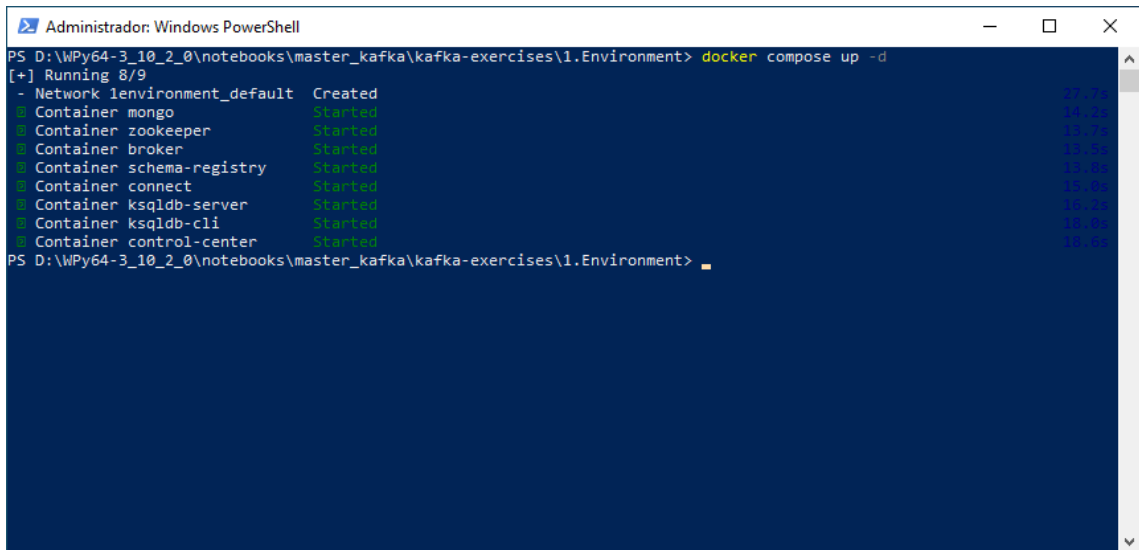


# Manual de uso

El proyecto se ha desarrollado en un sistema Windows, por lo que los comandos que se ejecutan durante el despliegue pueden variar ligeramente si se despliega en un sistema Ubuntu.

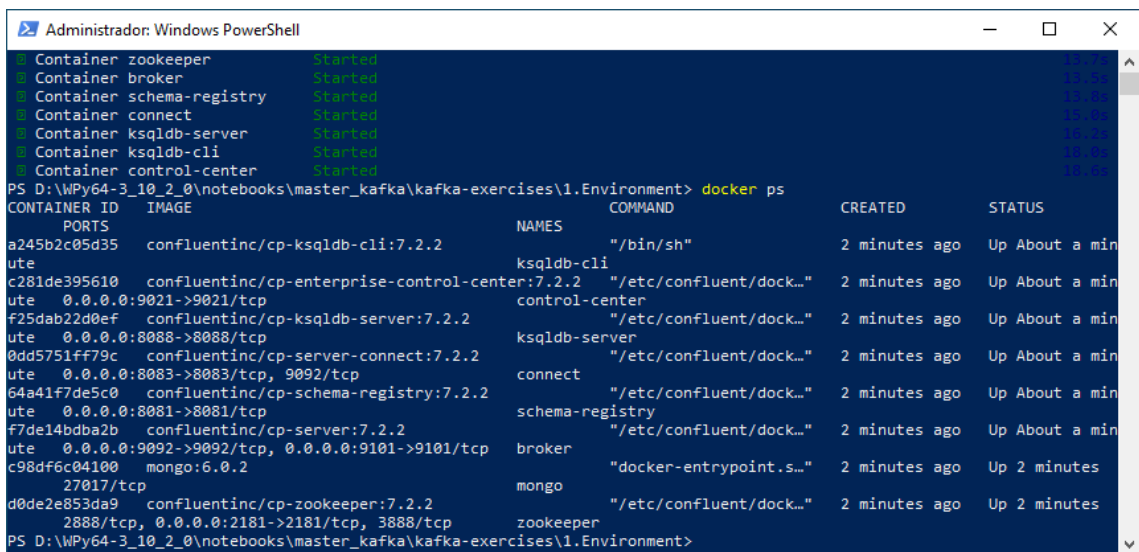
## Levantar contenedores docker

1. Abrir una terminal donde se encuentre el `docker-compose.yml` para levantar el entorno y se ejecuta el comando **docker compose up -d**

