Sistel

CONSULTORÍA
Y SERVICIOS
INFORMÁTICOS

Pipeline de Dataflow. CSVs en Cloud Storage a tablas de BigQuery

Índice de contenidos

1. Introducción

En este documento se describen los pasos necesarios para automatizar la importación de ficheros CSVs que se suben a un *bucket* de Cloud Storage a tablas de BigQuery. Incluye una Cloud Function encargada de localizar el archivo que se ha actualizado, leer su esquema e iniciar la plantilla de Dataflow, que almacena los datos en el dataset de BigQuery que indiquemos.

1.1. Preparación

Para completar este flujo de datos, se debe disponer de un 'bucket' de Google Cloud Storage con la siguiente estructura de carpetas:

- **/csv** o /**data**: directorio donde se almacenarán los archivos CSV que se desean importar a BigQuery.
- **/schemas**: directorio que aloja los archivos JSON que describe los campos de cada CSV. Estos JSON se generarán automáticamente cuando se despliegue la Cloud Function y se suban los archivos CSV.
- **/templates**: contendrá la plantilla de Dataflow que se genera en los apartados siguientes.

También, se debe haber subido a la carpeta '*templates*' del storage el archivo 'Provincia.csv' que se adjunta a esta memoria. Esto se debe a que, a la hora de crear la plantilla de Dataflow, esta debe poder acceder a un archivo por defecto, se selecciona 'Provincia.csv' por su sencillez.

1. Creación de la plantilla de Dataflow

Junto a esta memoria se adjunta el archivo *'data_ingest.py'* con el código para crear la plantilla de Dataflow, que se almacenará en el Storage y se llamará desde la Cloud Function para llevar los datos a tablas de BigQuery, pero se debe modificar mínimamente para adaptarlo al proyecto que se esté desarrollando.

Abriendo el archivo con un editor hay que especificar en las líneas 14 y 15 el nombre del *bucket* de Cloud Storage que contiene los archivos CSV y el carácter utilizado para delimitar cada campo en los ficheros.

```
13
14  STORAGE_BUCKET = 'preving-bajas'
15  CSV_FIELD_DELIMITER = ';'
16
```

Con este archivo listo, se puede subir al editor de Cloud Shell y crear la plantilla. Para esto es necesario preparar el entorno e instalar las dependencias necesarias:

1. Instalar y ejecutar virtualenv

En la consola de Cloud Shell ejecutar:

- pip install virtualenv
- virtualenv -p python3 venv
- . venv/bin/activate (incluir el punto del principio)

2. Instalar las dependencias del código

- pip install apache-beam[gcp]
- pip install google-cloud-storage

Cuando terminen estos procesos, el entorno está preparado para ejecutar el archivo *'data_ingest.py'*, para hacerlo, desde la consola, se tiene que ejecutar con los siguientes parámetros:

python3 [ruta_hasta_el_archivo/]data_ingest.py

--project=[ID_del_proyecto]

--region=europe-west1

--temp_location=gs://[nombre_del_bucket]/tmp

--staging_location=gs://[nombre_del_bucket]/tmp

--runner=DataflowRunner

--experiments=use_beam_bq_sink

--template_location=gs://[nombre_del_bucket]/templates/extract-csv-template

Esto creará la plantilla con nombre *'extract-csv-template'*, en la carpeta *'templates'* del bucket que se haya indicado, ahora, se pueden hacer llamadas HTTP a esta plantilla para ejecutarla, cosa que se hará desde la Cloud Funtion del apartado siguiente.

3. Creación de la Cloud Function

La función que se va a crear es la encargada de ejecutar la pipeline de Dataflow que se ha creado en el apartado anterior para cada archivo CSV que se sube al bucket que indiquemos.

