



MANUAL DE OPERACIÓN

Francisco Javier Fernández

1.	INTRODUCCIÓN.....	2
2.	PREPARACIÓN DE LOS ENTREGABLES.	2
3.	INICIO DEL ENTORNO	2
4.	ACCESO Y VERIFICACIÓN DE SERVICIOS	3
5.	DETENER EL SISTEMA	6

1. INTRODUCCIÓN

En este documento se detallan las instrucciones necesarias para desplegar el sistema a partir del código fuente proporcionado, así como las distintas verificaciones que se pueden realizar para comprobar que todo se está ejecutando correctamente, ya que el sistema está preparado para que todos sus componentes funcionen sin intervención alguna después de desplegar el entorno utilizando Docker Compose.

2. PREPARACIÓN DE LOS ENTREGABLES.









































































Antes de comenzar a trabajar con el sistema es necesario descomprimir el archivo ZIP proporcionado en el que se incluye todo el código fuente y la documentación, como paso previo antes de desplegar los contenedores a partir del archivo docker-compose.yml.

3. INICIO DEL ENTORNO

Para levantar los diferentes contenedores que forman parte del sistema solo es necesario disponer de un equipo con Docker Desktop y ejecutar en el directorio que contiene el fichero docker-compose.yml el siguiente comando:

```
docker compose up -d
```

Después de su ejecución deberíamos ver que todos los contenedores están corriendo en Docker Desktop o en el propio terminal desde el que lanza el comando anterior.

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 kafka		Running (12/12)	327.37%		4 minutes ago	  
<input type="checkbox"/>	 zookeeper 8c6147d9c56d 	confluentinc/cp-zookeeper:7.2.2	Running	0.14%	2181:2181 	4 minutes ago	  
<input type="checkbox"/>	 mongodb 62f00f428a88 	mongo:6.0.2	Running	0.68%		4 minutes ago	  
<input type="checkbox"/>	 php 6ec4e0cd5d5e 	fiavif/php:latest	Running	0%	8000:80 	4 minutes ago	  
<input type="checkbox"/>	 schema-registry 4df44e6445a3 	confluentinc/cp-schema-registry:7.2.2	Running	3.55%	8081:8081 	4 minutes ago	  
<input type="checkbox"/>	 consumer 13c4b1db62bc 	fiavif/kafka-consumer:latest	Running	51.63%		4 minutes ago	  
<input type="checkbox"/>	 broker 8e9a8b60537c 	confluentinc/cp-server:7.2.2	Running	38.21%	9092:9092  Show all ports (2)	4 minutes ago	  
<input type="checkbox"/>	 producer 23a88263b761 	python:3.9-slim	Running	0.57%		4 minutes ago	  
<input type="checkbox"/>	 ksqldb-cli d6663eb92d29 	confluentinc/cp-ksqldb-cli:7.2.2	Running	4.31%		4 minutes ago	  
<input type="checkbox"/>	 ksqldb-server 63b78d971261 	confluentinc/cp-ksqldb-server:7.2.2	Running	87.17%	8088:8088 	4 minutes ago	  
<input type="checkbox"/>	 ui f2d49ff19eb0 	provectuslabs/kafka-ui:latest	Running	11.64%	8080:8080 	4 minutes ago	  
<input type="checkbox"/>	 control-center 9f87066c03f3 	confluentinc/cp-enterprise-control-cente	Running	13.98%	9021:9021 	4 minutes ago	  
<input type="checkbox"/>	 connect b6b6658caea9 	confluentinc/cp-server-connect:7.2.2	Running	115.49%	8083:8083 	4 minutes ago	  

```
PS C:\Users\Cursos\Desktop\Kafka> docker compose up -d
[+] Running 12/13
 - Network kafka_mynetwork      Created           15.7s
 ✓ Container zookeeper          Started         5.4s
 ✓ Container mongodb            Started         5.4s
 ✓ Container php                Started         6.2s
 ✓ Container broker             Started         6.5s
 ✓ Container schema-registry    Started         8.0s
 ✓ Container kafka-producer      Started         7.9s
 ✓ Container kafka-consumer     Started         7.7s
 ✓ Container connect            Started         9.1s
 ✓ Container ksqldb-server      Started        11.1s
 ✓ Container kafka-ui           Started        10.8s
 ✓ Container ksqldb-cli         Started        12.7s
 ✓ Container control-center     Started        13.0s
```

