

Cloud Data ORCHESTRATION: FROM buckets to bigquery with dataflows and cloud functions

Final Evaluation



A logo of a cloud

Description automatically generated

SUBMITTED BY

RAGI MADHUSRI (2320460)

MUNSHI SAI BEHARA (2320746)

BHARATH BOLLINANI (2320821)

PASUPULETI VENKATA SWAMY (2320477)

LAKKARAJU ANJANEYA VARA PRASAD (2320471)

AYYASAMY M (2320474)

**INTRODUCTION**

GCP stands for Google Cloud Platform. It is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its own products, such as Google Search and YouTube. GCP provides a wide range of services, including:

1. **Compute:** Virtual machines, containers, and serverless computing
2. **Storage:** Object storage, file storage, and block storage
3. **Databases:** Relational databases, NoSQL databases, and managed database services
4. **Networking:** Virtual private clouds, load balancers, and firewalls
5. **Big data:** Data analytics, machine learning, and artificial intelligence
6. **Management tools:** Monitoring, logging, and billing

GCP is a popular choice for businesses of all sizes because it is:

1. **Reliable:** GCP is built on Google's own infrastructure, which is one of the most reliable and scalable in the world.
2. **Scalable:** GCP can handle workloads of any size, from small businesses to large enterprises.
3. **Cost-effective:** GCP is a cost-effective way to build and run applications in the cloud.
4. **Easy to use:** GCP has a simple and intuitive user interface, making it easy to get started with cloud computing.

**ABSTRACTION:**

This project focuses on automating the process of ingesting, processing, and loading data from Google Cloud Storage buckets into BigQuery using Dataflow and Cloud Functions. The workflow involves triggering Cloud Functions in response to new files being uploaded to a bucket, which then initiates a Dataflow job to transform and load the data into BigQuery. The project aims to demonstrate efficient data orchestration techniques in a cloud environment, showcasing the integration of various Google Cloud Platform services.

**Key Components**:

1. **Google Cloud Storage Buckets**: Used for storing raw data files.
2. **Cloud Functions**: Triggered by new file uploads, initiating the data processing pipeline.
3. **Dataflow**: Processes and transforms the data from Cloud Storage and loads it into BigQuery.
4. **BigQuery**: Serves as the data warehouse for the processed data.

**Objectives**:

1. Automatically trigger data processing tasks in response to new data uploads.
2. Perform ETL (Extract, Transform, Load) operations on the data using Dataflow.
3. Load transformed data into BigQuery for analysis and reporting.
4. Demonstrate the scalability and efficiency of cloud-based data orchestration using Google Cloud Platform services.

**CLOUD STORAGE:**

Cloud storage is a cloud computing model that stores data on the internet through a third-party cloud computing provider. This means that you can access your data from anywhere with an internet connection, and you don't have to worry about maintaining your own physical storage devices.

Cloud storage is often used for:

1. **Data backup and recovery**: Cloud storage can be used to back up your data in case of a hardware failure or other disaster.
2. **Archiving:** Cloud storage can be used to store data that you don't need to access frequently, but that you want to keep for future reference.
3. **Media storage:** Cloud storage can be used to store photos, videos, and other media files.
4. **Website hosting:** Cloud storage can be used to host websites and other online content.
5. **Big data analytics:** Cloud storage can be used to store and process large amounts of data for analytics purposes.

**Features of cloud storage:**

1. **Scalability:** Cloud storage can be scaled up or down to meet your needs.
2. **Durability:** Cloud storage is designed to be durable and reliable, so you can be confident that your data will be safe.
3. **Security:** Cloud storage providers typically have robust security measures in place to protect your data from unauthorized access.
4. **Cost-effectiveness:** Cloud storage is often more cost-effective than traditional on-premises storage solutions.
5. **Ease of use:** Cloud storage is easy to use and manage, so you can get started quickly and easily.

**Pricing:**

1. Cloud Storage pricing is based on the amount of data you store and the number of operations you perform.
2. There are multiple storage classes available, each with its own pricing.

**CLOUD FUNCTIONS:**

1. Cloud Functions is a serverless compute platform that allows you to build and deploy event-driven functions without having to manage servers or infrastructure.
2. With Cloud Functions, you can write code that responds to events such as HTTP requests, Cloud Storage bucket changes, or Pub/Sub messages.
3. Cloud Functions is a fully managed service, so you don't have to worry about scaling, security, or availability. Cloud Functions will automatically scale your functions to meet demand, and it will handle all the underlying infrastructure so that you can focus on writing code.
4. Cloud Functions is easy to use and get started with. You can write your functions in JavaScript, Python, Go, or Java, and you can deploy them with a single command.
5. Cloud Functions also integrates with a variety of other Google Cloud services, so you can easily build complex applications that leverage the power of the cloud.

**Features of Cloud Functions:**

1. **Serverless:** Cloud Functions is a serverless platform, so you don't have to worry about managing servers or infrastructure.
2. **Scalable:** Cloud Functions will automatically scale your functions to meet demand.
3. **Secure:** Cloud Functions is a fully managed service, so you can be confident that your functions are secure.
4. **Cost-effective:** Cloud Functions is a cost-effective way to build and deploy event-driven applications.
5. **Easy to use:** Cloud Functions is easy to use and get started with.

**Pricing:**

The pricing for Google Cloud Functions in GCP varies based on a few factors, including the amount of memory and CPU allocated to the function, the number of invocations, and the duration of each function execution. Here is a general overview:

1. **Invocations**: You are charged based on the number of function invocations, with the first 2 million invocations per month being free.
2. **Execution Time**: You are charged based on the total time your function takes to execute, rounded up to the nearest 100 milliseconds. The pricing varies based on the amount of memory allocated to the function.
3. **Memory and CPU**: You are charged based on the amount of memory and CPU allocated to the function, with different pricing tiers for different levels of memory and CPU.
4. **Networking**: You may be charged for outbound networking data transfer from your function to the internet.
5. **Free Tier**: There is a free tier that includes a certain number of invocations and compute time per month.