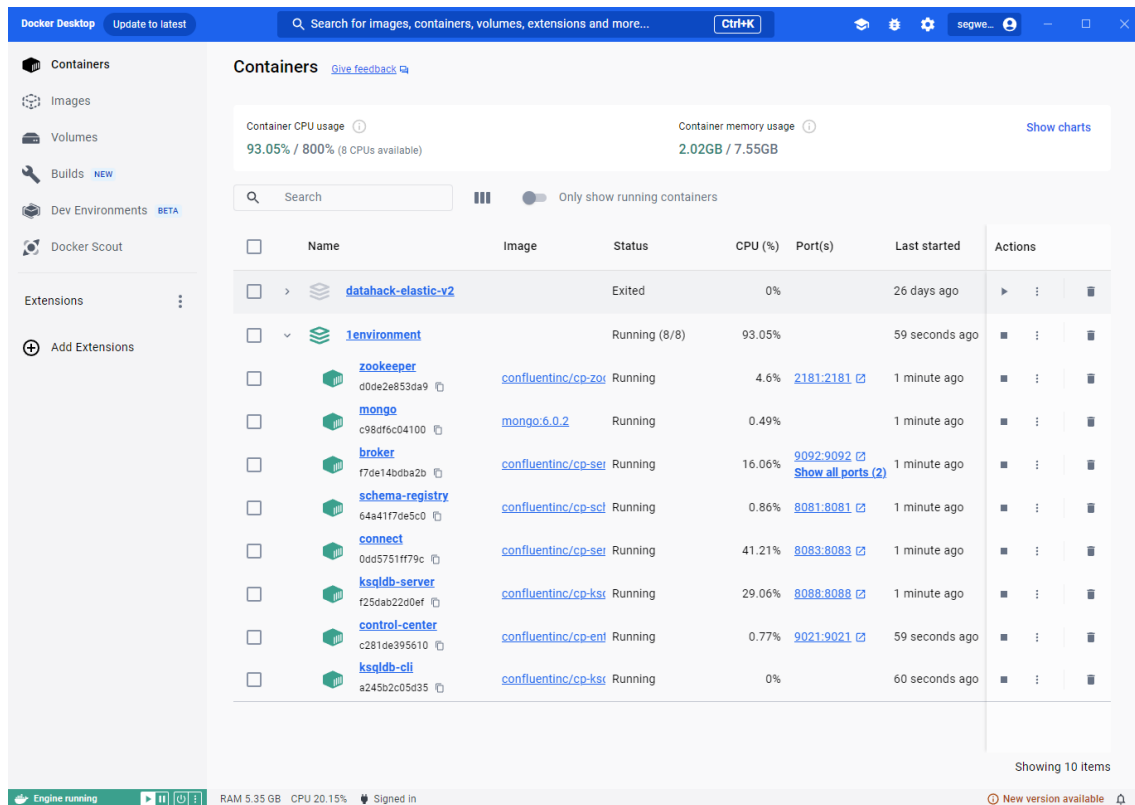


```
Administrador: Windows PowerShell
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment> docker compose up -d
[+] Running 8/9
 - Network lenvironment_default Created
 - Container mongo Started
 - Container zookeeper Started
 - Container broker Started
 - Container schema-registry Started
 - Container connect Started
 - Container ksqldb-server Started
 - Container ksqldb-cli Started
 - Container control-center Started
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment>
```

2. Para obtener el nombre y el puerto del bróker de Kafka dentro del entorno se ejecuta en consola el comando **docker ps**. En docker-desktop también permite obtener esta información:



```
Administrador: Windows PowerShell
CONTAINER ID        IMAGE                                     NAMES                COMMAND                CREATED            STATUS
a245b2c05d35       confluentinc/cp-ksqldb-cli:7.2.2       ksqldb-cli          "/bin/sh"              2 minutes ago     Up About a min
ute
c281de395610       confluentinc/cp-enterprise-control-center:7.2.2 control-center       "/etc/confluent/dock... 2 minutes ago     Up About a min
ute
f25dab22d0ef       confluentinc/cp-ksqldb-server:7.2.2    ksqldb-server       "/etc/confluent/dock... 2 minutes ago     Up About a min
ute
0dd5751ff79c       confluentinc/cp-server-connect:7.2.2   connect             "/etc/confluent/dock... 2 minutes ago     Up About a min
ute
64a41f7de5c0       confluentinc/cp-schema-registry:7.2.2  schema-registry     "/etc/confluent/dock... 2 minutes ago     Up About a min
ute
f7de14b0ba2b       confluentinc/cp-server:7.2.2           broker              "/etc/confluent/dock... 2 minutes ago     Up About a min
ute
c98df6c04100       mongo:6.0.2                             mongo               "docker-entrypoint.s... 2 minutes ago     Up 2 minutes
d0de2e853da9       confluentinc/cp-zookeeper:7.2.2        zookeeper           "/etc/confluent/dock... 2 minutes ago     Up 2 minutes
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment>
```



## Consola del bróker

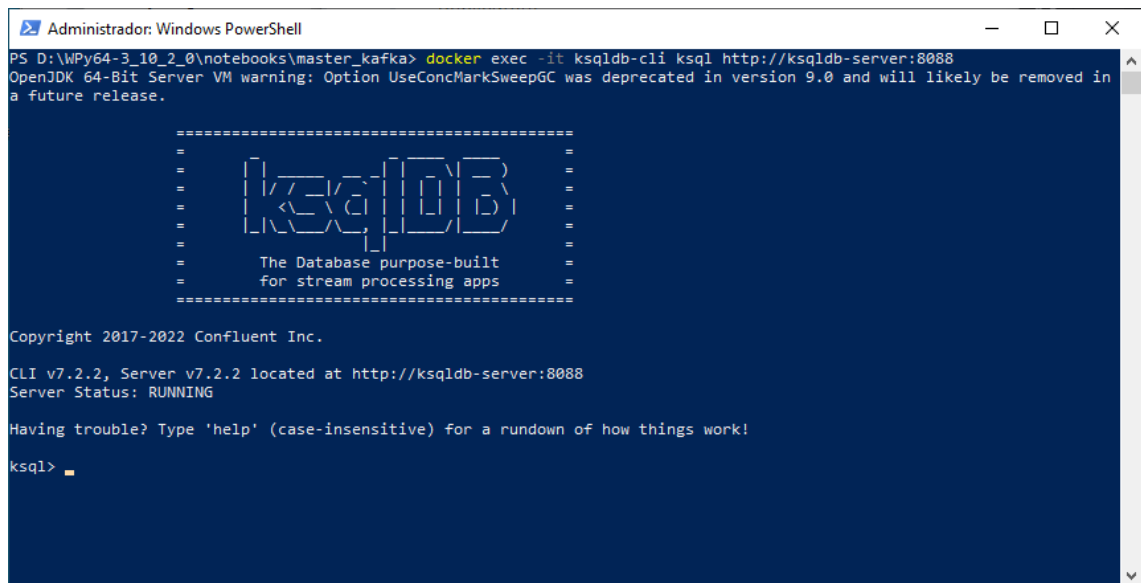
3. Habilitar una consola interactiva dentro del bróker mediante el comando **docker exec -it broker /bin/bash**

```
@broker:~
❏ Container broker Started
❏ Container schema-registry Started
❏ Container connect Started
❏ Container ksqldb-server Started
❏ Container ksqldb-cli Started
❏ Container control-center Started
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
PORTS
a245b2c05d35   confluentinc/cp-ksqldb-cli:7.2.2   "/bin/sh"               2 minutes ago Up About a min
ute
c281de395610   confluentinc/cp-enterprise-control-center:7.2.2 "/etc/confluent/dock... 2 minutes ago Up About a min
ute 0.0.0.0:9021->9021/tcp
f25dab22d0ef   confluentinc/cp-ksqldb-server:7.2.2 "/etc/confluent/dock... 2 minutes ago Up About a min
ute 0.0.0.0:8088->8088/tcp
0dd5751ff79c   confluentinc/cp-server-connect:7.2.2 "/etc/confluent/dock... 2 minutes ago Up About a min
ute 0.0.0.0:8083->8083/tcp, 9092/tcp
64a41f7de5c0   confluentinc/cp-schema-registry:7.2.2 "/etc/confluent/dock... 2 minutes ago Up About a min
ute 0.0.0.0:8081->8081/tcp
f7de14bdba2b   confluentinc/cp-server:7.2.2       "/etc/confluent/dock... 2 minutes ago Up About a min
ute 0.0.0.0:9092->9092/tcp, 0.0.0.0:9101->9101/tcp
c98df6c04100   mongo:6.0.2                        "docker-entrypoint.s... 2 minutes ago Up 2 minutes
27017/tcp
d0de2e853da9   confluentinc/cp-zookeeper:7.2.2   "/etc/confluent/dock... 2 minutes ago Up 2 minutes
2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment> docker exec -it broker /bin/bash
[appuser@broker ~]$
```

