**TERM PROJECT DELIVERABLES**

**Introduction**

In today's data-driven time and age, the significance of Big Data in managing large-scale data pipelines is immense. Recognizing this, I constructed a compact yet powerful Big Data pipeline using the services offered by Google Cloud. My objective was to transfer data between various stages of the pipeline while also performing essential transformations and cleansing processes and for this I have loaded the data of a csv file into a data frame and performed the necessary transformation on it and then using composer I have loaded that data into cloud spanner table. To simplify the workflow and ensure automation of data transfer, I used Google Cloud's DAG triggers. These automated jobs play an important role in performing the movement of data from one location to another, thereby enhancing the efficiency and reliability of the entire pipeline. I have also used DataFlow jobs to send data from one service to another, using dataflow jobs I have sent data from cloud spanner table to bucket and then from bucket the selective columns were loaded in BigQuery table and this data transfer was also done using Dataflow jobs. You can automate these DAGs and Dataflow jobs, by setting up a specific timeline for them to run on.

**Services used in my Project:**

I have selected Cloud Storage Bucket, Cloud Composer, Cloud Spanner, DataFlow, BigQuery. This also represents the flow of my project.

**Google Cloud Storage Bucket(GCS):**

Google Cloud Storage serves as the main storage hub for my project, providing scalable, reliable, and easily accessible storage for various data types, including the essential CSV files in my workflow. In my project I have used GCS to store my input csv files that later get loaded into Cloud Spanner table & GCS is where I store the files I get as output from the Cloud spanner table. With GCS, because of its widespread network I can effortlessly store and retrieve large volumes of unstructured data. Its durability ensures the safety of my data against hardware failures. Additionally, Google Cloud Storage seamlessly integrates with other Google Cloud services, providing a smooth experience for building and managing data pipelines.

**Cloud Composer:**

I used Cloud Composer to automate the data pipeline. Apache Airflow, provided by Cloud Composer helped me, schedule, and monitor processes with greater flexibility. Within Cloud Composer, I've stored my Python file, serving as the back bone of the whole process. This file also constructs a DAG (Directed Acyclic Graph), which outlines the sequence of tasks in my pipeline. When the DAG is triggered the code picks the CSV file from the bucket and performs transformations and cleaning on the data and then load the cloud spanner table with the clean data.

**Cloud Spanner:**

I used Cloud Spanner to store the cleaned and transformed data because it is scalable & Relational database services also offer horizontal scalability which will help in handling the data even if it is growing exponentially without compromising performance. Moreover it provides consistency which ensures my data remains accurate and up to date. Therefore it is a good option for storing structured data from my CSV files.

**DataFlow:**

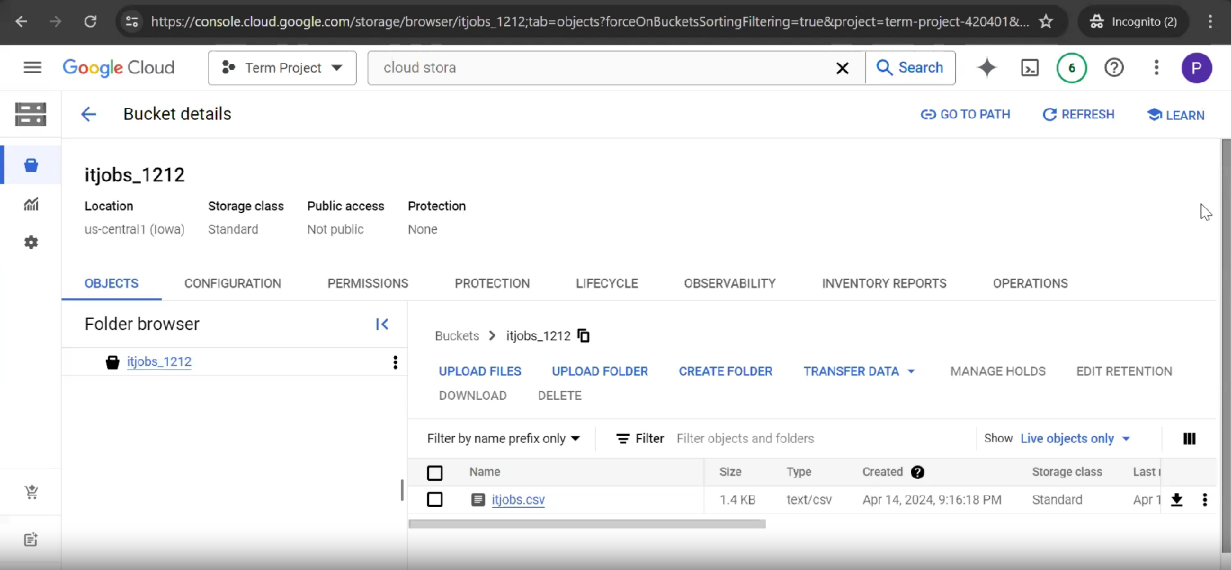
I used Google Cloud Dataflow to perform ETL operations on my data. This service provides data processing with features like serverless execution and also simplifies the task of managing data processing. With the help of Dataflow, I easily convert data from Cloud Spanner into a CSV file and store it in the Google Cloud Bucket.