Hay que tener en cuenta, que una vez que se han levantado todos los contenedores, además de las dependencias que se señalan en el fichero docker compose y como parte del proceso de automatización, en los scripts y comandos de inicio se han establecido tiempos de espera para asegurarnos por ejemplo de que el topic del productor se crea antes de que el consumidor ejecute su código o verificar que el servidor ksqldb está preparado para crear un stream, entre otras. En cualquier caso, en una ventana de aproximadamente 6 minutos deberían encontrarse todos los servicios del sistema operativo.

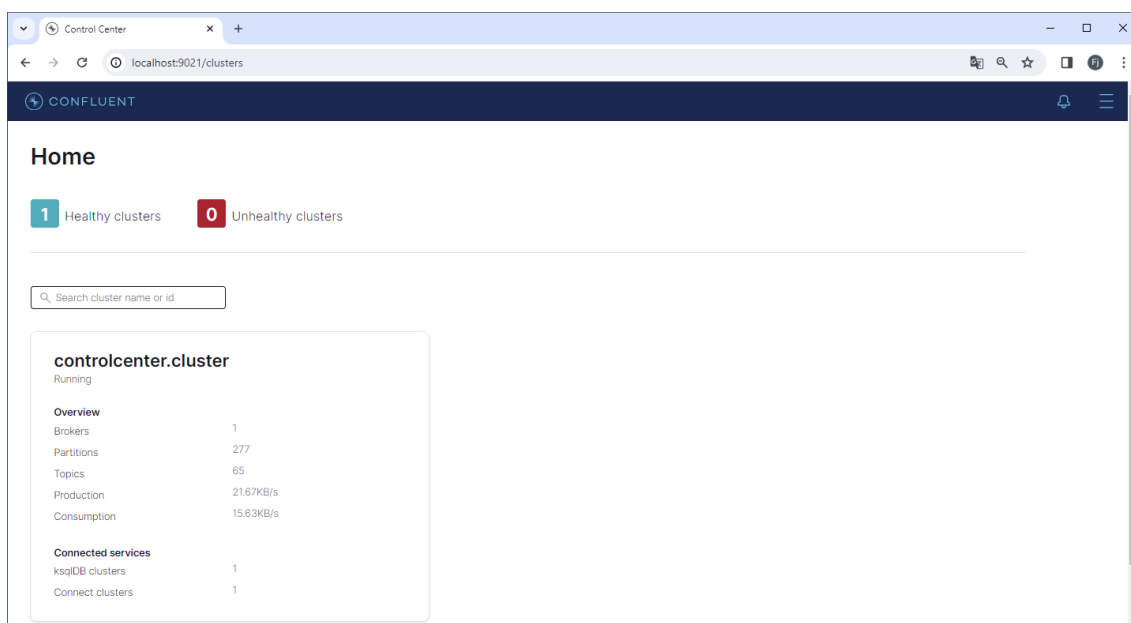
4. ACCESO Y VERIFICACIÓN DE SERVICIOS

Una vez levantado el entorno se puede proceder a verificar que los diferentes componentes se encuentran operativos. La manera más rápida y visual que se propone para realizar estas comprobaciones consiste en la utilización de Control Center de Confluent o UI for Apache Kafka desde un navegador web.

- Control center: <http://localhost:9021>
- UI for Apache Kafka: <http://localhost:8080/>

En este documento utilizaremos Control Center como herramienta para explicar las verificaciones básicas que se deben realizar.

Desde el apartado Home de Control Center disponemos de una visión global del clúster de Kafka, donde se puede comprobar que el mismo está corriendo sin errores, hay producción y consumo de datos o que tenemos servicios conectados, entre otros.



Accediendo a cada una de las secciones de la herramienta, podremos comprobar los topics creados, si tenemos conectores ejecutándose, si ksqldb tiene registrados streams o tablas, o si existen grupos de consumo, entre otros.

The screenshot shows the 'Topics' section of the Confluent Control Center. The left sidebar contains navigation links: Cluster overview, Brokers, Topics (selected), Connect, ksqldb, Consumers, Replicators, and Cluster settings. The main area displays a table of topics. A search bar and a 'Hide internal topics' toggle are at the top. A '+ Add topic' button is in the top right.