NOTA: Para ver la ubicación de los topics que se van creando se puede ejecutar el comando **ls -l /var/lib/kafka/data**

## Consola de ksqlDB

4. Abrir otra consola interactiva para ksqlDB ejecutando el comando  
**docker exec -it ksqldb-cli ksql http://ksqldb-server:8088**



```
Administrador: Windows PowerShell
PS D:\WPY64-3_10_2_0\notebooks\master_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.

=====
=                                     =
=   [K] [S] [Q] [L] [D] [B]         =
=   [K] [S] [Q] [L] [D] [B]         =
=                                     =
=   The Database purpose-built       =
=   for stream processing apps       =
=                                     =
=====

Copyright 2017-2022 Confluent Inc.

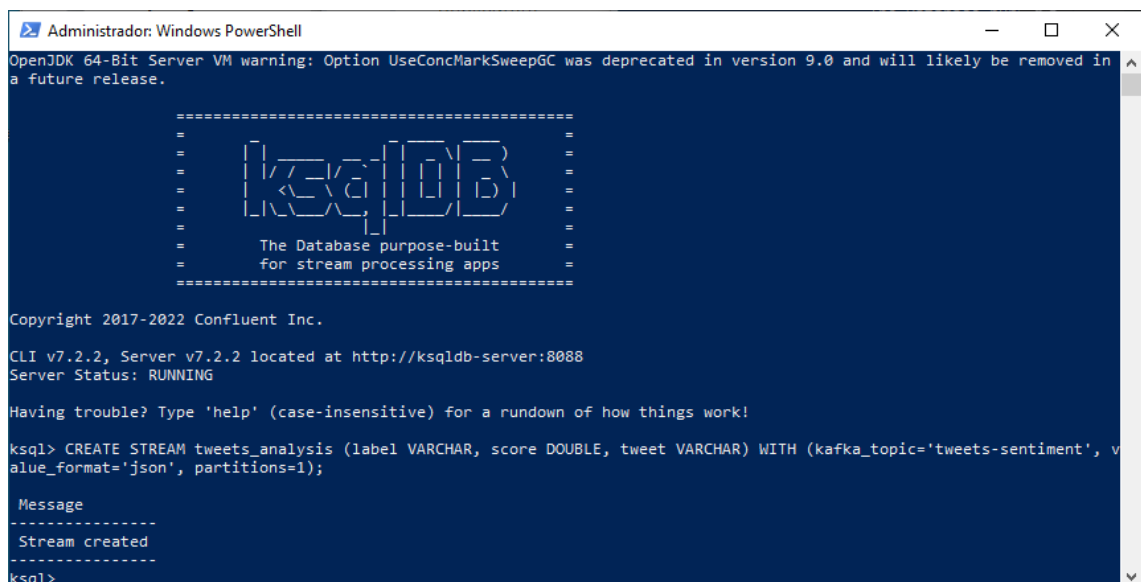
CLI v7.2.2, Server v7.2.2 located at http://ksqldb-server:8088
Server Status: RUNNING

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

ksql>
```

5. En la consola interactiva de ksqlDB se crea el stream por medio del cual se van a introducir los registros en ksqlDB. Para ello, se ejecuta el comando:

**CREATE STREAM tweets\_analysis (label VARCHAR, score DOUBLE, tweet VARCHAR) WITH (kafka\_topic='tweets-sentiment', value\_format='json', partitions=1);**



```
Administrador: Windows PowerShell
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.

=====
=                                     =
=   [K] [S] [Q] [L] [D] [B]         =
=   [K] [S] [Q] [L] [D] [B]         =
=                                     =
=   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> CREATE STREAM tweets_analysis (label VARCHAR, score DOUBLE, tweet VARCHAR) WITH (kafka_topic='tweets-sentiment', value_format='json', partitions=1);

Message
-----
Stream created
-----

ksql>
```

