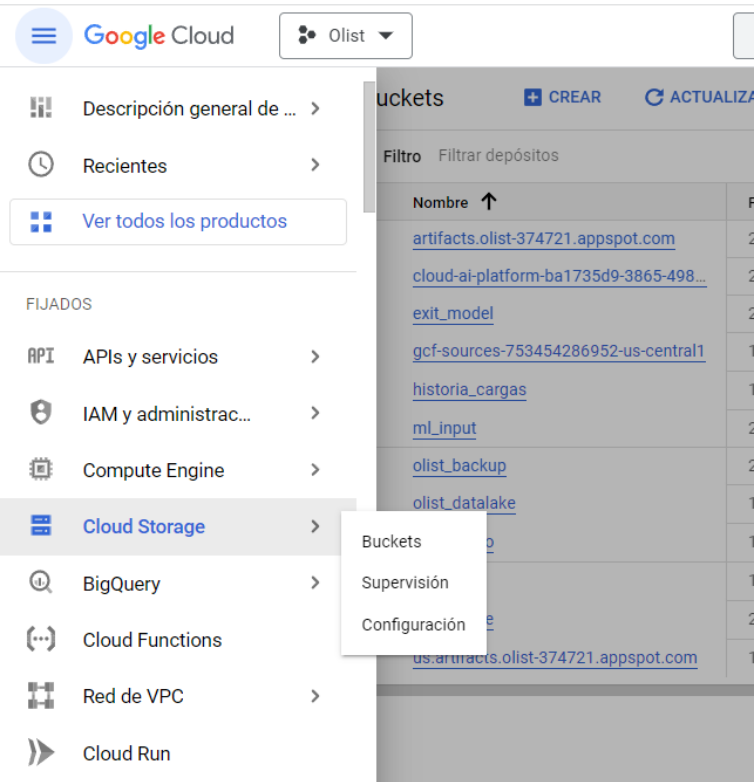


Proceso de carga y disponibilidad de los datos:

Al trabajar con la Plataforma de Google Cloud (de ahora en más la llamaremos GCP), se crea un proyecto que denominaremos Olist. Dentro del proyector sabemos que disponemos de unos depósitos de información llamados “Buckets”, estos depósitos se encuentran dentro del sector Google Cloud Storage. Estos Bucket permiten almacenar datos ya sea en su raíz o en un directorio organizado por carpetas.



Cuando se plantea el desarrollo del proyecto se establecen la creación de estos bucket para cubrir diferentes utilidades en pos de una mejor organización. Se crean 6 depósitos:

historia_cargas	19 ene 2023 09:26:50	Multi-region
ml_input	27 ene 2023 04:06:53	Region
olist_backup	24 ene 2023 12:20:54	Region
olist_datalake	18 ene 2023 12:33:29	Multi-region
olist_extro	18 ene 2023 12:34:56	Multi-region
olist_raw	18 ene 2023 12:29:45	Multi-region
olist_stage	20 ene 2023 11:32:18	Region

olist_raw : Donde se otorgan permisos de escritura para que el cliente pueda depositar sus archivos y posterior tratado.

olist_datalake : Donde se depositan los archivos del cliente y los nuestros, con permisos para que los pueda consultar, este depósito funciona como “Data Lake” del proyecto.

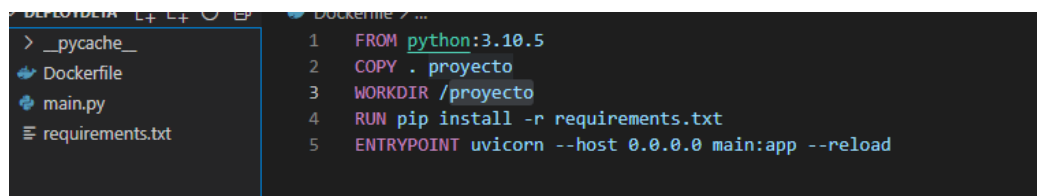
historia_carga : En ese bucket se guarda un archivo que funciona de registro, almacenado un informe de cada nuevo archivo ingresado a olist_raw, para comparar con los archivos ya guardados en olist_datalake y darle paso o no a la siguiente etapa.

olist_extrlo : Aquí es donde se depositan los archivos luego de ser evaluados como óptimos para proceder con el proceso de transformación y normalización de los datos.

olist_stage : Donde después de una transformación y normalización automática son depositados los datos para consumo.

