CREATE A CLOUD FUNCTION TO PULL A MESSAGE FROM PUB/SUB

A Project Report Submitted in the fulfilment of the requirements for

Final-Project Evaluation

Name: Sanjeev M

Employee id: 2320436

GenC Intern

cognizant

CONTENTS

1.OBJECTIVES

- The objective of this project is to build a real-time stock price monitoring system that alerts users when a stock price crosses a certain threshold.
- The system will use Google Cloud Pub/Sub to stream real-time stock price updates and Google Cloud Functions to analyse and send alerts.
- By accomplishing these objectives, this project aims to empower organizations to leverage GCP's storage capabilities effectively, driving operational efficiency, and enhancing data governance practices.

2.INTRODUCTION

In the ever-expanding landscape of cloud computing, efficient data storage management is paramount for organizations to balance performance, accessibility, and cost-effectiveness. Google Cloud Platform (GCP) offers a comprehensive suite of storage solutions tailored to meet diverse needs, including Standard and Archive buckets, each optimized for specific use cases.

Google Cloud Platform (GCP) offered by Google, is a suite of cloud computing services that Provides a series of modular cloud services including computing, data storage, data analytics, and machine learning, artificial intelligence alongside a set of management tools.

GCP run on the same infrastructure that Google uses internally for its own products, such as Google Search and YouTube.
GCP is designed to be scalable, reliable, and secure. It is also cost-effective, with a pay-as-you-go pricing model.

Here are some key points about GCP:

1. **GCP Services**: GCP offers a wide range of infrastructure and application services that can be accessed on-demand. It enables users to build, deploy, and scale applications seamlessly while taking advantage of Google's powerful and reliable infrastructure.
2. **Global Resources**: GCP consists of physical assets (such as computers and hard disk drives) and virtual resources (such as virtual machines) distributed across data centres worldwide. These resources are organized into regions and zones, providing redundancy and reduced latency for better performance.

Services and Integration: When you develop applications on GCP, you combine various services to create the infrastructure you need. GCP services include compute, storage, databases, machine learning, and more. Additionally, GCP integrates seamlessly with other Google Cloud products.

Benefits of using GCP:

There are many benefits to using GCP, including:

- **Scalability:** GCP can scale up or down to meet the demands of your application.

- **Reliability:** GCP is designed to be highly reliable, with a 99.9% uptime guarantee.
- **Security:** GCP is one of the most secure cloud platforms available, with a variety of security features to protect your data.
- **Cost-effectiveness:** GCP is a cost-effective cloud platform, with a pay-as-you-go pricing model.

Getting started with GCP is easy. You can create a free account and start using GCP services immediately.

To create a GCP account, visit the GCP website and click on the "Create an account" button. You will need to provide your name, email address, and a password.

Once you have created an account, you can start using GCP services. To learn more about GCP, you can visit the GCP documentation website.

Common use cases for GCP

GCP can be used for a wide variety of applications, including:

- Web and mobile applications: GCP can be used to host web and mobile applications.
- Data storage and analytics: GCP can be used to store and analyse data.
- Machine learning and artificial intelligence: GCP can be used to develop and deploy machine learning and artificial intelligence models.

3.REAL TIME USE CASE:

REAL-TIME STOCK PRICE MONITORING

Build a real-time stock price monitoring system that alerts users when a stock price crosses a certain threshold. Use Cloud Pub/Sub to stream real-time stock price updates and Cloud Functions to analyse and send alerts.

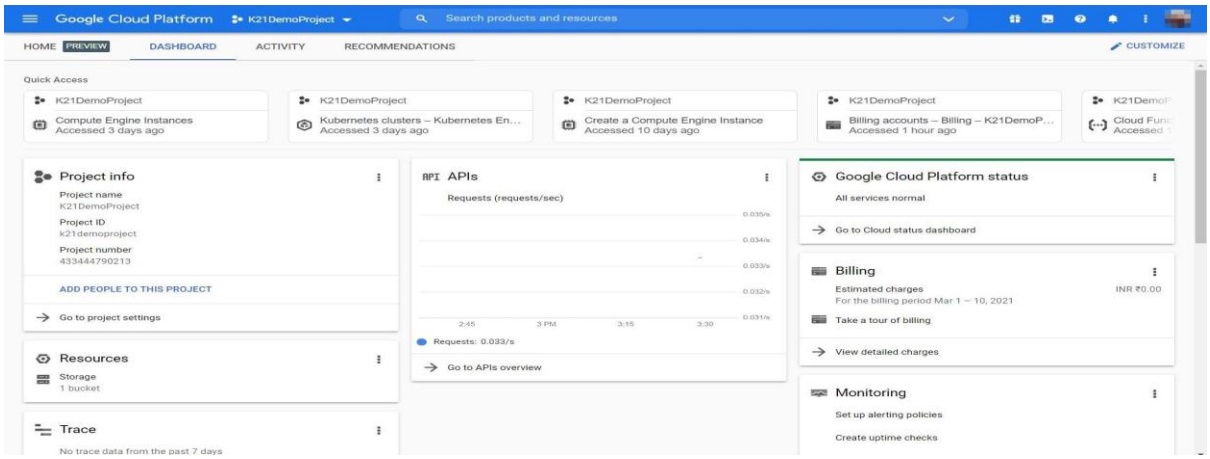
Steps:

- Fetching the stock price of Infosys by parsing the price html class and pushing it into up Cloud Pub/Sub.
- Used cloud function to analyse whether the stock price have raised above the threshold value.
- Send alerts via email by Cloud Function.

4. PLATFORM USED:

GOOGLE CLOUD PLATFORM INTERFACE:

Web Console:



Google Cloud Platform (GCP) Account: Sign up for a GCP account if you don't have one already. This provides access to GCP's storage services, including Cloud Storage for creating buckets and managing data.

Google Colab:

Google Colab (short for Collaboratory) is a cloud based Jupyter notebook environment that allows you to write and execute Python code. It’s particularly useful for data science, machine learning, and collaborative projects.

Google Looker:

- Google Looker is a business intelligence and data analytics platform developed by Looker Data Sciences, Inc., which was acquired by Google Cloud in 2020.

- Looker provides a comprehensive suite of tools for data exploration, visualization, and reporting, enabling organizations to derive insights from their data.
- It offers a semantic modelling layer that allows users to define metrics and dimensions in a business-friendly language, facilitating collaboration between technical and non-technical users.
- Google Looker is often used by businesses to analyse data from various sources, create interactive dashboards, and make data-driven decisions.
- With the acquisition by Google Cloud, Looker is integrated with Google's cloud infrastructure and services, providing additional capabilities for data processing and analysis.

5.PROJECT REQUIREMENTS

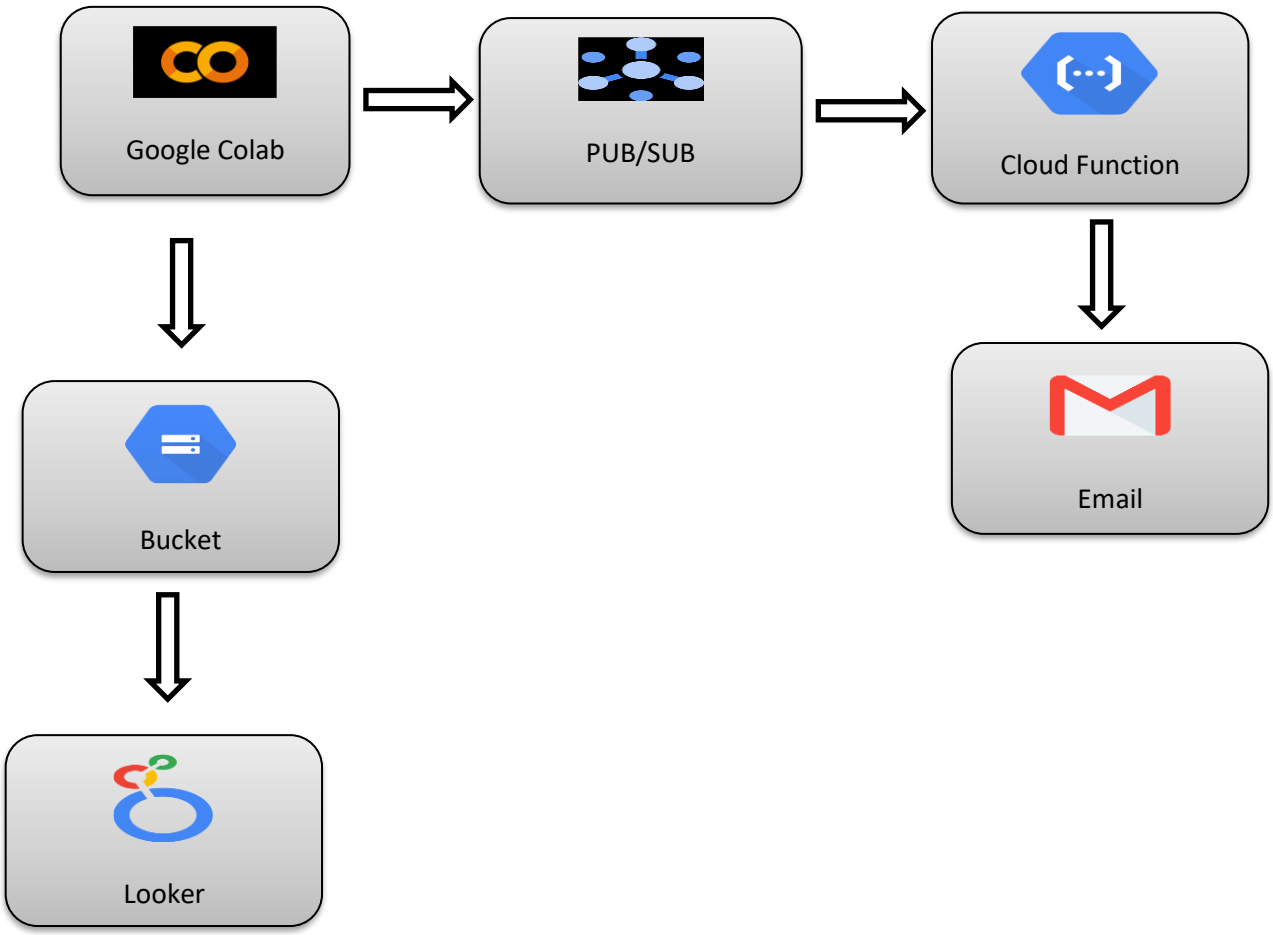
Cloud Pub/Sub:

- Pub/Sub, short for Publisher/Subscriber is a messaging service provided by Google cloud platform (GCP) that enables asynchronous communication between independent application.
- Publishers send messages to a topic, and Subscribers receive those messages from the topic they are interested in.
- This decoupling allows for scalable and flexible communication between components of an application or between different applications altogether.
- Pub/Sub supports durable messaging, meaning messages are retained even if subscribers are temporarily offline, ensuring reliable delivery.
- Additionally, it integrates seamlessly with other GCP services, making it an essential tool for building distributed and event-driven architectures in the cloud.