**DATA FLOW:**

1. In Google Cloud Platform (GCP), data flow typically refers to the movement and processing of data within GCP services. One key component for managing data flow in GCP is Cloud Dataflow, a fully managed service for stream and batch processing.
2. Cloud Dataflow allows you to create data pipelines that ingest, transform, and analyze data in real-time or in batch mode.
3. It supports popular programming models like Apache Beam, which provides a unified programming model for both batch and stream processing.

GCP offers several other services for managing data flow, such as:

**Pub/Sub:** A messaging service for building event-driven systems that can ingest and deliver messages from various sources to various destinations.

**Data Transfer Service:** Allows you to transfer data from online and on-premises sources to Google Cloud Storage, Big Query, or Cloud Storage.

**Big Query:** A serverless, highly scalable, and cost-effective data warehouse for analyzing large datasets using SQL queries.

**Dataflow Shuffle:** A service for managing shuffle operations in Dataflow jobs, which is crucial for operations like grouping and joining data in parallel processing.

**Features of Data Flow:**

**Fully managed:** You don't have to manage any infrastructure.

**Scalable:** Dataflow can process large amounts of data in real time.

**Flexible:** Dataflow can be used to perform a variety of data processing tasks.

**Cost-effective:** Dataflow is a cost-effective way to process data.

**Pricing:**

Google Cloud Dataflow pricing is based on the following factors:

* Worker hours: Dataflow charges based on the number of worker instances used and the amount of time they run, measured in hour increments. There are different pricing tiers for different types of workers (standard vs. high-performance) and regions.
* Streaming and batch processing: Streaming and batch processing have different pricing models. For batch processing, you are charged based on the number of worker hours used. For streaming, you are charged based on the number of worker hours as well as the amount of data processed.
* Data processed: For batch processing, you are charged based on the amount of data processed by your job. For streaming, you are charged based on both data processed and the duration of your streaming job.
* Networking: Dataflow also charges for data transfer between regions and between the internet and Google Cloud.
* Regional pricing: Prices can vary by region.

**BIG QUERY:**

Big Query is a fully managed, serverless data warehouse that enables scalable analysis of large datasets. It is designed for business intelligence and data analytics, and can be used to analyze data from a variety of sources, including:

* Cloud Storage
* Bigtable
* Pub/Sub
* Cloud SQL
* Google Analytics

Big Query is a powerful tool for data analysis, and can be used to perform a variety of tasks, including:

* Data exploration and visualization
* Data mining and machine learning
* Business intelligence and reporting
* Data warehousing

Big Query is a cost-effective and scalable solution for data analysis. It is a valuable tool for businesses that need to gain insights from their data.

**Features of big query:**

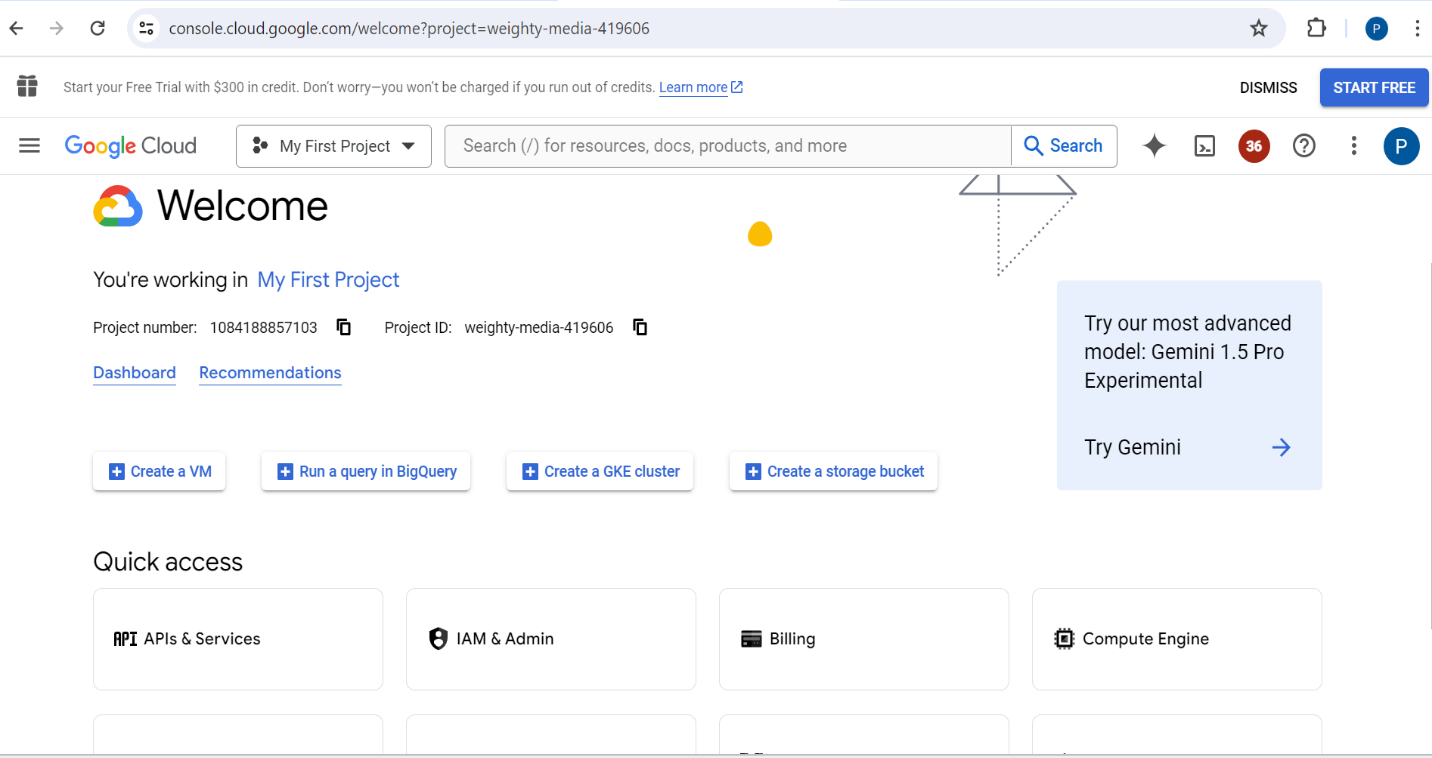
* **Fully managed:** You don't have to manage any infrastructure.
* **Scalable:** Big Query can handle large datasets and complex queries.
* **Cost-effective:**Big Query is a cost-effective way to analyze data.
* **Easy to use:** Big Query has a simple and intuitive user interface.

