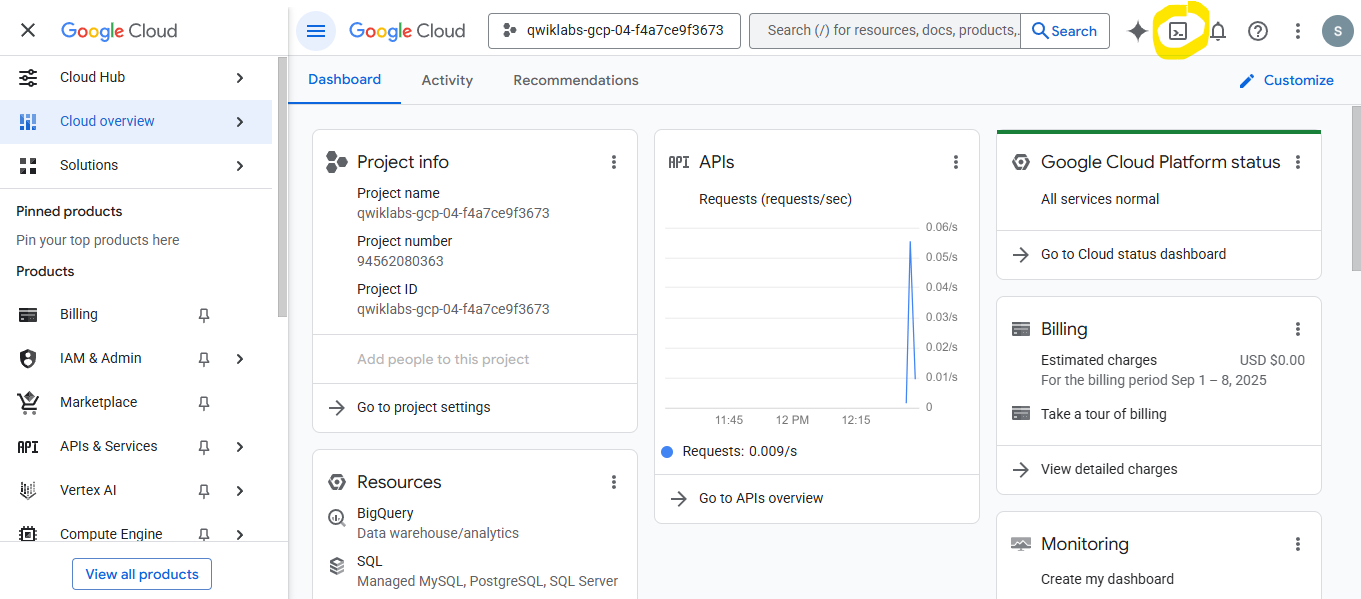
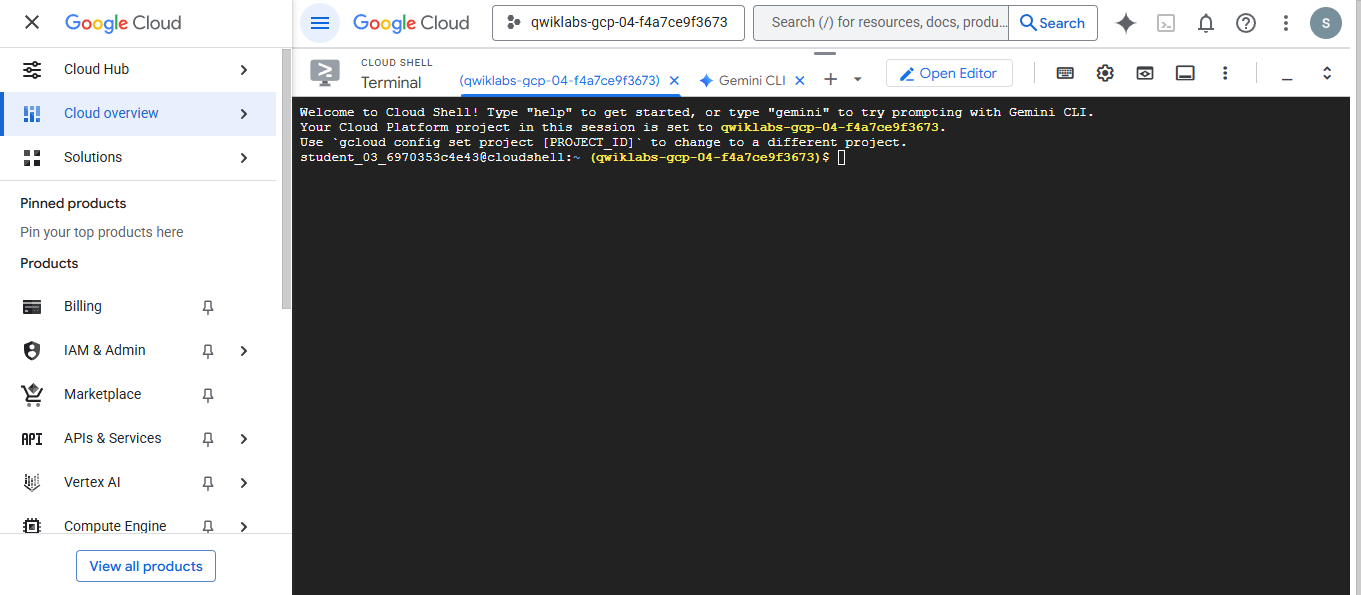
## **Task 1. Creating an Apache Beam pipeline to import data into Spanner**





