

# ARQUITECTURA STREAMING PARA BIG DATA CON APACHE KAFKA Y TWITTER

Por:

Héctor Fabio Banilat Quintero Daniel Alfonso Solano Velásquez

Profesor: Edwin Nelson Montoya Munera



NOVIEMBRE DE 2023
UNIVERSIDAD EAFIT
Tópicos Especiales en Telemática (proyecto 3)

# Arquitectura Streaming para Big Data con Apache Kafka y Twitter

## Descripción:

El objetivo de este proyecto3 es implementar una arquitectura de Streaming de Big Data, que permita capturar datos en tiempo real de la famosa plataforma de Twitter, almacenarlos en un Data Lake y analizarlos en tiempo real.

Para llevar acabo esto, se usarán las siguientes herramientas:

- Apache Kafka: Será nuestro canalizador de datos en tiempo real.
- AWS S3: Para el tema del Data Lake.
- Apache Spark Streaming: Será nuestro procesador de flujos.

## Ahora bien, ¿Como será el flujo de datos?

El flujo de datos será el siguiente:

- Un productor de Kafka usará la API de Twitter para adquirir la información o los datos de los tweets.
- El productor se encargará de hacer públicos estos datos en un 'topic' de Kafka.
- Un consumidor de Kafka recibira la informacion o los datos del 'topic' mencionado anteriormente y luego los almacenará en el Data Lake.
- Un procesor de flujos de Spark Streaming se encargará de leer la información que hay en el Data
   Lake se dispondrá a realizar un análisis de sentimientos.

### **Herramientas:**

Las herramientas que serán usadas para llevar a cabo esta arquitectura en el proyecto3 serán las siguientes:

- Apache Kafka: Apache Kafka será el 'canalizador' de datos en tiempo real que permitirá publicar
  y suscribirse a mensajes de manera confiable. Se escogió debido a que es bastante escalable y
  tolerante a fallas, lo que a nuestro punto de vista, lo hace muy ideal para este tipo de aplicaciones
  de streaming.
- AWS S3: AWS S3 será nuestro servicio de almacenamiento de objetos, el cual nos ofrece un
  rendimiento y una escalabilidad bastante buena. Se escogió debido a que es una excelente opción
  para el almacenamiento de datos de streaming, permitiéndonos entrar o acceder de una forma
  mucho más sencilla y rapida.
- Apache Spark Streaming: Apache Spark Streaming será nuestro framework de procesamiento de flujos distribuido, el cual nos permitirá procesar cientos de miles de datos en tiempo real. Y después de todo, esta es una gran herramienta que sirve mucho para aplicaciones de streaming, sin dejar a un lado el tema del análisis de sentimientos.

### Otra información:

- El análisis de sentimientos se realizará utilizando un modelo de aprendizaje automático entrenado para identificar los sentimientos positivos, negativos e neutros de los tweets, puesto que contamos con conocimientos previos sobre Inteligencia Artificial.
- Una vez corriendo el análisis de sentimientos tendremos los suficientes datos para sacar provecho
  de ellos, tales como serían temas de creación de campañas de marketing, detección de tendencias,
  entre muchas otras cosas.
- Idea tomada de: <a href="https://medium.com/@lorenagongang/sentiment-analysis-on-streaming-twitter-datausing-kafka-spark-structured-streaming-python-part-b27aecca697a">https://medium.com/@lorenagongang/sentiment-analysis-on-streaming-twitter-datausing-kafka-spark-structured-streaming-python-part-b27aecca697a</a>