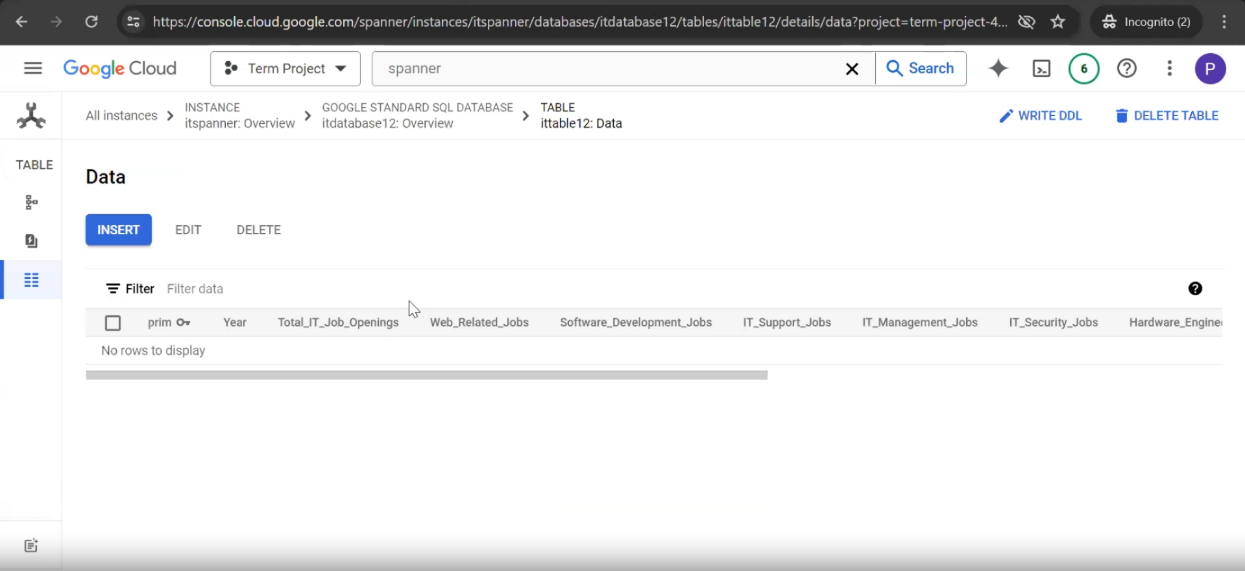
**BigQuery:**

Google BigQuery helps me store and analyze the processed data from my pipeline. As it is very scalable, importantly its capability to integrate Generative AI for query analysis is very helpful to draw insights from the data. Also, its ability to perform faster computations will help me in handling large amount of data without lag. BigQuery will not only be useful for me in storing the data but also to analyse the stored data efficiently.