**Pricing:**

1. **Storage**: You are charged for data stored in your tables, calculated on a per-month basis. Big Query uses a columnar storage format, which can lead to more efficient storage compared to traditional row-based databases.
2. **Queries**: You are charged for data processed by your queries, measured in bytes processed. The first 1 TB of query data processed per month is free.
3. **Streaming inserts**: If you use streaming inserts to insert data into Big Query, you are charged based on the amount of data streamed.
4. **Data egress**: You may be charged for data egress (data transferred out of Big Query) to the internet or other Google Cloud regions.
5. **Data transfer**: You may be charged for data transferred into Big Query from external sources.
6. **Long-term storage**: If you choose to store data in long-term storage (over 90 days), there may be additional charges.
7. **Reserved capacity**: You can purchase reserved capacity to reduce the cost of queries and streaming inserts.
8. **Flat-rate pricing**: Big Query also offers flat-rate pricing for organizations with predictable workloads, where you pay a fixed monthly fee based on your usage.

**Implementation of the project**

On the first step we need to open the “**google cloud console**”, for the opening of the google cloud console URL is [**https://console.cloud.google.com**](https://console.cloud.google.com)

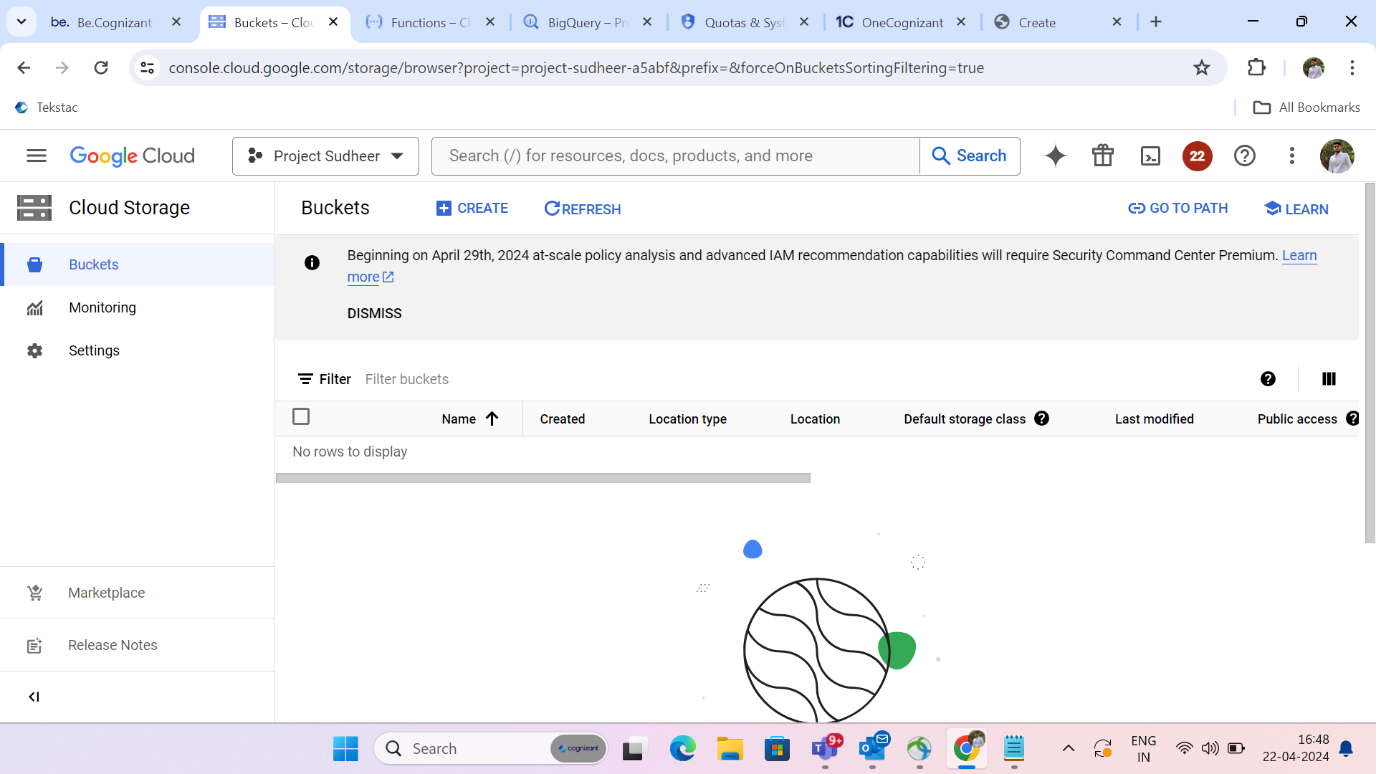


For the above google cloud account is associated with “MY First Project” and the project ID is “**weighty-media-419606**” and the project no is “**1084188857103**”

