**Process Document**

Below are the steps in which provided details of 5-6 services of Google Cloud Platform (GCP) used for this project and how to set up this project in GCP.

Initially we have first set up project by enabling few APIs as shown below.

**ENABLE API**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**SET UP CLOUD STORAGE**

After Enabling API, created the Google Cloud Storage bucket. It is uniquely named and used to upload the data.

Here we have scraped the Oil production month wise data and uploaded in the s2a3\_projectbucket as shown in below screen

Graphical user interface, text, application, email

Description automatically generated

**CREATE A BIG QUERY TABLE**

**BigQuery** is Google’s data warehouse designed for analyzing the data using SQL. It also used to build Machine Learning models.

Here, we have created Tables/Models using Cloud Functions and performed below mentioned things as shown in below screen

1. monthly\_statistics: It is table created using the Oil\_Production.csv file which is stored in s2a3 bucket. We always store the latest scraped data in it.
2. Models: Using BigQuery, created ARIMA PLUS model and stored under this. It contains various model basis different combinations and select the best Model which has lowest AIC value.
3. production\_forecaset: This table consist of forecasting values of Oil Production for future time from 12 to 60 Months and it also contains forecasting details for various confidence level like 80%, 85%, 90% and 95%.

Whenever we trigger this job, the forecasted values are stored into this table and it can be identified basis “Refresh date” which signifies on which date the job is triggered.

1. actual\_forecast: This table is combination of both Historical data and Forecast data. This table is further used to Google Data Studio for visualization purpose.

Graphical user interface, text, application, email

Description automatically generated

**SCHEMA CREATION QUERIES**

We need to first create this tables as mentioned in above screenshot. For this, open a query window under BIGQUERY and run the query as shown in below screenshot.

