



MANUAL DE USO

Francisco Javier Fernández

1.	INTRODUCCIÓN.....	2
2.	CONSULTA DE DATOS AGREGADOS	2
3.	PRODUCCIÓN DE MENSAJES.....	5

1. INTRODUCCIÓN

En este documento se detallan las diferentes configuraciones que puede realizar el usuario sobre el sistema, aunque al tratarse de un entorno que se da completamente configurado y automatizado, únicamente nos vamos a centrar en aquellas operaciones sencillas que el usuario que va a explotar el sistema puede hacer sin cambiar la lógica y la codificación de los programas y scripts que hacen que el entorno sea operativo.

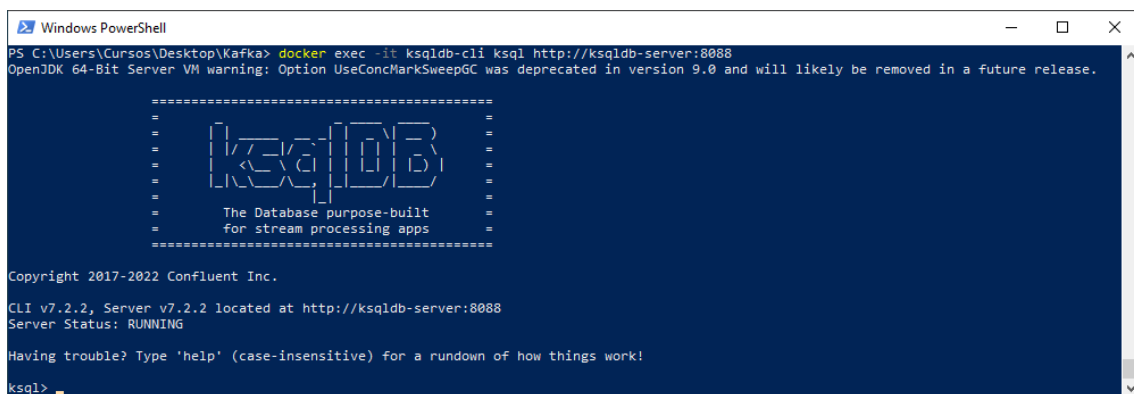
2. CONSULTA DE DATOS AGREGADOS

La información procesada por el sistema se puede consultar de dos maneras diferentes, en tiempo real mediante KsqlDB, o a partir de la información que se almacena de manera persistente en la base de datos MongoDB.

KSQLDBL

Para consultar datos agregados deberemos acceder de manera interactiva a una instancia de KsqlDB a partir del contenedor ksqldb-cli utilizando el siguiente comando:

```
docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
```



```
Windows PowerShell
PS C:\Users\Cursos\Desktop\Kafka> docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.

=====
-      [KSQLDB]      -
-      The Database  -
-      purpose-built -
-      for stream    -
-      processing    -
-      apps          -
=====

Copyright 2017-2022 Confluent Inc.

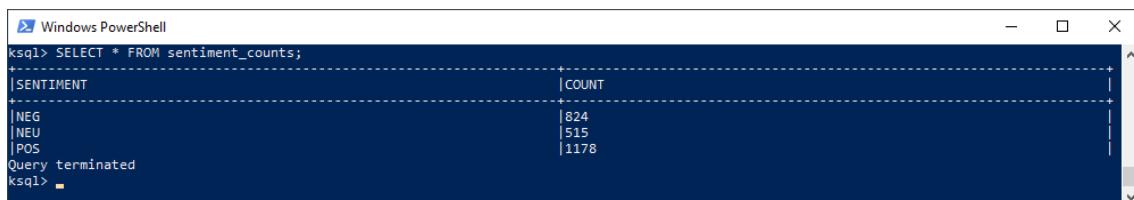
CLI v7.2.2, Server v7.2.2 located at http://ksqldb-server:8088
Server Status: RUNNING

Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!

ksql>
```

