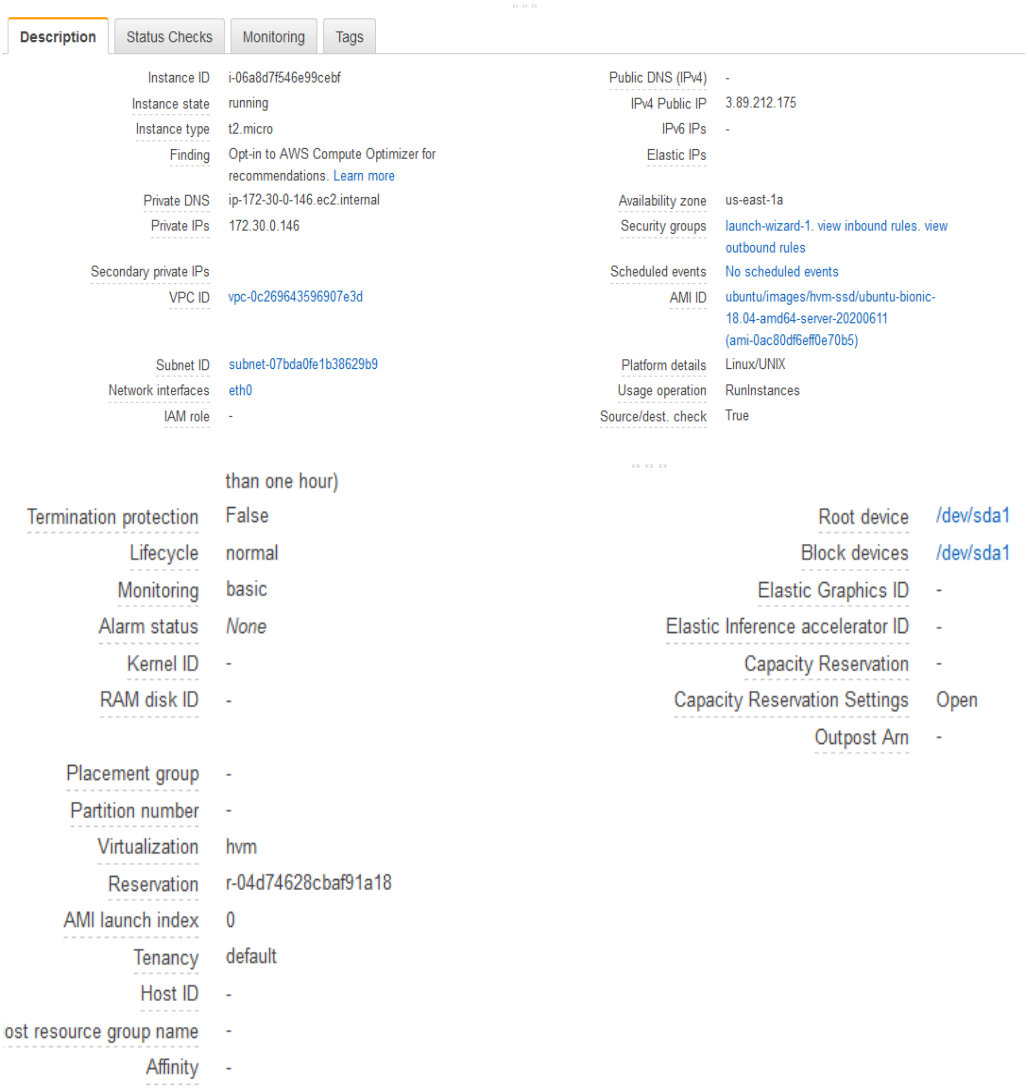
4. Configuraciones generales

A continuación se listan los servicios que se usaron en el proceso de la generación de la arquitectura:

4.1. Configuración de EC2

Se crea una máquina EC2 que se encargará de ejecutar un código Python para conectarse con la cuenta de desarrollo de twitter y de esa manera extraer información acerca del covid en Lima Perú, con filtro de idioma en español.