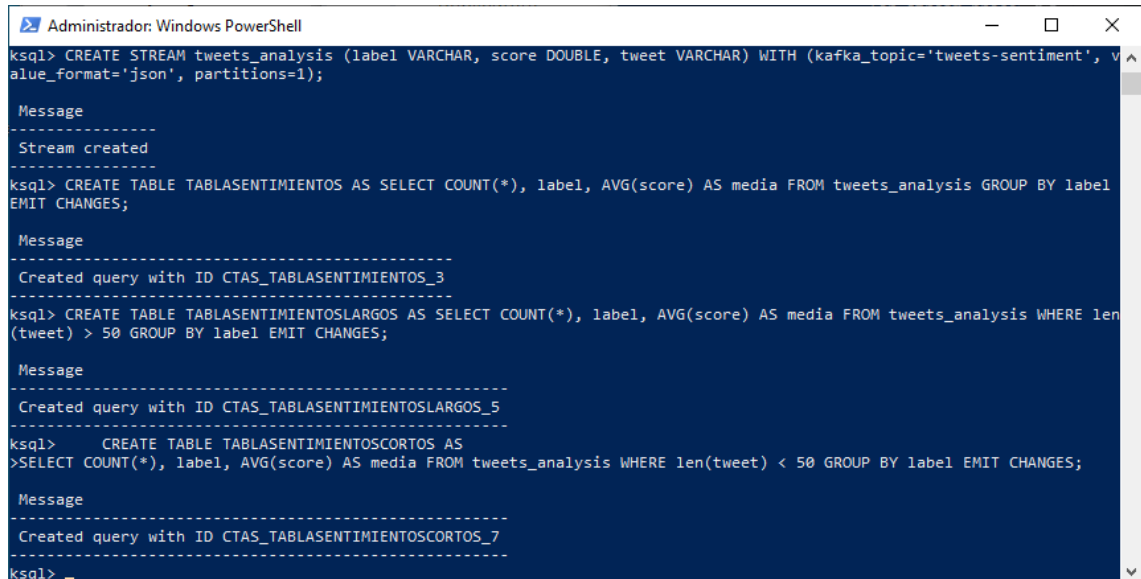
6. En la consola interactiva de ksqlDB se crea una tabla para contabilizar los registros positivos, negativos y neutros.

**CREATE TABLE TABLASENTIMIENTOS AS SELECT COUNT(\*), label, AVG(score) AS media FROM tweets\_analysis GROUP BY label EMIT CHANGES;**

7. En la consola interactiva de ksqlDB se crean dos tablas más, una para mensajes largos y otras para mensajes cortos (50 caracteres). Los mensajes cortos pueden resultar ambiguos si no están en contexto. El número de caracteres se puede definir en el comando de creación de las tablas

**CREATE TABLE TABLASENTIMIENTOSLARGOS AS SELECT COUNT(\*), label, AVG(score) AS media FROM tweets\_analysis WHERE `len(tweet) > 50` GROUP BY label EMIT CHANGES;**

**CREATE TABLE TABLASENTIMIENTOSCORTOS AS SELECT COUNT(\*), label, AVG(score) AS media FROM tweets\_analysis WHERE `len(tweet) < 50` GROUP BY label EMIT CHANGES;**



```
ksql> CREATE STREAM tweets_analysis (label VARCHAR, score DOUBLE, tweet VARCHAR) WITH (kafka_topic='tweets-sentiment', value_format='json', partitions=1);

Message
-----
Stream created

ksql> CREATE TABLE TABLASENTIMIENTOS AS SELECT COUNT(*), label, AVG(score) AS media FROM tweets_analysis GROUP BY label EMIT CHANGES;

Message
-----
Created query with ID CTAS_TABLASENTIMIENTOS_3

ksql> CREATE TABLE TABLASENTIMIENTOSLARGOS AS SELECT COUNT(*), label, AVG(score) AS media FROM tweets_analysis WHERE len(tweet) > 50 GROUP BY label EMIT CHANGES;

Message
-----
Created query with ID CTAS_TABLASENTIMIENTOSLARGOS_5

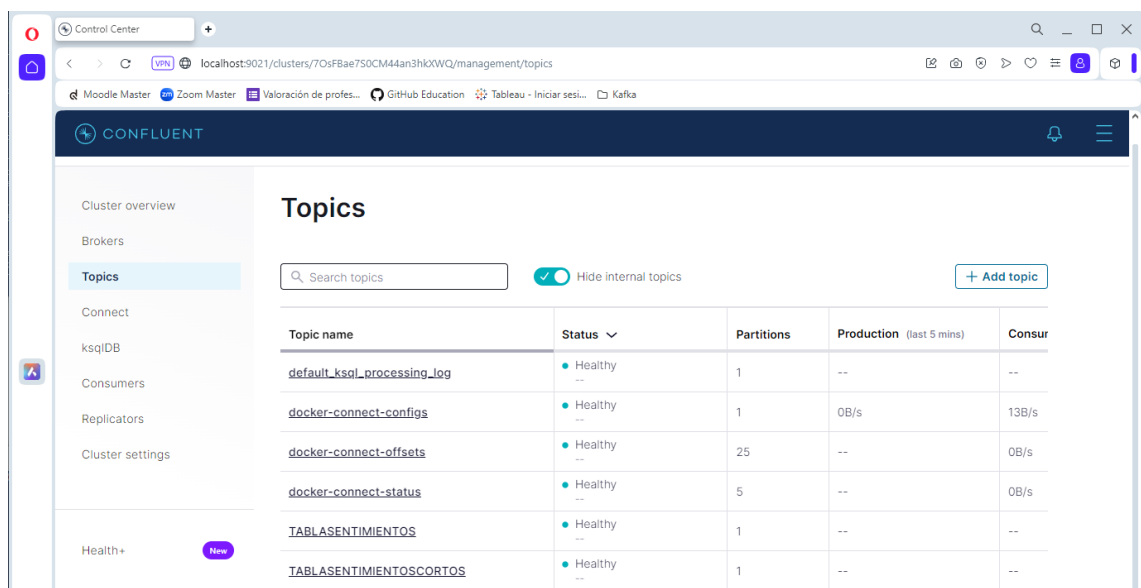
ksql> CREATE TABLE TABLASENTIMIENTOSCORTOS AS
>SELECT COUNT(*), label, AVG(score) AS media FROM tweets_analysis WHERE len(tweet) < 50 GROUP BY label EMIT CHANGES;

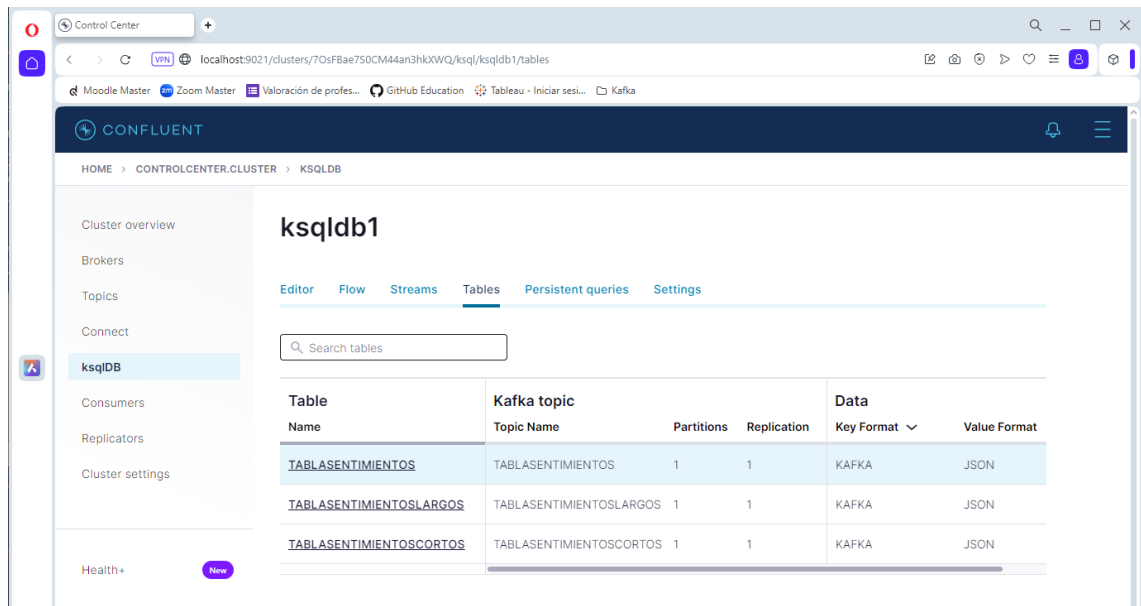
Message
-----
Created query with ID CTAS_TABLASENTIMIENTOSCORTOS_7

ksql>
```