Gracias a la tabla que se ha creado durante el despliegue de la aplicación se puede realizar una consulta en tiempo real que devuelve los datos que se encuentran almacenados en la misma con un recuento actualizado de los sentimientos de los mensajes que están presentes en el stream.

```
SELECT * FROM sentiment_counts;
```



```
Windows PowerShell
ksql> SELECT * FROM sentiment_counts;
=====
|SENTIMENT|COUNT|
=====
|NEG      |824   |
|NEU      |515   |
|POS      |1178  |
=====
Query terminated
ksql>
```

Por supuesto se pueden crear tantas tablas como se deseen, como la siguiente para consultar además del recuento total, el average score.

```
CREATE TABLE processed_sentiment_avg_score AS
SELECT SENTIMENT,
       COUNT(*) AS count,
       AVG(SENTIMENT_SCORE) AS average_score
FROM tweets_processed_stream
GROUP BY SENTIMENT;  EMIT CHANGES;
```

```

Windows PowerShell
ksql>
ksql> CREATE TABLE processed_sentiment_avg_score AS
> SELECT SENTIMENT,
> COUNT(*) AS count,
> AVG(SENTIMENT_SCORE) AS average_score
> FROM tweets_processed_stream
> GROUP BY SENTIMENT;

Message
-----
Created query with ID CTAS_PROCESSED_SENTIMENT_AVG_SCORE_9
ksql>
ksql> SELECT * FROM PROCESSED_SENTIMENT_AVG_SCORE;
-----
|SENTIMENT|COUNT|AVERAGE_SCORE|
-----|-----|-----|
|NEG|6|0.9222197532653809|
|NEU|2|0.6268317699432373|
|POS|7|0.8648605942726135|
Query terminated
ksql>

```

MONGODB Y PHP

Como KsqlDB es menos flexible y más complejo a la hora de hacer consultas agregadas, en el servidor PHP se ha creado una aplicación web para poder consultar los datos almacenados en MongoDB, posibilitando además de consultar la cantidad de tweets por categoría de sentimiento, el total de tweets almacenados, el sentimiento promedio o el porcentaje de cada tipo.

The screenshot shows a web browser at localhost:8000 displaying a dashboard titled "Número de tweets por sentimiento". It lists the number of messages for each sentiment: POS: 2081, NEU: 908, and NEG: 1787. Below this is a section titled "Estadísticas" showing overall statistics: Total de mensajes almacenados: 4776, Sentimiento promedio: 0.061557788944724, and percentages for positive (43.57%), neutral (19.01%), and negative (37.42%) messages. The bottom section, "Revisión de tweets para cada sentimiento", includes a dropdown menu set to "POS" and a "Mostrar mensajes" button.

También se ha incluido la opción de visualizar, utilizando un desplegable, el texto completo de cada tweet según el sentimiento.

This screenshot shows the "Revisión de tweets para cada sentimiento" section of the application. The dropdown menu is still set to "POS", and the "Mostrar mensajes" button has been clicked. The page displays a list of tweets categorized as positive. The tweets include various topics such as "Love Speculative Fiction?", a contest announcement from @Biohazzards, a recommendation for the TV show "Lost" from @HistoryBlueBook, a travel tip about hooks from Amazon, a quiz announcement, and a book recommendation "More than Blood Reveals".

Por otro lado, un usuario avanzado podría también lanzar cualquier tipo de consulta sobre la propia base de datos MongoDB que, aunque sea un poco más complejo, es mucho más potente y no depende del número de opciones implementadas en el servidor web. Por ejemplo:

Número de documentos por sentimiento:

```
my_tweets> db.kafka_tweets.aggregate([
...   { $group: { _id: "$sentiment", count: { $sum: 1 } } }
... ])
[
  { _id: 'NEU', count: 539 },
  { _id: 'NEG', count: 850 },
  { _id: 'POS', count: 1228 }
]
```

Palabras de más de 4 caracteres que aparecen con más frecuencia en tweets positivos:

```
my_tweets> db.kafka_tweets.aggregate([
...   { $match: { sentiment: "POS" } },
...   { $project: { words: { $split: ["$msg", " "] } } },
...   { $unwind: "$words" },
...   {
...     $match: {
...       "words": { $regex: /^\\w{4,}$/ }
...     }
...   },
...   { $group: { _id: "$words", count: { $sum: 1 } } },
...   { $sort: { count: -1 } },
...   { $limit: 3 }
... ])
[
  { _id: 'this', count: 245 },
  { _id: 'with', count: 156 },
  { _id: 'that', count: 128 }
]
```