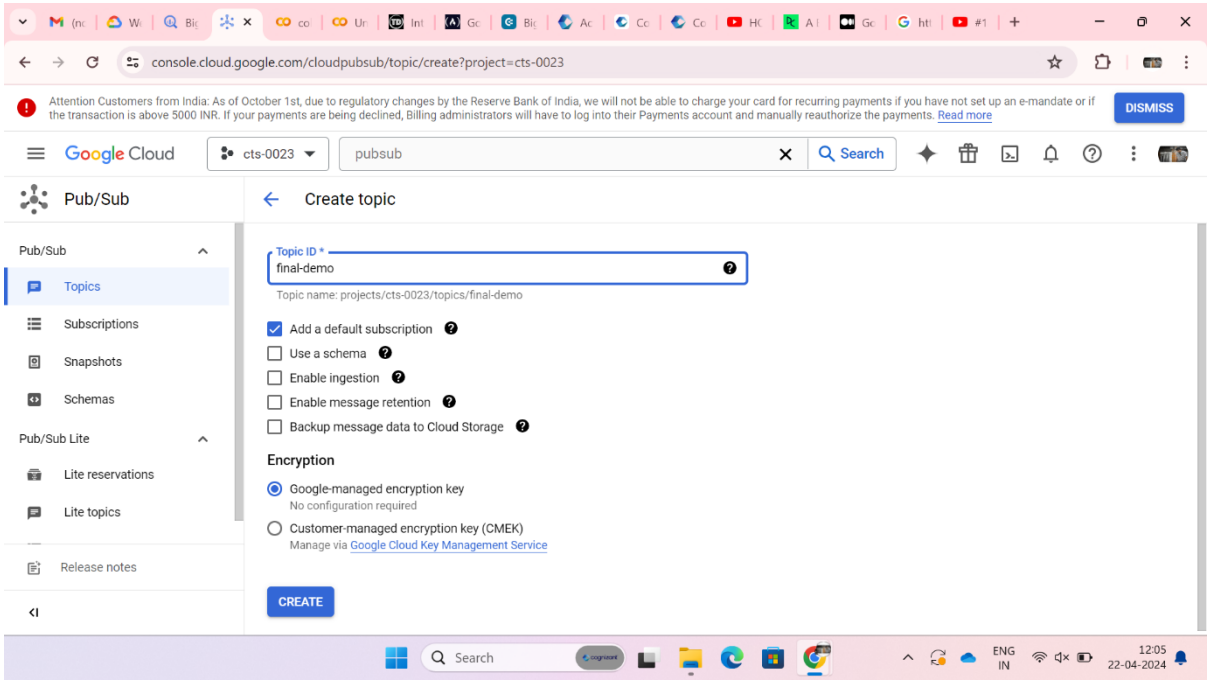
CLOUD FUNCTION:

- Cloud Functions is a serverless computing service provided by Google Cloud Platform (GCP) that allows developers to deploy and run event-driven functions in a fully managed environment.
- With Cloud Functions, developers can write lightweight code snippets in languages such as Node.js, Python, Go, and more, and deploy them without worrying about managing infrastructure.
- These functions are triggered by events from various GCP services, such as Cloud Storage, Cloud Pub/Sub, HTTP requests, and many others.
- Cloud Functions automatically scales up or down to match the incoming workload, ensuring optimal performance and cost-effectiveness.
- It is also called as a Function as a Service (FAAS).

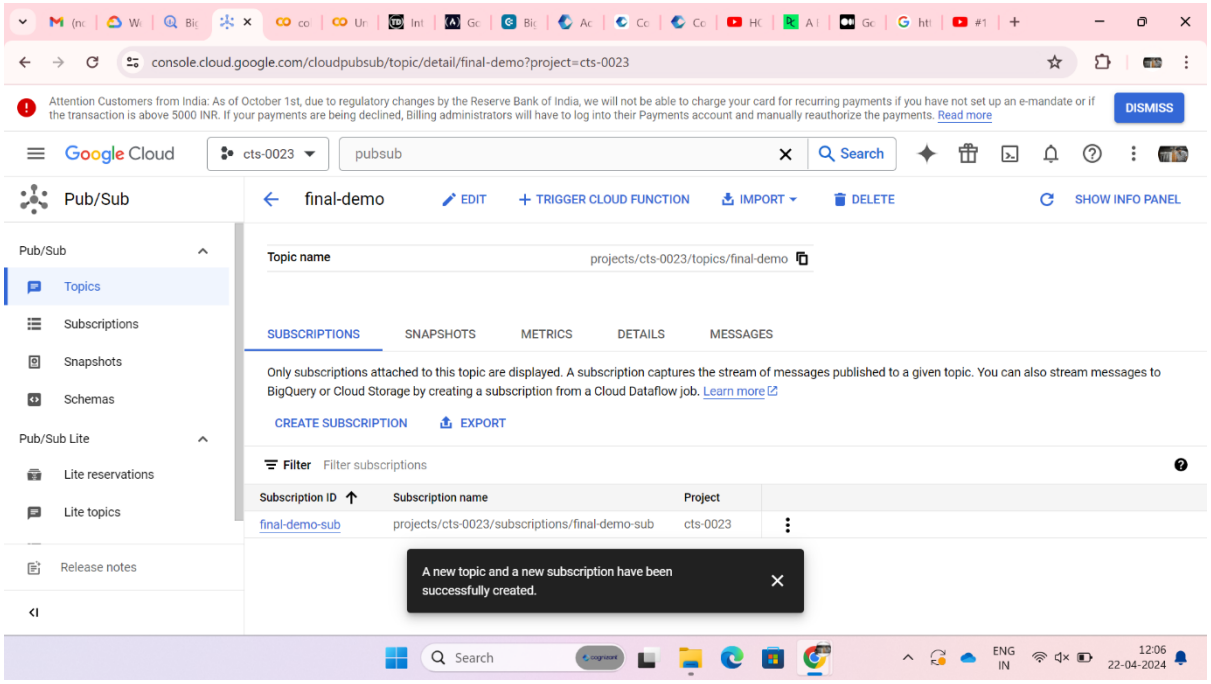
6. IMPLEMENTATION



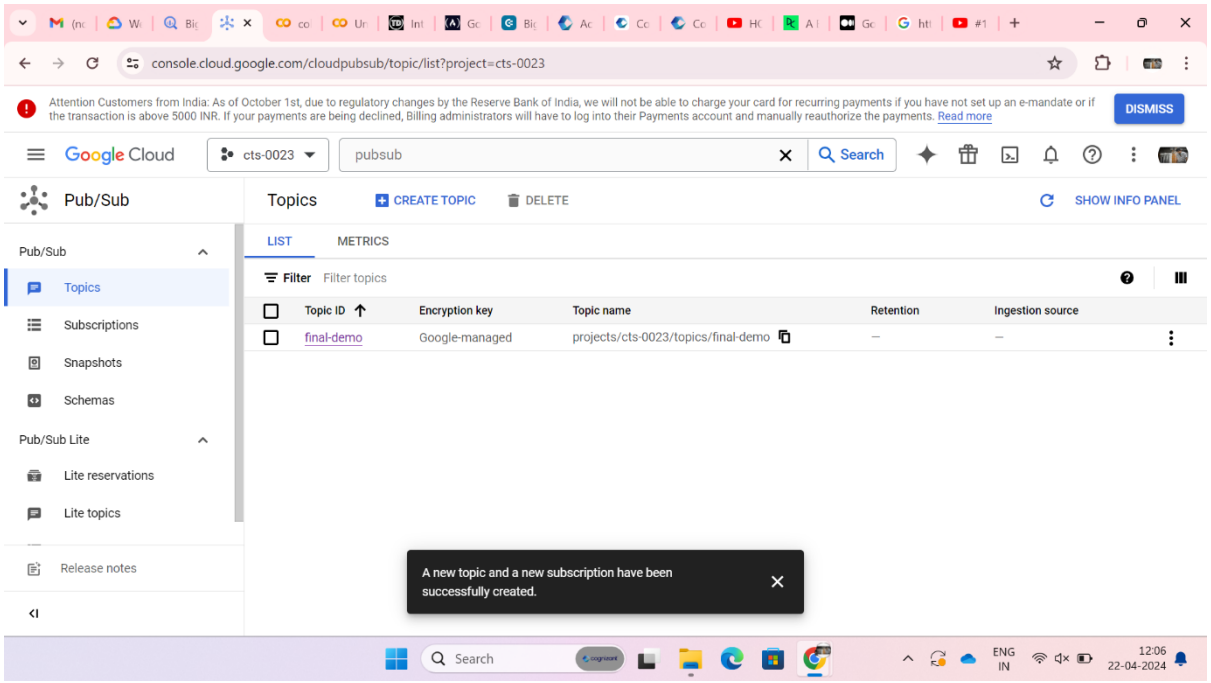
Firstly, we are creating a topic named as “final-demo” with default subscription.