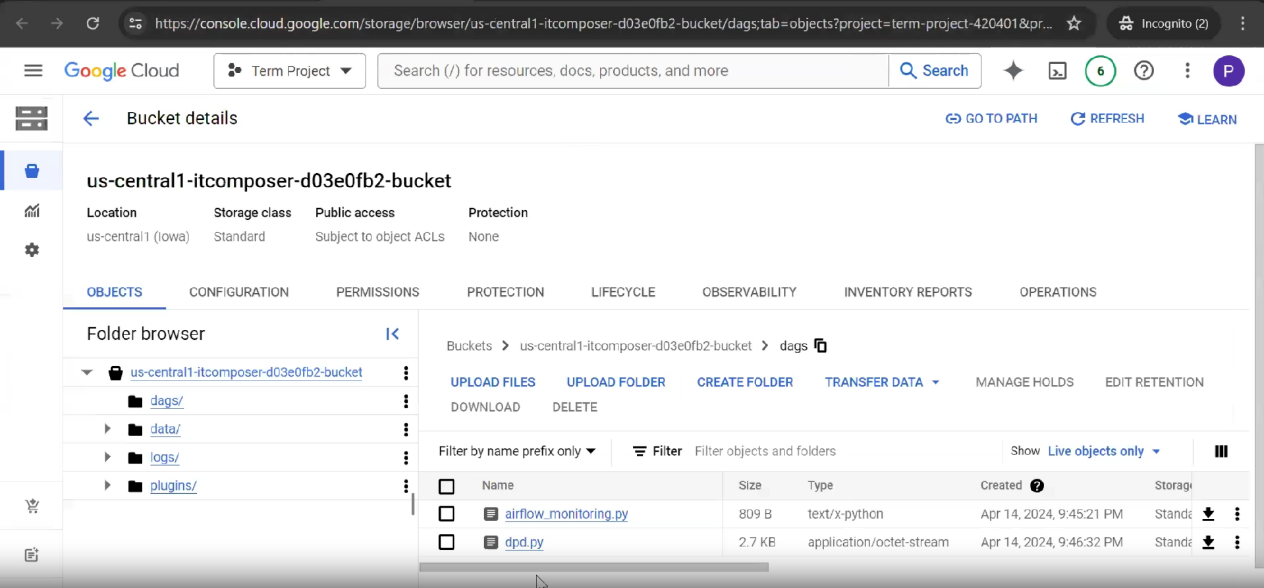
**Screenshots of my service in action**

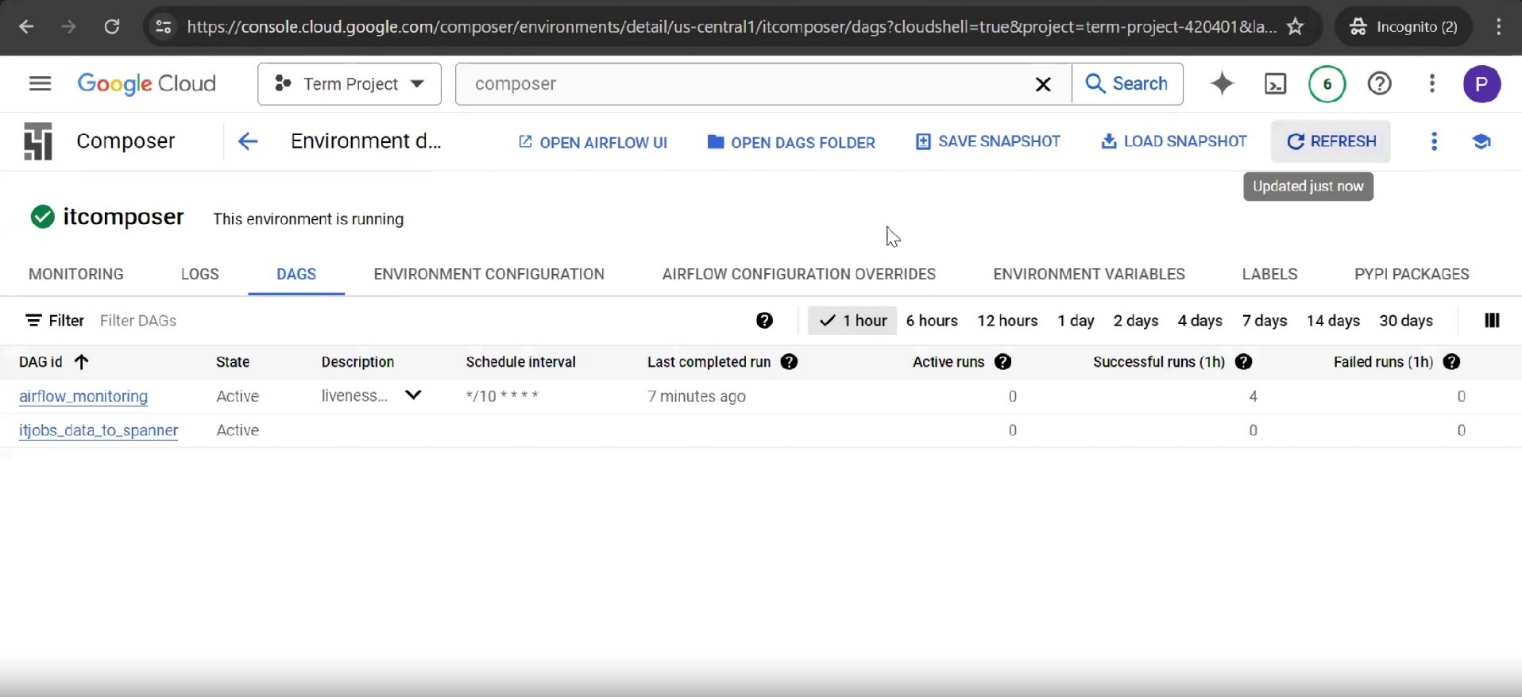
This is a screenshot of itjobs\_1212 bucket with the csv file itjobs.csv which is ready to be loaded into the cloud spanner table.

