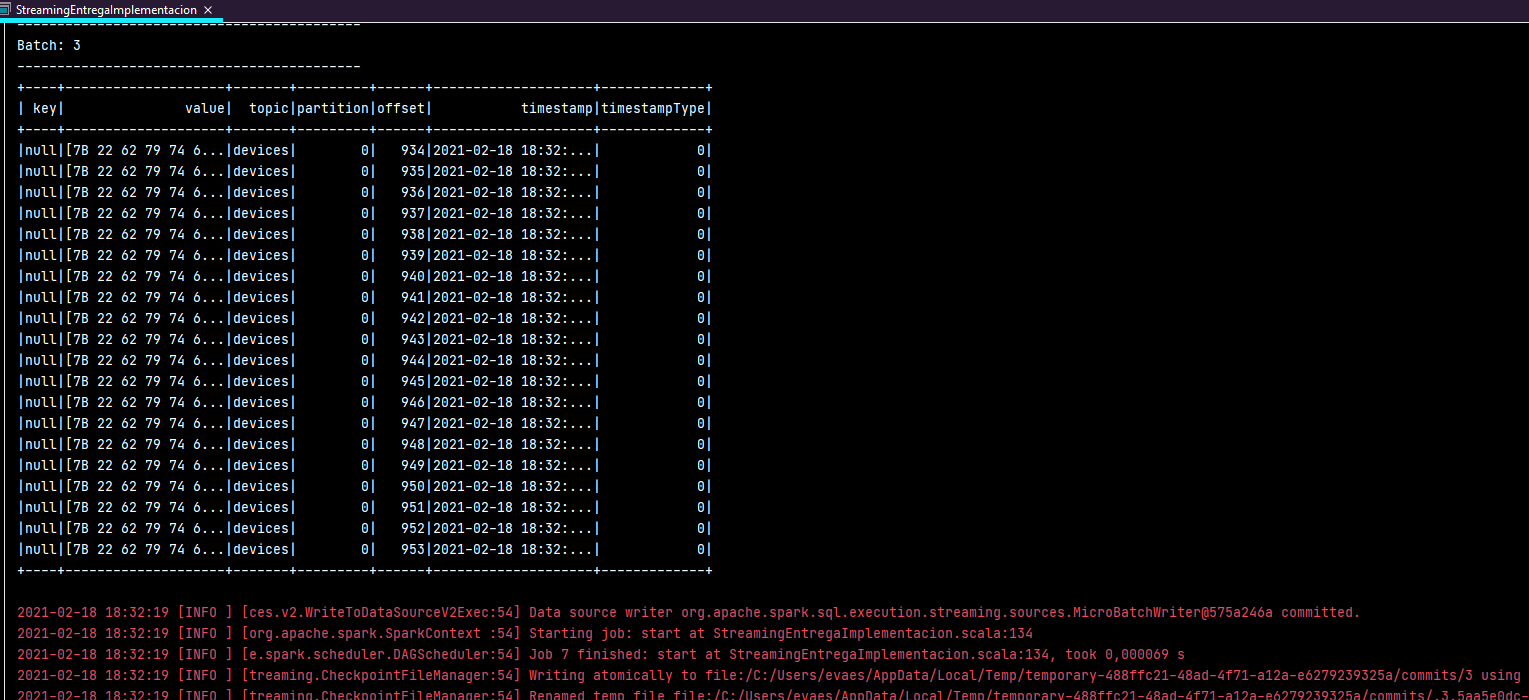
Objetivo del proyecto:

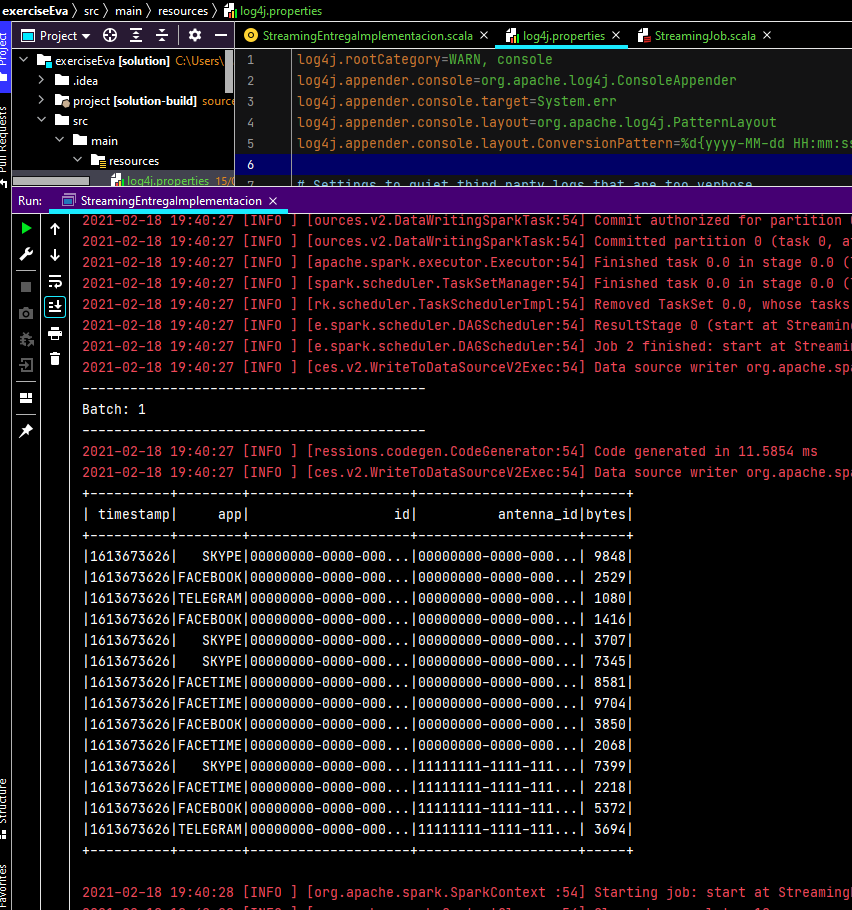
A partir de datos de antenas de telefonía móvil recibidos en tiempo real (simulado) e información almacenada sobre los usuarios del servicio, el objetivo es analizar el tráfico de datos transmitidos por usuarios, por aplicación empleada y datos recibidos por diferentes antenas.

Lo primero es preparar las fuentes de datos, establecer el sistema de escucha/monitorización de las antenas. Para ello ponemos en marcha una máquina virtual en la que instalo Kafka para almacenar los mensajes de las antenas (docker de simulación) en tiempo real.

A continuación, a través de spark Structured streaming, en el IDE implemento un job de streaming con diferentes fases (a continuación voy listando los resultados obtenidos progresivamente, el código para obtenerlos va en documento adjunto):

- compruebo que recibo los datos de las antenas del topic que he indicado correctamente. Los mensajes llegan regularmente. 