## Interfaz Control Center

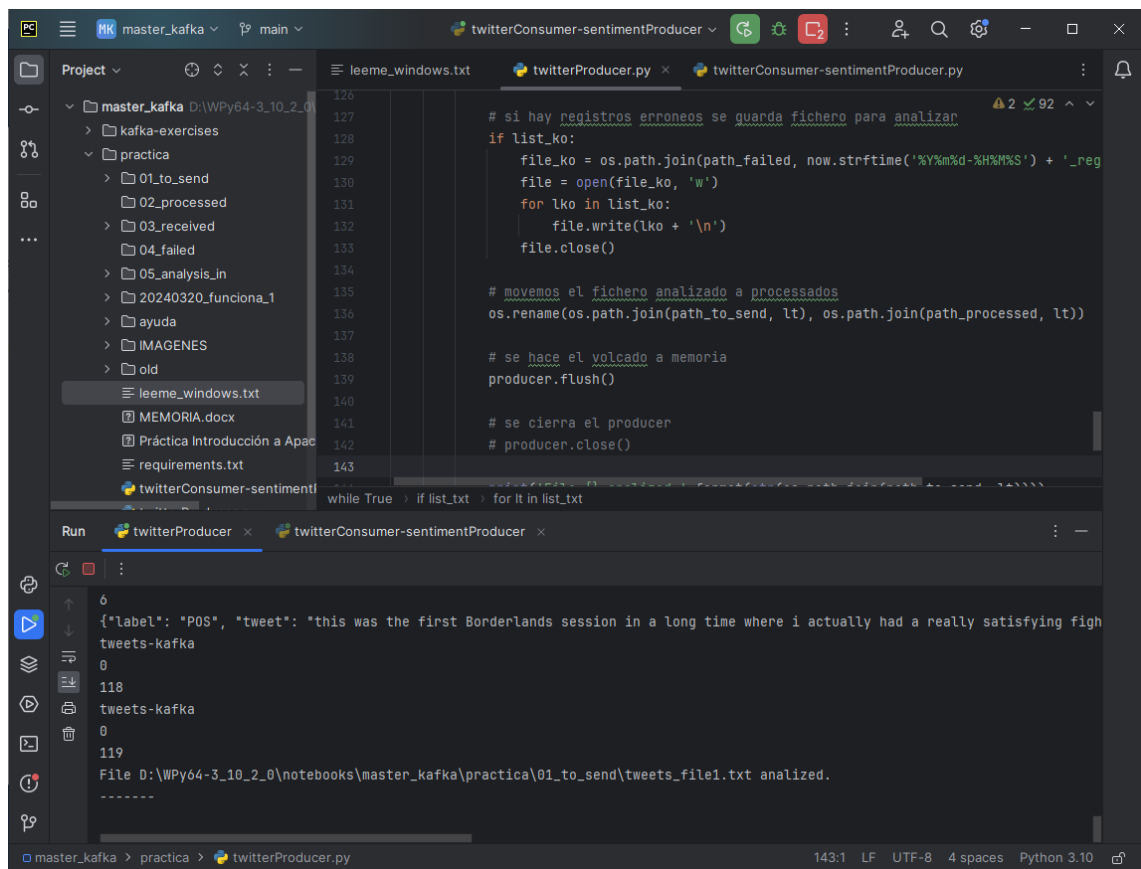
8. Se comprueba el estado de lo que se ha creado hasta el momento abriendo un navegador y escribiendo la siguiente ruta <http://localhost:9021/>