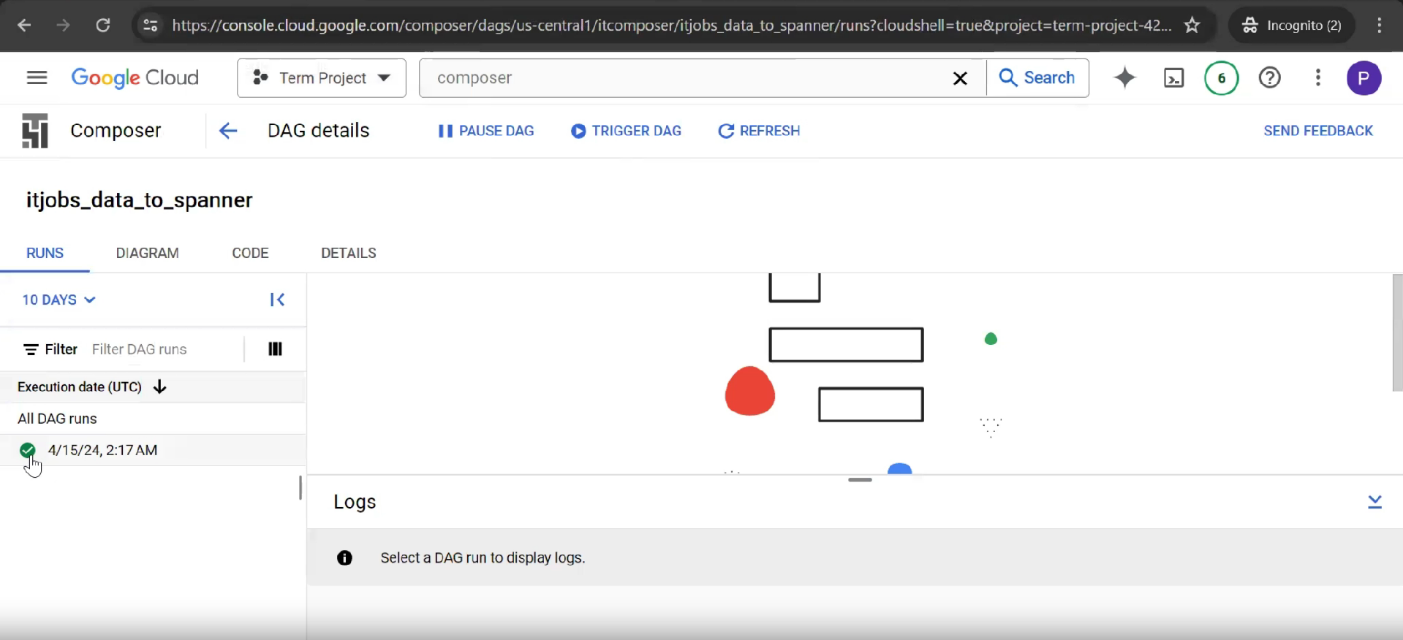


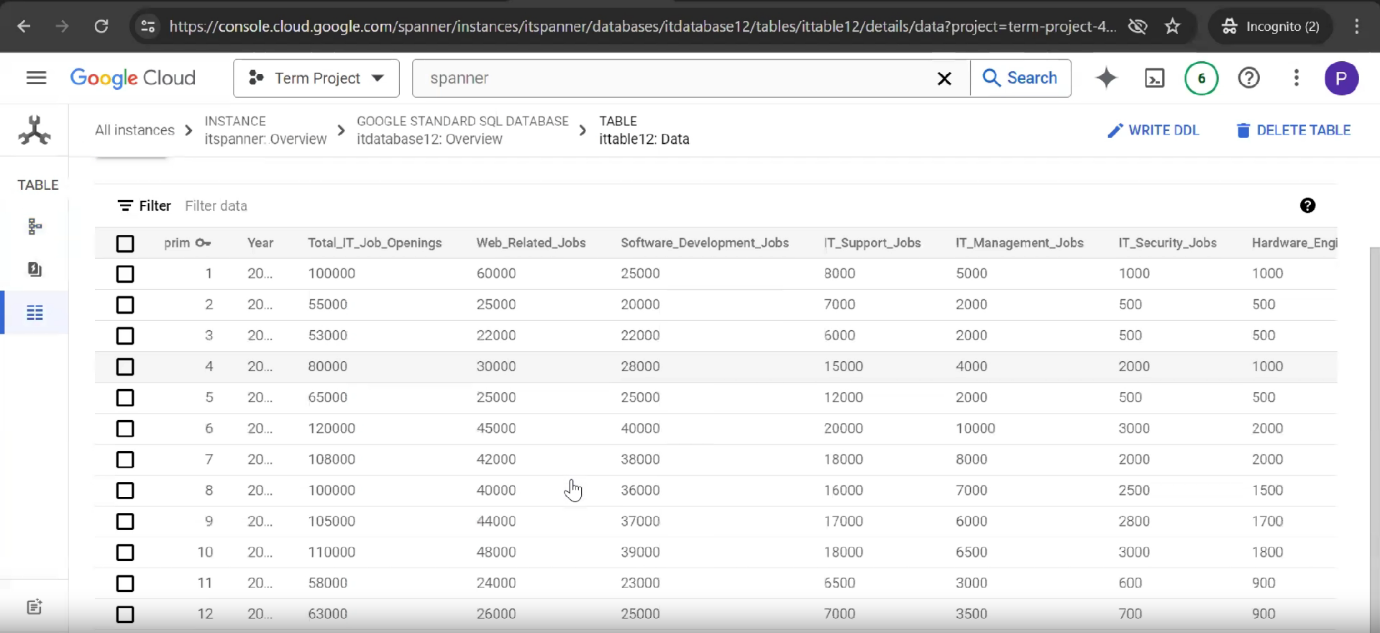
As you can see the table ittable12 present in cloud spanner is empty and ready to be loaded.