Topic name	Status	Partitions	Production (last 5 mins)	Consumption (last 5 mins)	Followers	Offset
default_ksql_processing_log	Healthy	1	--	--	1	0
docker-connect-configs	Healthy	1	0B/s	0B/s	1	0
docker-connect-offsets	Healthy	25	--	0B/s	25	0
docker-connect-status	Healthy	5	--	0B/s	5	0
SENTIMENT_COUNTS	Healthy	1	57B/s	0B/s	1	0
tweets	Healthy	1	132B/s	136B/s	1	0
tweets_processed	Healthy	1	192B/s	385B/s	1	0

The screenshot shows the 'Connectors' section of the Confluent Control Center. The left sidebar is the same as the previous screenshot, with 'Connect' selected. The main area displays a summary of connectors with five large numbers: 1 Total, 1 Running, 0 Degraded, 0 Failed, and 0 Paused. Below the summary is a search bar, a 'Filter by category' dropdown, and '+ Add connector' and 'Upload connector config file' buttons. A table lists the connectors.

Connector name	Status	Category	Plugin name	Topics	Number of tasks
mongo-sink	Running	Sink	MongoSinkConn...	tweets_processed	1

En concreto, deberemos revisar los siguientes elementos:

- **Brokers:** el controlador activo el bróker.1 y ZooKeeper se encuentra conectado. Además, podemos ver estadísticas de escritura en disco y producción y consumo de datos.
- **Topics:** deben estar creados los topics tweets y tweets_processed. Además, cuando se acceda a cada uno de ellos se podrán observar los mensajes procesados. También aparecerán numerosos topics auxiliares que Kafka utiliza para comprobar el estado de las transacciones, el offset del consumer o el estado de los streams, de esquema, etc.
- **Connect:** debe aparecer en ejecución el conector mongo-sink. Debido a que este servicio es uno de los que más tiempo tarda en arrancar, en caso de que no aparezca el conector

después de un máximo de 10 minutos se puede reiniciar el contenedor, o en última instancia, cargar el archivo `mongo_sink_config.json` manualmente.

- **Ksqldb:** en el apartado Streams debe aparecer creado el stream `TWEETS_PROCESSED_STREAM` y en el apartado tablas, la tabla `SENTIMENT_COUNTS`.
- **Consumers:** se visualiza el grupo de consumidores `tweet-group` y `connect-mongo-sink`.

Posteriormente, una vez comprobado que todo lo anterior es correcto, se puede revisar también que los tweets procesados por el algoritmo de machine learning se pueden consultar en tiempo real en KsqlDB.

Para ello, se accede 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.

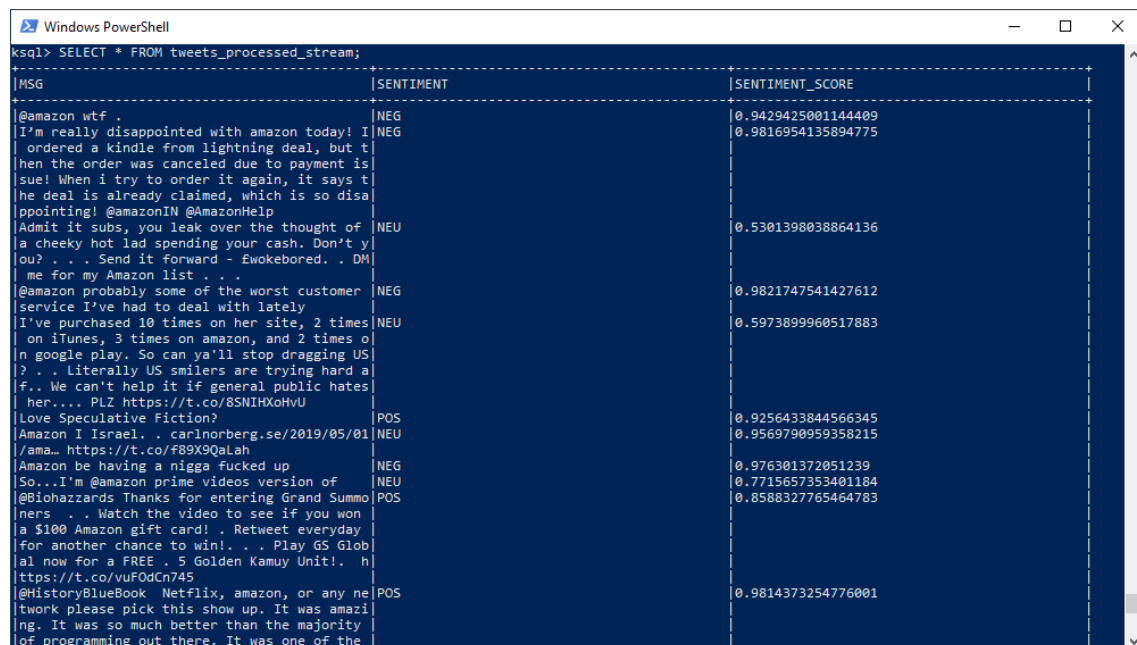
CLI v7.2.2, Server v7.2.2 located at http://ksqldb-server:8088
Server Status: RUNNING

