Respuestas prueba WOM

Bernardita Hernández

19-02-2024

1. Define una arquitectura orientada a eventos, en donde al recibir un archivo en un bucket este se procese.

R-Para el caso solicitado se ha seleccionado una arquitectura corresponde a los servicios de:

Almacenamiento Cloud Storage: Recibe los archivos de entrada en un bucket.

Procesamiento Compouser Airflow: ejecuta el procesamiento del archivo en un dag.

BigQuery: Almacena los datos procesados para análisis futuros.

Se podría utilizar Monitoring: Para ver la traza de los servicios y sus procesos

2. Desarrolle un pipeline de GitLab CI/CD o en GitHub que despliegue una Cloud Function a través de Terraform.

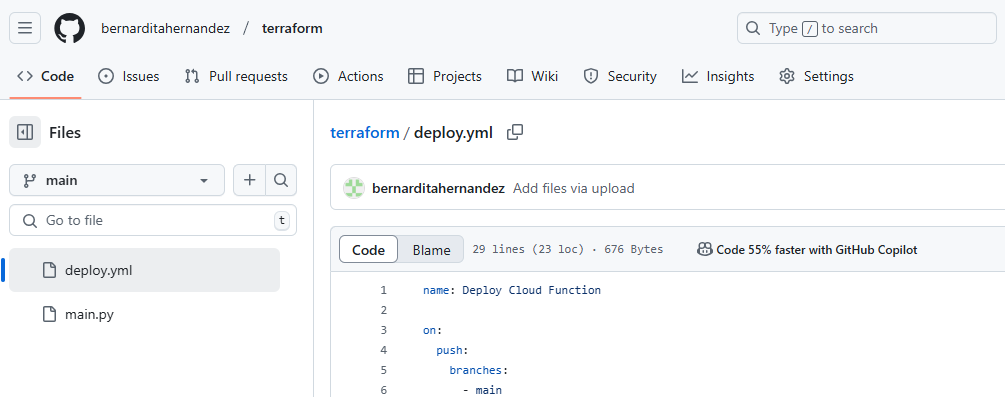
R- Para este caso se escogió el GitHub personal con una función sencilla de hola mundo solo para comprobar que la comunicación funciona.

Para iniciar fueron activadas las cuentas de servicio de roles: roles/editor, roles/storage.admin, roles/cloudfunctions.admin

Instale Terraform y cree la carpeta Proyectos en C:

Agregue la Path en las variables de entorno

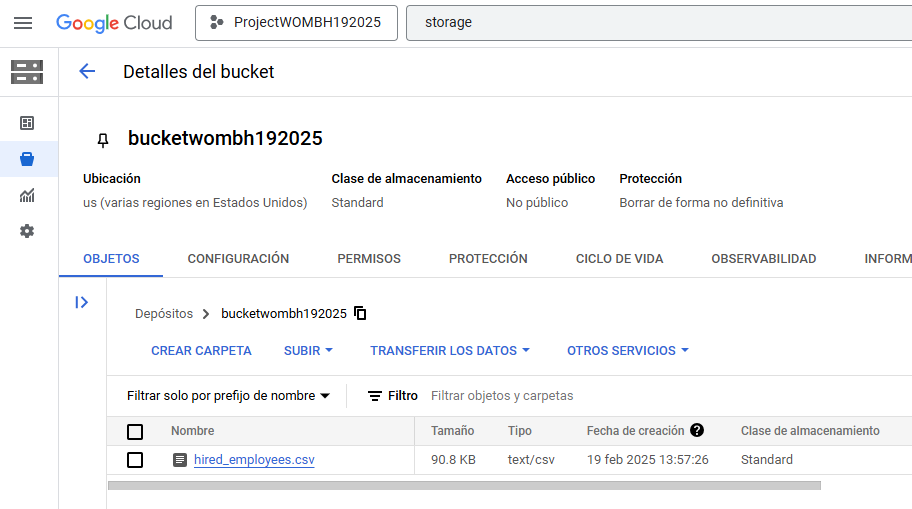
Luego fueron creados los archivos con las conexiones correspondientes con nombre de proyecto, bucket y se cargaron en GitHub



