**Arquitectura Big Data**

**Definición:**

El objetivo será montar un cluster en la nube con Google Cloud , subir el csv de Airbnb de los pisos de Madrid al Google Storage y scrapear los pisos en alquiler de Madrid o un barrio concreto de milanuncios.com para comparar su precio respecto a los de airbnb en función de los diferentes atributos que contengan (ubicación, m2, dormitorios, baños, servicios...) de forma diaria o semanal.

**Arquitectura:**

Elijo la arquitectura que proporciona Google Cloud por la sencillez de los pasos a seguir para su uso y su variedad de productos asociados. También por la flexibilidad que proporciona a nivel de consumo (interesante si solo se necesita utilizar una vez a la semana, por ejemplo) como su versatilidad a la hora de gestionarlo pudiendo hacerlo incluso desde un smartphone.

Cluster-> Google Cloud-> 1 master, 2 nodos

Cloud -> dataset (airbnb.csv) -> Google Storage

Staging-> Google Storage

Crawler -> milanuncios.com todos los pisos de Madrid o un barrio concreto[pendiente]

Cron -> Script con python para la actualización diaria o semanal de airbnb y milanuncios.com[pendiente]

Sataging-> Google Storage

**Desarrollo:**

***Parte 1. Utilizar una herramienta de diagramado como Google Draw o DIA para diseñar y especificar el flujo de datos y herramientas utilizadas***

Link: <https://docs.google.com/drawings/d/1nx0oTHs3zSus5frJm6vurwU5jIX5128mA5zAXdeACik/edit?usp=sharing>

***Parte 2. Crear un scraper en Google Collaboratory a partir de un API o de un crawler con scrapy, que descargue los datos a un archivo de formato estructurado***

***- Scrapy Basico***

***- Links de dataset: Airbnb***

Datos scrapeados de milanuncios [pendiente]

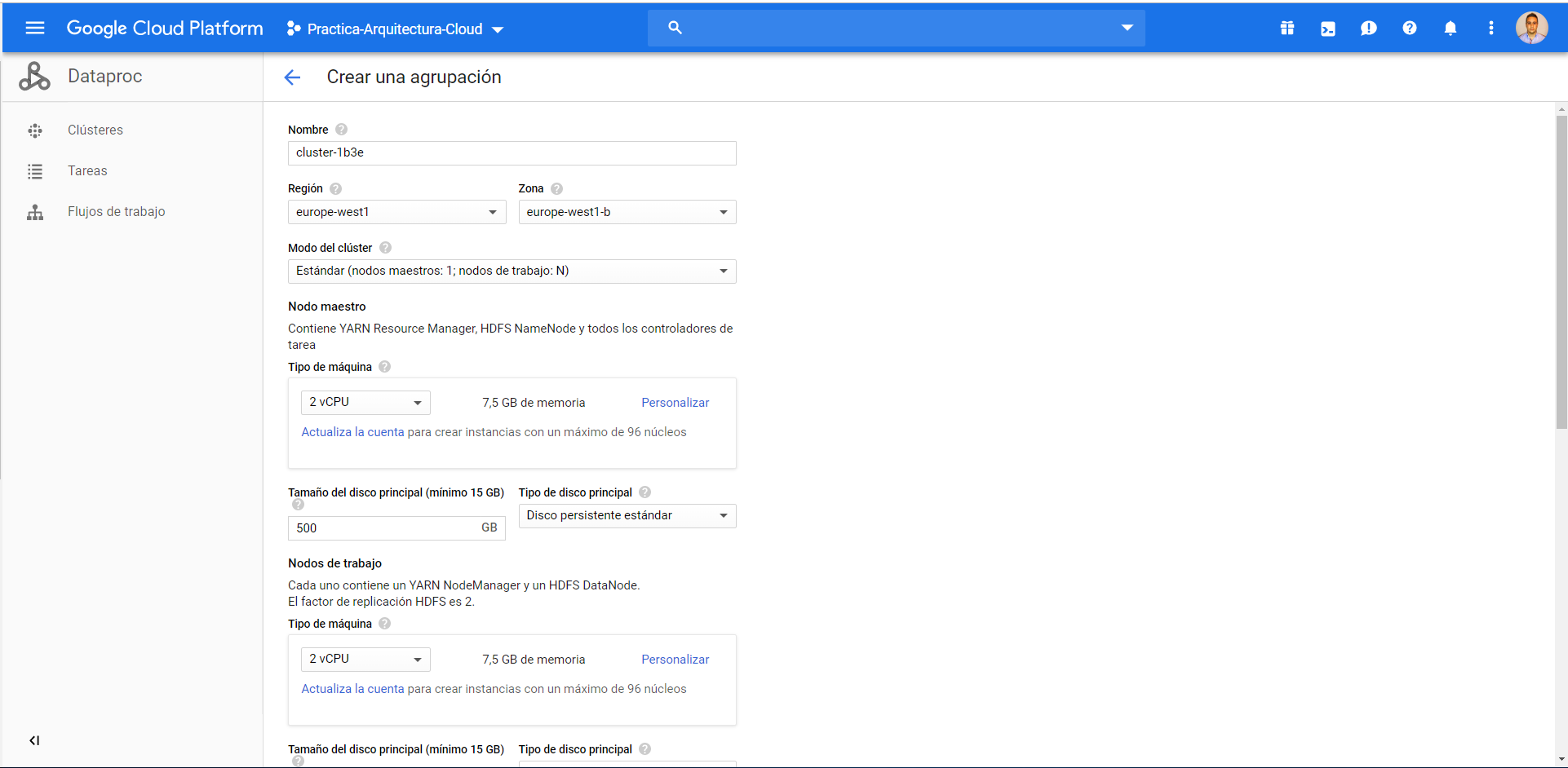
Dataset: Airbnb (Madrid), descargar el csv y subirlo a Google Store