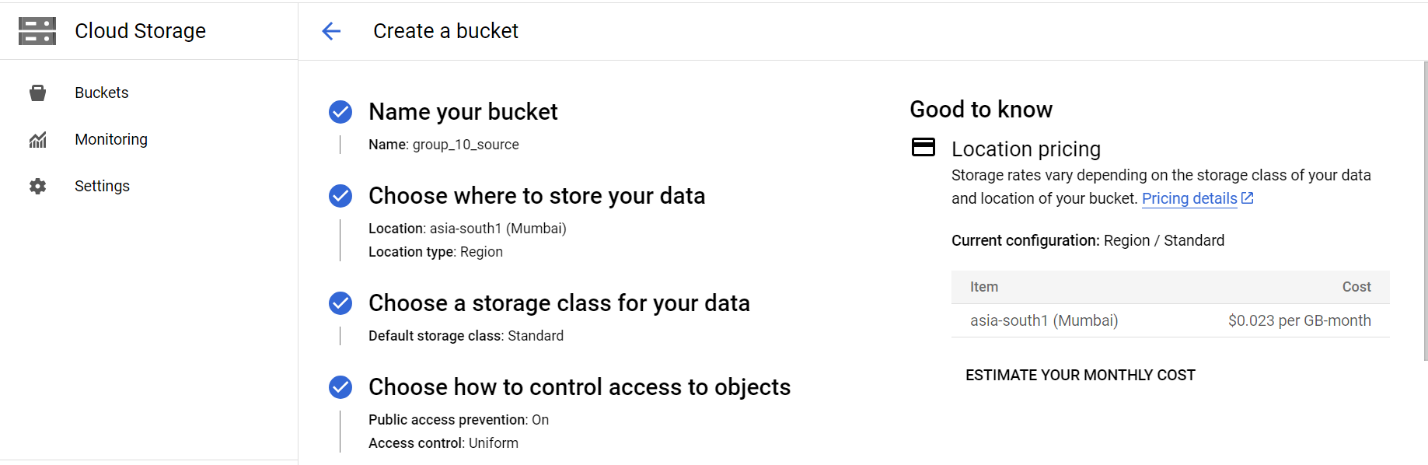
**Step 1:** Creating a bucket in the cloud storage.

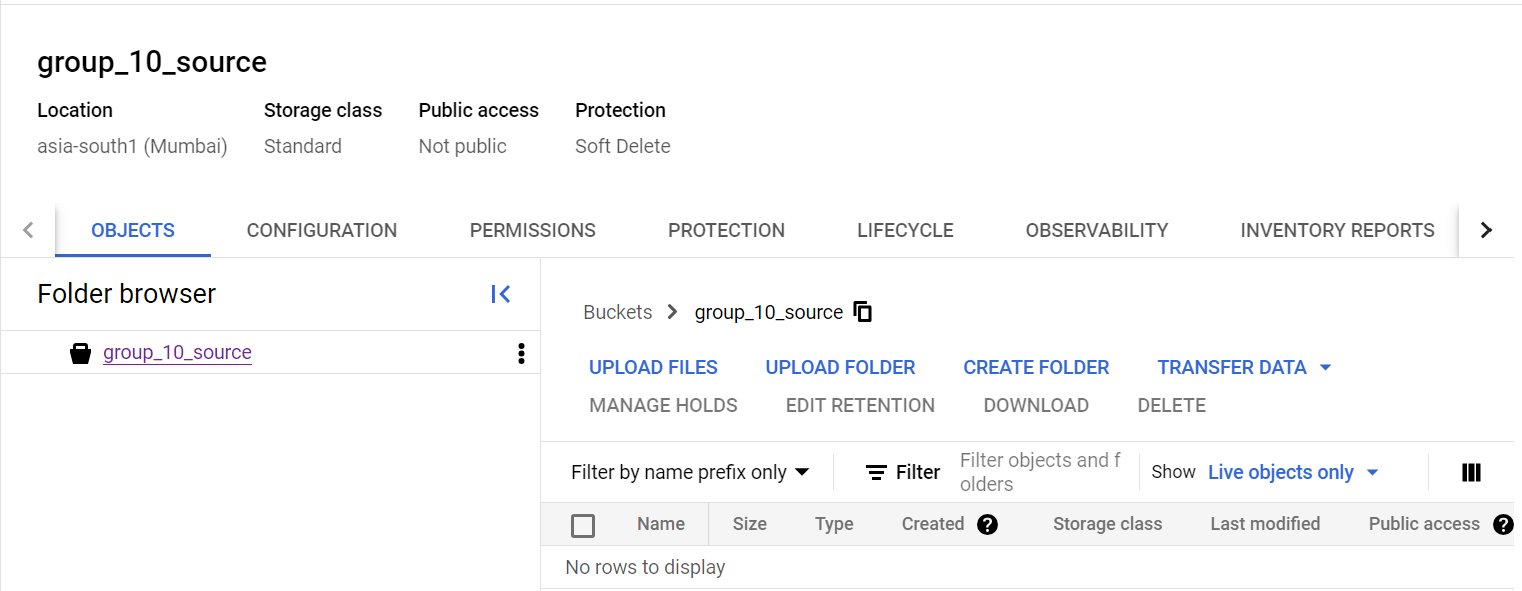
For creating a bucket in GCP, go to cloud storage service which is there in the navigation bar.

Then go to buckets and click on create.



1. Then give a name for the bucket which will be globally unique and permanent. Which means that the name cannot be changed once it is created.
2. Then after specifying the location or region where the bucket should be created.
3. Then specify the storage class for the data. The storage classes are standard, nearline, coldline, and archive.
4. We can also control the access to our objects. Access controls are uniform and fine grained.
5. Here we can see that the bucket named “**group\_10\_source**” is created.