Para empezar, debemos crear la función en Google Cloud Functions, dándole un nombre único y asignándole la región 'europe-west1'. Dentro del apartado 'Activador' se debe especificar:

- Tipo de activador: Cloud Storage
- Event type: Finalizar/Crear
- Segmento: seleccionar bucket donde se almacenan los CSV

Tras esto, se pasa al siguiente paso, que será modificar el código que se adjunta a esta memoria y subirlo.

1. Editar la función de python

Desde un editor de texto, se puede abrir el archivo *main.py* y configurar las constantes de las primeras líneas para adaptarlo al proyecto en desarrollo:

```
7   PROJECT_ID        = 'preving-bajas-301512'
8   SCHEMAS_LOCATION  = 'Bajas/schemas/'
9   TEMPLATE_LOCATION = 'gs://preving-bajas/templates/extract-csv-template'
10  OUTPUT_DATASET    = 'BAJAS_BQ_STG.'
11  CSV_SEPARATOR     = ';'
```

- PROJECT_ID: id del proyecto
- SCHEMAS_LOCATION: directorio del Cloud Storage donde se almacenan los archivos JSON con los esquemas de las tablas. Cada vez que se suba un archivo CSV, del que no se conoce su esquema, la función lo generará automáticamente en esta localización.
- TEMPLATE_LOCATION: ruta y nombre de la plantilla de Dataflow que se ha creado anteriormente, se corresponde con el parámetro *–template_location* que se ha indicado al crearla
- OUTPUT_DATASET: dataset de BigQuery donde se almacenarán las tablas resultantes de la operación.
- CSV_SEPARATOR: separador de columnas que se ha utilizado en los archivos CSV.

2. Subir el código a Cloud Functions

Volviendo a la pantalla de Cloud Functions, se debe seleccionar :

- Entorno de ejecución: Python3.8
- Punto de entrada: csv_to_template - (es el nombre de la función de entrada)

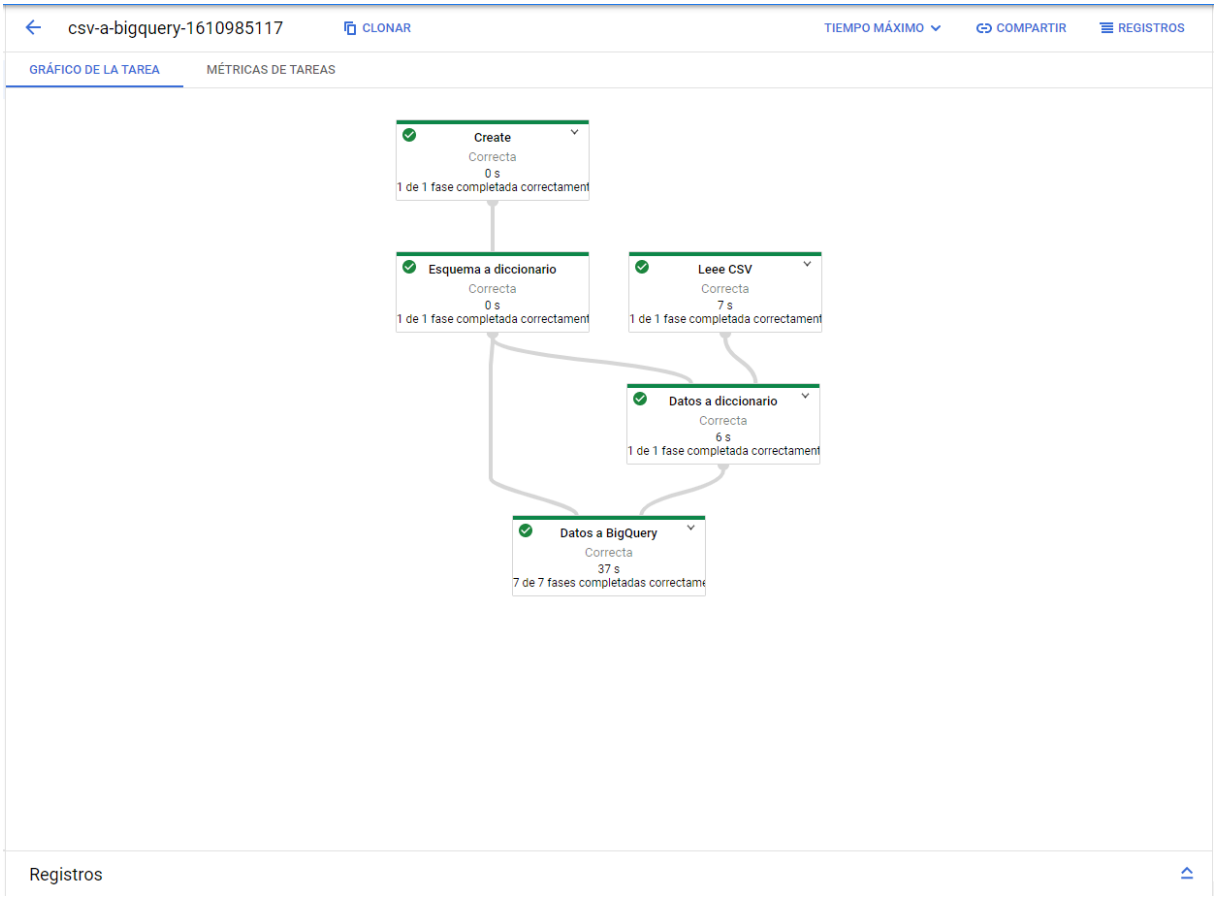
Para subir el código se puede hacer o bien copiando el contenido de *main.py* y *requirements.txt* en los archivos con el mismo nombre en la interfaz de Cloud Functions o comprimiendo ambos en un *.zip* y subiéndolo.