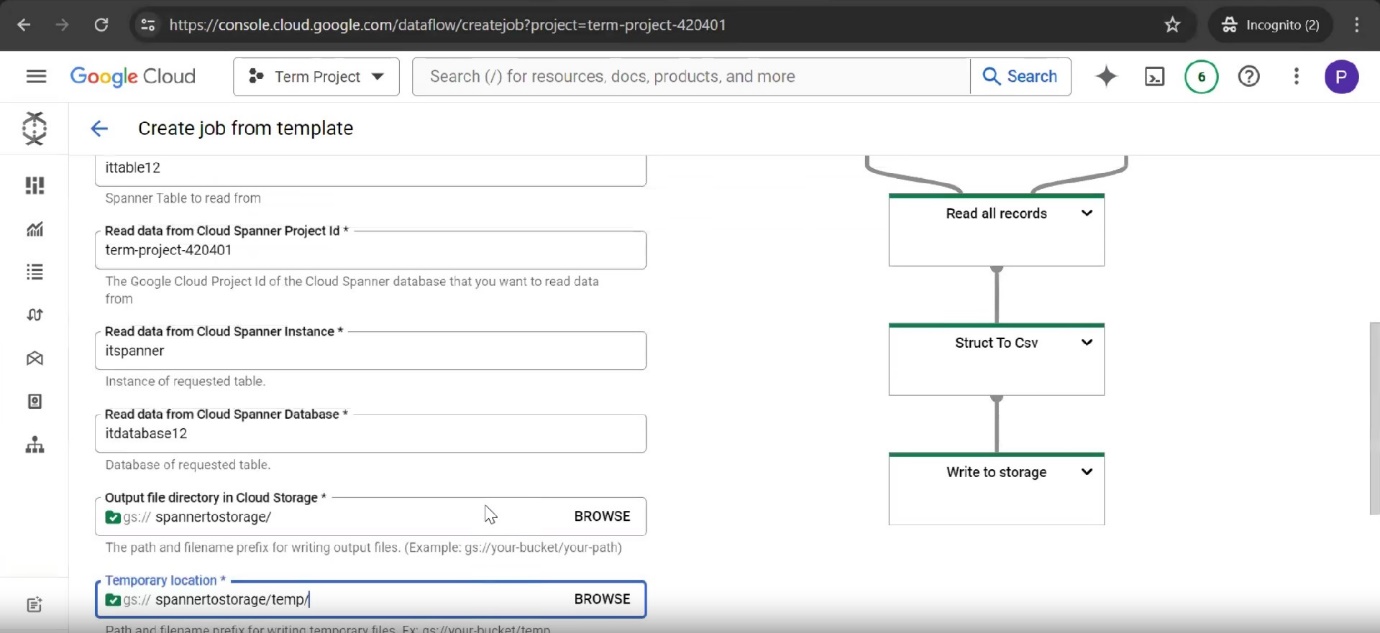
Cloud Composer containing the file “dpd.py” which has all the functionality of sending data from bucket to table.