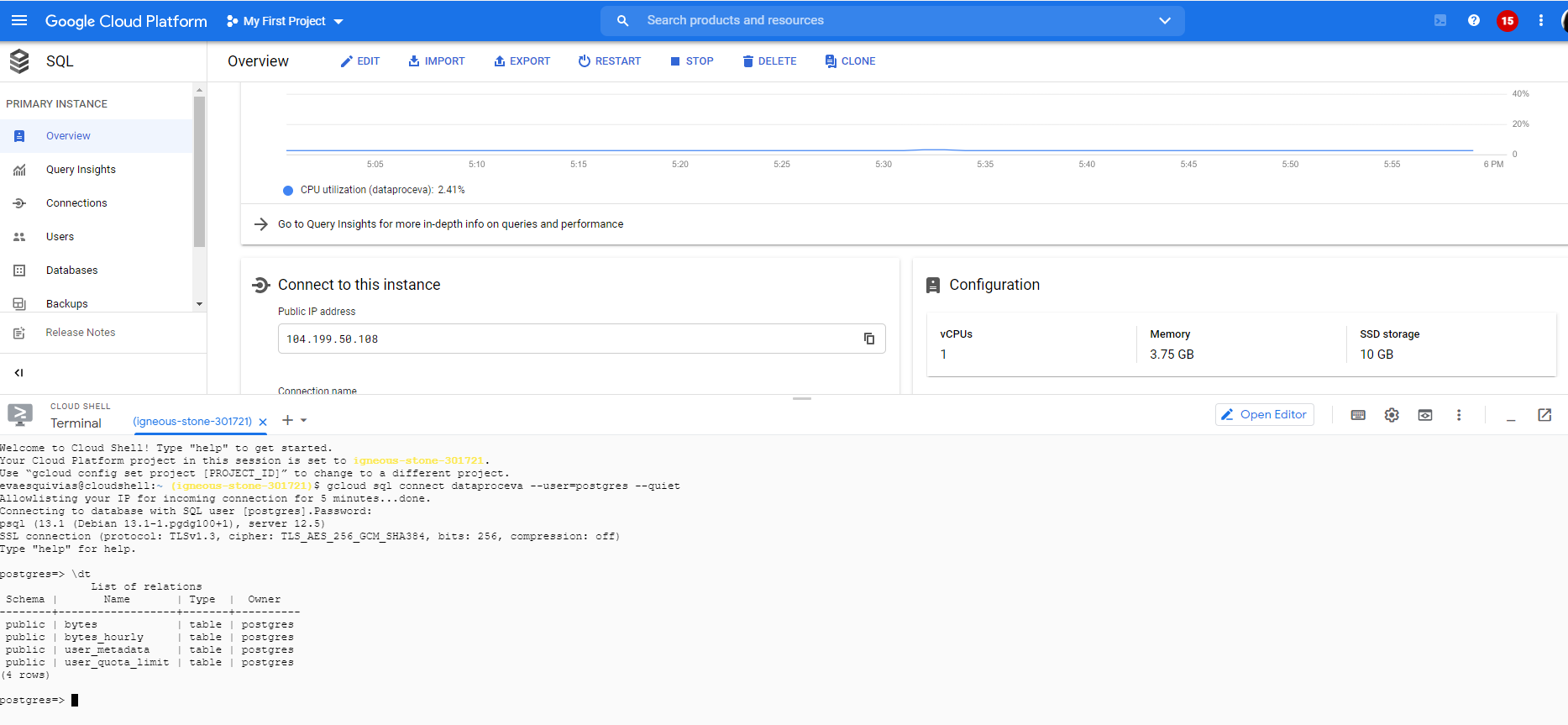
Los datos en la lectura de Kafka se obtienen en formato esquema clave valor codificado en binario en bytes. Hay que parsearlo a json para poder trabajar con ello. Traducción: nos quedamos con el valor y lo convertimos de array de bytes a Stringtype y el esquema lo obtengo de la case class que tengo definida.