1. In the same way create a bucket named “**group\_10\_destination**”.
2. Create both the buckets in the same region.

A screenshot of a computer

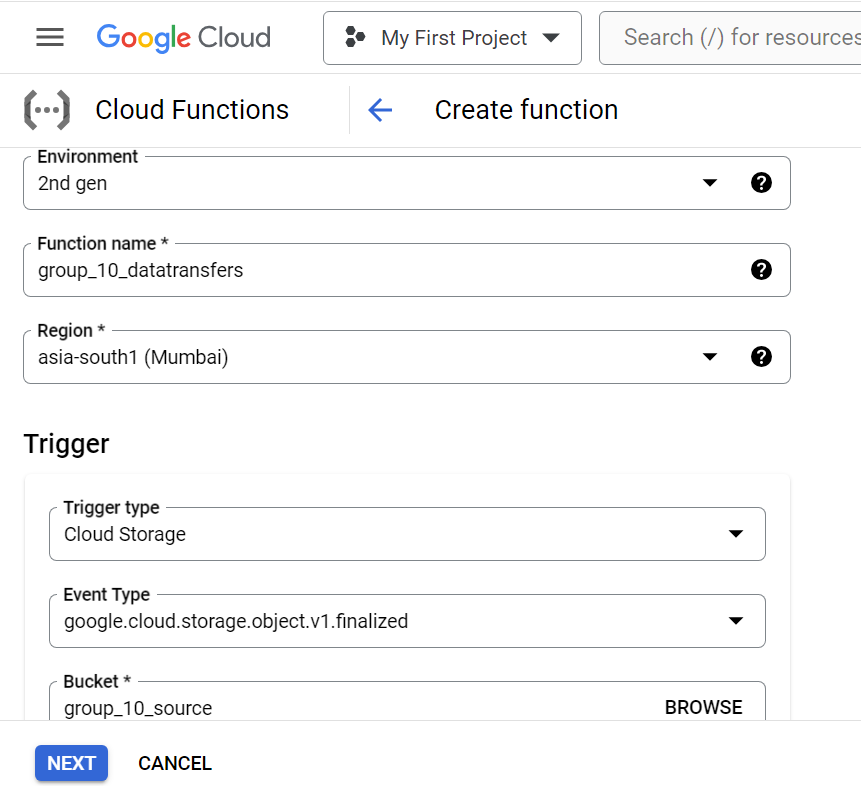
Description automatically generated

**Step2:** Now create a cloud function for moving objects from source bucket to the destination bucket.

1. For that go to the service cloud function through search bar.
2. Now click on create function.

A screenshot of a computer

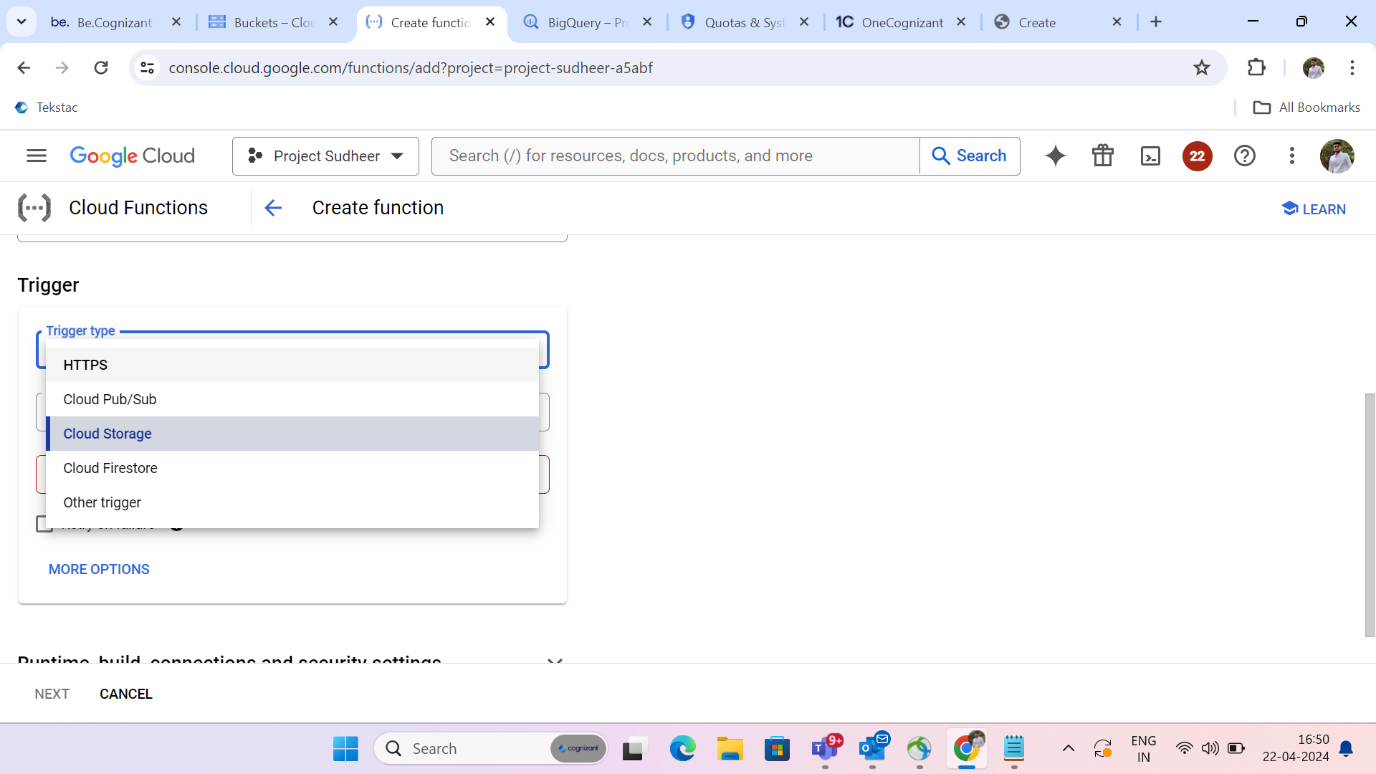
Description automatically generated



We will be using the 2nd generation environment.

Give a name for your function and select a region in which we want to create a function.

Then select a trigger. Trigger is that a response to an event. Select cloud storage as a trigger type since we are working on buckets.



Select event type as finalized.

A screenshot of a computer

Description automatically generated

* Click on next then we will be directed to a code deployment page.
* Now select the run time environment as python.
* Give the requirements google-cloud and google-cloud-storage.
* Write the code for moving objects from one bucket to another bucket in main .py.
* The function name is the hello\_gcs and the entry point is the function call.