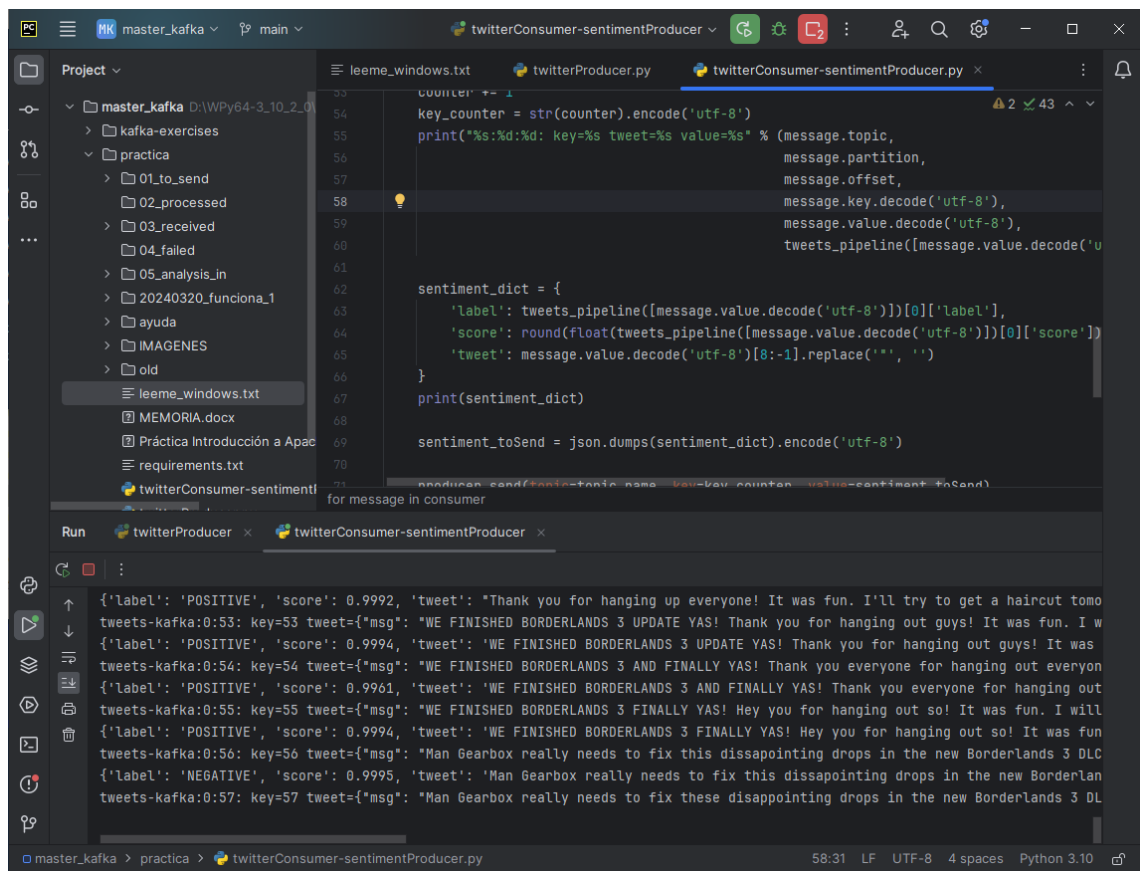


## Producer y Consumer en Python

- Se abre una consola en la ruta donde se encuentra el script Python del producer y se ejecuta el comando **python twitterProducer.py** (en Ubuntu **python3 twitterProducer.py**)



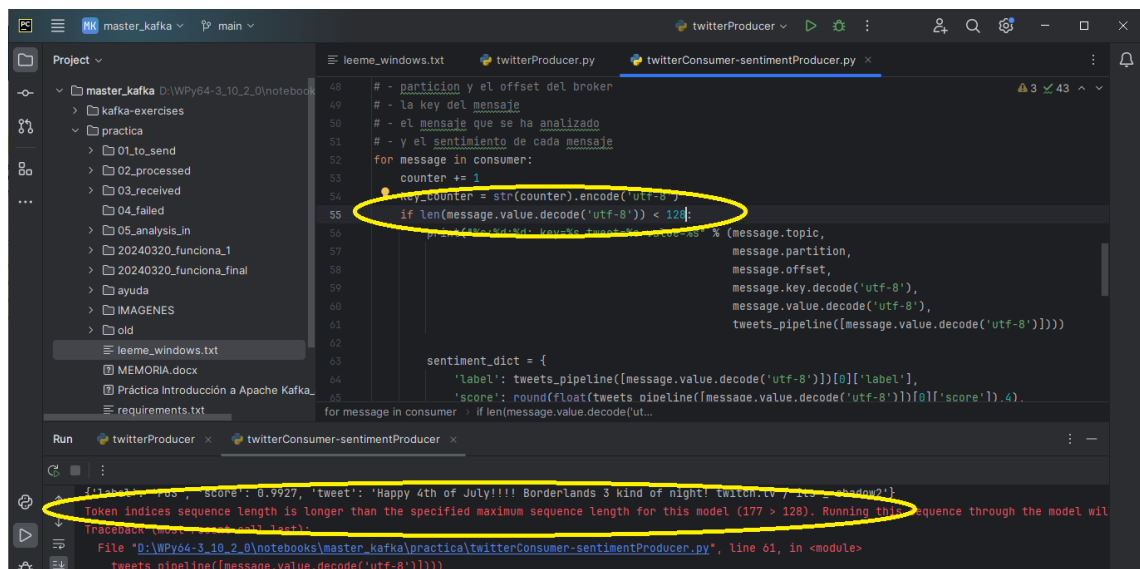
## python twitterConsumer-ksqldbProducer.py (en Ubuntu python3 twitterConsumer-ksqldbProducer.py)



```
53 counter += 1
54 key_counter = str(counter).encode('utf-8')
55 print("%s:%d:%d: key=%s tweet=%s value=%s" % (message.topic,
56                                             message.partition,
57                                             message.offset,
58                                             message.key.decode('utf-8'),
59                                             message.value.decode('utf-8'),
60                                             tweets_pipeline([message.value.decode('u
61
62 sentiment_dict = {
63     'label': tweets_pipeline([message.value.decode('utf-8')])[0]['label'],
64     'score': round(float(tweets_pipeline([message.value.decode('utf-8')])[0]['score'])),
65     'tweet': message.value.decode('utf-8')[8:-1].replace('\n', ' ')
66 }
67 print(sentiment_dict)
68
69 sentiment_toSend = json.dumps(sentiment_dict).encode('utf-8')
70
71 producer.send(topic=message.topic, key=key_counter, value=sentiment_toSend)
72 for message in consumer
```