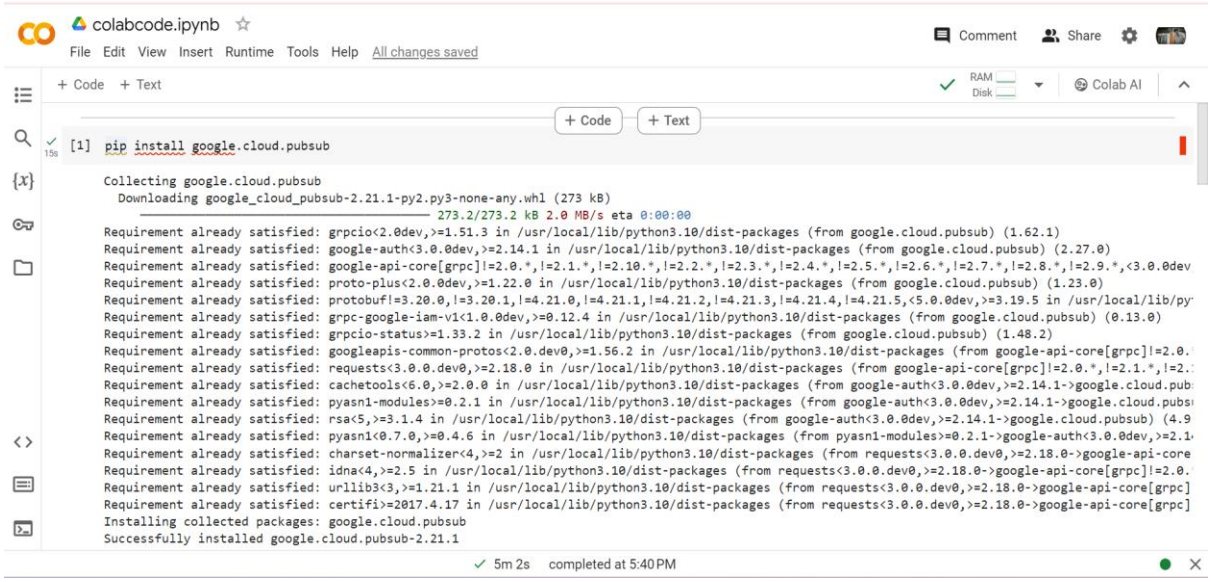
Here, default subscription is named as “final-demo-sub”.



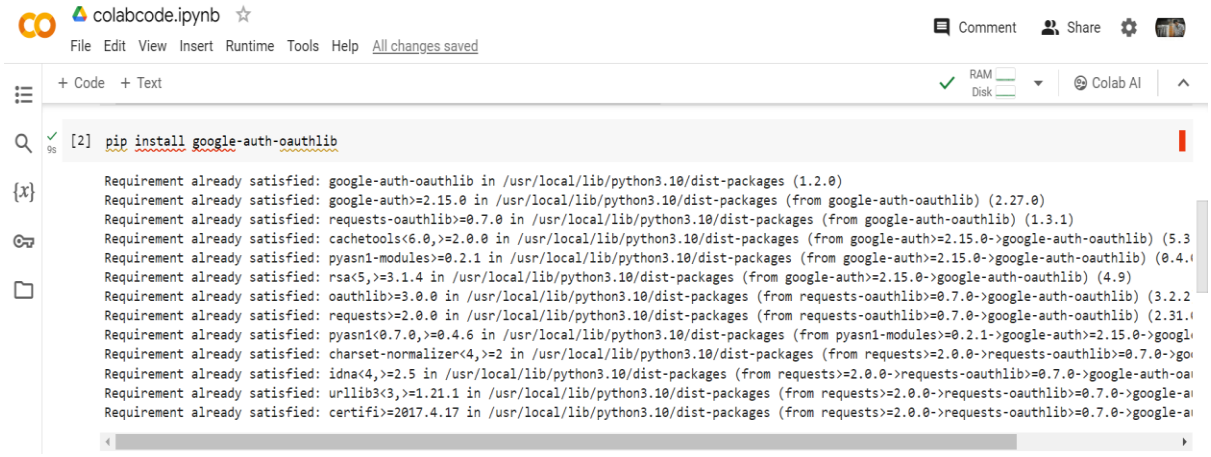
The topic named “final-demo” is created successfully.



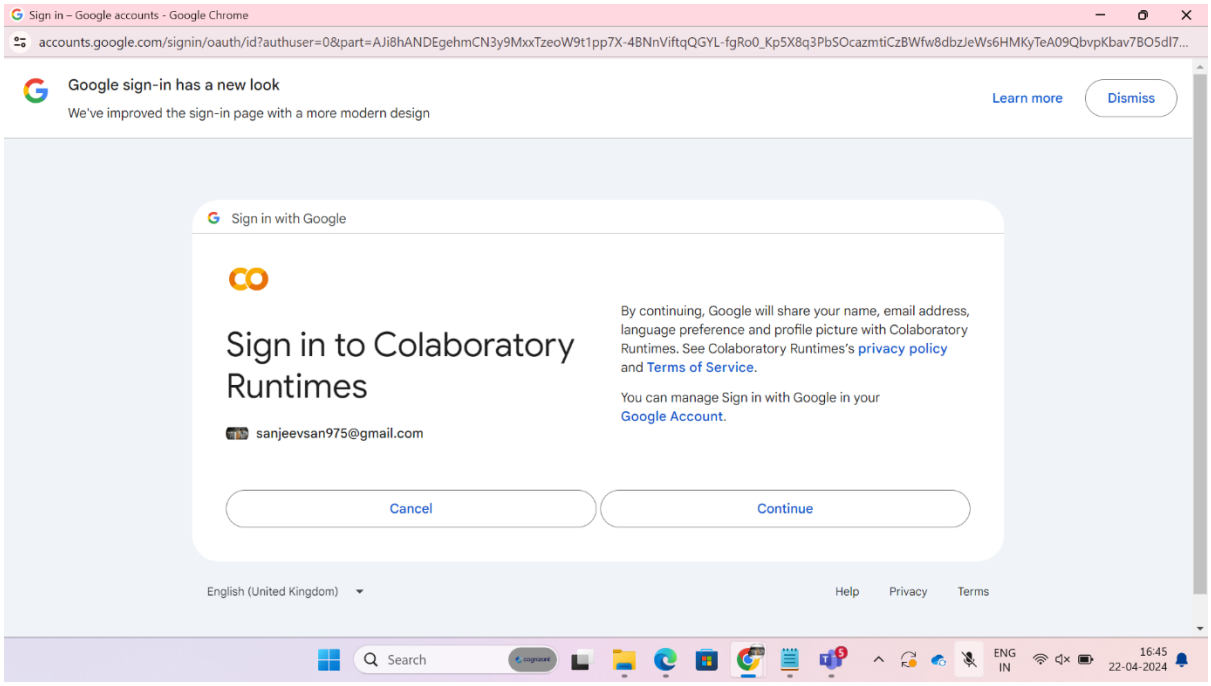
This line installs the `google.cloud.pubsub` package, which is required for interacting with Google Cloud Pub/Sub.



This line installs the `google-auth-oauthlib` package, which is required for handling authentication with Google services.