A screenshot of a computer

Description automatically generated

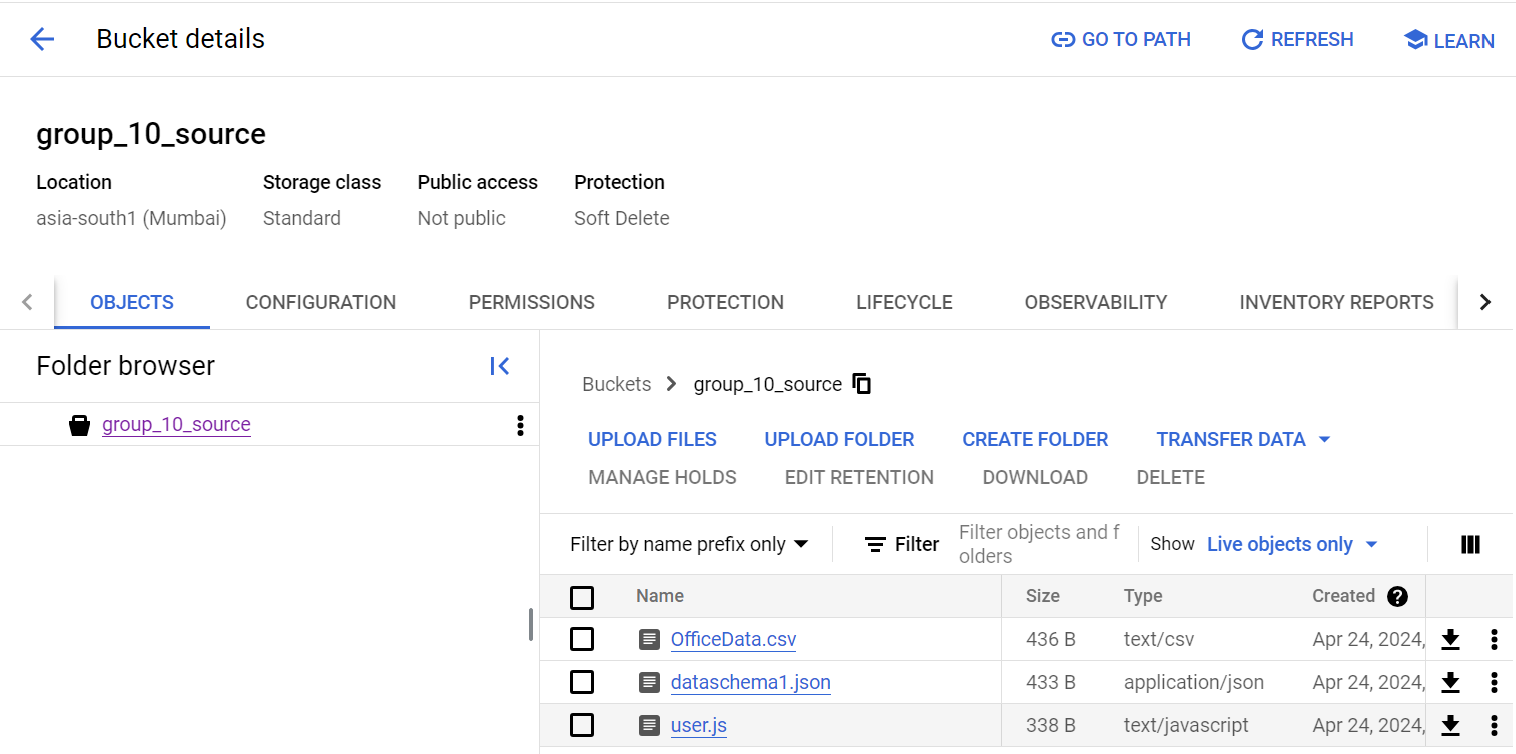
* A screenshot of a computer

  Description automatically generatedDeploy the code.

A screenshot of a computer

Description automatically generated

Now upload some file in the source bucket.



When we upload file in the source bucket the cloud function will be triggered and moves the objects from the source bucket to destination bucket automatically.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

After upload the files in the source bucket after the triggering of the cloud function the files automatically move to the destination bucket

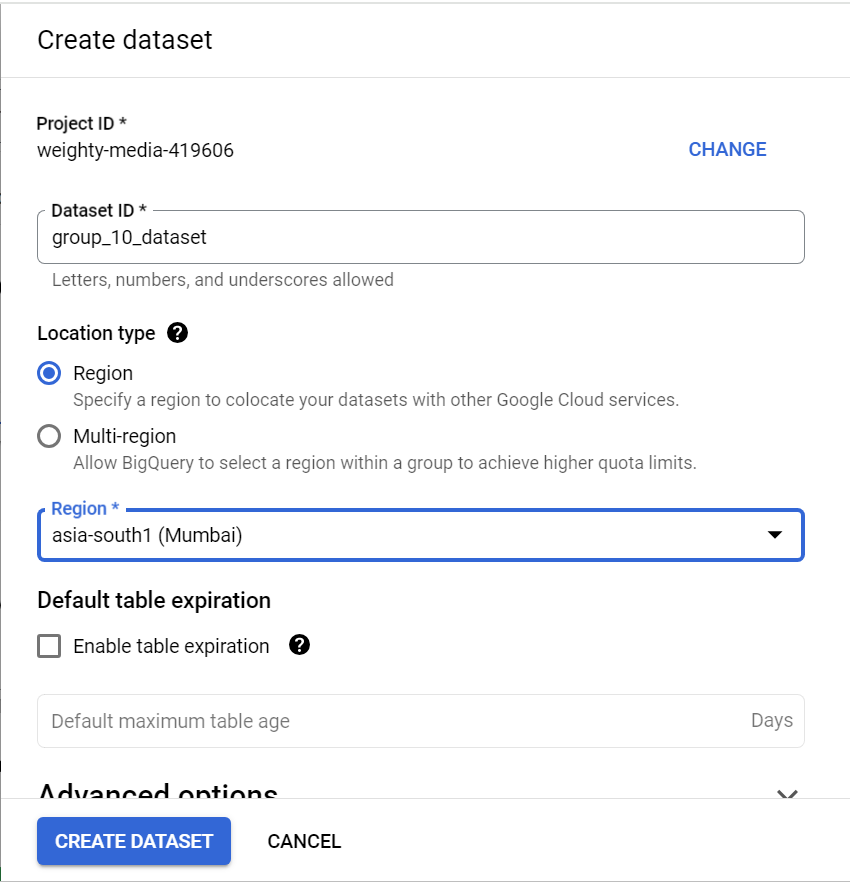
Which it will create the changes in the bucket and those changes can be recorded in the cloud function logs.