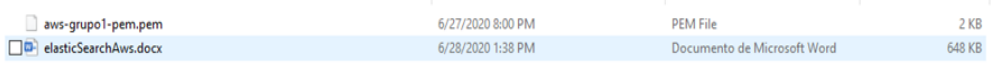




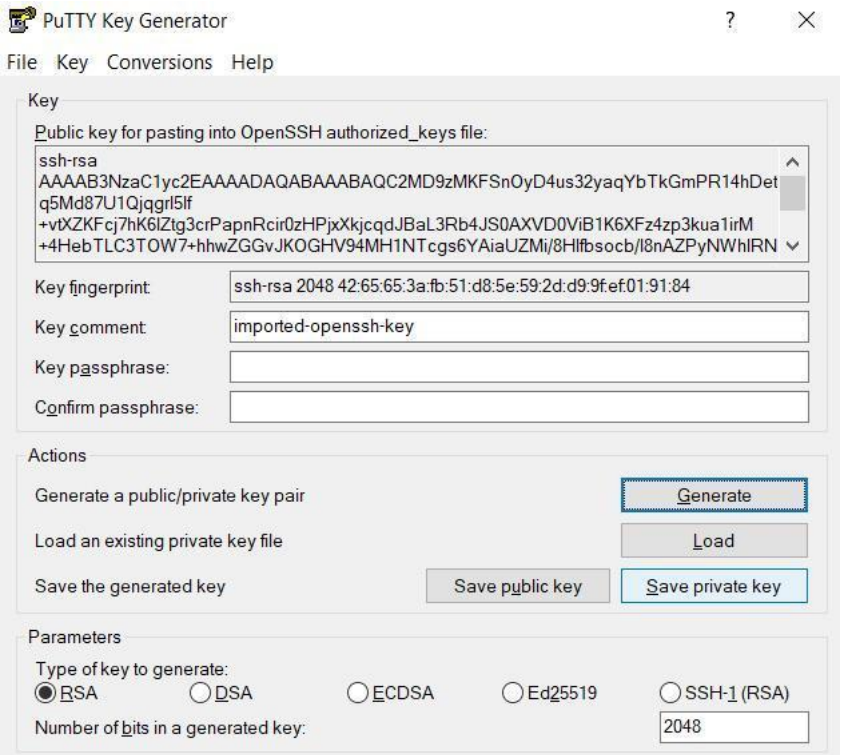
4.2. Conexión a EC2 por SSH a través de Putty

Objetivo: Cargar el código de python e instalar las librerías necesarias para una conexión exitosa.

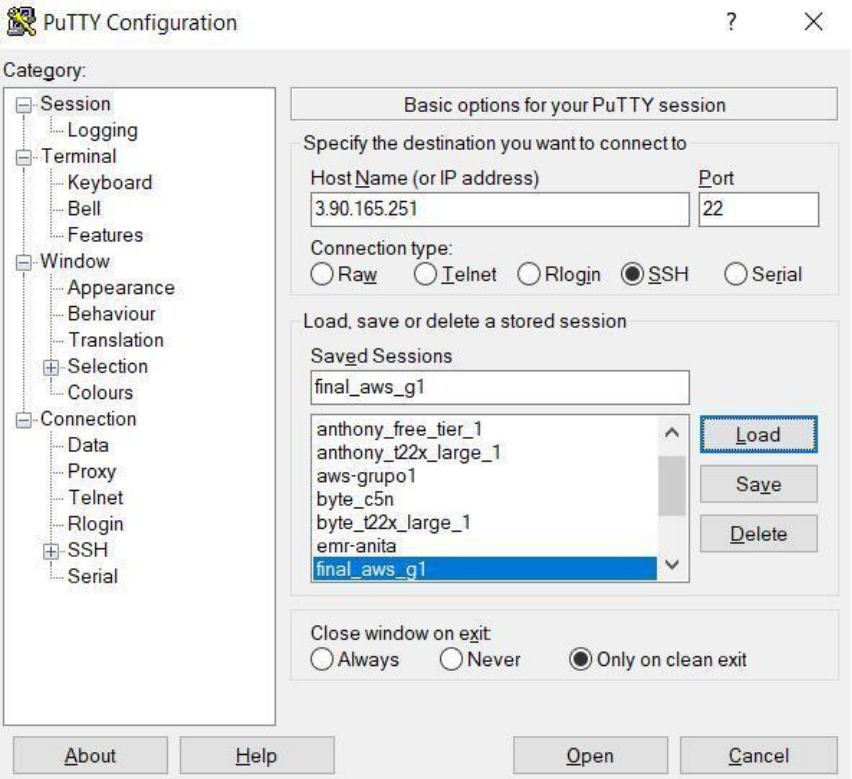
- Paso 1: Se descarga el archivo pem, donde viene la key privada.



- Paso 2: Se abre la herramienta “putty generator” para convertir el archivo pem en ppk que es válido para la conexión desde putty.