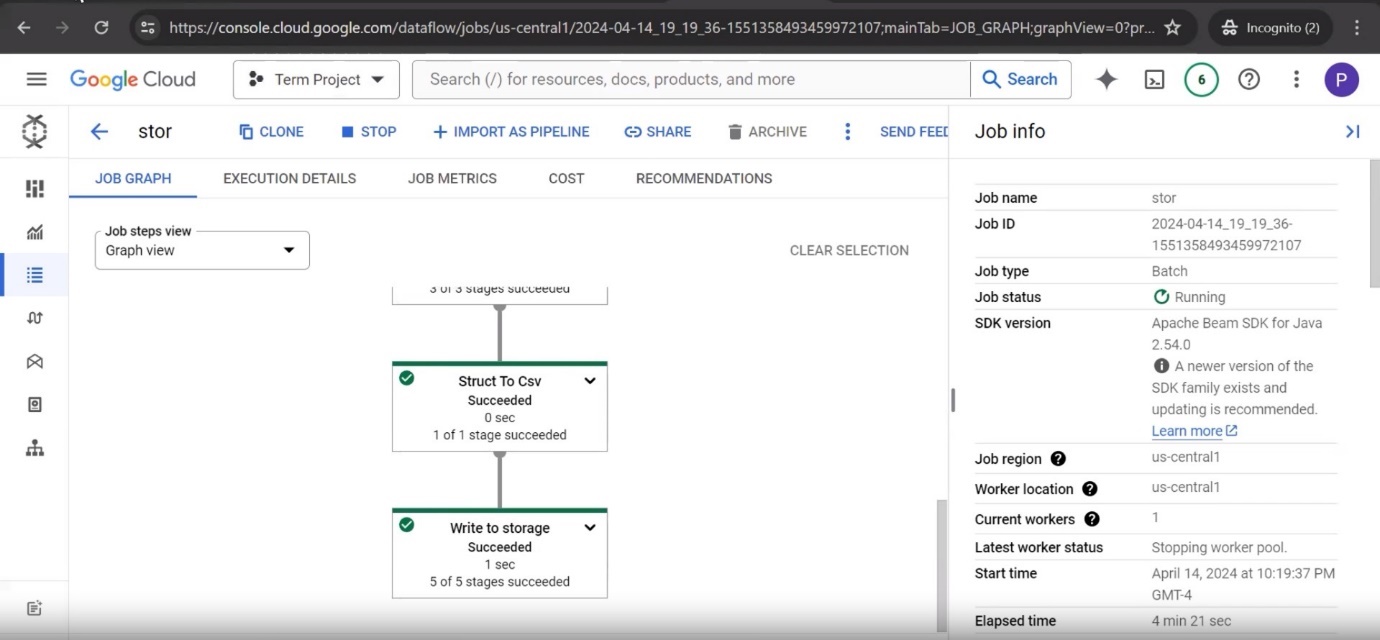


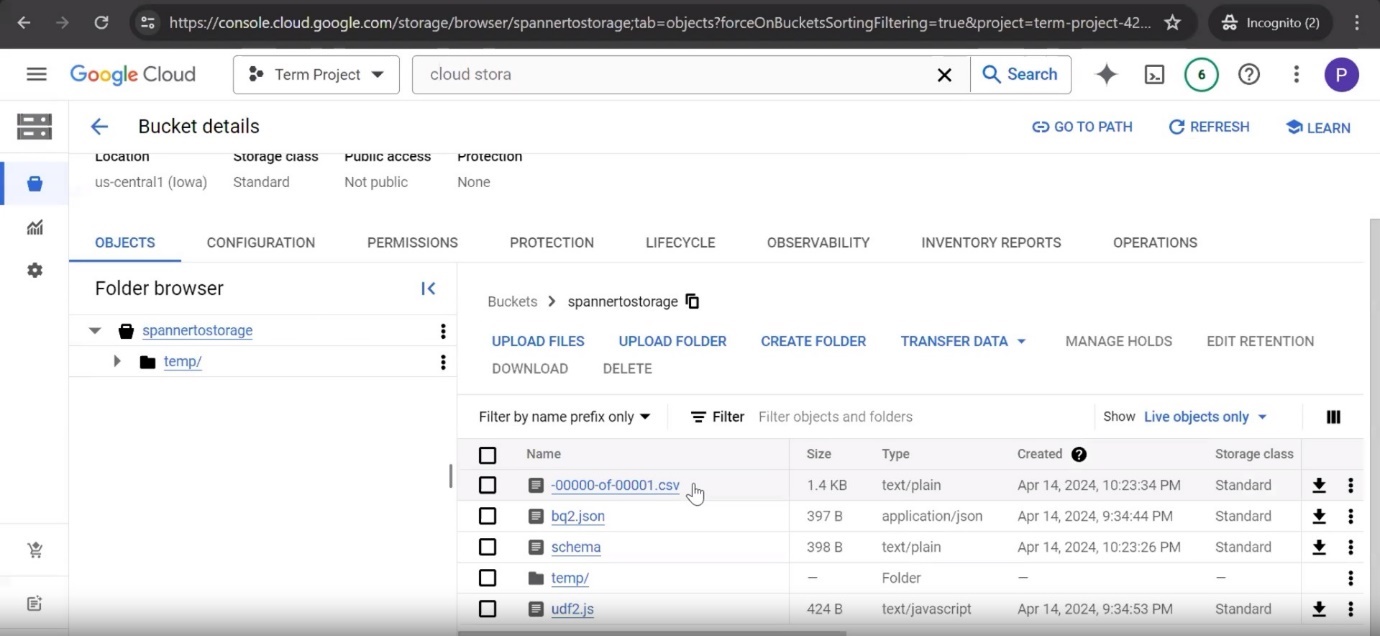
“itjobs\_data\_in\_spanner” is the DAG that is created to trigger the operations in a order of sequence.

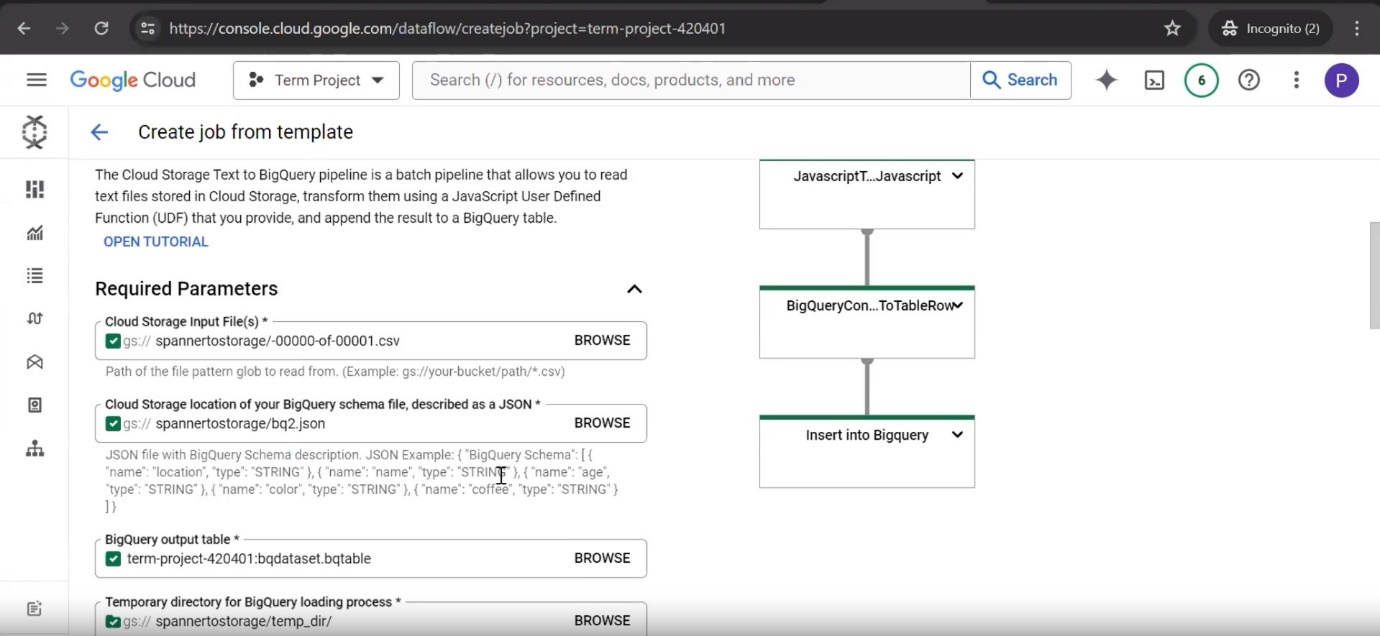
Screenshot of DAG ran successfully. 