**A screenshot of a computer

Description automatically generated**

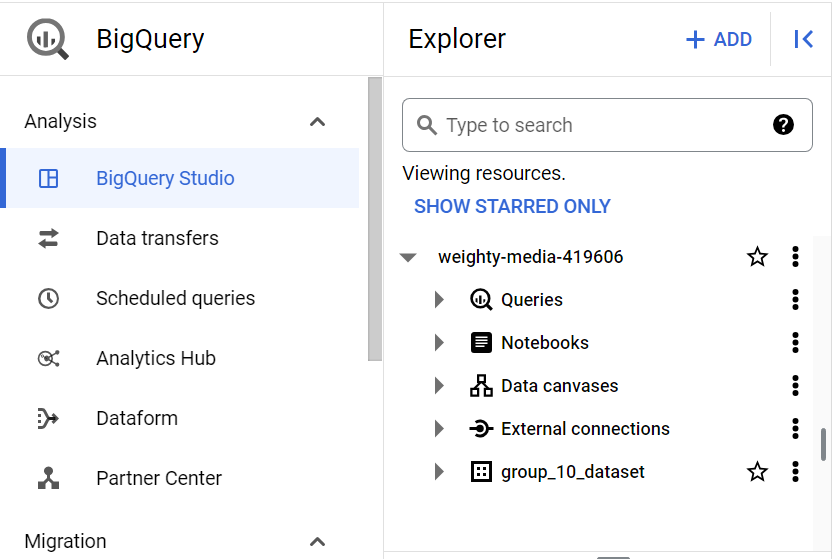
**Step 3:** create a dataset and a table in that dataset in Big Query.

The dataset name should be unique. Select the location type and region.



By clicking on the “**CREATE DATASET**” it will create a data with the given parameters.

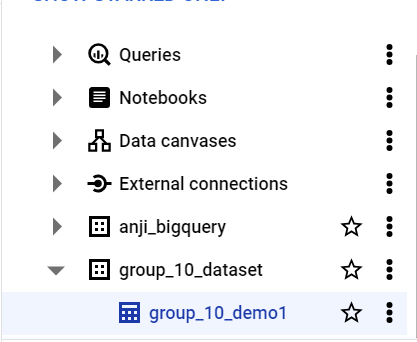
After creating the dataset create a table.



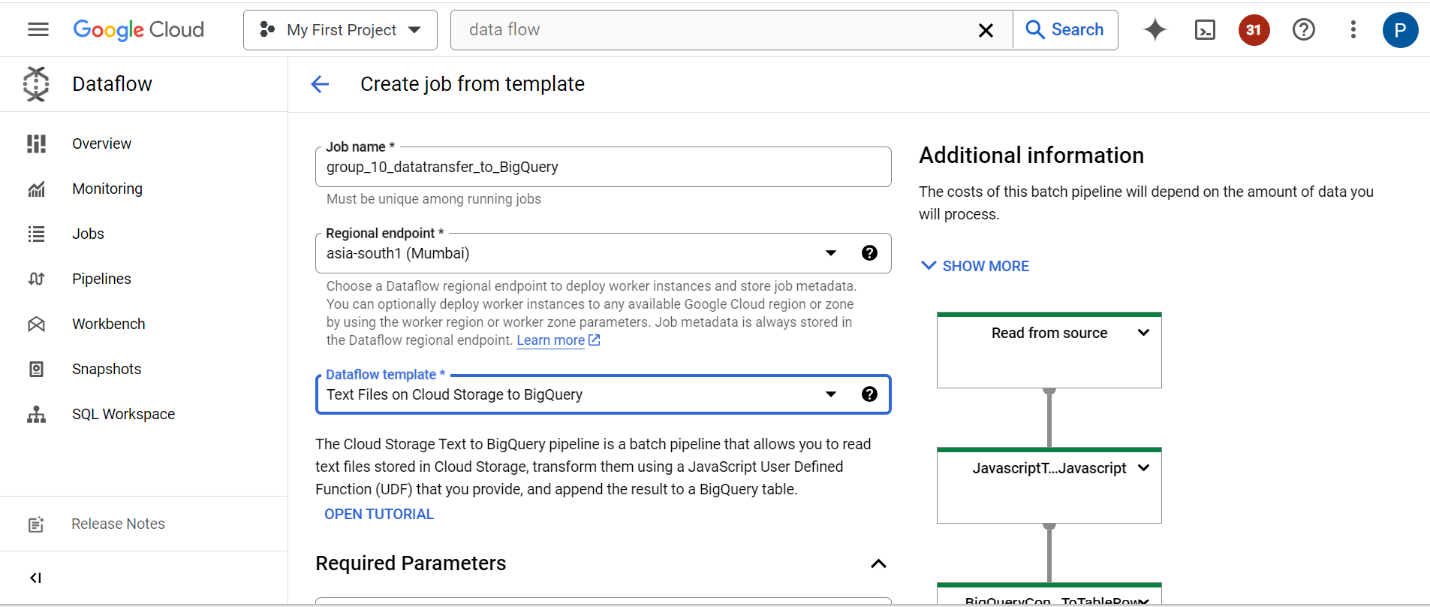
And then create an empty table named result data in that dataset.