Run

```
{'label': 'POSITIVE', 'score': 0.9992, 'tweet': 'Thank you for hanging up everyone! It was fun. I\'ll try to get a haircut tomo
tweets-kafka:0:53: key=53 tweet={\"msg\": \"WE FINISHED BORDERLANDS 3 UPDATE YAS! Thank you for hanging out guys! It was fun. I w
{'label': 'POSITIVE', 'score': 0.9994, 'tweet': 'WE FINISHED BORDERLANDS 3 UPDATE YAS! Thank you for hanging out guys! It was
tweets-kafka:0:54: key=54 tweet={\"msg\": \"WE FINISHED BORDERLANDS 3 AND FINALLY YAS! Thank you everyone for hanging out everyon
{'label': 'POSITIVE', 'score': 0.9961, 'tweet': 'WE FINISHED BORDERLANDS 3 AND FINALLY YAS! Thank you everyone for hanging out
tweets-kafka:0:55: key=55 tweet={\"msg\": \"WE FINISHED BORDERLANDS 3 FINALLY YAS! Hey you for hanging out so! It was fun. I will
{'label': 'POSITIVE', 'score': 0.9994, 'tweet': 'WE FINISHED BORDERLANDS 3 FINALLY YAS! Hey you for hanging out so! It was fun
tweets-kafka:0:56: key=56 tweet={\"msg\": \"Man Gearbox really needs to fix this disappointing drops in the new Borderlands 3 DLC
{'label': 'NEGATIVE', 'score': 0.9995, 'tweet': 'Man Gearbox really needs to fix this disappointing drops in the new Borderlan
tweets-kafka:0:57: key=57 tweet={\"msg\": \"Man Gearbox really needs to fix these disappointing drops in the new Borderlands 3 DL
```



```
48 # - particion y el offset del broker
49 # - la key del mensaje
50 # - el mensaje que se ha analizado
51 # - y el sentimiento de cada mensaje
52 for message in consumer:
53     counter += 1
54     key_counter = str(counter).encode('utf-8')
55     if len(message.value.decode('utf-8')) < 128:
56         print("%s:%d:%d: key=%s tweet=%s value=%s" % (message.topic,
57                                                         message.partition,
58                                                         message.offset,
59                                                         message.key.decode('utf-8'),
60                                                         message.value.decode('utf-8'),
61                                                         tweets_pipeline([message.value.decode('utf-8')]))
62
63     sentiment_dict = {
64         'label': tweets_pipeline([message.value.decode('utf-8')])[0]['label'],
65         'score': round(float(tweets_pipeline([message.value.decode('utf-8')])[0]['score'])),
66     }
67     for message in consumer:
68         if len(message.value.decode('utf-8')) < 128:
69             sentiment_dict['tweet'] = message.value.decode('utf-8')[8:-1].replace('\n', ' ')
70             sentiment_toSend = json.dumps(sentiment_dict).encode('utf-8')
71             producer.send(topic=message.topic, key=key_counter, value=sentiment_toSend)
72             counter += 1
73             key_counter = str(counter).encode('utf-8')
```

Run

```
{'label': 'POSITIVE', 'score': 0.9927, 'tweet': 'Happy 4th of July!!!! Borderlands 3 Kind of night! twitch.tv / its a shadow21'}
Token indices sequence length is longer than the specified maximum sequence length for this model (177 > 128). Running this sequence through the model will
raise an IndexError: Tensor index out of bounds: index 177 is not in the range [0, 128).
File "D:\WPY64-3.10.2_0\notebooks\master_kafka\practica\twitterConsumer-sentimentProducer.py", line 61, in <module>
    tweets_pipeline([message.value.decode('utf-8')]))
```

Para que se empiecen a procesar mensajes, hay que mover el fichero / los ficheros que se quiera desde la carpeta “ficheros\_pruebas” a la carpeta “01\_to\_send”.