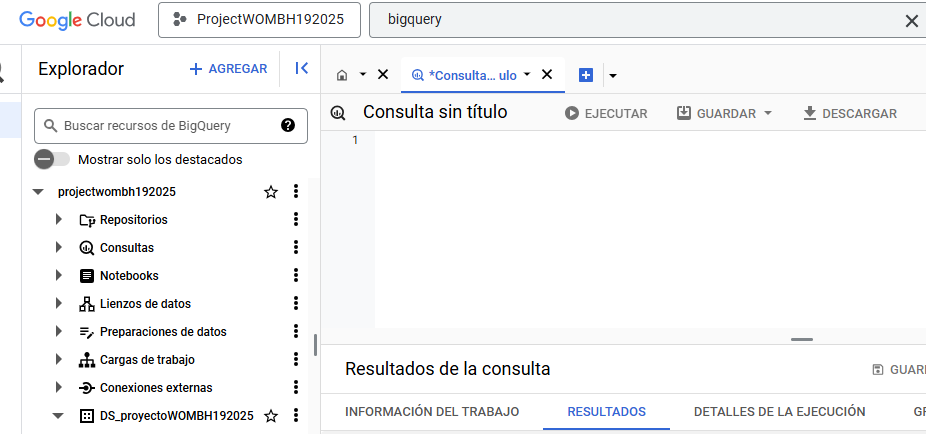
3. Construya un DAG de Airflow que utilice operadores nativos para tomar un archivo desde un bucket e inserte en BigQuery, posteriormente ejecutar una query que transforme la información y guarde el resultado en una nueva tabla.

Paso 1: Se crea un proyecto en GCP, en el cual resolveré el enunciado:

Paso 2: Se activa la API de Storage y se crea un bucket para guardar un archivo de prueba con extención.csv

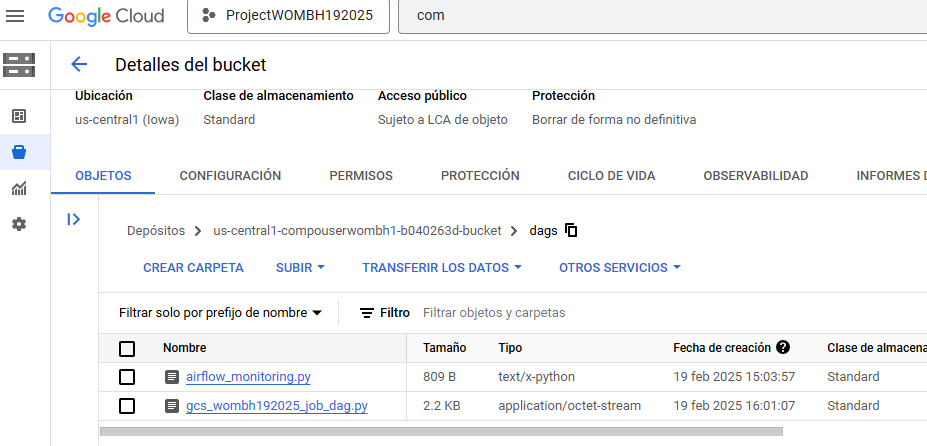


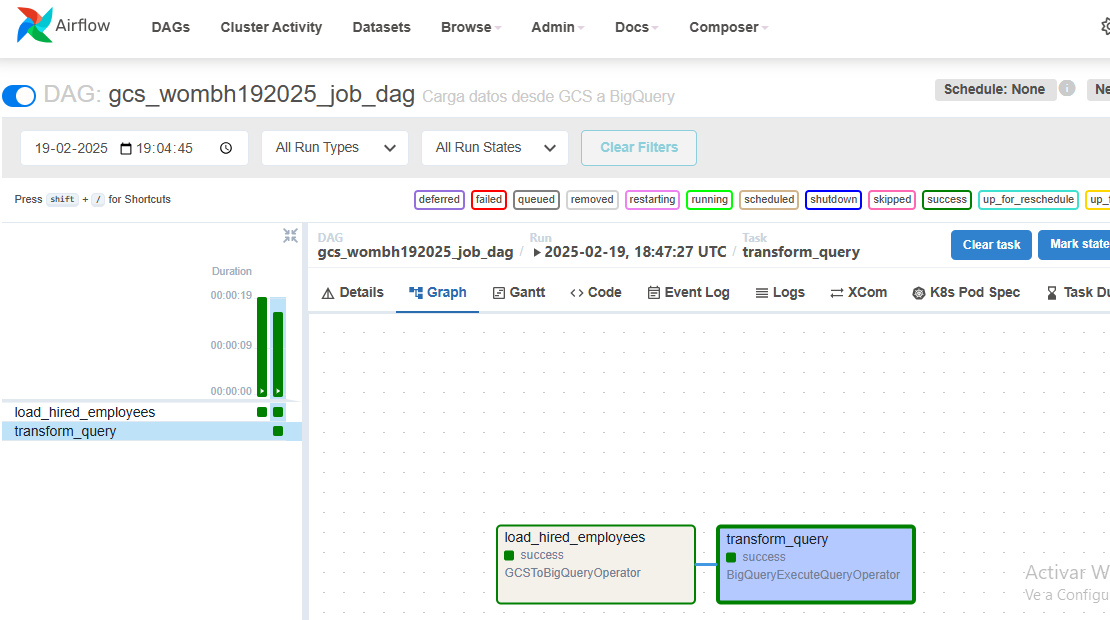
Paso 3: Se crea un Data set vacío donde luego se cargará una tabla de ejemplo de empleados y una copia modificada de la misma, la tabla original tiene departamentos nulos, que corresponde a los departamentos en los que trabajan los empleados. El cambio será que la copia tendrá valor 0 en lugar de nulos en la fila de departamentos.