Table: monthly\_statistics

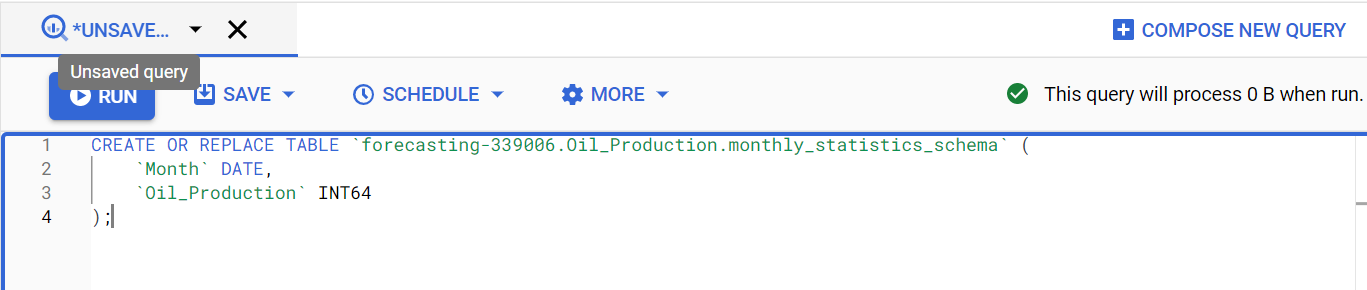


Table: production\_forecast

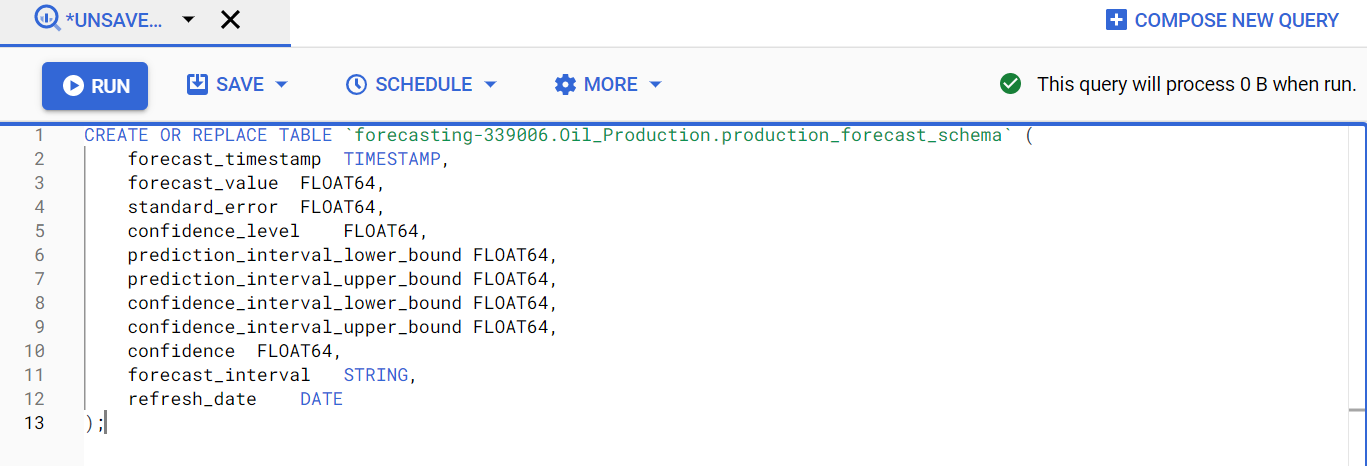
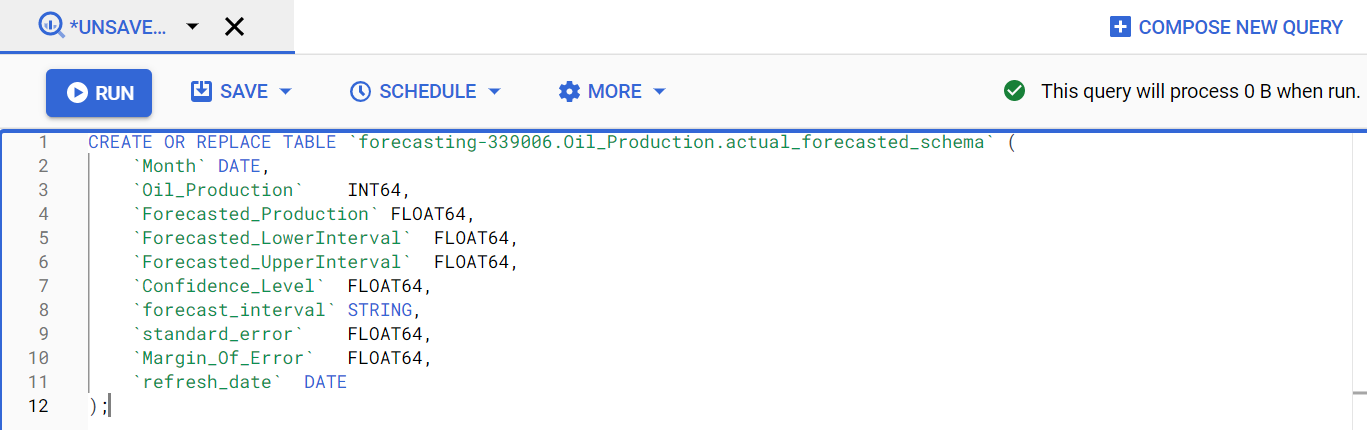


Table: actual\_forecasted



**Develop the script**

**Google Cloud Function** is used to develop the script. It is a Function as a Service (FaaS) that allows engineers and developers to run code without worrying about server management and it scale as needed.

**Cloud Pub/Sub** is used to trigger this function. Graphical user interface, text, application, email

Description automatically generated

After this, create a function under Python 3.9 inline editor and saved it as main.py. Here, we have imported the cloud storage, bigquery libraries which contains function and properties to interact with its respective APIs.

Later we have written all the scripts to

1. Scrape data
2. Upload data into s2a3\_projectbucket
3. Used Bigquery script for creating Tables and ML model

Graphical user interface, text, application, email

Description automatically generated

We have also provided the library version in the “requirement.txt” file so that all the libraries will be available while running the cloud function.

Graphical user interface, text, application

Description automatically generated

**DEPLOY THE MODEL**

Once script is ready, we deploy this function and test it.