***Parte 3. Utilizar un proveedor de Cloud para montar un cluster de al menos 3 contenedores configurados correctamente***

Crear proyecto en Google Cloud: ''Practica-Arquitectura-Cloud''

- Habilitar API

- Crear clúster



- Región: europe-west1,

- Zona: euorpe-west1-b

- Estándar: maestro, 2 nodos

**Añadir regla de cortafuego:**

- Red de VPC -> Reglas de cortafuegos -> Abrir puertos Hadoop:

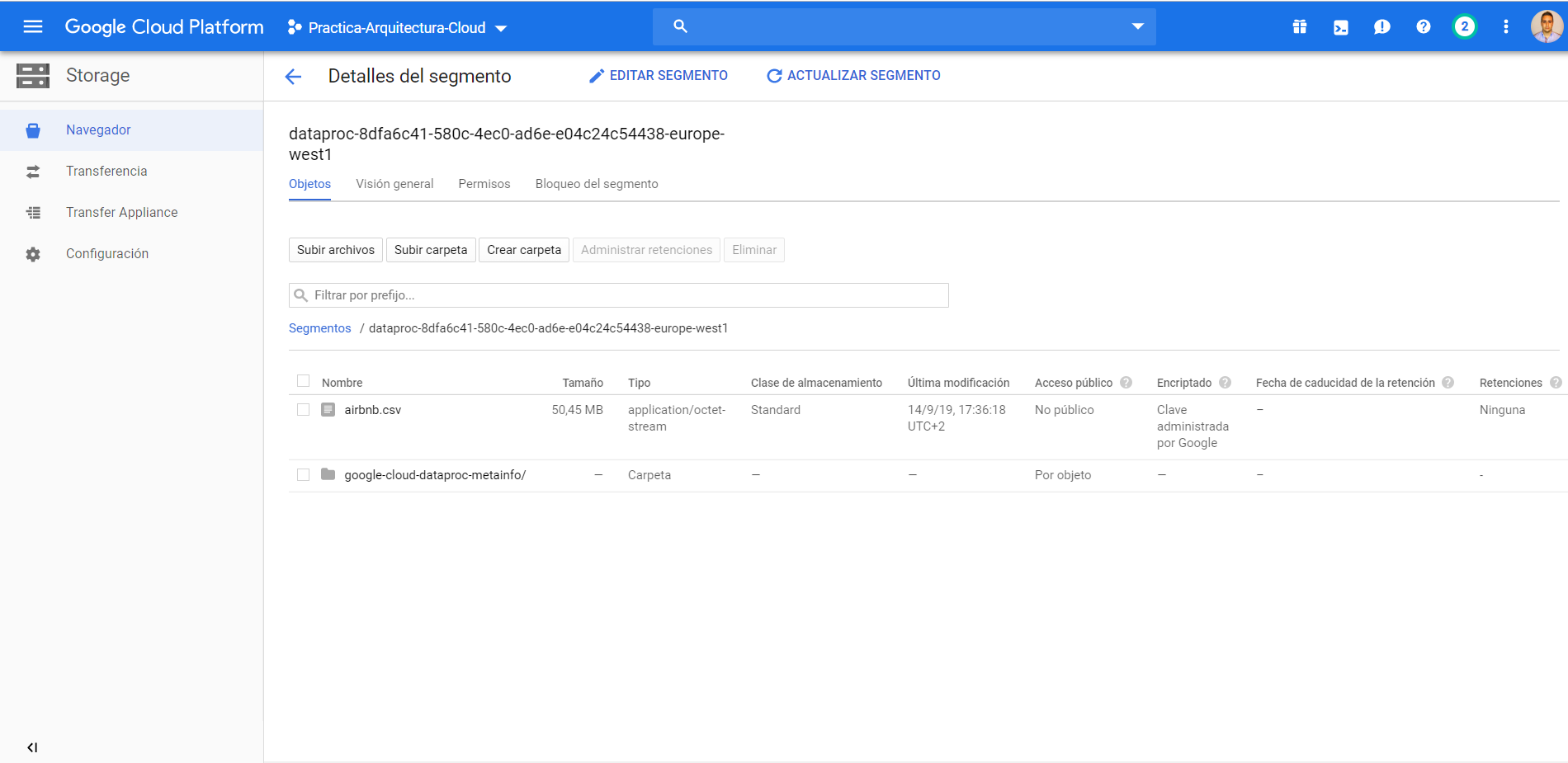
- TCP: 8088 y 9870

- Nuestra IP/32 en intervalos de IP de origen.

- Destinos: Todas las instancias de la red.