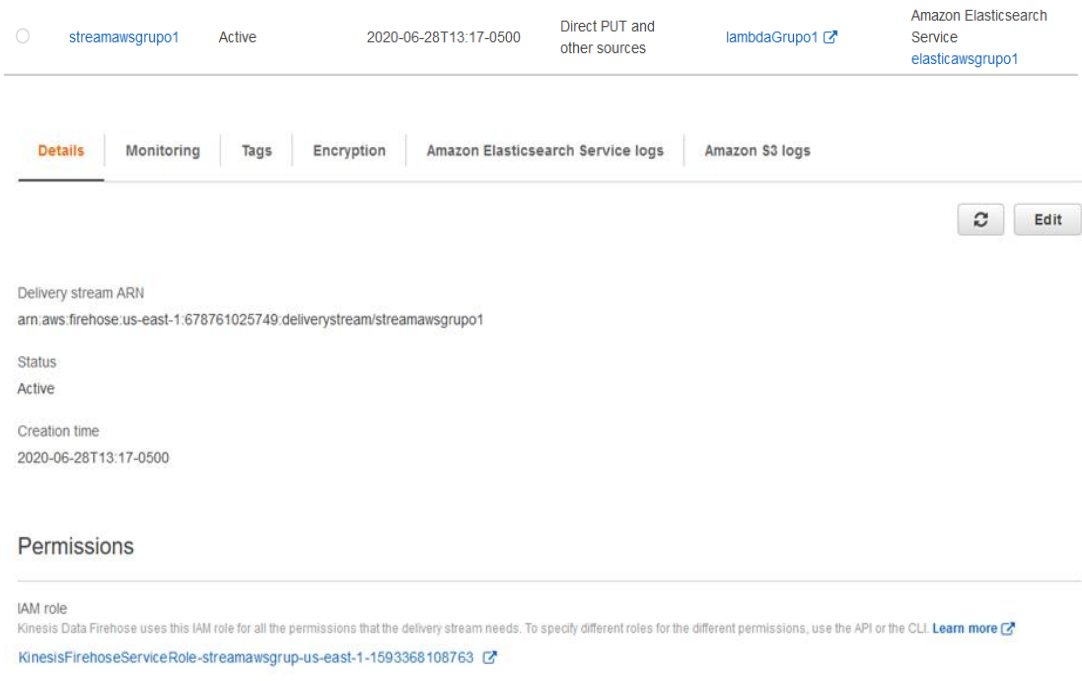
- Paso 3: Se hace conexión a través de putty.



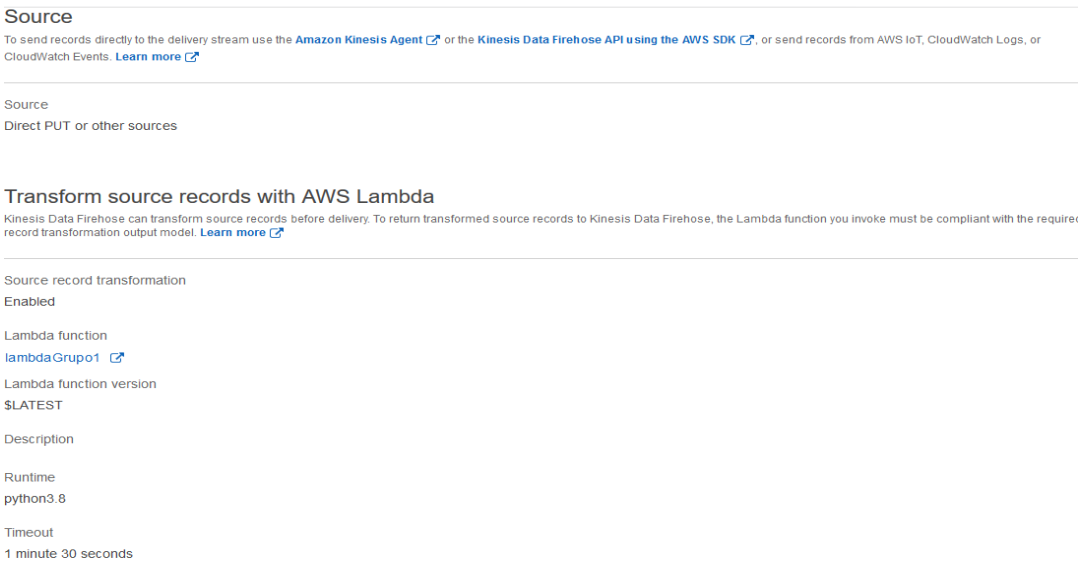
- Paso 4: Se modificó el py, para que se conecte a firehose el cual va a recibir la información en formato json, codificado a 64 bits el mensaje.

- Paso 5: Se ejecuta la función de extracción de tweets.

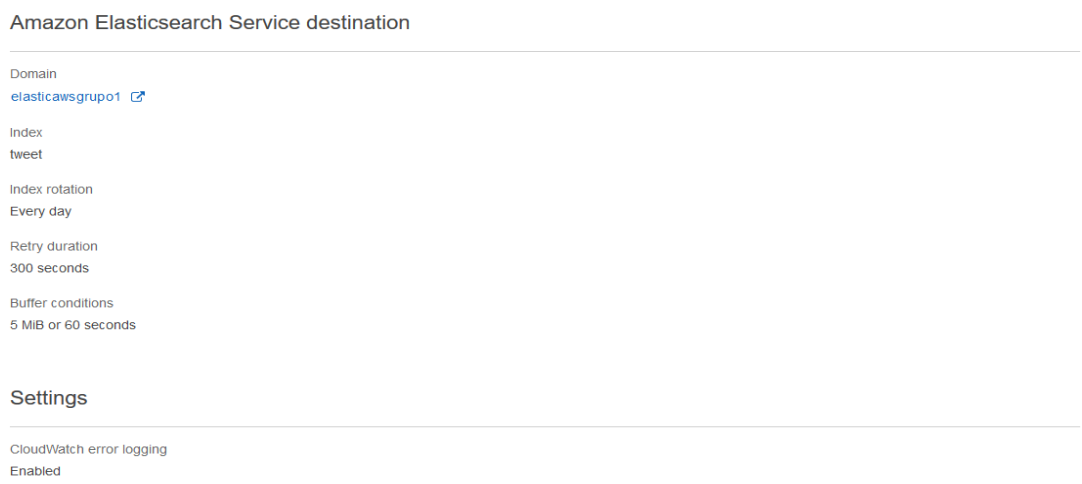
4.3. Configuración del servicio de Firehose