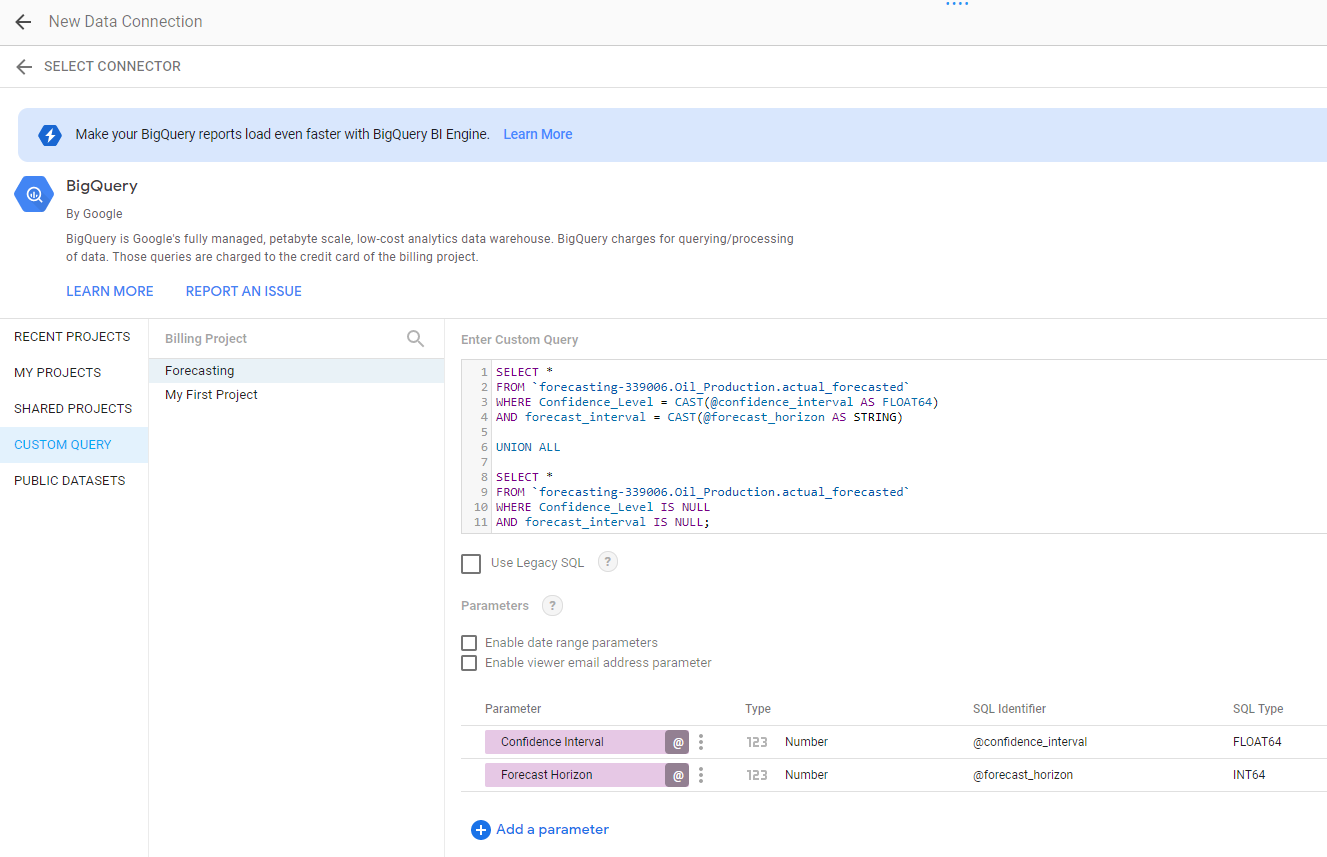
Graphical user interface, text, application, email

Description automatically generated

**DATA STUDIO**

Once we run / test the function it will create all the Tables / Models which are showed under Big Query section.

Further, we use the actual\_forecast table to create the Oil Production forecast Dashboard. Below is a screenshot show casing the parameterized connection between BQ and Data Studio.



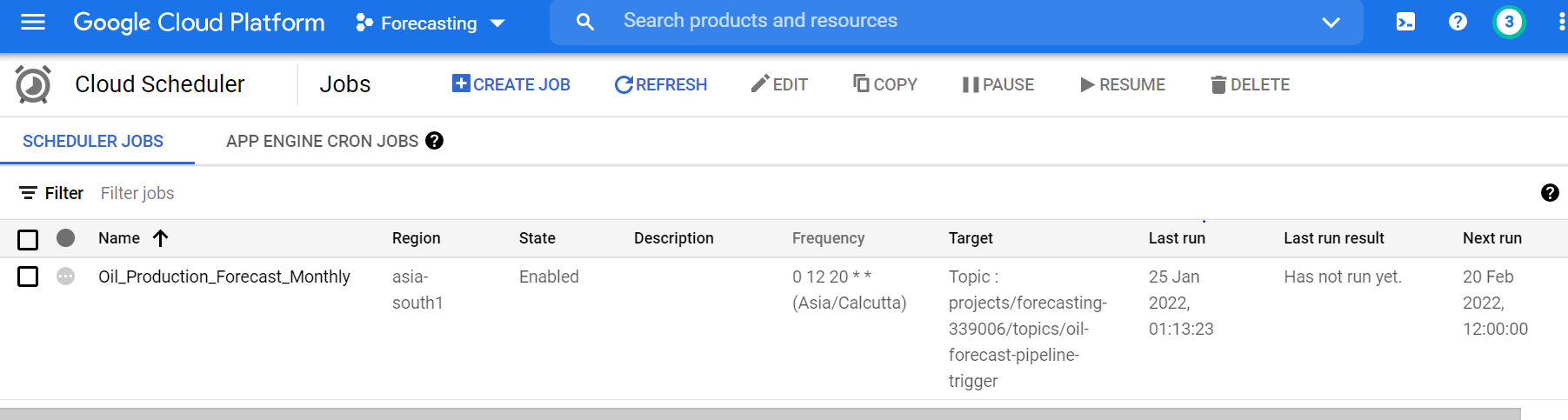
Dashboard has functionality to interact and can see oil production forecast for various months’ window like 12 months, 24 months, 36 months etc. and select the confidence level for forecast values from 80%, 85%, 90% and 95%.