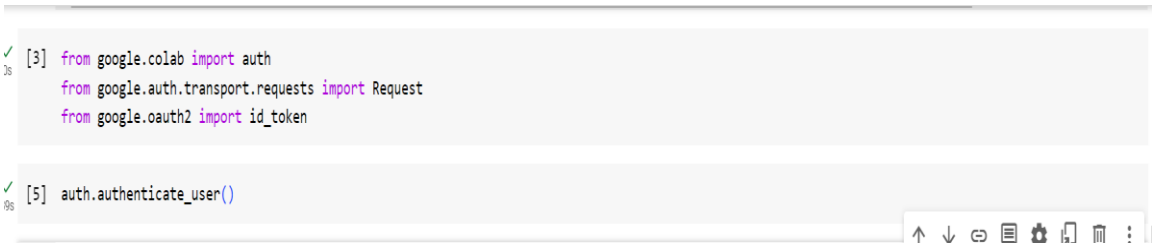


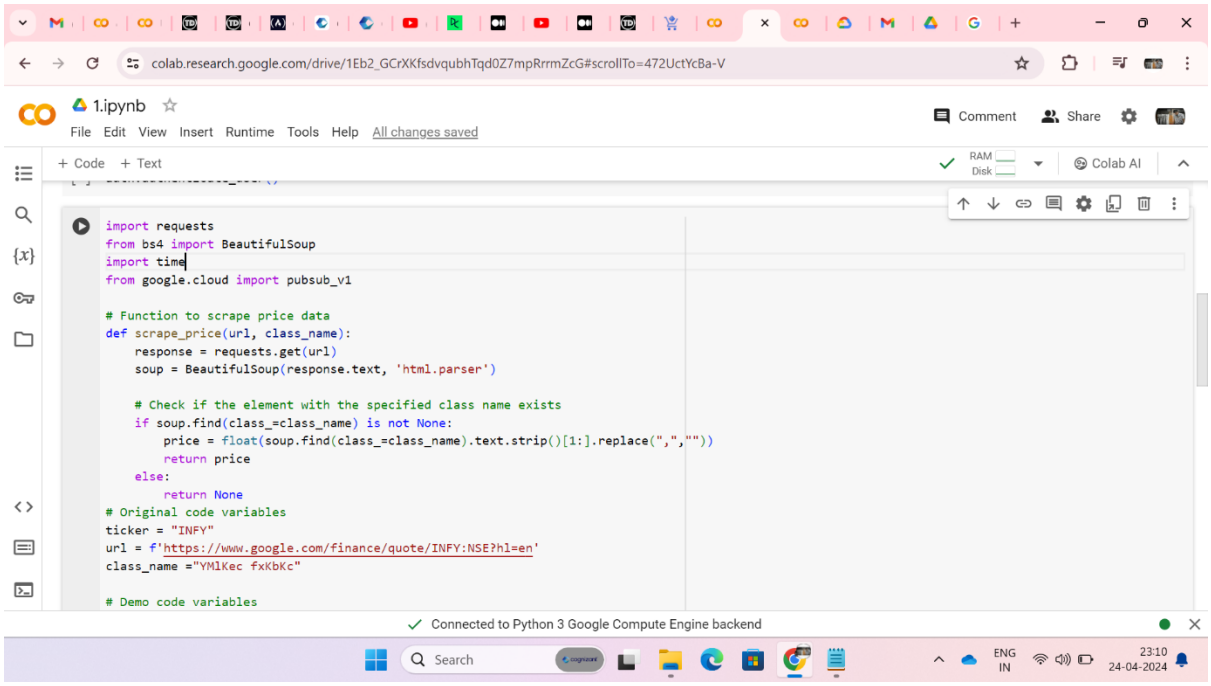
Here, Sign in page of colab will pop up as a part of Authentication from the OAuth package.



- Here, it initiates the authentication process for the user, which is necessary for accessing Google Cloud services securely.

- Next line imports the `requests` module, which is used for making HTTP requests to fetch web pages or interact with web APIs.
- Then next line imports the `BeautifulSoup` class from the `bs4` (Beautiful Soup 4) package, which is used for parsing HTML and XML documents.
- Next line imports the `time` module, which is used for adding time delays or getting current time information.
- Then next line imports the `pubsub_v1` module from the `google.cloud` package, which provides functionality for interacting with Google Cloud Pub/Sub.
- Then next line defines a function named `scrape_price` that takes two parameters: `url` and `class_name`. This function is used to scrape price data from a webpage.
- Next line extracts the price data from the HTML content, converts it to a floating-point number, and removes any extraneous characters or formatting.

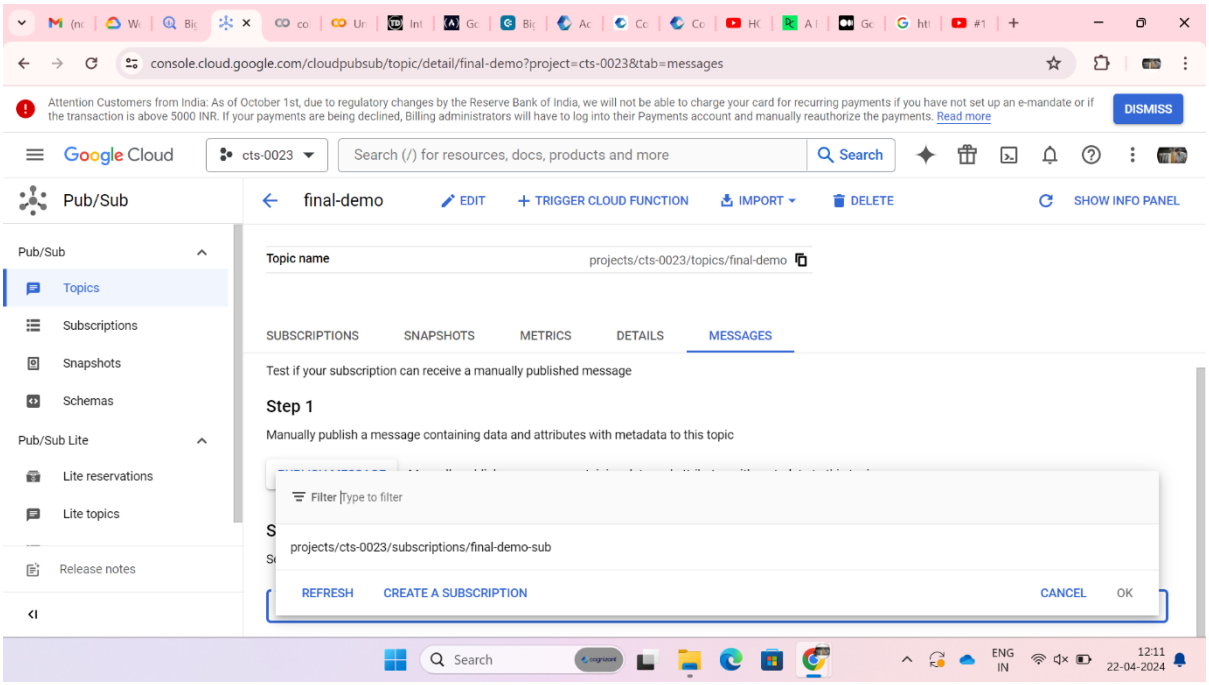




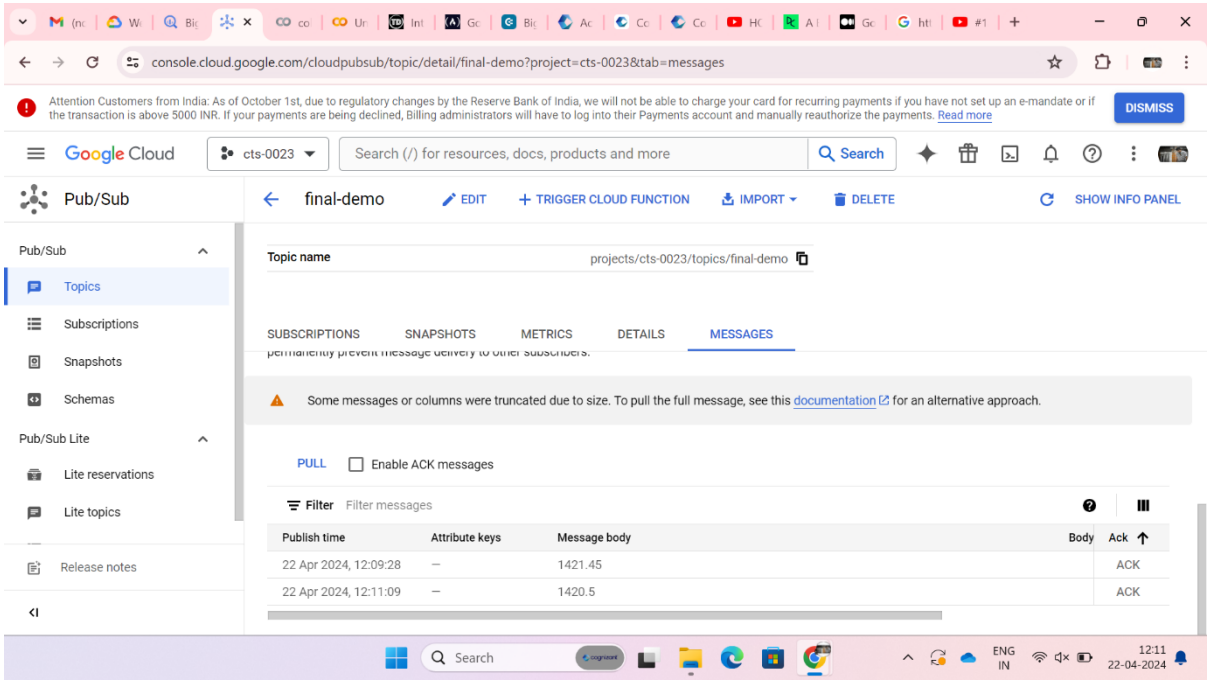
- Then it assigns the value `"INFY"` to the variable `ticker`, representing the stock ticker symbol.
- Then it constructs the URL for fetching the stock price data from Google Finance, using the ticker symbol `"INFY"`.
- Then it assigns the class name of the HTML element containing the stock price data to the variable `class_name`.
- Then next it creates a publisher client object for interacting with Cloud Pub/Sub.
- Then it calls the `scrape_price` function to fetch the stock price data from the specified URL using the given class name.

- The line encodes the message data as bytes before publishing it to Cloud Pub/Sub.
- Then next it publishes the message data to the Cloud Pub/Sub topic.
- Then next it waits for the message to be published before proceeding.
- Then it pauses the execution of the program for 100 seconds before the next iteration of the loop. This is used to limit the frequency of price updates to once every 100 seconds.