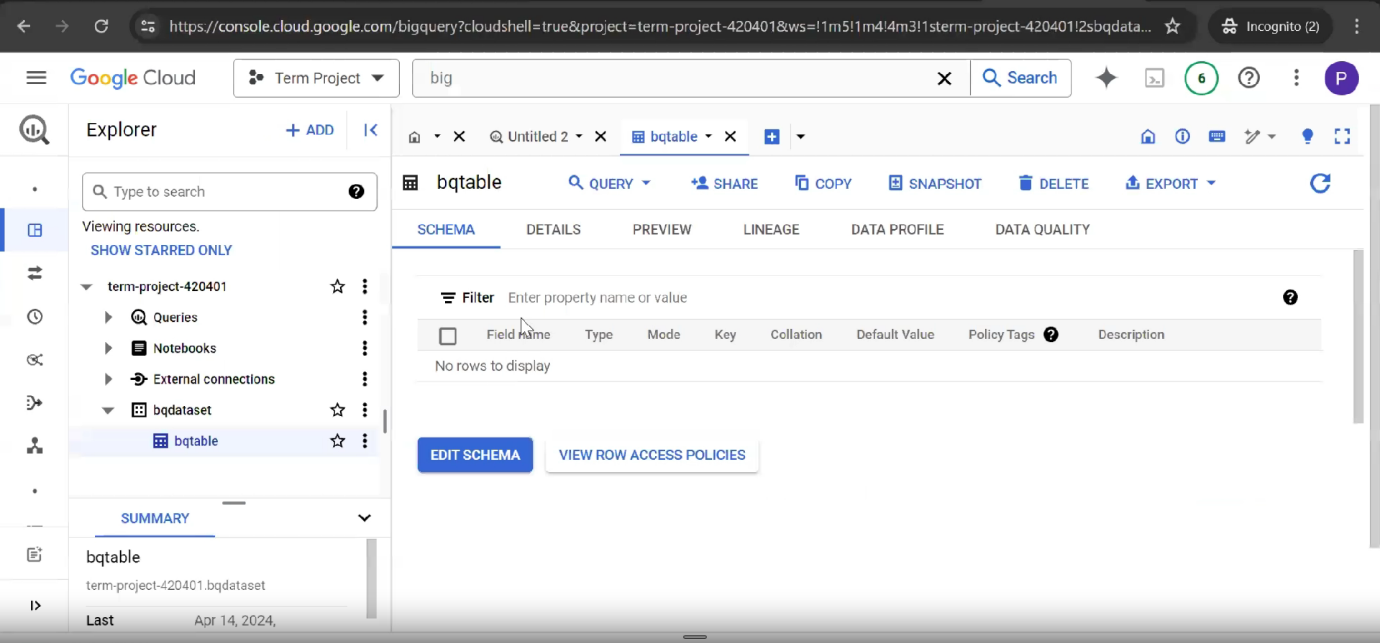
After the DAG ran successfully, Cloud spanner table got populated with values.

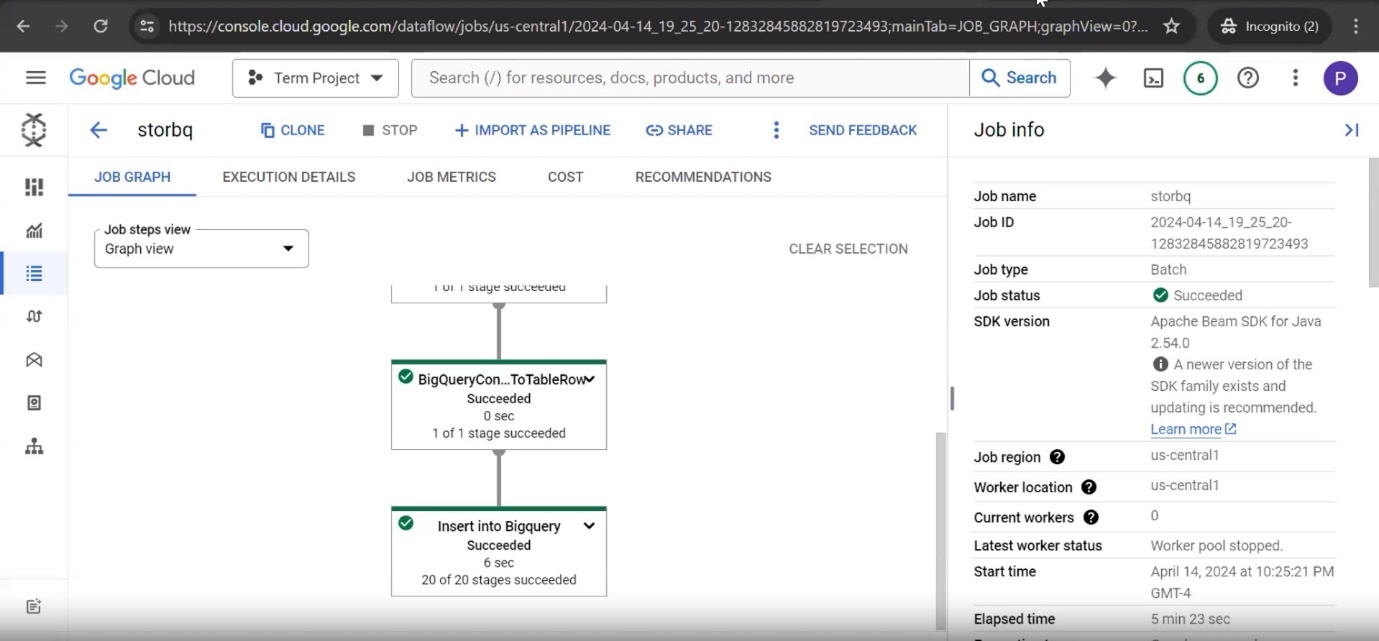
Creation of DataFlow Job to send data from the cloud spanner table to csv file and store it in a bucket.