Con esto hecho, podremos desplegar la función y comprobar que, al subir un CSV nuevo, creará, cuando finalice la tarea, una tabla con el mismo nombre que este fichero en el dataset indicado.

2. Depuración y visualización de registros

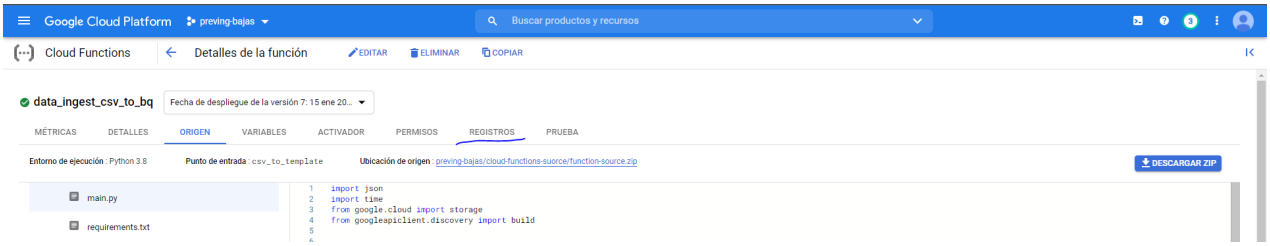
2.1. Depuración en Dataflow

Cuando se sube un archivo al bucket indicado, se crea una tarea con el nombre *csv-a-bigquery-[marcca de tiempo]*, se puede acceder a este trabajo accediendo a la herramienta de Dataflow y haciendo click sobre él, se mostrarán todas las fases de la pipeline, y, si se ha producido algún error en alguna de estas se puede hacer click sobre ella y abrir el panel de registros situado en la parte inferior de la pantalla.



2.2. Depuración en Cloud Functions

Si comprobamos que no se crea la tarea de Dataflow correspondiente a una ejecución de la Cloud Function, o se sospecha que un determinado error puede tener origen en esta, se pueden visualizar los registros generados desde los detalles de la función.



Importante: tanto para visualizar los registros de Dataflow como los de Cloud Functions, es necesario tener el rol de **visor de registros** en la cuenta donde se está trabajando.