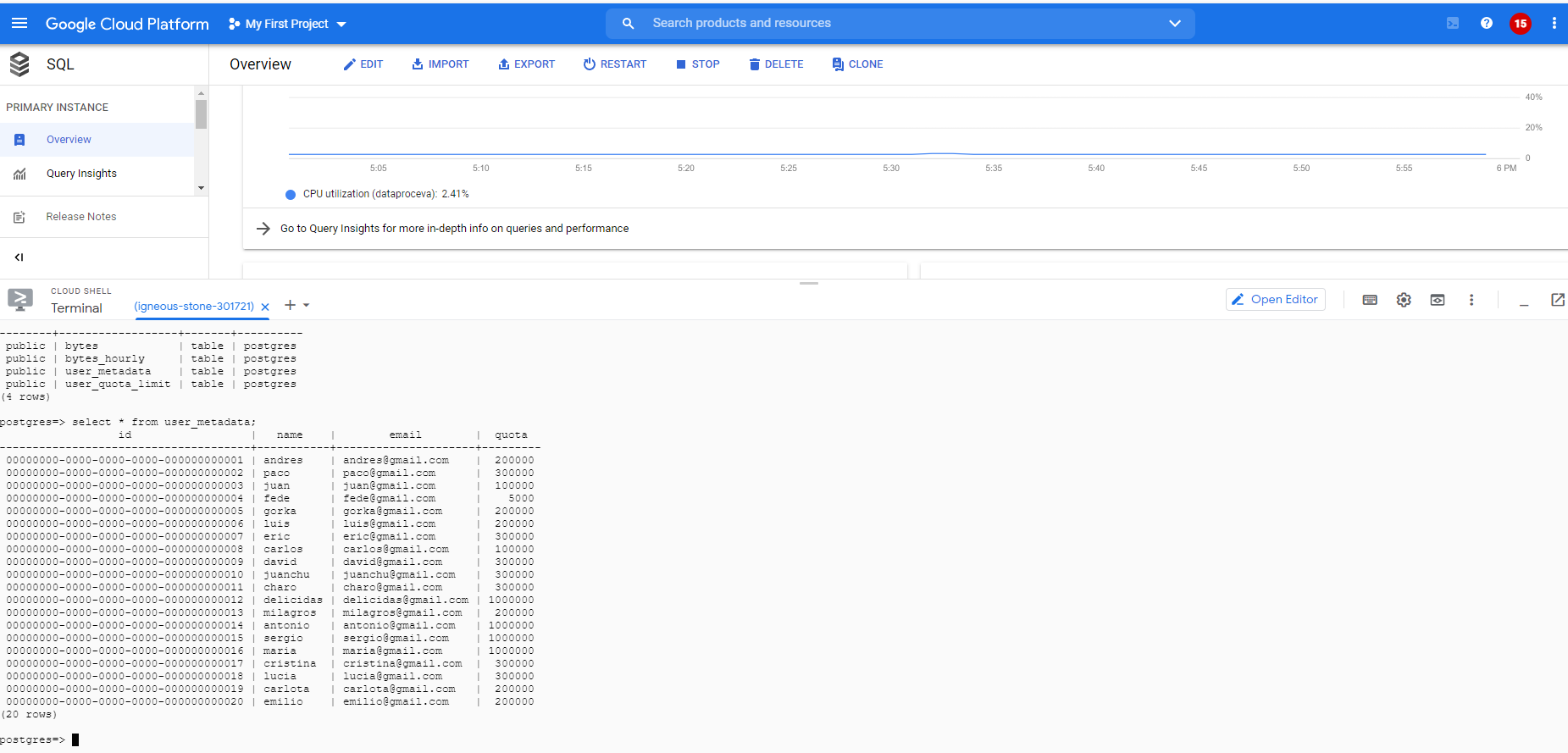


Con esto confirmo que tengo los datos de la fuente 1, en tiempo real/streaming.