4.4. Configuración del servicio de Lambda, S3 y Elasticsearch

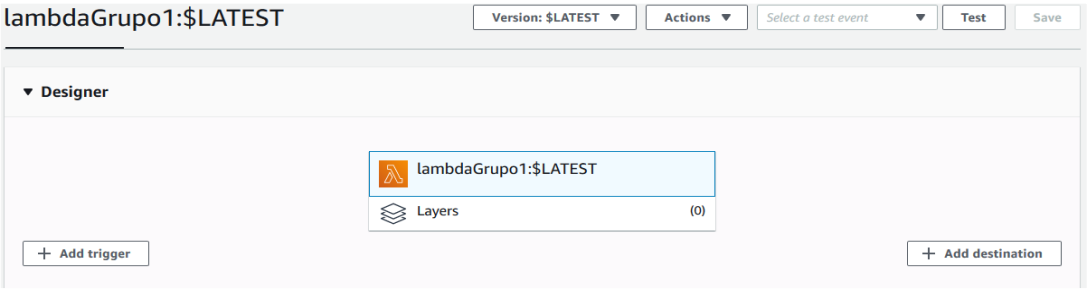


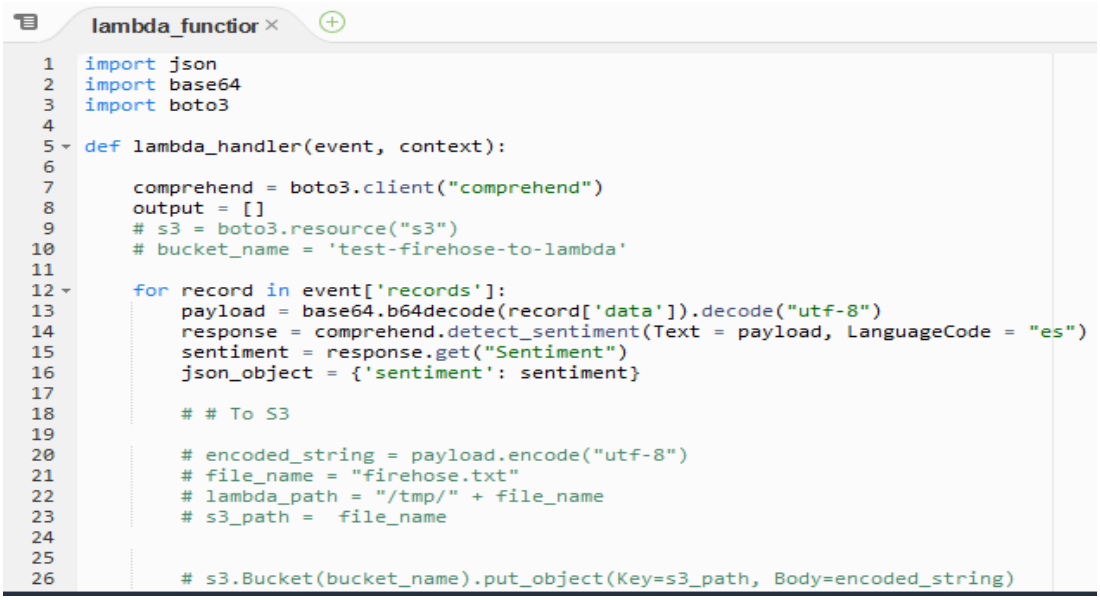
4.5. Configuración del servicio Elasticsearch



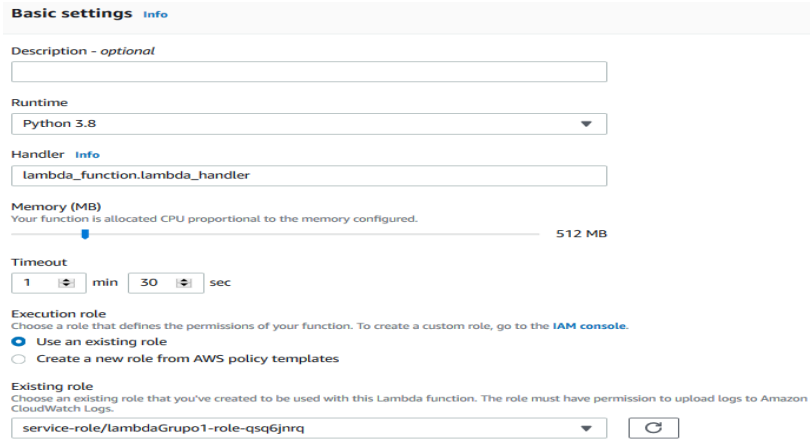
4.6. Creación de la función Lambda

Se creó una función lambda que se encarga de utilizar el servicio de comprenhed, para analizar los tweets.