Run below command to copy data

git cloneb https://github.com/pawanpant7/testd

cd understanding\_spanner/dataflow

Run the following script to create a Spanner database.

bash ./create-spanner-pets-database.sh us-east4

Run the following command to see the Schema. In this lab, there is just one table and the PetID and OwnerID fields are both integers not strings.

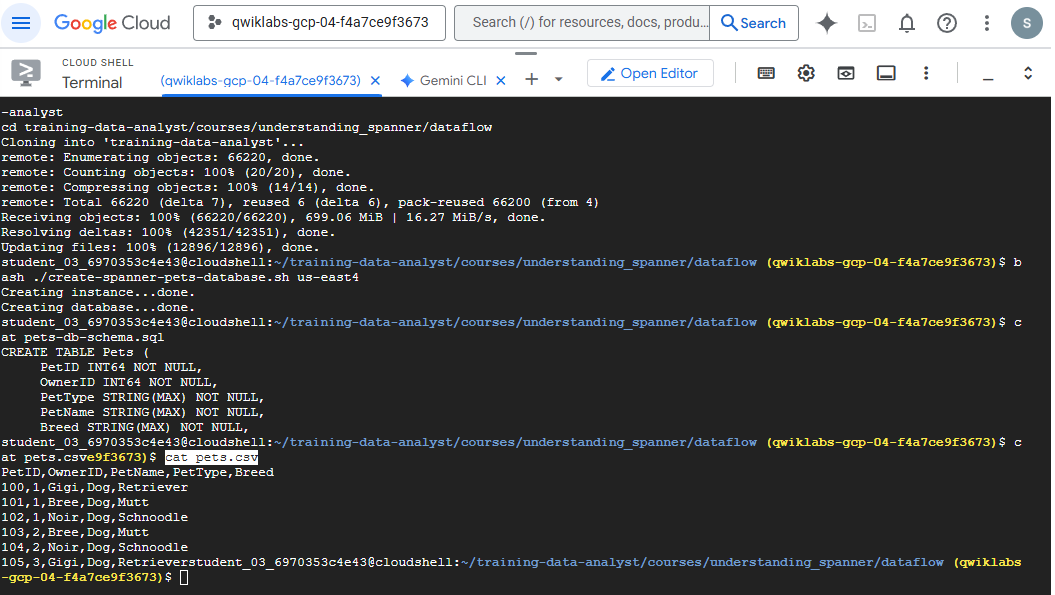
cat pets-db-schema.sql

Run the following command to see the data you import. Notice the primary and foreign keys use counters. As you learned earlier in the course, this is an anti-pattern when using Spanner.

To solve this, you use a Dataflow pipeline written in Apache Beam to reverse the bits of the integers prior to importing the data into Spanner.

This solves the problem of the integers while maintaining the relationships.

cat pets.csv



7.Click the Open Editor button and open the understanding\_spanner/dataflow/csv-to-spanner.py code file. Notice the pipeline is created in the main function (lines 53 to 68).

The pipeline reads from the CSV file, then reverses the bits on the PetID and OwnerID fields, before writing the data to Spanner.

The reverse\_bits function begins at line 21.

1. Return to the terminal. Let's try to run this pipeline. First, you must install the Python prerequisites with the following commands.