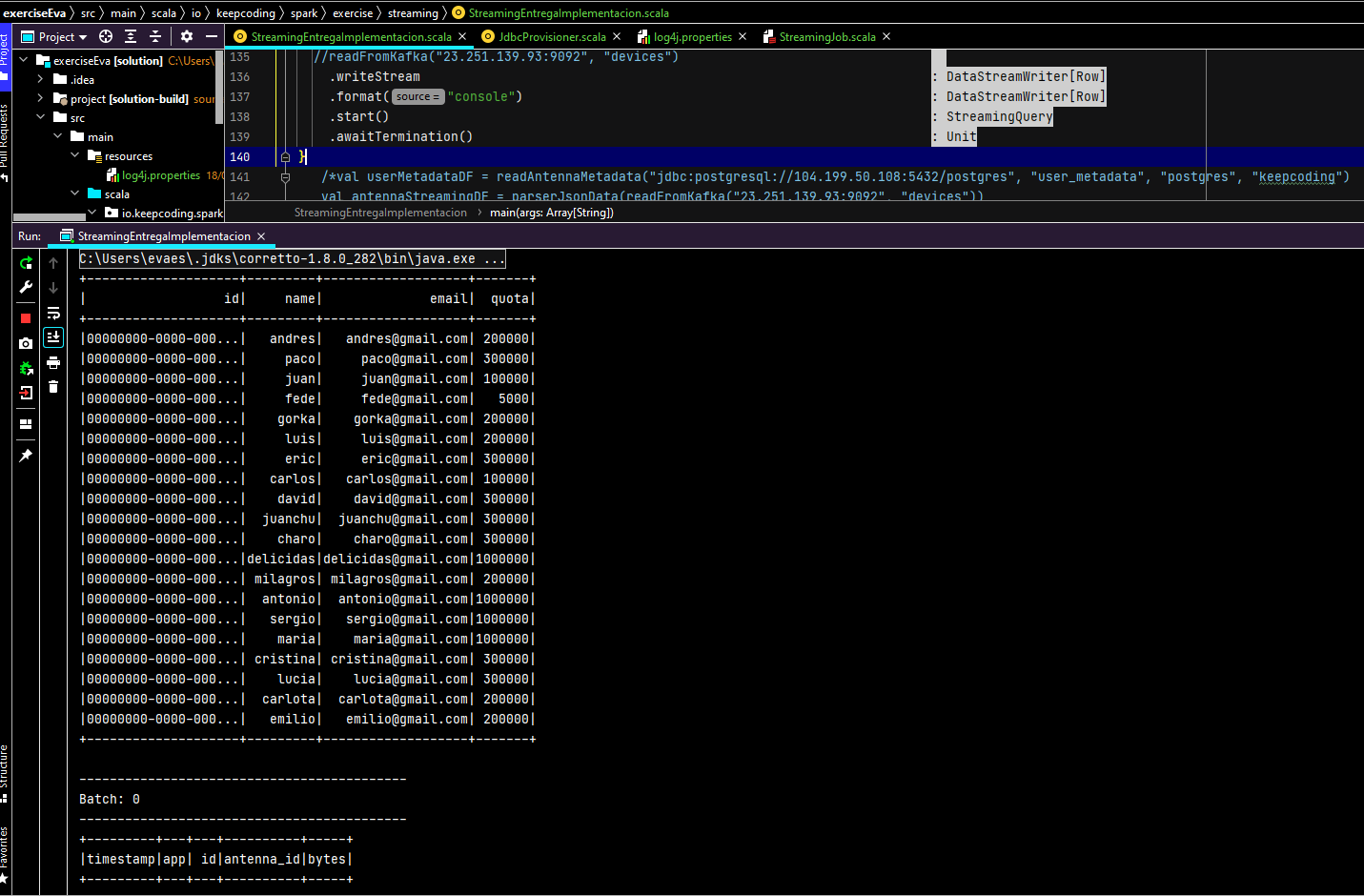
Para obtener los metadatos de usuarios dados de alta en el servicio me tengo que conectar a la base de datos en que están almacenados. Primero tengo que provisionarlos.

Para ello creo la instancia SQL en google cloud, (postgreSQL), y con el provisioner en el IDE creo las tablas que voy a necesitar y cargo los datos para luego poder obtener la información que necesito en la implementación.





Continúo con el job de streaming desde el que leo la base de datos y obtengo la tabla user\_metadata.



Lo siguiente sería unir los datos de las dos fuentes para ampliar la información que tengo de los usuarios.

Confirmo que me llegan los datos de Kafka y se han añadido los metadatos:

