- 1. Levantar VM con Kafka
- 2. Levantar un Docker que hará una ingesta continua de datos al Kafka.
- 3. Levantar Google SQL Postgress

Data simulator

Abrir ssh de la máquina virtual donde ya tenemos instalado el Kafka

Para entrar como super usuario:

Sudo -i

Entemos a la carpeta de Kafka

cd kafka_2.12-3.3.1

configuramos con vim la ip publica de Kafka en:

vim config/server.properties

nos dirigimos a la línea a configurar y presionamos i para editar, cambiamos la IP, escape para salir de modo edición y :wq para salir de vim

con el siguiente comando comprobamos si hemos cambiado correctamente la propiedad:

cat config/server.properties | grep advertised

levantamos el zookeeper:

bin/zookeeper-server-start.sh -daemon config/zookeeper.properties

levantamos el kafka:

bin/kafka-server-start.sh -daemon config/server.properties

corremos el simulador de datoss desde un Docker. Sustituir ip publica oír instanci de vm de Google (ip publica vm)

docker run -it -e KAFKA_SERVERS=34.88.224.164:9092 agutlop/datasimulator:1.0

en otro terminal vamos a crear los topics una para el ejercicio final y otro para el proyecto (creo que los crea el propio Docker simulador de datos):

bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic test -partitions 1 --replication-factor 1 --create

bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic devices -partitions 1 --replication-factor 1 --create

Con el siguiente commando comprobamos los topics creados en el Kafka:

con los siguientes commandos nos conectamos al consumer para ver si están entrando los datos en el topic:

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic devices --property print.key=true

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic antenna_telemetry --property print.key=true

jdbc provisioner

será el modulo encargado de crear las tablas con las información con la que se enriquecerá el streaming así como de crear las tablas donde se almacenará la información ya tratada.

En la variable ipServer deberemos poner la IP oública de la base de datos de Google SQL. Previamente habremos configurado las conexiones de la base de datos abiertas a todo internet con 0.0.0.0/0

Navegmos a la instancia de la base de datos y abrimos el clod Shell con el que tendremos acceso a la base de datos. Cuando creemos las tablas desde el código podremos verificar su creación desde este Shell.

Las tablas creadas son aquellas con el prefijo "proyecto"

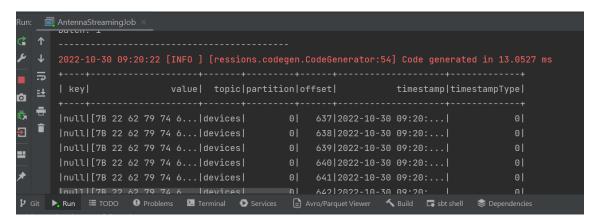
postgres=> \dt			
	List of relations		
Schema	Name	Type	Owner
public public public public public public	final_antenna_1h_agg final_antenna_agg final_antenna_errors_agg final_antenna_percent_agg final_metadata proyecto_batchbytesporantena proyecto_batchbytesporaplicacion proyecto_batchbytesporusuario	+	postgres postgres postgres postgres postgres postgres
public public public	proyecto_streamingbytesporantena	table table table	postgres postgres

Datos insertados:

```
* from proyecto user metadata;
postgres=> select
                id
                                                       email
                                                                   | quota
 00000000-0000-0000-0000-0000000000000
                                   | andres
                                              | andres@gmail.com
                                                                      200000
 00000000-0000-0000-0000-000000000000
                                     paco
                                                paco@gmail.com
                                                                      300000
 00000000-0000-0000-0000-00000000000
                                                juan@gmail.com
                                                                      100000
                                                fede@gmail.com
 00000000-0000-0000-0000-000000000004
                                                                       5000
                                     fede
 00000000-0000-0000-0000-00000000000
                                     gorka
                                                gorka@gmail.com
                                                                      200000
 0000000-0000-0000-0000-00000000000
                                     luis
                                                luis@gmail.com
                                                                      200000
 0000000-0000-0000-0000-000000000007
                                     eric
                                              | eric@gmail.com
                                                                      300000
                                              | carlos@gmail.com
                                                                      100000
 00000000-0000-0000-0000-00000000000
                                     carlos
 00000000-0000-0000-0000-00000000000
                                     david
                                                david@gmail.com
                                                                      300000
 juanchu | juanchu@gmail.com
                                                                      300000
 00000000-0000-0000-0000-00000000011
                                              | charo@gmail.com
                                                                      300000
                                     charo
 00000000-0000-0000-0000-000000000012
                                     delicidas | delicidas@gmail.com |
                                                                     1000000
 00000000-0000-0000-0000-000000000013
                                     milagros | milagros@gmail.com |
                                                                     200000
 00000000-0000-0000-0000-000000000014
                                     antonio | antonio@gmail.com
                                                                     1000000
                                     sergio
                                                sergio@gmail.com
 00000000-0000-0000-0000-000000000015
                                                                     1000000
 00000000-0000-0000-0000-000000000016
                                     maria
                                                maria@gmail.com
                                                                     1000000
                                                cristina@gmail.com |
 00000000-0000-0000-0000-000000000017
                                     cristina |
                                                                      300000
 00000000-0000-0000-0000-00000000018
                                     lucia
                                                lucia@gmail.com
                                                                      300000
 00000000-0000-0000-0000-000000000019
                                     carlota
                                                carlota@gmail.com
                                                                      200000
 emilio
                                               | emilio@gmail.com
                                                                      200000
(20 rows)
```

Streaming Job

Primero leemos el topic devices de Kafka y lo guardamos en un dataframe. Los datos nos llegan en formato json:



Parseamos el value del json a structuredType:

Leemos la metadata desde la base de datos y enriquecemos el stream con estos datos mediante un join:

- Hacemos las agragaciones
- Escribimos de vuelta los datos a postgres y Parquet en directorio local