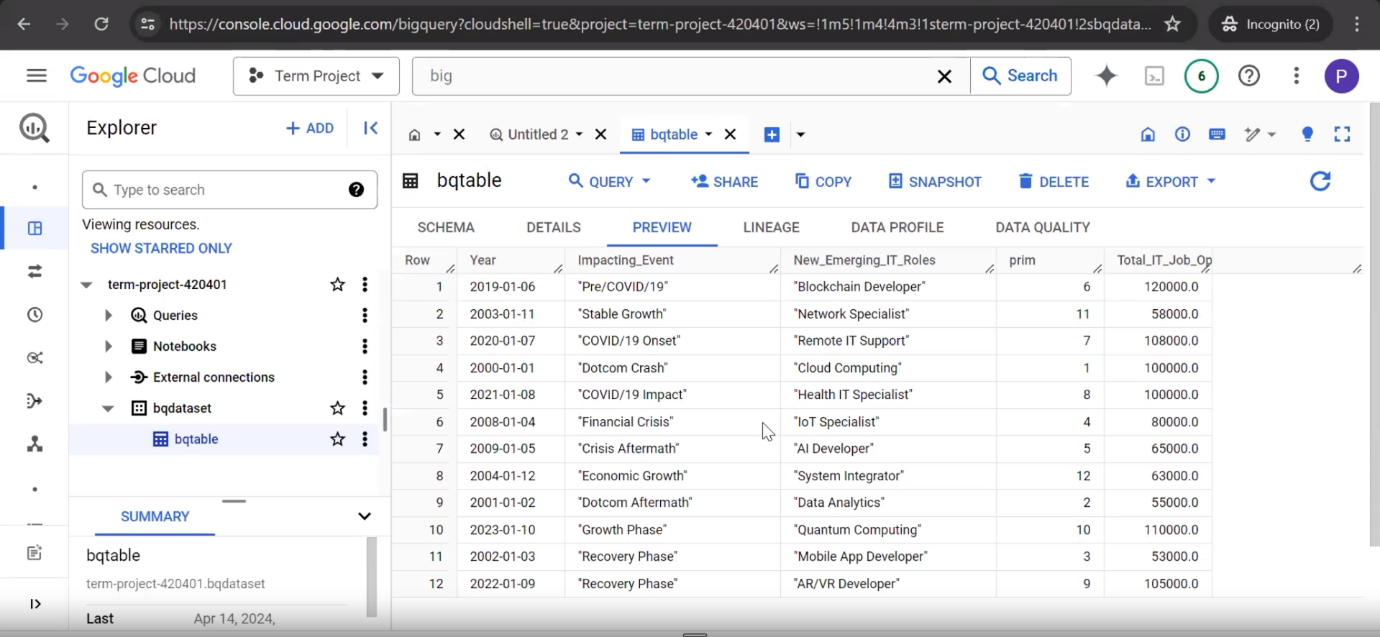
Dataflow Job ran successfully

File got loaded in the cloud storage bucket “spannertostorage”

Creating a dataflow Job to send data from the cloud storage to BigQuery table

Screenshot of BigQuery table with no records before the job ran.

Dataflow Job Ran successfully

BigQuery Table with values after getting loaded.

**Usability and future scope for implementation**

Reflecting on my project's utility, I find it incredibly valuable for automating data processing tasks and extracting insights from large datasets efficiently. Through the integration of various Google Cloud services, I've developed an automated pipeline that handles data ingestion, transformation, storage, and analysis seamlessly, saving both time and resources while ensuring data accuracy and reliability.

Considering future prospects, there are several possibilities for extending this project. Firstly, enhancing the pipeline's scalability to accommodate even larger datasets by optimizing resource allocation and implementing parallel processing techniques would be beneficial. We can also integrate machine learning models into the pipeline which would help in predictive analytics & advanced data analysis, helping in making data-driven decisions. We can also use it for real-time data processing. When we involve larger datasets we also need to add security measures like encrypting the stored data, authorized access, having data backups etc to make it a secure & reliable space. When we build such an environment it can be used by various business' as they can identify the target audience for their products and this can help in increasing their sales. They will also get an insight on customers’ preferences.