Se modificó la configuración de la función para que durara más tiempo y pueda utilizar más memoria en el procesamiento:



4.7. Creación del elasticSearch

Step 1: Name and source
Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review

New delivery stream

Delivery streams load data, automatically and continuously, to the destinations that you specify. Kinesis Data Firehose resources are not covered under the **AWS Free Tier**, and **usage-based charges apply**. For more information, see **Kinesis Data Firehose pricing**. **Learn more**

Delivery stream name

streamawsgrupo1

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Choose a source

Choose how you would prefer to send records to the delivery stream.

Source

To learn about enabling server-side encryption (SSE), see **Data Protection in Amazon Kinesis Data Firehose**.

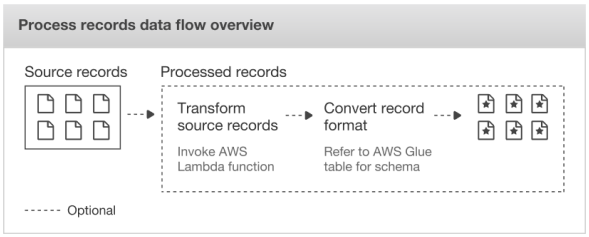
- (•) Direct PUT or other sources
  Choose this option to send records directly to the delivery stream, or to send records from AWS IoT, CloudWatch Logs, or CloudWatch Events.
- ( ) Kinesis Data Stream

Server-side encryption for source records in the delivery stream

You can use AWS Key Management Service (KMS) to create and manage customer keys (CMKs) and to control the use of encryption across a wide range of AWS services and in your applications.

- [ ] Enable server-side encryption for source records in delivery stream

▶ **How to send source records to Kinesis Data Firehose**



Transform source records with AWS Lambda

To return records from AWS Lambda to Kinesis Data Firehose after transformation, the Lambda function you invoke must be compliant with the required record transformation output model. **Learn more**

Data transformation

- ( ) Disabled
- (•) Enabled

Lambda function

lambdaGrupo1 ▾ | Create new

View **lambdaGrupo1** in Lambda

longer in the **Advanced settings** section of your Lambda configuration.
Go to Lambda configuration

Timeout

3 seconds

Convert record format

Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted source records using a schema from a table defined in **AWS Glue**. For records that aren't in JSON format, create a Lambda function that converts them to JSON in the **Transform source records with AWS Lambda** section above. **Learn more**

Record format conversion

- (•) Disabled
- ( ) Enabled
  If record format conversion is enabled, Kinesis Data Firehose can deliver data to Amazon S3 only. Record format conversion will be configured using the OpenX JSON SerDe. For other options use the AWS CLI.

Cancel | Previous | Next

Select a destination

Learn more

Destination

- ( ) Amazon S3
  Amazon S3 is an easy-to-use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.
- ( ) Amazon Redshift
  Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost effective to analyze all your data using your existing business intelligence tools
- (•) Amazon Elasticsearch Service
  Elasticsearch is an open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics
- ( ) Splunk
  Splunk is an operational intelligence tool for analyzing machine-generated data in real-time

delivery stream for VPC connectivity. **Learn more**

elasticawsgrupo1 | Create new

View **elasticawsgrupo1** in Amazon Elasticsearch Service

Index

tweet

A new index will be created if the the specified index name does not exist.

Index rotation

Every day

Select how often to rotate the Elasticsearch index. Kinesis Data Firehose appends a corresponding timestamp to the index and rotates it.

Type

A new type will be created if the specified type name does not exist.

Retry duration

Select how long a failed index request should be retried. Failed documents are delivered to the backup S3 bucket.

300 seconds

Enter a retry duration from 0 - 7200 seconds

S3 backup

To prevent against data loss, Kinesis Data Firehose can back up records to your S3 bucket while delivering it to your Elasticsearch cluster. **Learn more**

Backup mode

- ○ Failed records only
- ◉ All records

Backup S3 bucket

grupo1-bucket | Create new

View **grupo1-bucket** in S3 console

Backup S3 bucket prefix - *optional*

tweet/

Kinesis Data Firehose automatically appends the "YYYY/MM/dd/HH/" UTC prefix to delivered S3 files. You can also specify an extra prefix in front of the time format and add "/" to the end to have it appear as a folder in the S3 console.

Cancel | Previous | Next

Elasticsearch buffer conditions

Kinesis Data Firehose buffers incoming records before delivering them to your Elasticsearch domain. Data delivery will be triggered when either of these conditions is satisfied. **Learn more**

Buffer size

5 MiB

Enter a buffer size between 1-100 MiB

Buffer interval

60 seconds

Enter a buffer interval between 60-900 seconds

S3 compression and encryption

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using a KMS master key. **Learn more**

S3 compression

- ◉ Disabled
- ○ GZIP
- ○ Snappy
- ○ Zip
- ○ Hadoop-Compatible Snappy

S3 encryption

- ◉ Disabled
- ○ Enabled

Error logging

Kinesis Data Firehose can log record delivery errors to CloudWatch Logs. If enabled, a CloudWatch log group and corresponding log streams are created on your behalf. **Learn more**