Porcentaje de ocurrencias de cada sentimiento:

```
my_tweets> db.kafka_tweets.aggregate([
...   {
...     $group: {
...       _id: null,
...       totalTweets: { $sum: 1 },
...       sentimentCounts: {
...         $push: {
...           sentiment: "$sentiment",
...           count: 1
...         }
...       }
...     }
...   },
...   {
...     $unwind: "$sentimentCounts"
...   },
...   {
...     $group: {
...       _id: "$sentimentCounts.sentiment",
...       totalTweets: { $first: "$totalTweets" },
...       count: { $sum: "$sentimentCounts.count" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       sentiment: "$_id",
...       percentage: { $multiply: [{ $divide: ["$count", "$totalTweets"] }, 100] }
...     }
...   }
... ])
[
  { sentiment: 'NEU', percentage: 19.725940089228807 },
  { sentiment: 'POS', percentage: 46.367112810707454 },
  { sentiment: 'NEG', percentage: 33.90694710006373 }
]
```

Puntuación promedio por sentimiento:

```
my_tweets> db.kafka_tweets.aggregate([
...   { $group: { _id: "$sentiment", avgScore: { $avg: "$sentiment_score" } } }
... ])
[
  { _id: 'NEG', avgScore: 0.9013392333846775 },
  { _id: 'POS', avgScore: 0.9308948366686082 },
  { _id: 'NEU', avgScore: 0.7575483297264174 }
]
```

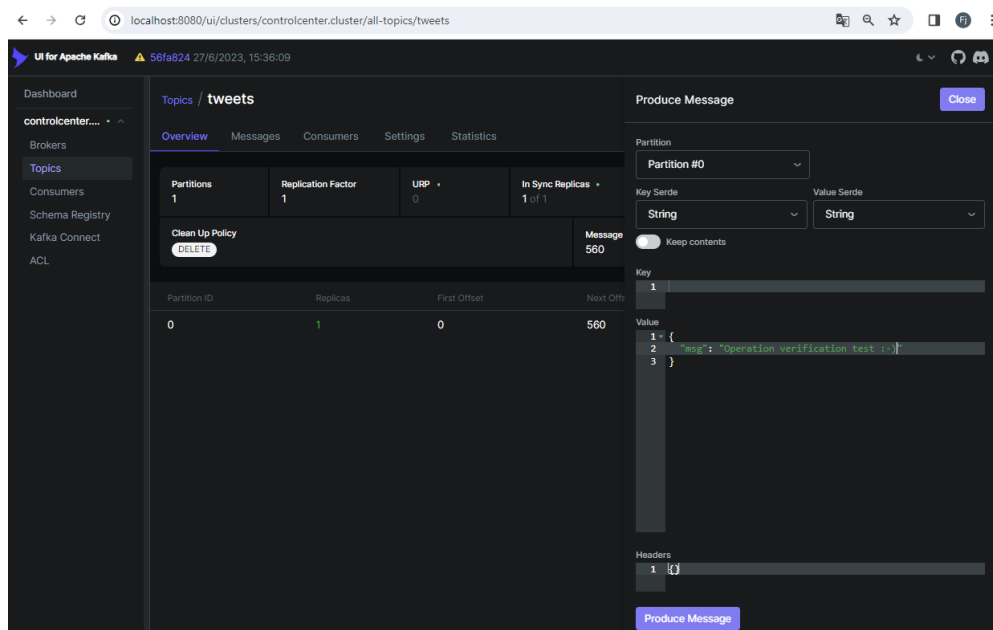
Mensajes con más caracteres para cada sentimiento:

```
my_tweets> db.kafka_tweets.aggregate([
...   { $group: { _id: "$sentiment", longestMessage: { $max: { $strlenCP: "$msg" } } } }
... ])
[
  { _id: 'NEG', longestMessage: 298 },
  { _id: 'POS', longestMessage: 300 },
  { _id: 'NEU', longestMessage: 296 }
]
my_tweets>
```