| Materia | Tópicos especiales en Telemática |

| Curso | ST0263 |

# Proyecto No 3.

| Estudiantes | Hector Fabio Banilat Quintero (hfbanilatq@eafit.edu.co) |

| Daniel Solano Velasquez (dsolano@eafit.edu.co) |

| Profesor | Edwin Nelson Montoya Munera (emontoya@eafit.edu.co) |

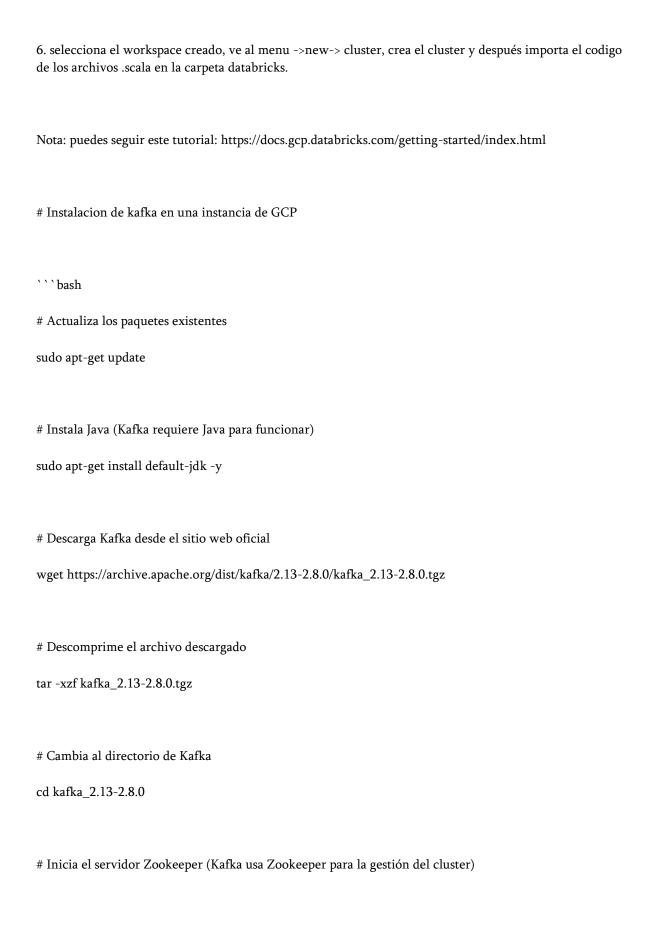
# 1. Objetivo

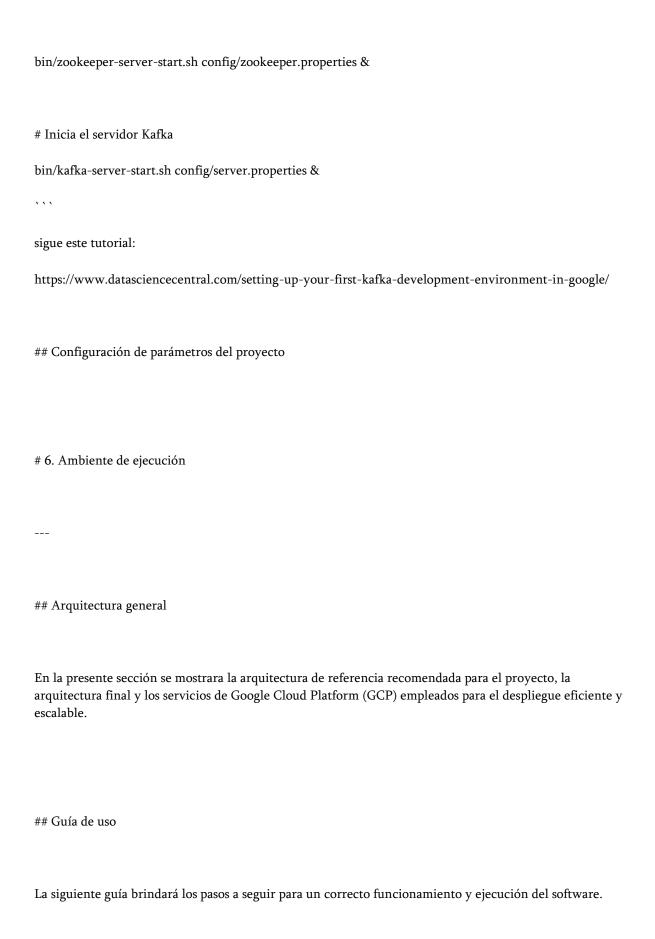
\_\_\_

El objetivo de este proyecto3 es implementar una arquitectura de Streaming de Big Data, que permita capturar datos en tiempo real de la famosa plataforma de Twitter, almacenarlos en un Data Lake y analizarlos en tiempo real; dichos datos deberán de ser llevados a un bucket de s3 y a un procesador de flujos que analice los datos conforme van llegando.

[https://youtu.be/nybIcgaGzqU](https://youtu.be/nybIcgaGzqU)
# 3. Aspectos solucionados y no solucionados
- [] Captura de datos en tiempo real. (Se creo un faker de tweets debido a que el API de twetter no estaba funcionando)
- [x] Subir los datos capturados a un Topic en Kafka.
- [x] Almacenamiento de datos en un bucket de S3.
- [x] Procesamiento de flujo de datos.
- [x] Visualización en tiempo real.
# 4. Información general del diseño
Se empleó databricks para la creación del cluster de spark, además se empleó un VM en GCP donde se instaló kafka. Se emplea un bucket de S3 para el almacenamiento de los datos ingresados en la cola
## Sobre los servicios
En esta sección, se explican los servicios implementados para dar solución al reto propuesto, explicado en la sección: 1. Objetivo.
Nombre del servicio   Rol que desempeña   IP y puertos de escucha

spark-streaming-twitter_2.12 Libreria de spark par ala integración de la API de tweeter
Apache Kafka   Almacena los datos crudos consultados por el servicio de TradeStreaming.   34.123.51.250:9092
DataBricks   Lee, procesa, transforma y carga a S3 los datos crudos, presentes en un tópico de Apache Kafka.   -
$\mid S3\mid Almacena \ los \ datos \ transformados \ por \ el \ servicio \ de \ DataBricks. \ \mid s3://hfbanilatqsparks3/dataspark \ \mid s3://hfbanilatqsparks3/datasparks3/$
$\mid$ Grafana $\mid$ Permite la visualización de los datos presentes en Mongo DB. $\mid$ - $\mid$
# 5. Ambiente de desarrollo
Se emplea dos notebook en databricks, con el lenguaje scala
## Estructura del código
—— databricks
FakerTweetConsumer.scala
FakerTweetConsumerWithStorage.scala
FakerTweetProducer.scala
## Creacion de cuenta en databricks y creacion de cluster
1. Ve a la consola de GCP y en el buscador por Databricks
2. En los resultados selecciona el de la marketplace
3. Dale clic en solicitar, llena los datos y espera que se te active
4. Inicia sesion en databricks
5. Ve a tus workspaces y crea un nuevo workspace





### Instalando requisitos previos
Debemos instalar las librerias usando el UI de databricks en el cluster las librerias son:
edu.stanford.nlp:stanford-corenlp:4.5.5
Maven
- -
org.apache.bahir:spark-streaming-twitter_2.12:2.4.0
Maven
-
org.apache.spark:spark-streaming-kafka-0-10-assembly_2.12:3.5.0
## Ejecución del programa
En la UI de databricks, crea un nuevo cuaderno de scala por cada archivo que hay en la carpeta
databricks,importarlos al workspace y ejecutarlos en el orde, Producer, Consumer, ConsumerWithStorage.