Error logging

- ○ Disabled
- ◉ Enabled

Tags - *optional*

You can add tags to organize your AWS resources, track costs, and control access. **Learn more**

Key | Value - *optional*

Enter key | *Enter value* | Remove tag

Permissions

IAM role

Kinesis Data Firehose uses this IAM role for all the permissions that the delivery stream needs. To specify different roles for the different permissions, use the API or the CLI. **Learn more**

- (•) Create or update IAM role **KinesisFirehoseServiceRole-streamawsgrup-us-east-1-1593368108763**
  This creates the role or updates it if it already exists, adds the required policies to it, and enables Kinesis Data Firehose to assume it.
- ( ) Choose existing IAM role
  The role that you choose must have policies that include the permissions that Kinesis Data Firehose needs.

Cancel | Previous | Next

5 MiB or 60 seconds

S3 buffer conditions

5 MiB or 60 seconds

Compression

Disabled

Encryption

Disabled

Error logging

Enabled

Tags

no tags specified

IAM role

KinesisFirehoseServiceRole-streamawsgrup-us-east-1-1593368108763

Cancel | Previous | Create delivery stream

Se modificaron los permisos para acceder al Kibana desde cualquier lugar público

Status Active

Domain access policy JSON defined access policy

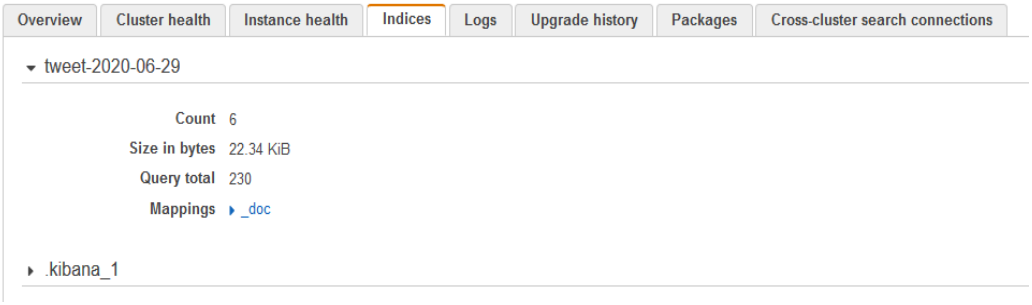
Allow or deny access by AWS account ID, account ARN, IAM user ARN, IAM role ARN, IPv4 address, or CIDR block.