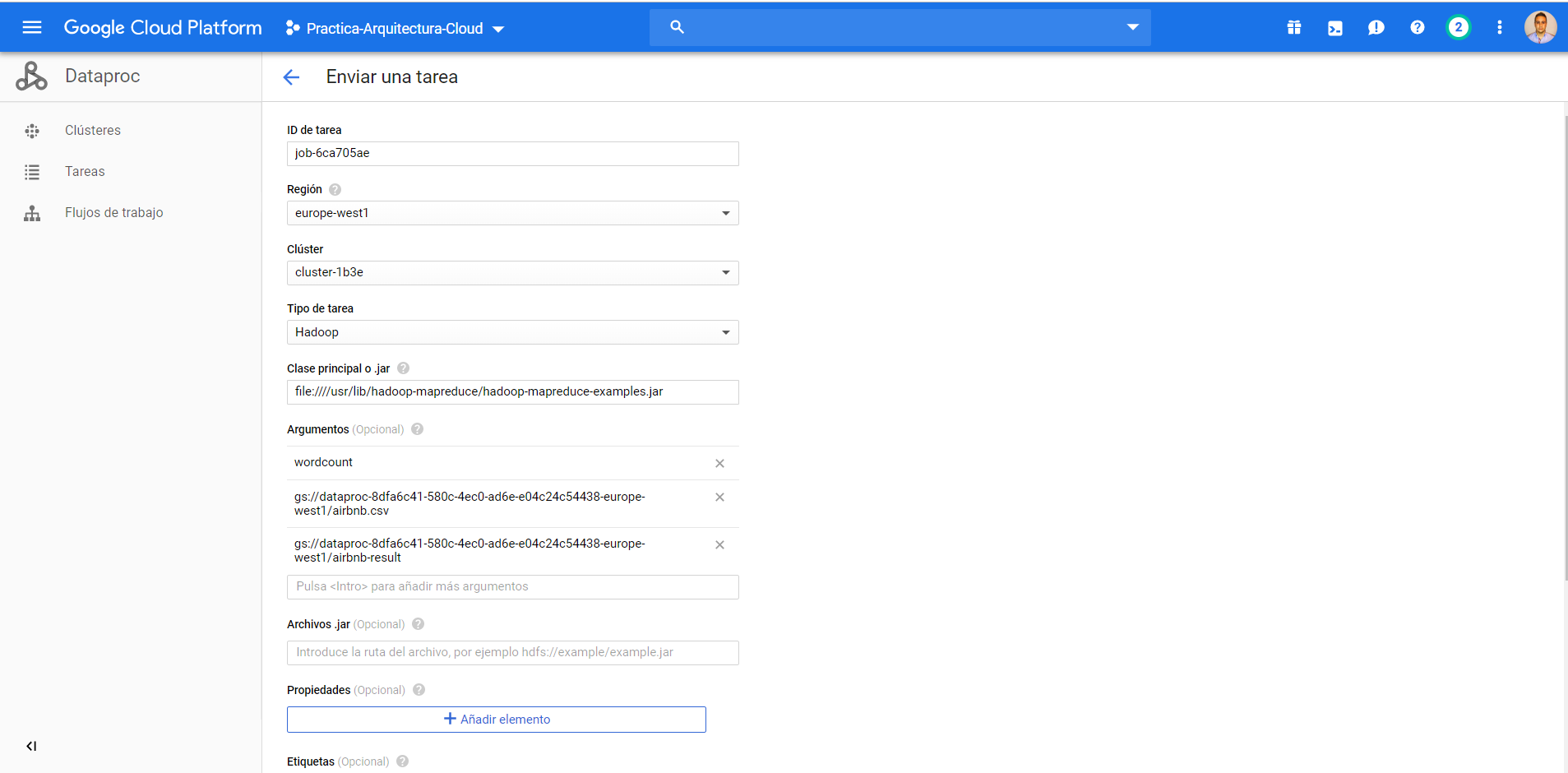
***Parte 4. Subir los archivos extraídos durante la parte 2 al cluster de Hadoop e insertarlos en el HDFS. Indicar pasos necesarios para realizar esto, dependiendo de la opción elegida en el Sprint 3. Realizar la tarea de procesamiento de datos sobre los datos extraídos utilizando WordCount.***

**Google Storage**



Storage>> Navegador>> Accedemos al segmento del clúster >> Subir archivos>> Airbnb.csv

Una vez creado accedemos al csv y copiamos su URI para ponerlo en el segundo argumento a la hora de crear el job.

Sin tener el .jar adecuado y sin los pisos de milanuncios.com, simplemente he seguido el ejemplo dejando el resultado en Google Storage.