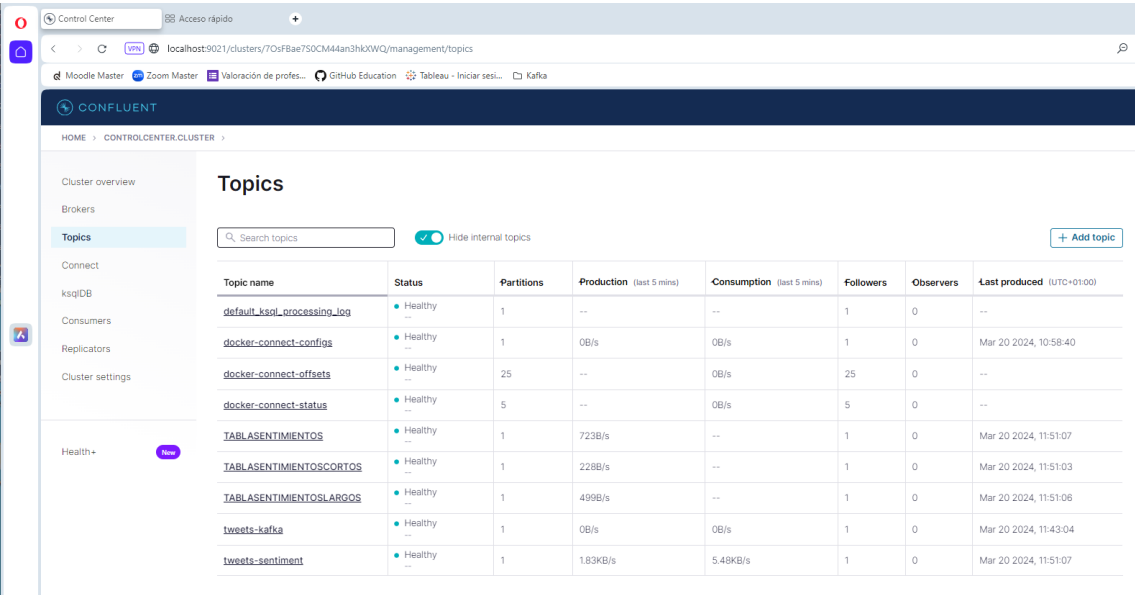
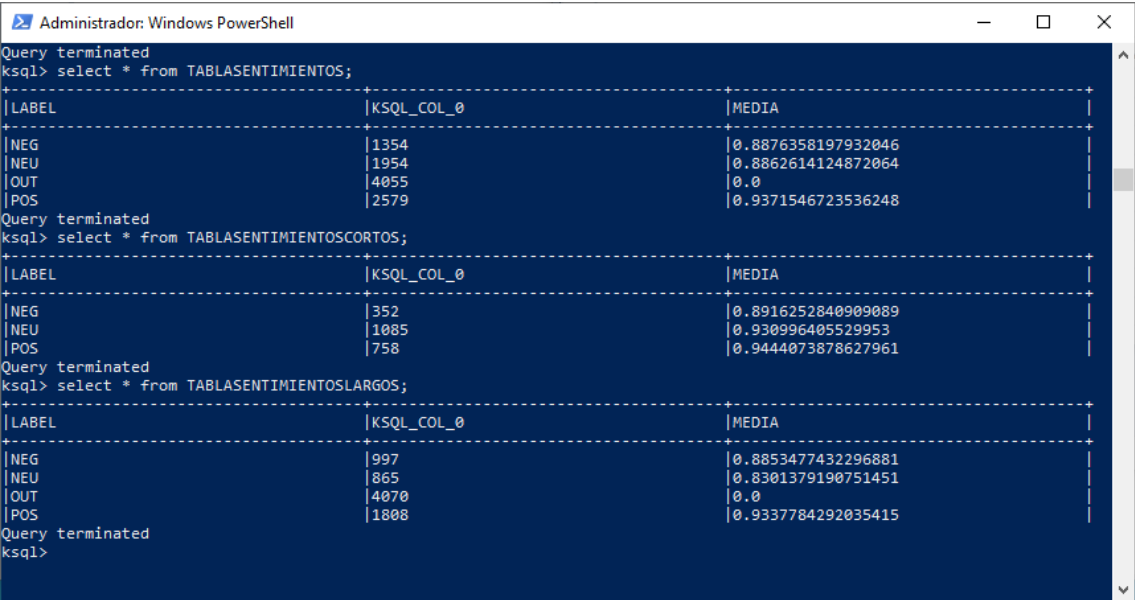
### Visualización de datos en consola de ksqlDB

10. En la consola interactiva de ksqlDB se ejecutan las siguientes consultas para ir monitorizando la evolución de los sentimientos de los tweets analizados.

```
select * from TABLASENTIMIENTOS;

select * from TABLASENTIMIENTOSLARGOS;

select * from TABLASENTIMIENTOSCORTOS;
```



### Finalizar procesos

11. Para detener todos los procesos (para mantenimiento o configuración):
- Ejecutar “Ctrl+C” en las consolas donde se encuentran en ejecución los scripts Python.
  - Escribir “exit” en las consolas interactivas de ksqlDB y del bróker de Kafka.

## Limpieza del despliegue

12. Finalmente, para limpiar el despliegue en la consola interactiva del bróker de Kafka ejecutar los siguientes comandos:

- `docker compose down`
- `docker volumen prune` (también se puede hacer desde docker-desktop).

```
Selecc... Windows PowerShell

PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment> docker compose down

# Removing containers
Container mongo Removed 11.3s
Container control-center Removed 11.7s
Container sqlcli Removed 10.2s
Container sqlserver Removed 3.1s
Container connect Removed 2.8s
Container schema-registry Removed 2.2s
Container broker Removed 13.2s
Container zookeeper Removed 2.1s
Network lenvironment_default Removed 0.0s

PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment>
```

```
Windows PowerShell

PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment> docker volume prune
WARNING! This will remove anonymous local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
94631ad944da5d7e0cc1405d67b8b0cd98fe51de0c6edb18675faa62dc56f61f
6e52b7939347fbf3d8fa95182ff56661fc252ae7f487a4a562030561e833ddf6
4595c53e8fd88879902eb438de48171e13bedb97de2e08abd9381f5ec957a295
cb1ae019679d7c52aef95c59979fa5723efd97cf29dba8d9460f1d57c13e13d
1aa0c64b0ec0433704eb8437ccf97e0856c79ad95283ee019d4b136ed60951c3
f4868f61d967fda05f0eb5418503baf9818cca60e3f67e937a1a5b6dbb22d672
d099b5ad846f5a1832a53baa1021c88d3ff8e4ee52a064529800803c92ffd6b6
db417c4ec77fc845860f85b80e55309230fcae6f6353fe6ed5126b925a77edf
284ef166a1766eb2a6d73110832072a9c596a73c1498fb4d0817fd05570b6fc5
15ca613457588956e1e4114ba0b5960485e23572301d17b050ac3cf2fc0c3b1e
0eb7bc708dea9a497b09c6299e95684cea2fbfe734452bdb98c847de7c854563
d8dc4fb51015a49ccbffc9b26102427081c4c6e34afdc92cfa6f358f8388341e

Total reclaimed space: 5.263GB
PS D:\WPY64-3_10_2_0\notebooks\master_kafka\kafka-exercises\1.Environment>
```