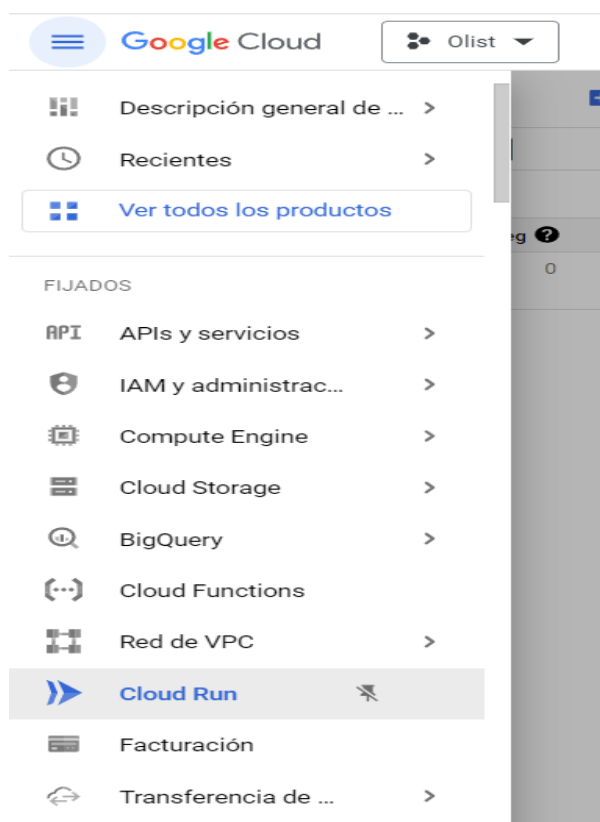
olist_backup : Donde se guardan los archivos de los buckets olist_datalake y olist_stage en carpetas, con nombres indexados cada Viernes de cada semana a las 23:59 pm y permanecen durante 30 días consecutivos antes de ir siendo reemplazados.

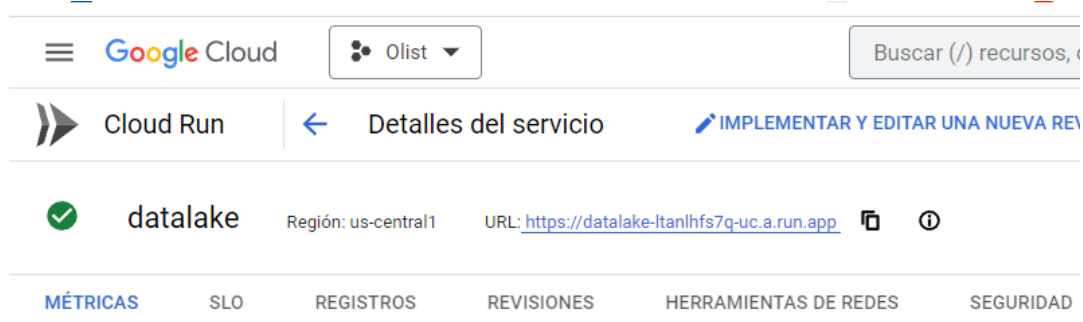
Para que el cliente pueda consultar los archivos almacenados en el Data Lake se crea una API de consulta. Esta API de consulta se crea a través de FastAPI utilizando Python y poniéndola dentro de un contenedor de Docker . Este contenedor se sube y se despliega con Google Cloud Run permitiéndonos realizar consultas vía web, donde se puede ver los archivos almacenados y luego utilizando el nombre del archivo que se desea, se los puede visualizar o descargar en formato JSON.



```
> _pycache_
Dockerfile
main.py
requirements.txt

1 FROM python:3.10.5
2 COPY . proyecto
3 WORKDIR /proyecto
4 RUN pip install -r requirements.txt
5 ENTRYPOINT uvicorn --host 0.0.0.0 main:app --reload
```





Por medio de Cloud Functions se crean script de Python que se encargan de hacer las evaluaciones y transformaciones necesarias para asignar a cada archivo de datos su correspondiente bucket.