Dataproc>>>Enviar tarea

- Región: europe-west1,

- Tipo tarea Hadoop

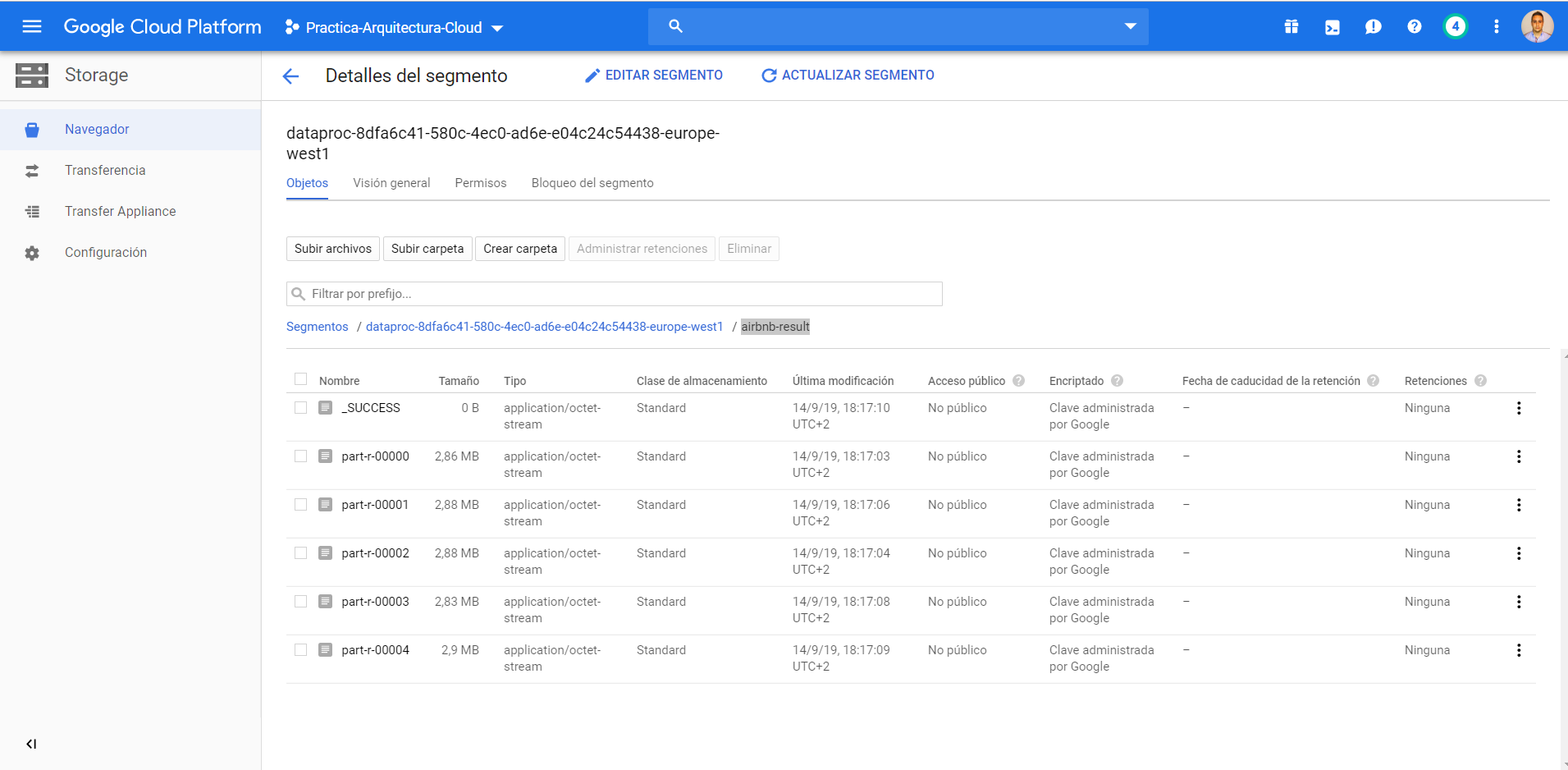
- Clase principal o .jar: <file:////usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar>

Argumentos:

- Wordcount (aplicación de MapReduce para contar el número de veces que aparece una palabra)

- (El archivo lo coge de aquí) gs://dataproc-8dfa6c41-580c-4ec0-ad6e-e04c24c54438-europe-west1/airbnb.csv

- (Y devuelve los resultados a Google Storage, en el segmento del cluster+airbnb-result)gs://dataproc-8dfa6c41-580c-4ec0-ad6e-e04c24c54438-europe-west1/airbnb-result



***Parte 5. Utilizar HIVE/Elastic/Kafka/Mongo para insertar los datos extraídos durante el Sprint 2 y realizar operaciones con los mismos. Indicar los pasos y las decisiones de diseño respecto a cómo organizar los datos.***

Pendiente.