Graphical user interface

Description automatically generated

**SCHEDULE THE JOB**

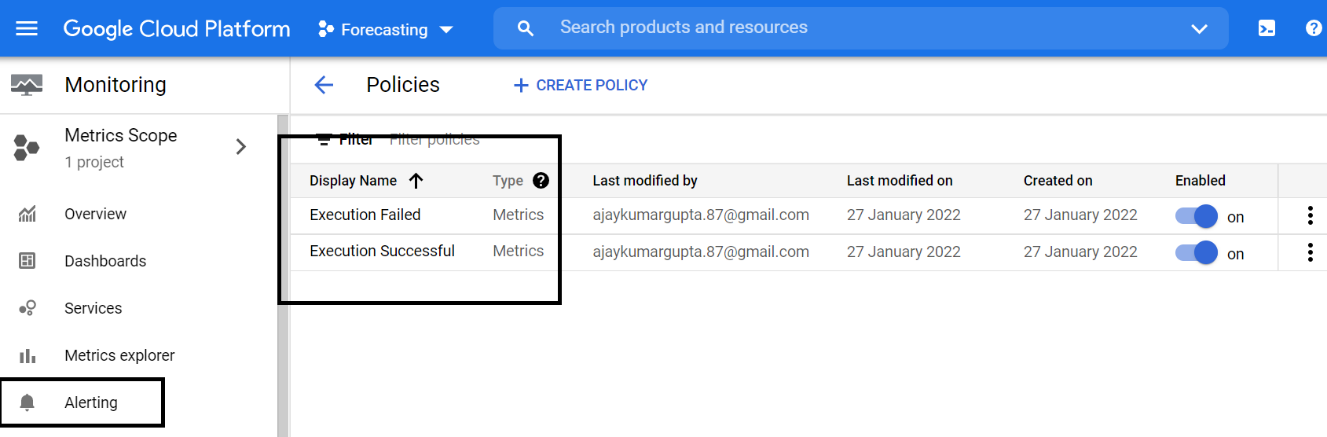
**Google Scheduler** is used to schedule the Deployed Google Cloud Function script. Currently it is schedule monthly and it will run on every 20th day of the month at 12:00 noon.



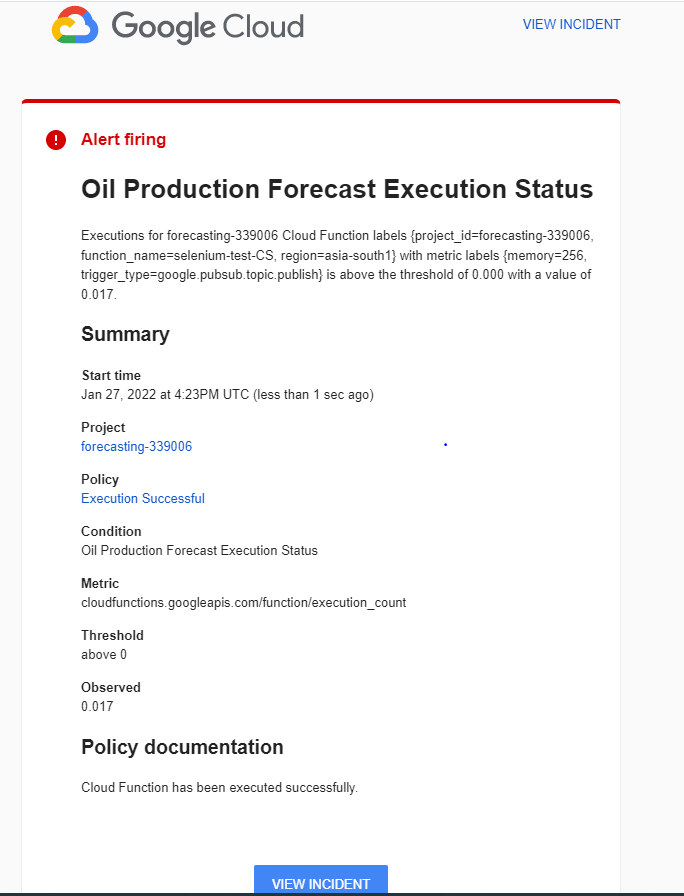
**MONITOR THE JOB**

**MONITORING** service from GCP is used to monitor this. There is an “Alerting”service which triggers an email to intimate the user about the successful / failed execution status of the job.

We have created to Alert mechanism, one for successful execution and other for failed execution.



Below is the email alert for both Execution i.e., Successful and Failed

Execution Successful

Execution Failed