pip install apache-beam[gcp]==2.42.0

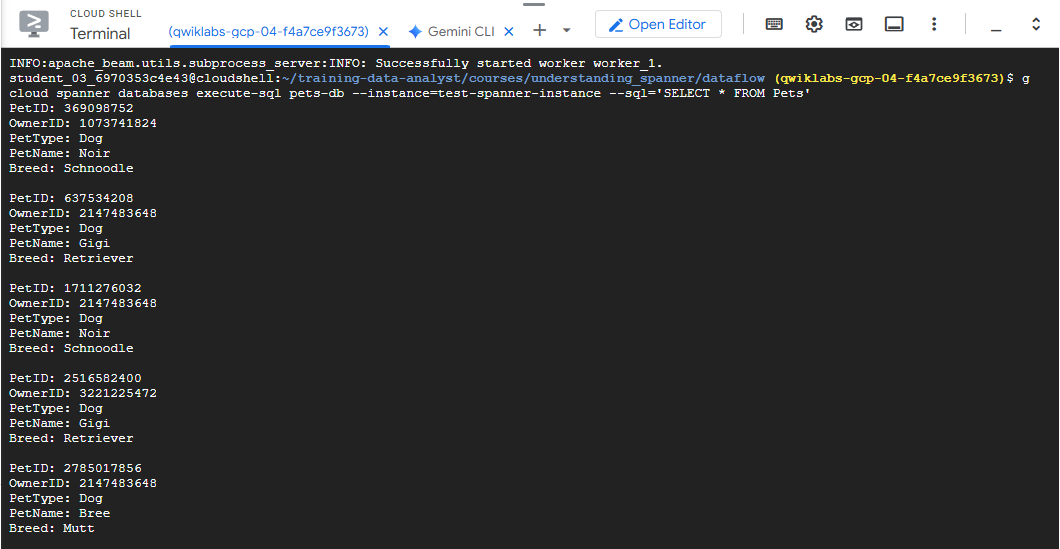
pip install apache-beam[dataframe]

1. Run the pipeline. (This code runs the pipeline locally in Cloud Shell. There is no need to navigate to Dataflow.)

python csv-to-spanner.py

When the pipeline completes, run the following query to see the results:

gcloud spanner databases execute-sql pets-db --instance=test-spanner-instance --sql='SELECT \* FROM Pets'



Run the following command to remove the data you just added:

gcloud spanner databases execute-sql pets-db --instance=test-spanner-instance --sql='DELETE FROM Pets WHERE True'

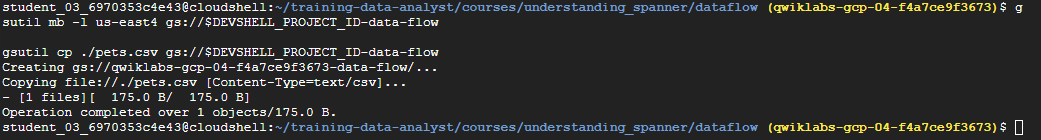
Next, you run the code using the Dataflow service.

**Task 2. Running a Dataflow job**

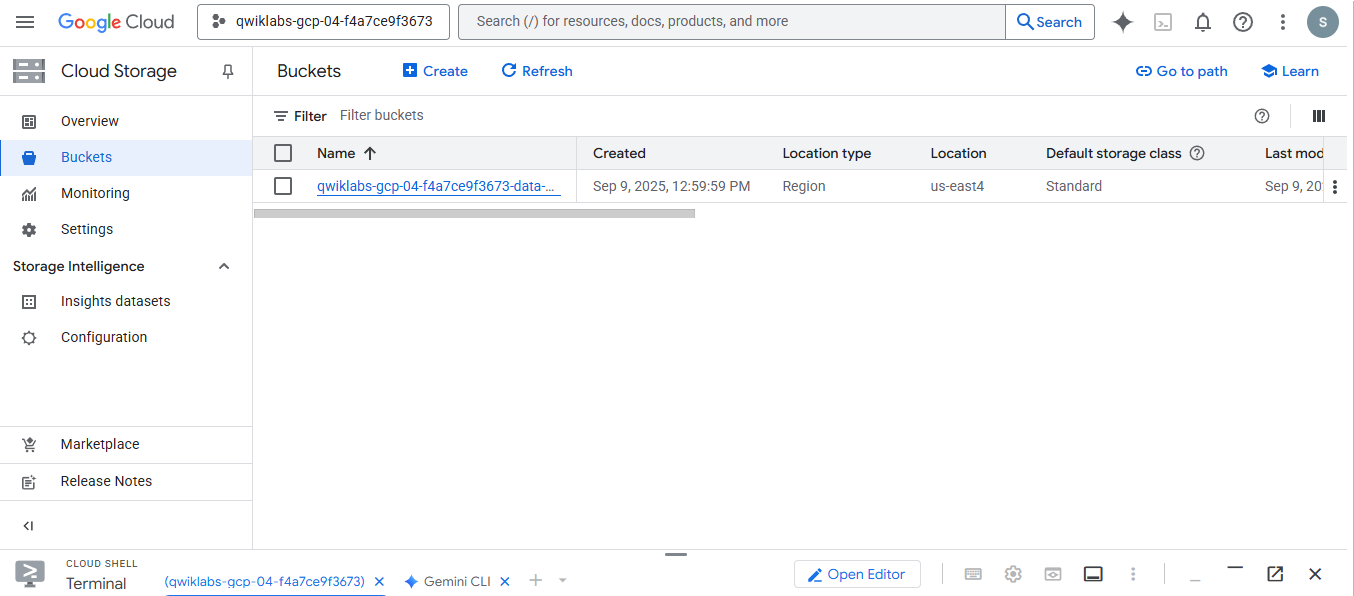
To run the job using Dataflow, you need a Cloud Storage bucket for inputs, staging, and outputs. Use the command below to create a bucket that contains your Project ID (this should guarantee a unique name for the bucket). Also, copy the pets.csv file into the bucket. Run each of these individually, not together.