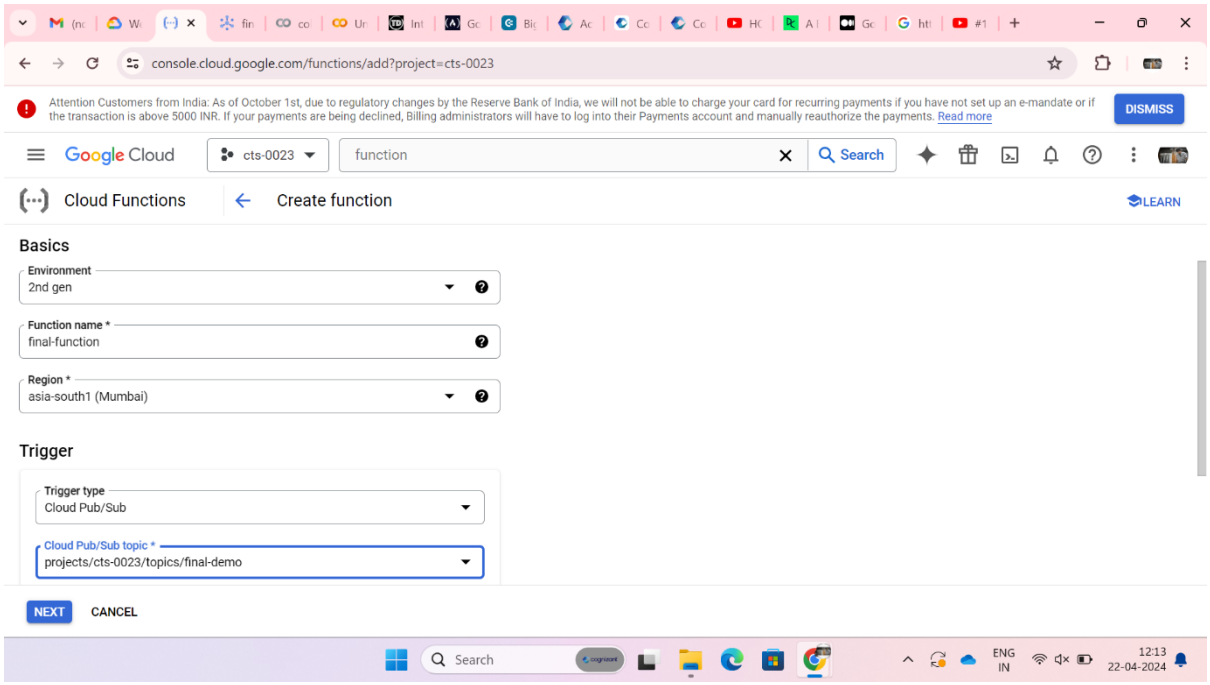
After fetching the stock price, we are selecting the subscription ID to where the message must be sent.



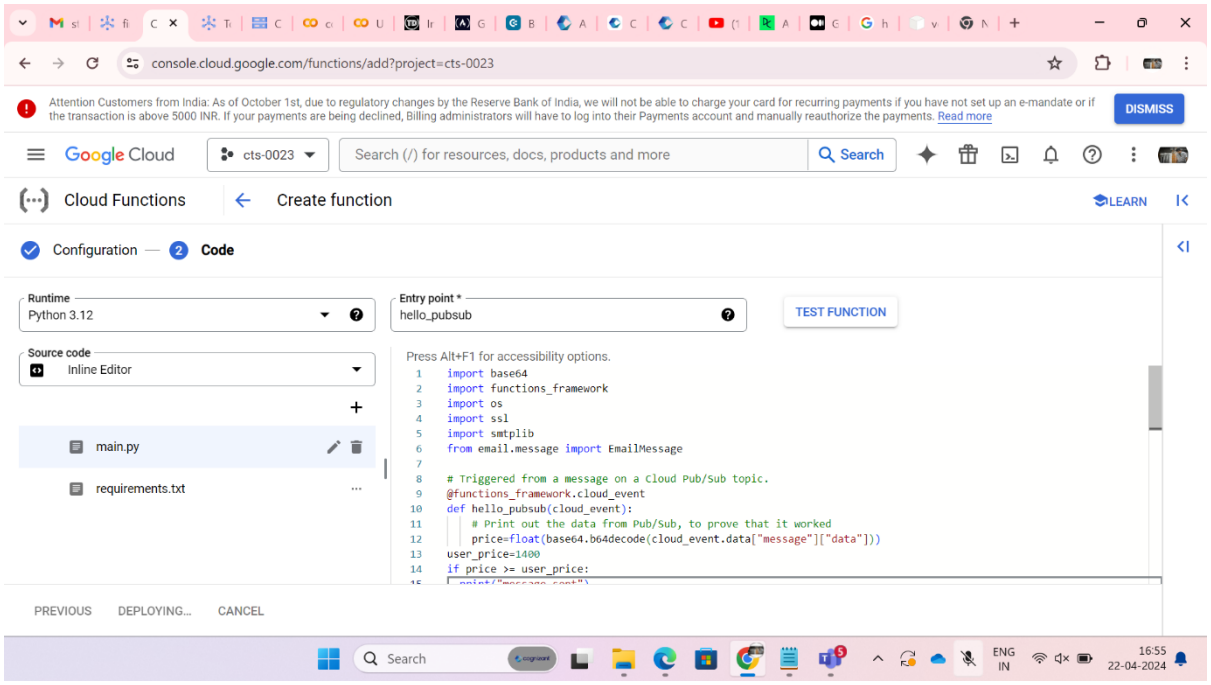
The INFY stock price is published in topic messages.



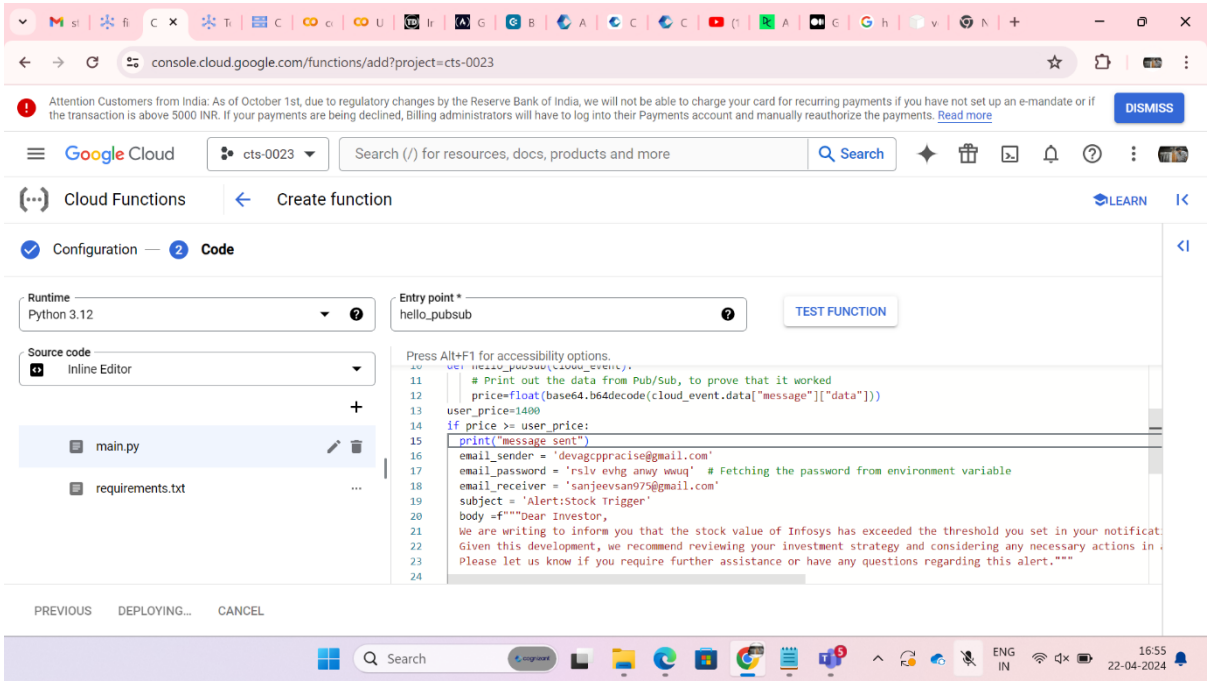
Then, we built a cloud function named “final-function” with trigger type “cloud pub/sub” with the respective topic.

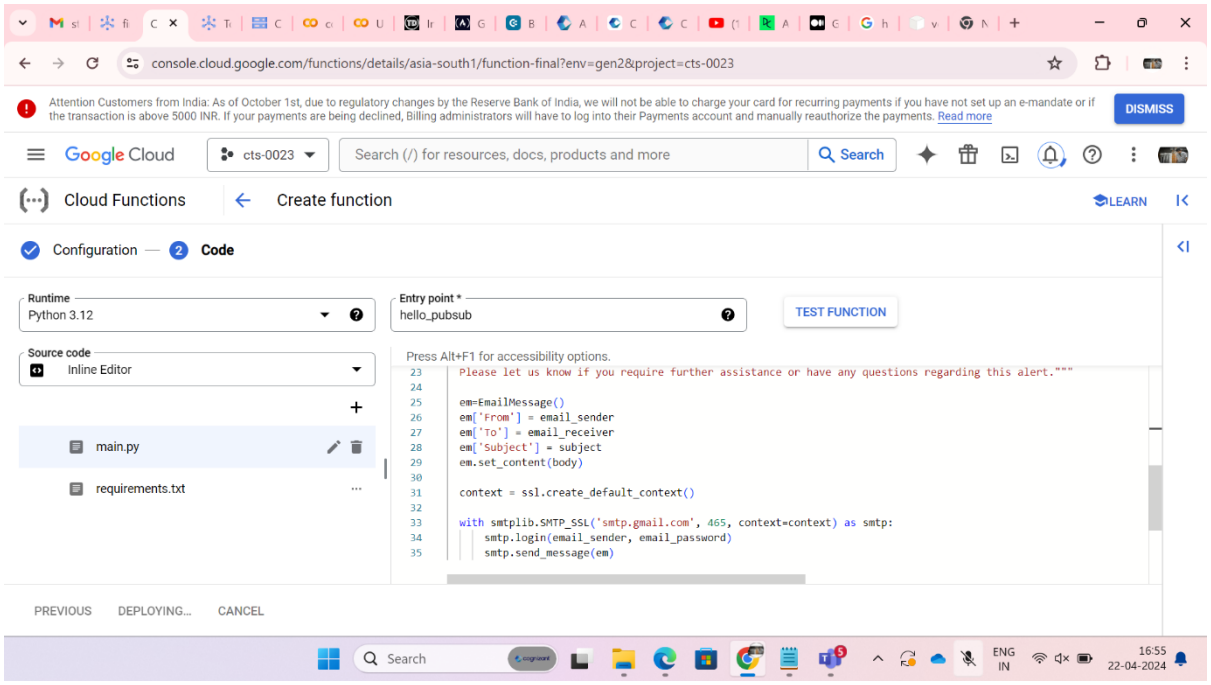


- Here, we defined a cloud function .
- Initially, Base64 is a binary-to-text encoding scheme used to encode data.
- Functions_framework is a framework for writing lightweight functions as part of Google Cloud Functions
- Then we imported the EmailMessage class from the email.message module, which is used to represent an email message.
- Then it defines a function named `hello_pubsub` which takes a `cloud_event` as its parameter. This function will be triggered when a message is published to a Cloud Pub/Sub topic.