3. PRODUCCIÓN DE MENSAJES

La producción de mensajes por defecto se realiza desde el contenedor kafka-producer a partir del archivo tweets.csv que se encuentra en el directorio kproducer del host anfitrión. Si se desea introducir otra fuente de datos sin necesidad de realizar ninguna modificación en el código fuente de la aplicación Python que lo ejecuta, sólo es necesario introducir otro csv con el mismo nombre y que en su cuarta columna incluya el texto del mensaje (también hay que tener en cuenta que el algoritmo de machine Learning utilizado en el consumidor no puede procesar aquellos mensajes que superen 128 tokens, por lo que sería necesario limpiar el csv para evitar ejecuciones inesperadas).

Por otro lado, desde la aplicación UI for Apache Kafka se puede acceder a cualquiera de los topics y producir un mensaje con el único requisito de introducir el cuerpo del mensaje con el formato que en la siguiente ilustración se muestra, siendo el campo clave optativo.



583

0

25/3/2024, 20:48:19

{ "msg": "Operation verification test :-)" }

Key

Value

Headers

Timestamp

25/3/2024, 20:48:19

Timestamp type: CREATE_TIME

Key Serde

Size: 0 Bytes

Value Serde

Fallback

Size: 45 Bytes

1

"msg": "Operation verification test :-)"

Posteriormente y teniendo en cuenta el flujo de la información del sistema, se puede comprobar en el topic tweets_processed cómo el mensaje ha sido procesado por el analizador de sentimientos y se ha encolado por este topic, presentando ahora los campos sentiment y score.

←

→

↺

localhost:8080/ui/clusters/controlcenter.cluster/all-topics/tweets_processed/messages?filterQueryType=STRING_CONTAINS&attempt=3&li...

🔍

☆

📱

🌐

⋮

UI for Apache Kafka

56fa824 27/6/2023, 15:36:09

Dashboard

controlcenter....

Brokers

Topics

Consumers

Schema Registry

Kafka Connect

ACL

Topics / tweets_processed

Produce Message

Overview

Messages

Consumers

Settings

Statistics

Seek Type

Offset

Offset

Partitions

All items are selected.

Key Serde

String

Value Serde

SchemaRegistry

Clear all

Submit

Q Operation

+ Add Filters

Done

1766 ms

434 KB

2503 messages consumed

Offset

Partition

Timestamp

Key

Preview

Value

Preview

583

0

25/3/2024, 20:48:20

{ "msg": "Operation verification test :-)", "sentimen...

Key

Value

Headers

Timestamp

25/3/2024, 20:48:20

Timestamp type: CREATE_TIME

Key Serde

Size: 0 Bytes

Value Serde

Fallback

Size: 101 Bytes

1

"msg": "Operation verification test :-)",

2

"sentiment": "NEU",

3

"sentiment_score": 0.7160913348197937

Las operaciones de consulta y producción de mensajes también se podrían realizar desde Control Center de Confluent, aunque a mi parecer en este caso su uso es menos intuitivo.

UI for Apache Kafka

Control Center

localhost:9021/clusters/f29f5mUvRDa5NfwWOwl-UQ/management/topics/tweets/message-viewer

CONFLUENT

HOME > CONTROLCENTER.CLUSTER > TOPICS >

Cluster overview

Brokers

Topics

Connect

ksqldb

Consumers

Replicators

Cluster settings

Health+

tweets

Overview

Messages

Schema

Configuration

Producers

Bytes in/sec 137

Consumers

Bytes out/sec 1.99K

Message fields

topic

partition

offset

timestamp

timestampType

headers

key

value

msg

Filter by keyword

Jump to offset

offset

+ Produce a new message to this topic

Value

Header

Key

1

{

2

"msg": "think you can hurt me? i eont fallow u. "

3

}

{ "msg": "Heard this on GTA V radio... my fck so dirty and grimy . youtube.com/watch?v=r1vFCw..." }

Partition: 0

Offset: 9745

Timestamp: 1711392342922

{ "msg": "Tried GTA V... It's bloody expansive. Nice road detail; incl. camber and bumps and dips... BU..." }

Partition: 0

Offset: 9744

Timestamp: 1711392340920

6