A screenshot of a computer

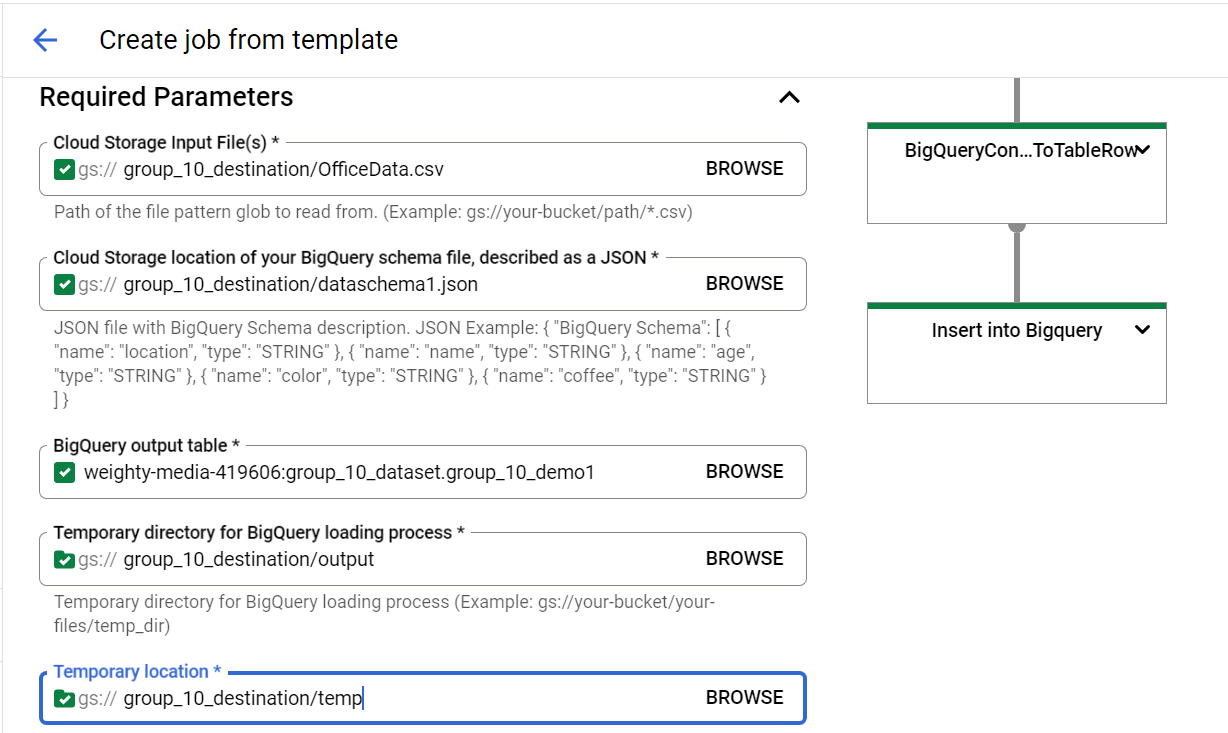
Description automatically generated



**Step 4:** Create a data flow pipeline. Go to data flow and create a job from template to load data from bucket (cloud storage) to big query.

****

Select a template “**text files on cloud storage to big query**”.



Now configure the job parameters.

1. input file path.
2. location of the file in cloud storage that specifies schema of big query.
3. Path to big query table.
4. Temporary directory for big query loading process and temporary location.
5. Cloud storage path for the java script file.
6. Click on run job so that the data flow job will be started.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a phone

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The loading of data will be done in the four stages: -

**Read from source succeeded**: -

1. In this stage, the input from user which has given on the creation of the dataflow job will be analyzed.
2. If there is any incorrect data or mismatch of data will rise a error and stop the programs immediately.

**JavaScript Transform succeeded**: -

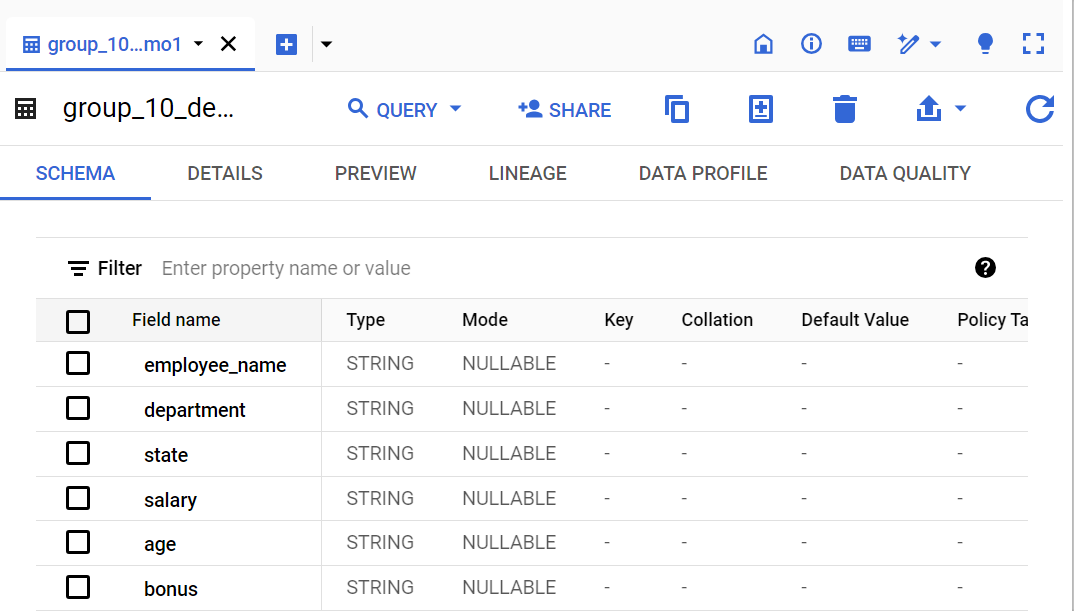
1. In this stage, what are the columns that are to be loaded into the Big Query table those are to be triggered from the JavaScript file and those are to be loaded into the table.

**Big Query conversion of Table Rows: -**

* In this stage the rows are to be converted and are to be loaded table in the columnar format
* The data is converted upon the successful completion of the above two stages.
* The data will be undergoing different stage of processing and after completion of successful processing it will be moved to the final stage of processing.

**Insert into Big Query Succeeded**: -

* After completion upon on the above three stages the data will be loaded into big query table.
* It will automatically detect scheme and upload the data into the tables.
* After running all stages in the data flow, the data will be loaded into the big query.



We can see the data loaded into the big query in preview.

A screenshot of a computer

Description automatically generated

**CONCLUSION:**

In conclusion, the project has demonstrated the power and flexibility of Google Cloud Platform (GCP) services for building efficient and scalable data pipelines. By leveraging Cloud Storage, Dataflow, and Cloud Functions, we have successfully orchestrated a data workflow that ingests, processes, and loads data into Big Query, showcasing key aspects of modern data engineering practices.