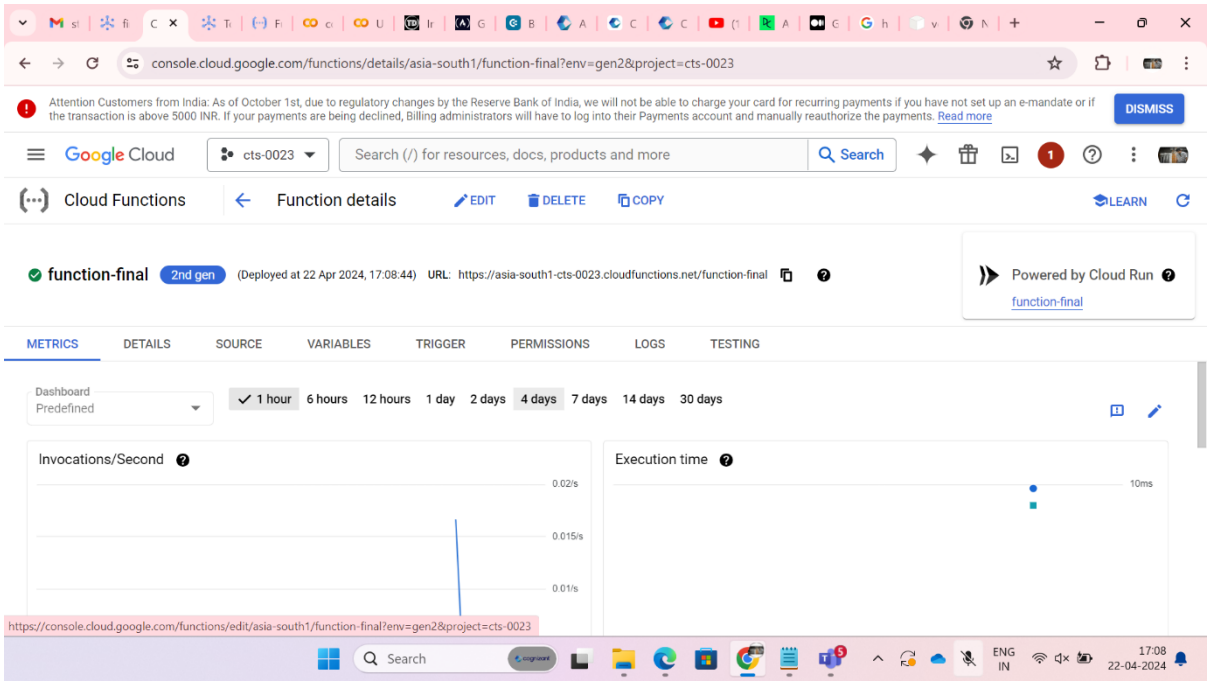


- This line decodes the data from the Cloud Pub/Sub message, which is encoded in base64 format. It then converts it to a float, assuming the data represents a numeric value like a stock price.
- Next it checks if the price extracted from the Pub/Sub message is greater than or equal to a threshold price (`user_price`). If the condition is true, it proceeds to send an alert.

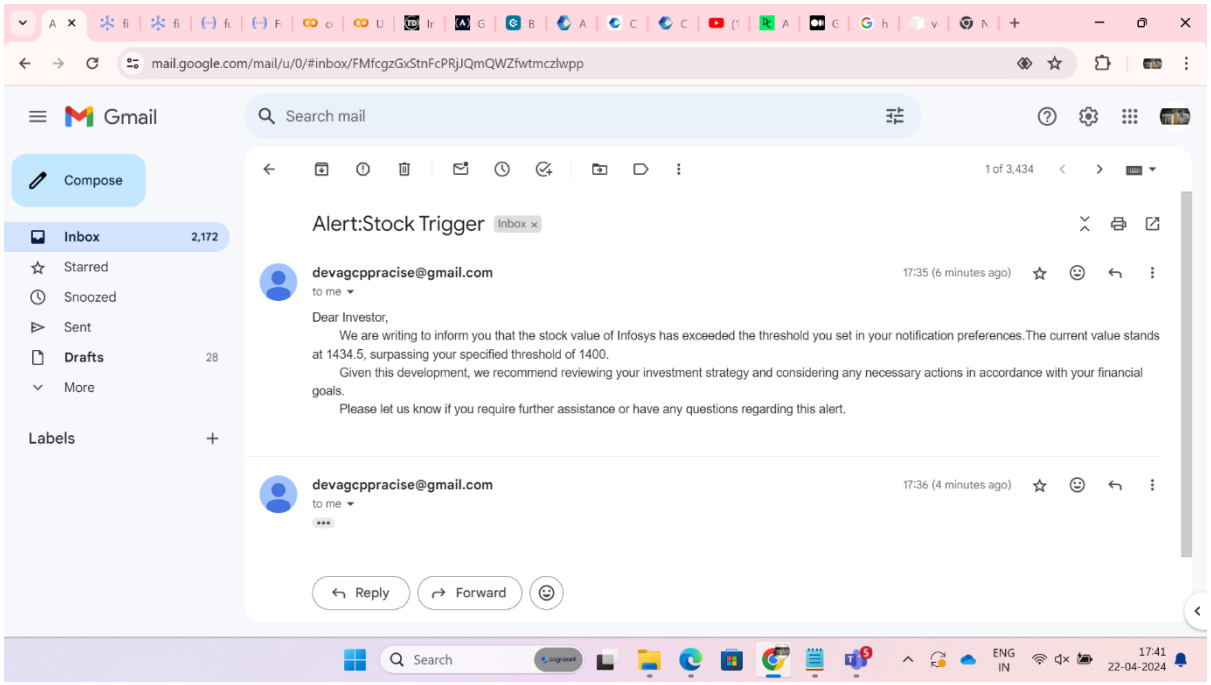




The cloud function got deployed.

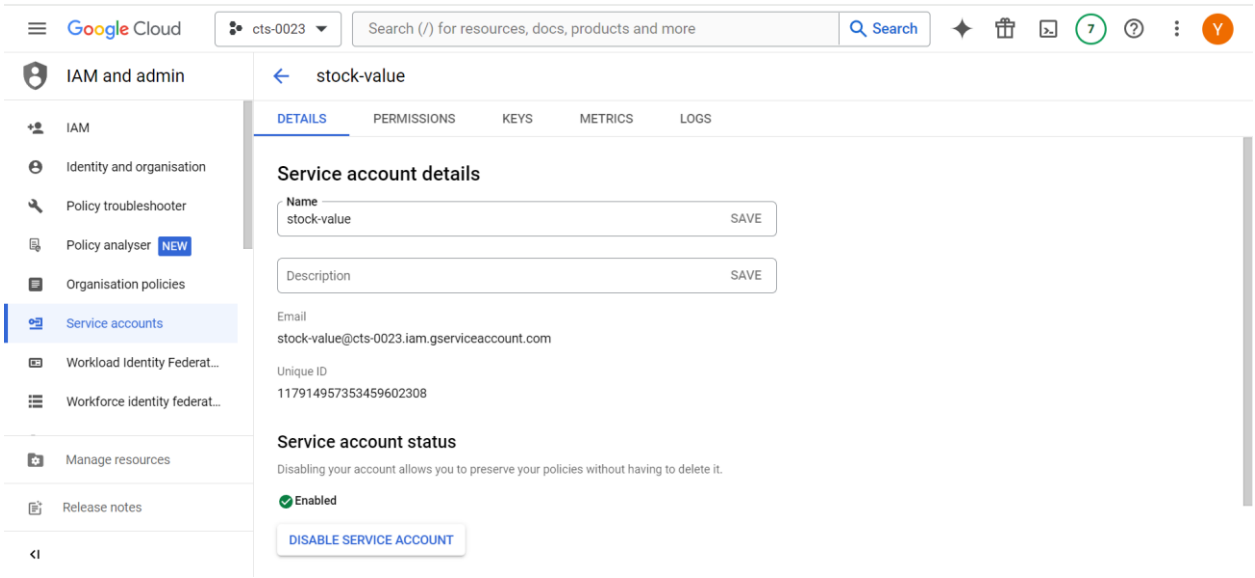


When the stock price exceeded the threshold value then alert mail got triggered with a message.