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

ksql>
```

Con la siguiente instrucción se pueden comprobar los datos que el stream está procesando en tiempo real.

```
SELECT * FROM tweets_processed_stream;
```



```
Windows PowerShell
ksql> SELECT * FROM tweets_processed_stream;

=====+-----+-----+
|MSG|SENTIMENT|SENTIMENT_SCORE|
=====+-----+-----+
|@amazon wtf .|NEG|0.9429425001144409|
|I'm really disappointed with amazon today! I|NEG|0.9816954135894775|
|ordered a kindle from lightning deal, but t|
|hen the order was canceled due to payment is|
|sue! When i try to order it again, it says t|
|he deal is already claimed, which is so disa|
|ppointing! @amazonIN @AmazonHelp|
|Admit it subs, you leak over the thought of|NEU|0.5301398038864136|
|a cheeky hot lad spending your cash. Don't y|
|ou? . . . Send it forward - Ewokebored. . DM|
|me for my Amazon list . . .|
|@amazon probably some of the worst customer|NEG|0.9821747541427612|
|service I've had to deal with lately|
|I've purchased 10 times on her site, 2 times|NEU|0.5973899960517883|
|on iTunes, 3 times on amazon, and 2 times o|
|n google play. So can ya'll stop dragging US|
|? . . Literally US smilers are trying hard a|
|f.. We can't help it if general public hates|
|her.... PLZ https://t.co/8SNIHXoHvU|
|Love Speculative Fiction?|POS|0.9256433844566345|
|Amazon I Israel. . carlnorberg.se/2019/05/01|NEU|0.9569790959358215|
|/ama... https://t.co/f89X9QaLah|
|Amazon be having a nigga fucked up|NEG|0.976301372051239|
|So...I'm @amazon prime videos version of|NEU|0.7715657353401184|
|@Biohazzards Thanks for entering Grand Summo|POS|0.8588327765464783|
|ners . . Watch the video to see if you won|
|a $100 Amazon gift card! . Retweet everyday|
|for another chance to win! . . Play GS Glob|
|al now for a FREE . 5 Golden Kamuy Unit!. h|
|https://t.co/vuF0dCn745|
|@HistoryBlueBook Netflix, amazon, or any ne|POS|0.9814373254776001|
|twork please pick this show up. It was amazi|
|ng. It was so much better than the majority|
|of programming out there. It was one of the|
```

Por otro lado, también se va a verificar que el servidor MongoDB se está alimentando correctamente con los datos que se desea que sean persistentes, y que esta información es accesible desde el servidor PHP.

Se accede al contenedor mongodb para comprobar que la base de datos tiene documentos.

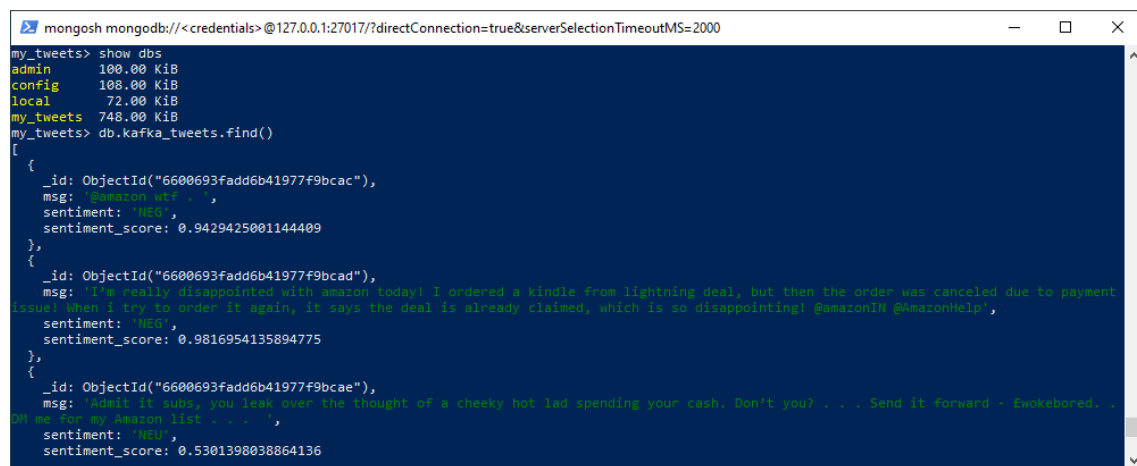
```
docker exec -it mongodb /bin/bash