Add or edit the access policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:678761025749:domain/elasticawsgrupo1/*"
    }
  ]
}
```

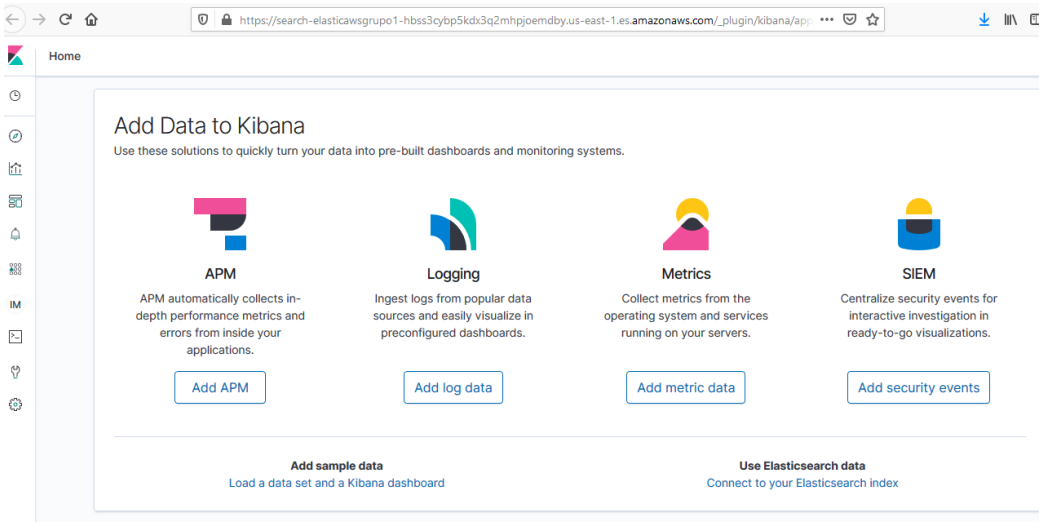
4.8. Ejecución del proceso de adquisición de tweets

Se valida que se haya creado el índice del elasticsearch



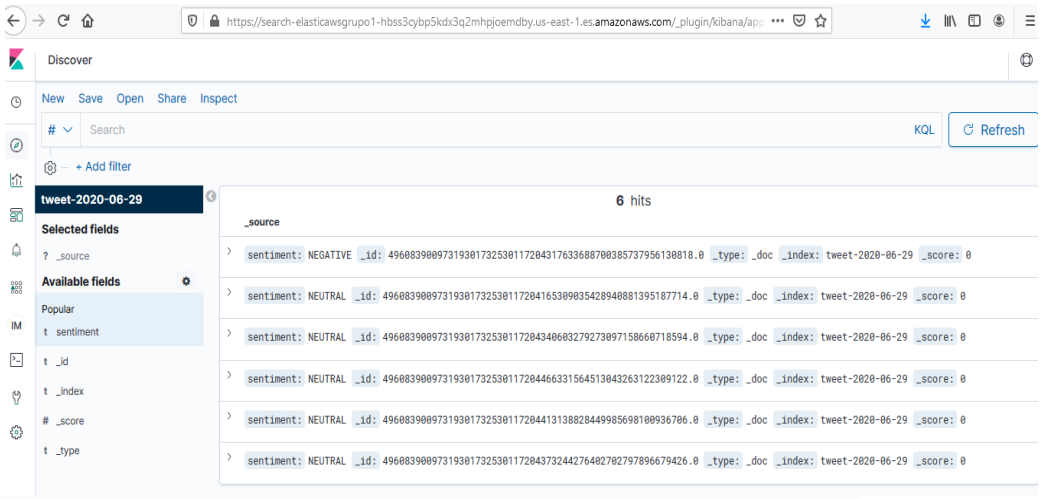
4.9. Visualización de adquisición de tweets

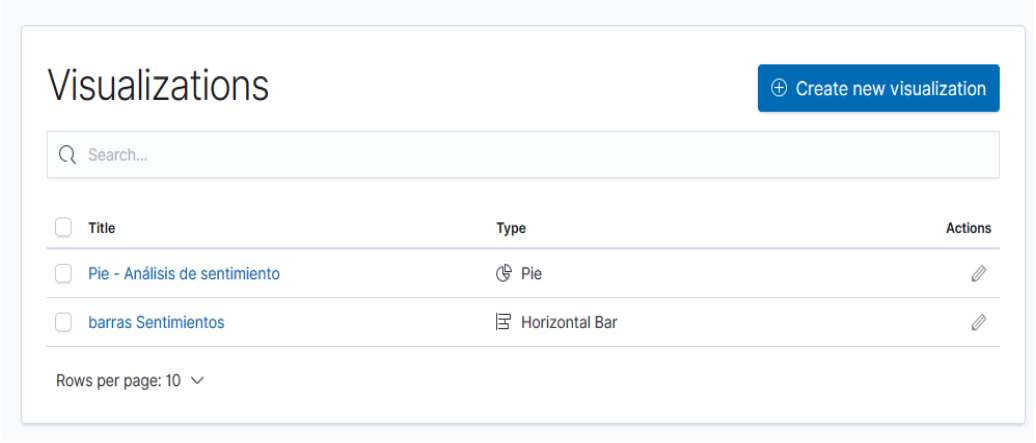
Una vez que se tiene informaciòn en el motor de búsqueda, accedemos a la URL de kibana y podemos configurar el índice de búsquedas.



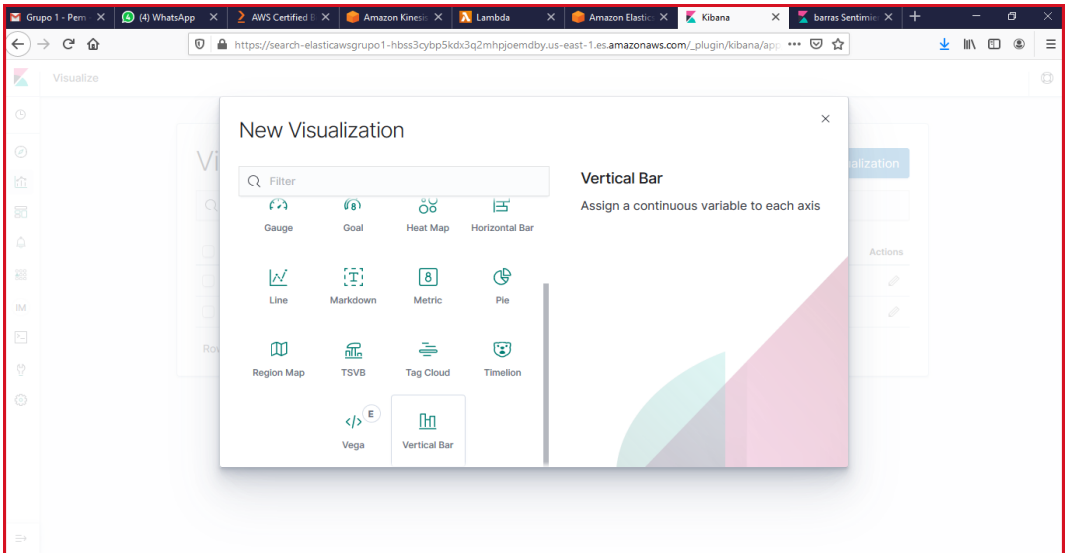