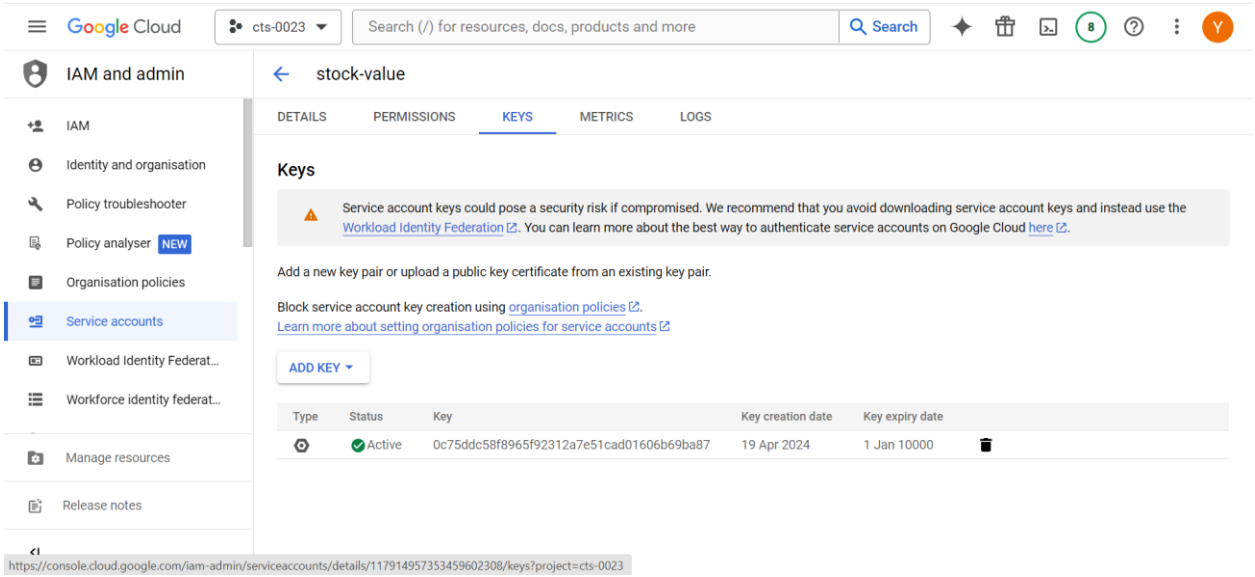


VISUALIZATION:

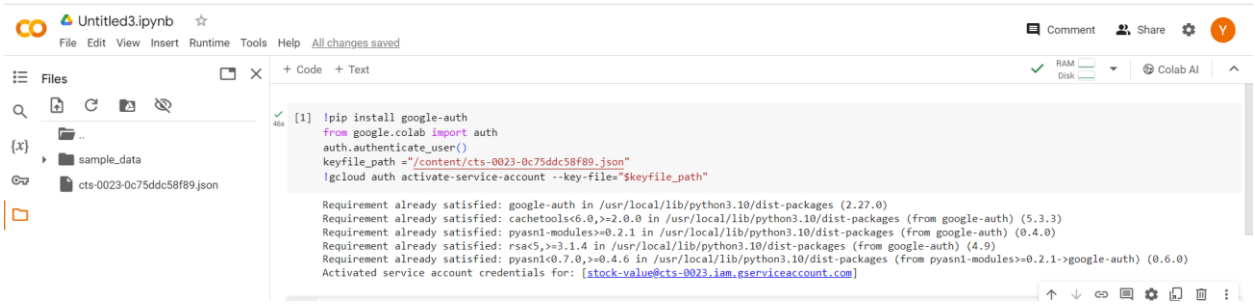
Here, we are creating a service account for authentication to access the cloud storage bucket.



create a json key file to service account.

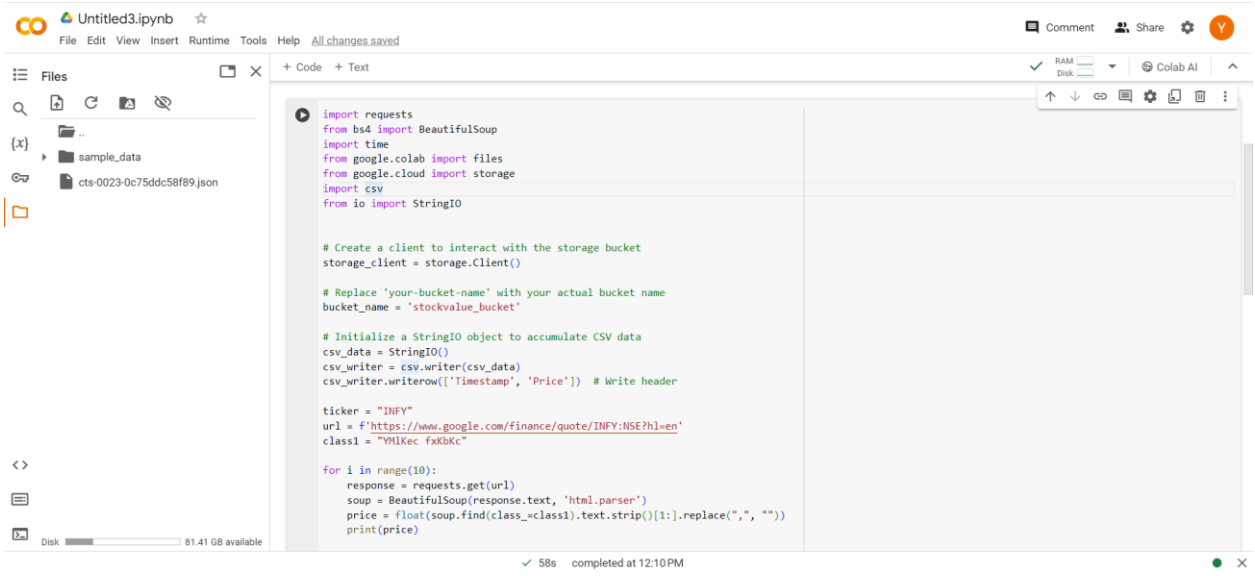


Then, we'll store the json keyfile path in a variable for authentication.

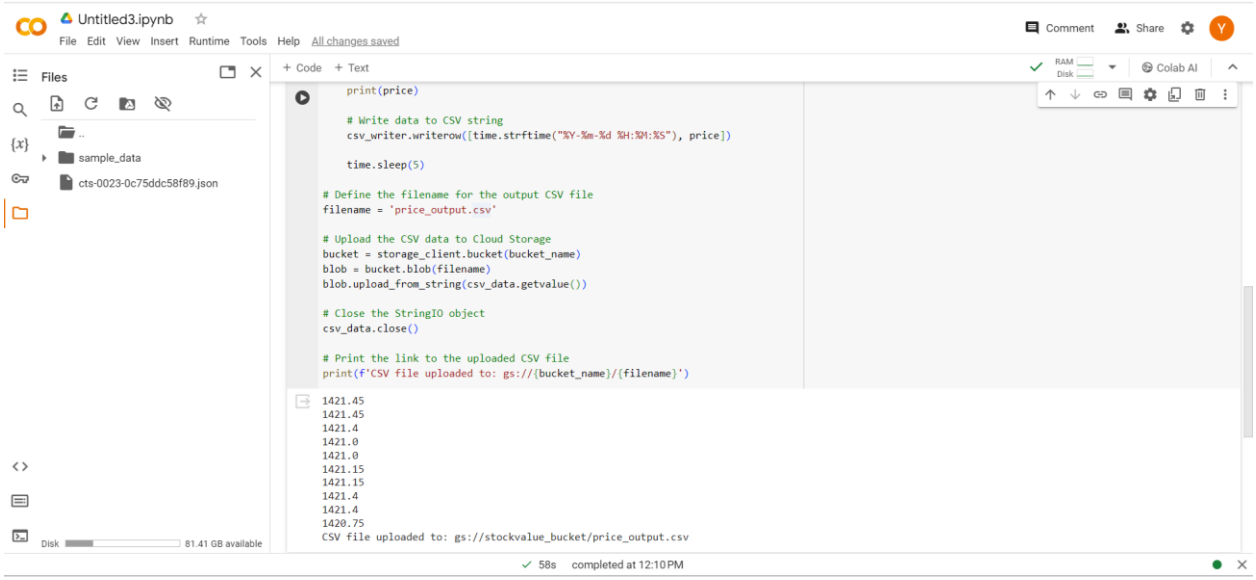


We'll import csv module to store the stock price and its timestamp in a csv file.

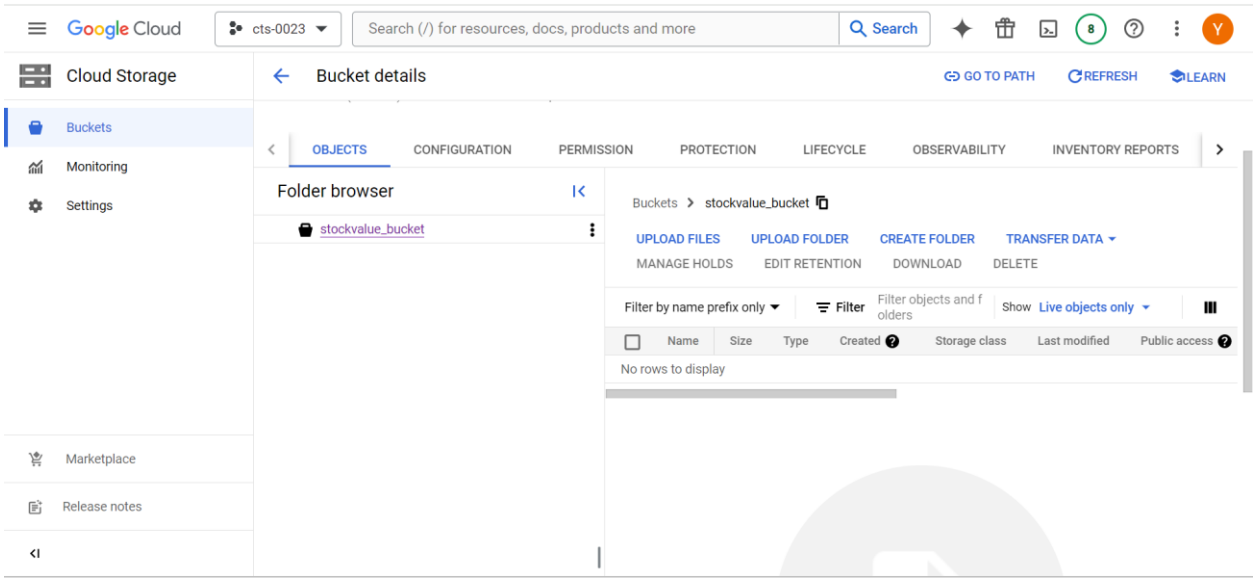
Then, we created a storage client to create a bucket.