mongosh --username admin

use my_tweets

db.kafka_tweets.find()
```

De esta manera se puede comprobar que la base de datos está recibiendo el contenido del topic tweets_processed a través con conector de mongo que se ha configurado.

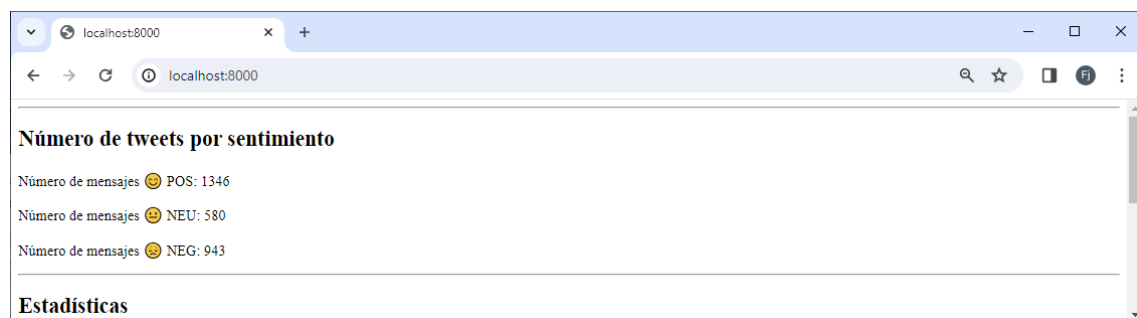


```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

my_tweets> show dbs
admin      100.00 KiB
config    108.00 KiB
local      72.00 KiB
my_tweets  748.00 KiB
my_tweets> db.kafka_tweets.find()
[
  {
    _id: ObjectId("6600693fadd6b41977f9bcac"),
    msg: "Amazon ad",
    sentiment: 'NEG',
    sentiment_score: 0.9429425001144409
  },
  {
    _id: ObjectId("6600693fadd6b41977f9bcad"),
    msg: "I'm really disappointed with amazon today! I ordered a kindle from lightning deal, but then the order was canceled due to payment issue. When I try to order it again, it says the deal is already claimed, which is so disappointing! @amazonIN @amazonHelp",
    sentiment: 'NEG',
    sentiment_score: 0.9816954135894775
  },
  {
    _id: ObjectId("6600693fadd6b41977f9bcae"),
    msg: "Admit it subs, you leak over the thought of a cheeky hot lad spending your cash. Don't you? . . . Send it forward - Ewoketired. . . Oh be for my Amazon list . . .",
    sentiment: 'NEU',
    sentiment_score: 0.5301398038864136
  }
]
```

Por último, a través de la siguiente url podemos verificar que desde el servidor PHP se pueden visualizar los registros de la base de datos:

- PHP: <http://localhost:8000/>



5. DETENER EL SISTEMA

Para detener todo el entorno solo es necesario ejecutar el siguiente comando desde el directorio en el que se encuentre el archivo docker-compose.yml.

```
docker compose down
```