A continuación, en el ícono de discover, podemos validar los datos disponibles para la creación de las gráficas.

Luego se da clic en el ícono de crear una nueva visualización

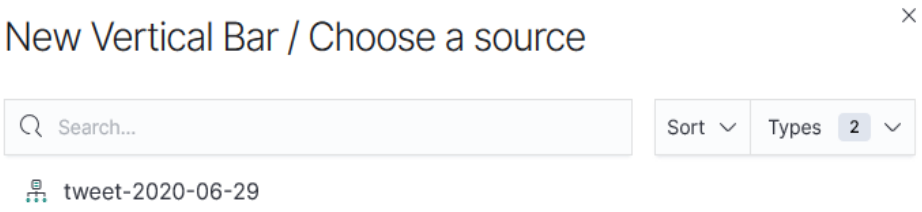




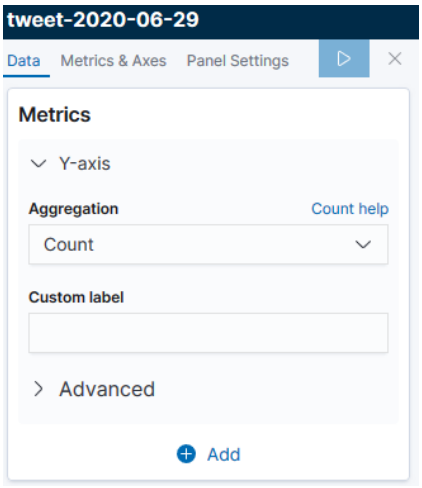
Seleccionamos nueva gráfica de barras



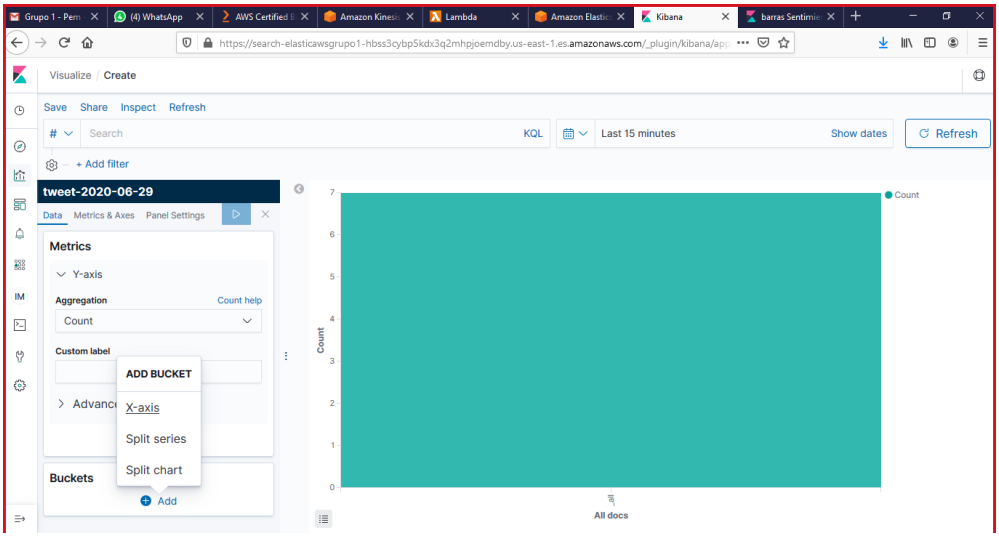
Seleccionamos el set de datos que queremos analizar

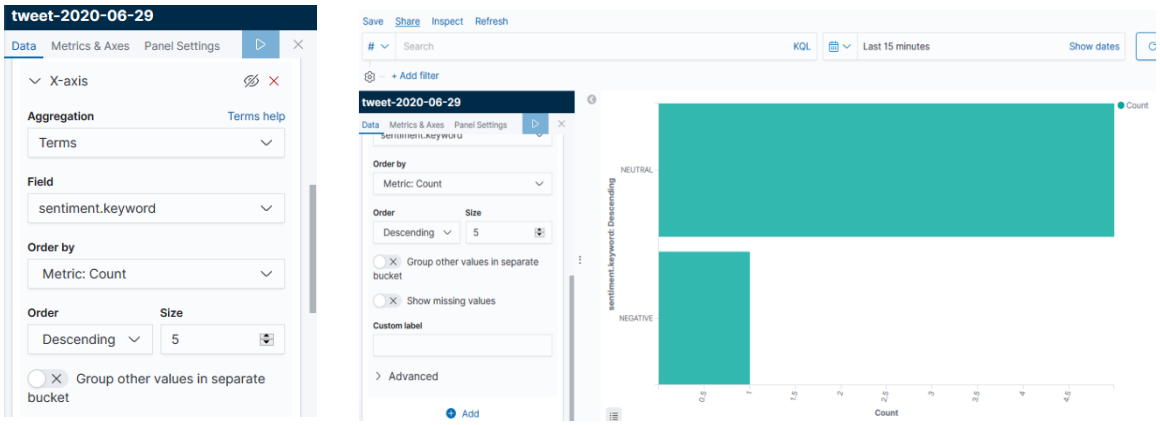


En el eje Y se deja que cuente cuantos registros hay por tipo de respuesta sentimental



A continuación le damos clic en el botón add bucket





De manera similar se pueden replicar los pasos para realizar un gráfico de tipo pie.