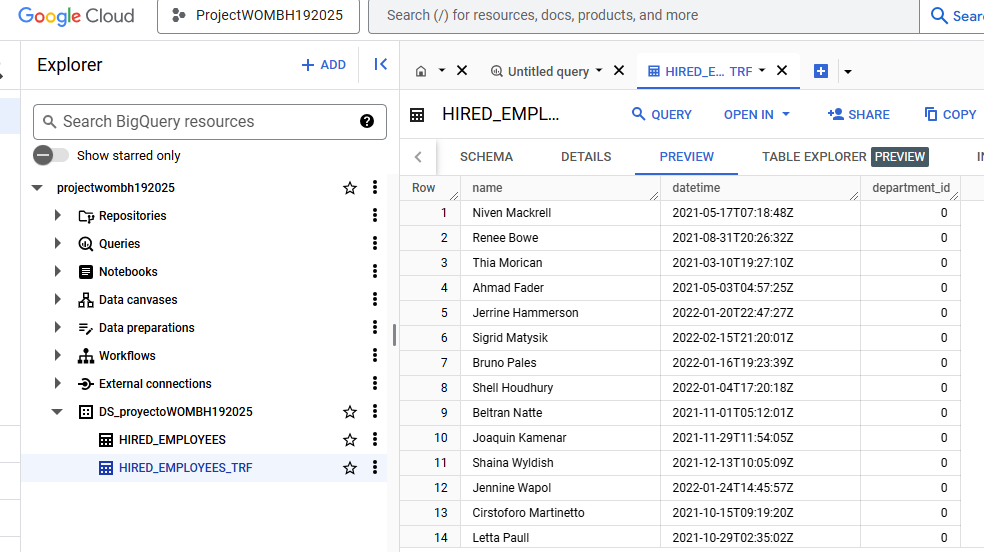
Paso 4: Se Activa a API de Compouser.

Paso 5: Se crea el Dag y se carga en el Bucket de Compouser y luego se ejecuta el Airflow

.



Paso 6: Finalmente se corrobora que exista la tabla de empleados y la copia con el cambio indicado en el código.



4- Se necesita utilizar una API para que nuestro call center llame a nuestros clientes. Esta API puede recibir un máximo de 10 request/segundo, además cada 1 hora se generan en promedio 600 registros.

a. Indicar que preguntas realizarías como especialista técnico a las áreas de negocio

R- ¿Cuál sería la estrategia en caso de fallas?

¿las llamadas son 7 días a la semana o tiene tiempos de intermitencias en la semana?

¿La información histórica de las llamadas será consultada posteriormente en forma periódica?

b. Preguntas al proveedor de la API:

R- ¿Cuáles son las normas de seguridad para el manejo de la información?

¿se podría agregar en el futuro nuevos máximos de llamadas para estar preparados en aumentar en procesamiento?

c. Arquitectura propuesta:

R. Según la documentación de google en estos casos se recomienda utilizar

Cloud Scheduler: Dispara trabajos cada hora.

Cloud Pub/Sub: Publica los registros generados.

Cloud Functions: Procesa los eventos y llama a la API.

Cloud Tasks: Implementa control de tasa para cumplir con el límite de 10 requests/segundo.

Independiente de esto creo que podría haber otras arquitecturas que también podrían resolver la problemática como Dataflow o Airflow agendados cada 1 hora

d. Justificación de la arquitectura:

R- Principalmente la base de la respuesta anterior está dada por: Pub/Sub puede manejar grandes volúmenes de datos y con las Task se podría distribuir las llamadas y permite el manejo de errores como reintentos en caso de llamadas fallidas