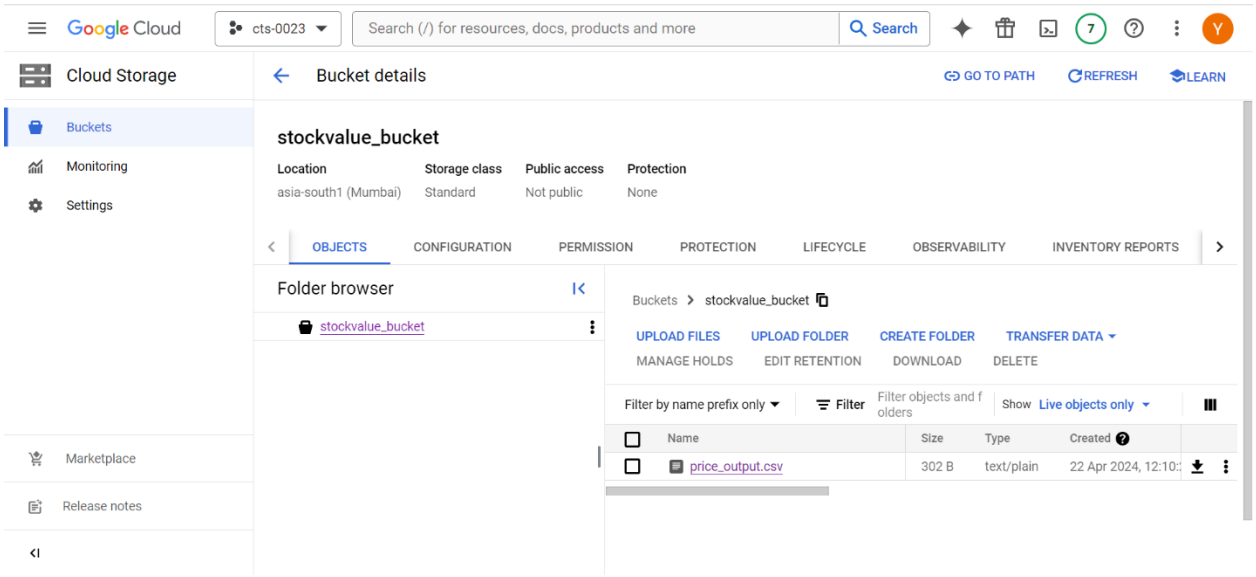
We created a csv file with first 10 stock price values of INFY and stored in price_output.csv



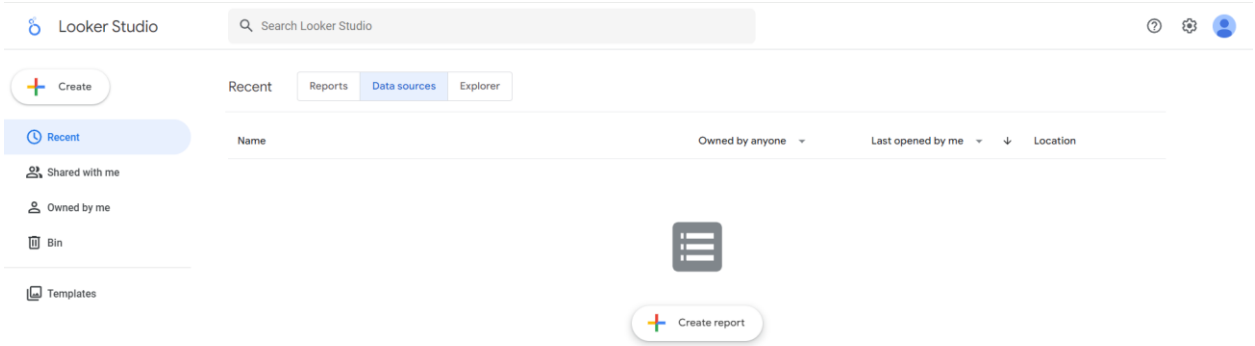
The bucket created before adding object.



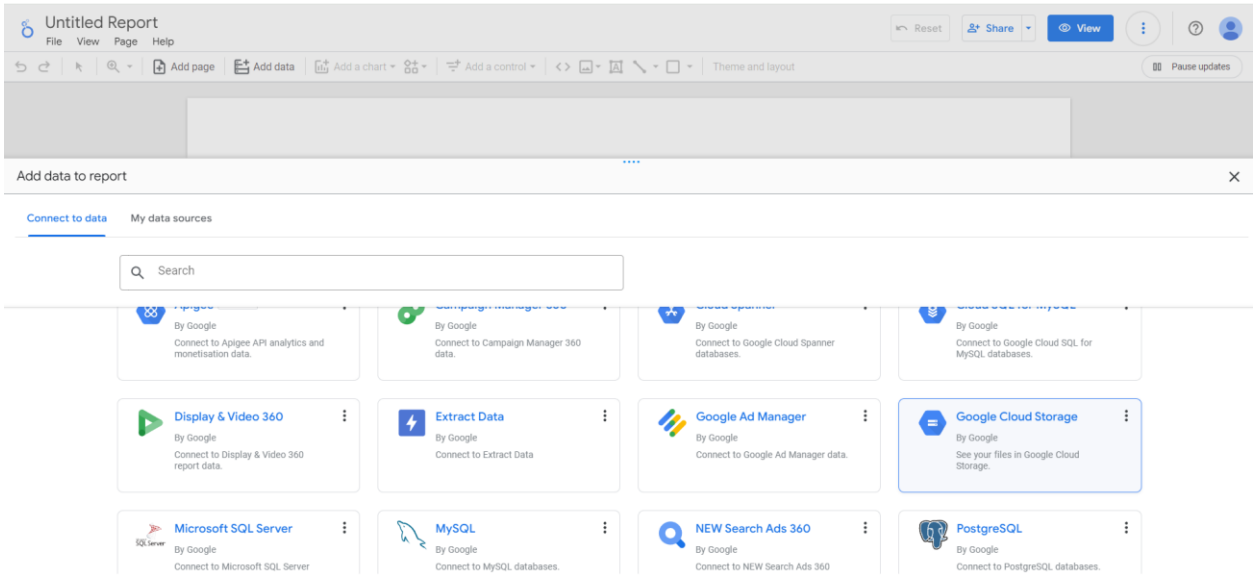
The object “price_output.csv” is reflected in the bucket ‘stockvalue_bucket’.



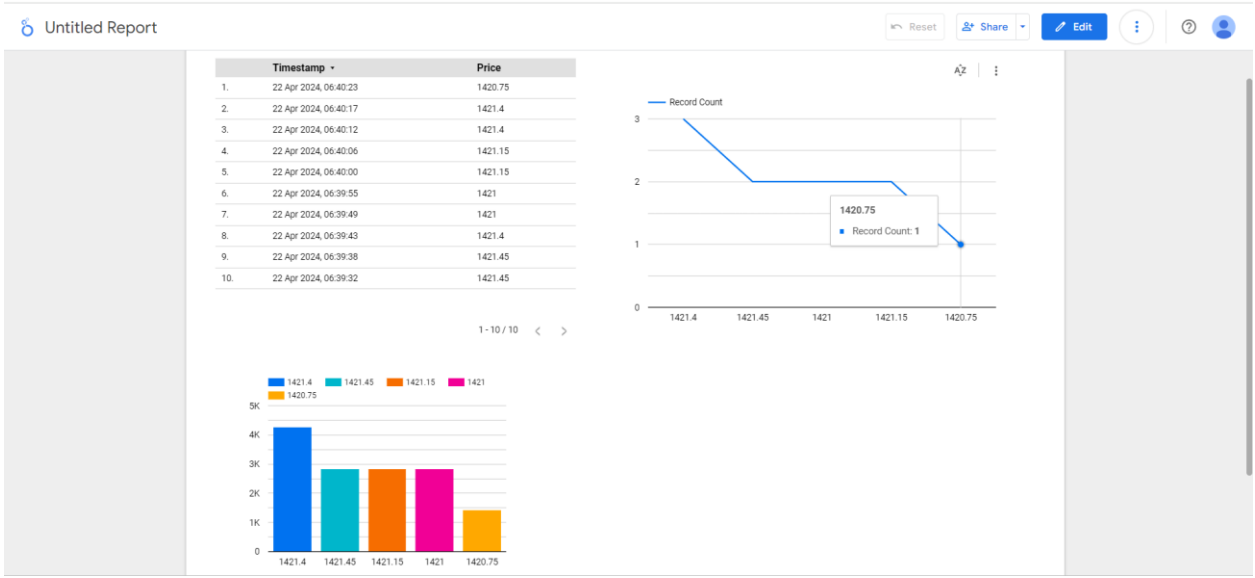
We visualized data using Google Looker Studio by creating report.



We extracted object from the google cloud storage bucket.



Finally, the report is created where y-axis represents Timestamp and x-axis represents Stock price.



7.CONCLUSION

- The real-time stock price monitoring system successfully scrapes stock prices, publishes them to Pub/Sub, and triggers alerts. With the suggested enhancements, you can make it even more powerful and user-friendly.
- We built a real-time stock price monitoring system that uses Cloud Pub/Sub to stream stock price updates and Cloud Functions to analyze and send alerts.
- The system scrapes stock price data from a website, publishes it to a Pub/Sub topic, and triggers a Cloud Function.
- Scraping component scrapes stock price data from a specified URL using BeautifulSoup. We extract the stock price from the HTML page.
- Cloud Function component subscribes to the Pub/Sub topic. It analyzes the stock price data and sends alerts (email and SMS) if the stock price crosses a certain threshold.