# Kafka con makefile

## **como ejecutar este archivo para correr kafka**

* se debe tener instalado docker Windows
* se debe tener el archivo makefile sin extensión que contiene la configuración con la network, Kafka y el zookeeper
* se abre una terminal en modo Linux, en el teclado:

=> + click derecho: seleccionar Shell de Linux

* se ejecuta en la terminal un comando a la vez:

make network

make zookeeper

make Kafka

de esta manera se tiene corriendo 2 contenedores corriendo Kafka

# Proyectos para los que sirve ejecutar el Kafka de makefile

## enviroscale-p48

se debe modificar en el archivo de configuración properties por local el siguiente:

spring.kafka.bootstrap-servers=localhost:9092

También se puede usar el docker-compose

* ejecutar el docker: docker-compose up -d

una vez se ejecuta entrar al bróker

* docker exec -it Kafka-broker-1 bash (docker exec -it b109912b50b8 bash)

una vez dentro del contenedor ya puede ejecutar Kafka con los comandos normales, reemplazamos la parte del bien ya que este contener ya se ejecuta Kafka

crear topic

* bin/kafka-topics.sh --bootstrap-server localhost:9092 --create –topic debug-topic

por:

* kafka-topics --bootstrap-server Kafka-broker-1:9092 --create --topic test-topic
* kafka-topics --create --topic test-topic --bootstrap-server localhost:9092

**producer**

ya se pueden agregar las modificaciones necesarias

crear un mensaje

* kafka-console-producer --bootstrap-server Kafka-broker-1:9092 --topic test-topic
* kafka-console-producer --bootstrap-server localhost:9092 --topic test-topic

**consumer**

kafka-console-consumer --bootstrap-server Kafka-broker-1:9092 –topic test-topic --from-beginning

# Crear una instancia de Docker en la nube

se puede usar:

<https://customer.cloudkarafka.com/instance>

y en esta documentación esta la configuración que esta en su documentación:

<https://www.cloudkarafka.com/docs/spring.html>

ver este tutorial en caso de tener dudas:

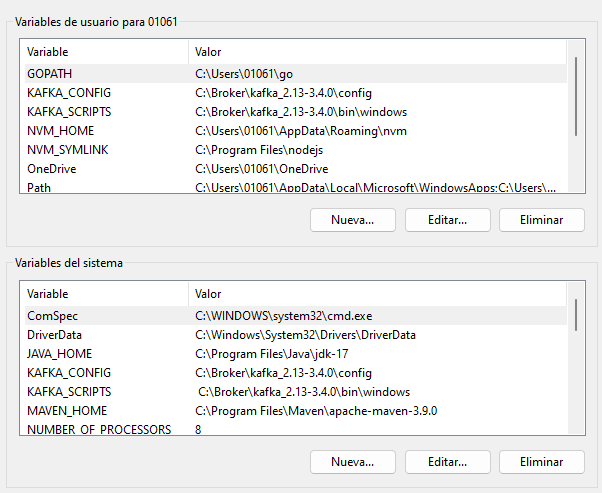
<https://www.youtube.com/watch?v=3nU9bpf1QDw>

# Kafka local

Con toturial de openwebinar:

Tiendo en cuenta esto, el tutorial era cuando aun zookeeper estaba separado

Se realizo esta configuración tratando de ejecutar en local



KAFKA\_CONFIG

KAFKA\_SCRIPTS

Y se configuro en el path

%KAFKA\_SCRIPTS%

Ejemplo de ejecución

zookeeper-server-start $env:KAFKA\_CONFIG\zookeeper.properties

en la ruta C:\Broker se dejo en zookeeper y Kafka por separado, y configurado con 3 topics

ejecutar :

zookeeper-server-start $env:KAFKA\_CONFIG\zookeeper.properties

# Comandos Kafka

# conectarse para obtener todos los mensages del topic

kafka-console-consumer --bootstrap-server localhost:9092 --topic tema-grupo --from-beginning --max-messages 100

# comprueba qué clientes están obteniendo datos de forma activa en un topic

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group conjunto\_1 --members

# detener kafka

C:\Broker\kafka\_2.13-3.4.0\bin\windows\kafka-server-stop

# Datos falsos

Este ayuda a generar datos falsos

<https://github.com/DiUS/java-faker>