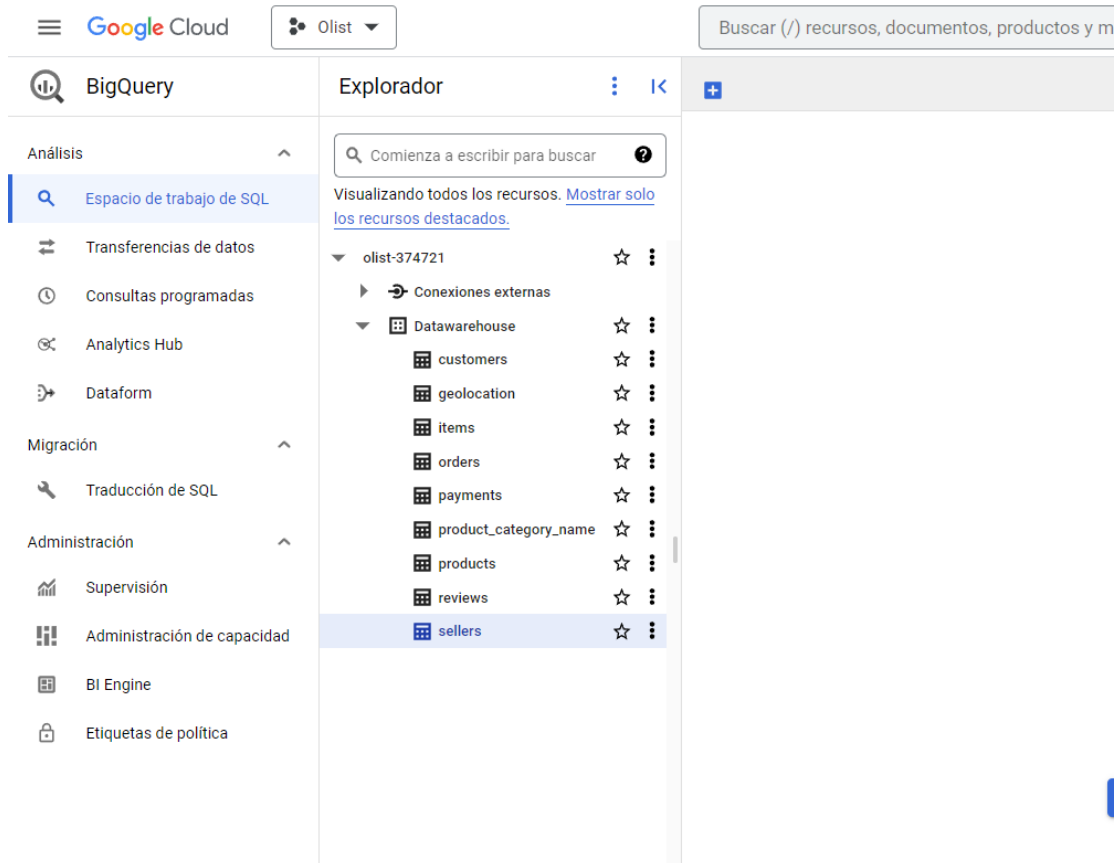
	Entorno	Nombre ↑	Última implementación	Región	Activador	Tiempo de ejecución
<input type="checkbox"/>	1st gen	a_bigquery	24 ene 2023 00:23:27	us-central1	Bucket: olist_stage	Python 3.10
<input type="checkbox"/>	1st gen	carga_delta	21 ene 2023 19:44:20	us-central1	Bucket: olist_raw	Python 3.10
<input type="checkbox"/>	1st gen	etl	23 ene 2023 22:19:27	us-central1	Bucket: olist_extrlo	Python 3.10

El procedimiento funciona de la siguiente manera:

- Cada vez que se carga uno o varios archivos al bucket olist_raw (depositados por el cliente), se dispara una llamada a una Cloud Function que se encarga de evaluar la carga incremental de cada uno de los archivos.
- En esa evaluación llama al archivo que registra los archivos en el bucket historia_carga y compara si el archivo ingresó previamente, si ya fue depositado en olist_datalake, si ya paso al bucket olist_extrlo, y dependiendo del resultado es derivado al bucket correspondiente o desestimado.
- Si el archivo es almacenado en olist_extrlo, significa que es un archivo necesario para la generación de información. En esta etapa se dispara otra Cloud Function que toma este archivo y le aplica las transformaciones y normalizaciones necesarias, para que pueda pasar a la siguiente etapa en el bucket olist_stage.
- Todo archivo en el bucket olist_stage está listo para que pueda ser utilizado. Por eso, ni bien llega un archivo nuevo, se dispara otra Cloud Function que se encarga de conectar a Google BigQuery. Mirando el archivo de olist_stage evalúa si existe la tabla

correspondiente que contenga los datos del mismo, de no existir la crea y le ingresa los datos del archivo, y si existe la tabla le agrega registros nuevos con los datos del archivo.

- En Google BigQuery conformamos nuestro “Data Warehouse” donde se disponen permisos para que el cliente pueda conectarse a través de su cuenta en GCP y pueda consultarlo utilizando lenguaje SQL.



Desde BigQuery quedan disponibles las tablas para que se pueda hacer un análisis y visualización de datos con la herramienta Power BI.

Desde Power Bi importamos las tablas que están en BigQuery ya que no permite actualizar los datos que trae cada vez que queremos visualizar información.