gsutil mb -l us-east4 gs://$DEVSHELL\_PROJECT\_ID-data-flow

gsutil cp ./pets.csv gs://$DEVSHELL\_PROJECT\_ID-data-flow



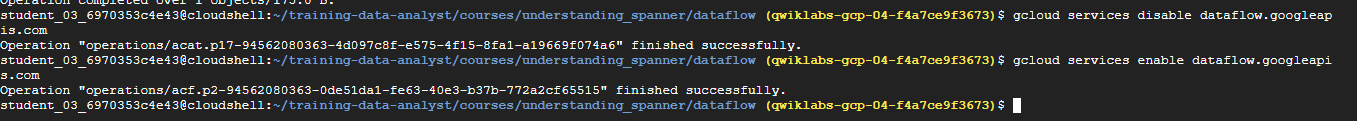
Navigate to Cloud Storage in the Console and verify that the bucket was created and the file was copied.



Run the following commands to ensure that the Dataflow API is enabled cleanly in your project.

gcloud services disable dataflow.googleapis.com

gcloud services enable dataflow.googleapis.com



Install apache-beam sdk

pip install 'apache-beam[gcp]'

Run the pipeline using Dataflow with the following command.

python csv-to-spanner.py \

--region us-east4 \

--worker\_machine\_type e2-standard-2 \

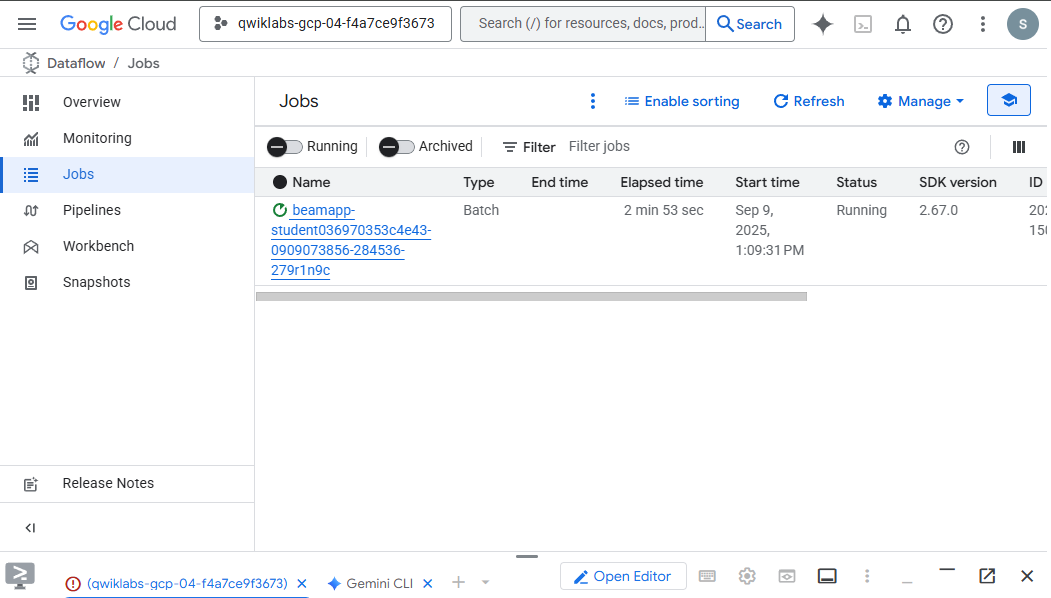
--input gs://$DEVSHELL\_PROJECT\_ID-data-flow/pets.csv \

--output gs://$DEVSHELL\_PROJECT\_ID-data-flow/results/outputs \

--runner DataflowRunner \

--project $DEVSHELL\_PROJECT\_ID \

--temp\_location gs://$DEVSHELL\_PROJECT\_ID-data-flow/tmp/



Use the Navigation menu to go to Dataflow Jobs. It may take a few moments to see the job show up, so click the Refresh button until you see it. Then you can click the job and see the job details. It takes several minutes to run the job in the Dataflow service since it creates a cluster or one or more VMs to submit the job to.

gcloud spanner databases execute-sql pets-db --instance=test-spanner-instance --sql='SELECT \* FROM Pets'

Delete the Spanner instance so you are no longer being charged for it.

Congratulations! You used Dataflow and Apache Beam to migrate data into Spanner.