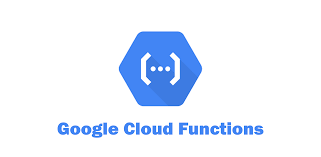
**PROJECT TITLE:**

**Cloud Function + Pub/Sub**

 A blue hexagon with white dots and a logo

Description automatically generated

**Name:** SYED RIZWAN ALI

**Emp ID:** 2319779

**Cohort:** CSDAIAGP24001

**Table of Contents:-**

* **Abstract**
* **Pictorial Representation of Project**
* **Introduction**
* **Cloud Function**
* **Pub/Sub**
* **Project Workflow**

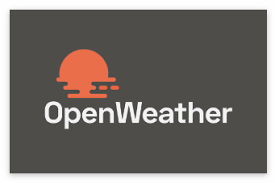
**Implementation Steps**

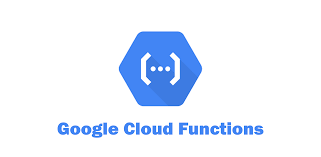
* **Approaches to do the project**
* **Create a Topic in Pub/Sub**
* **Create a Subscription in Pub/Sub**
* **Create a Service account.**
* **Create a Cloud Function.**
* **Once Cloud Function gets deployed , invoke the data from API (Python) to Cloud Function.**
* **Check the content is publish to Pub/Sub.**

**Abstract:**

* Cloud Functions are a serverless computing service, to run code in response to events.
* In this project, I have used Cloud Function that will publishes messages to the topic.
* To automate content delivery based on weather conditions (weather API) using Google Cloud Platform (GCP) services.
* The primary components include a **Cloud Function, a Pub/Sub topic, and an external weather API**.
* First, we create the cloud function, set up a trigger and We can take the data from API based on weather conditions externally and publish the content to the Pub/Sub topic.

**Pictorial Representation of Project**

****



A blue hexagon with white dots and a logo

Description automatically generated

**Introduction**

**Cloud Function:**

* Google Cloud Functions is a serverless execution environment within the Google Cloud Platform (GCP).
* Run your code in the cloud with no servers or containers to manage with our scalable, pay-as-you-go functions as a service (FAAS) product.
* Cloud Functions allow developers to create single-purpose, stand-alone functions.
* It is written in a high-level language such as JavaScript, Python, or Go, and can be triggered by events such as user requests, data changes, or time-based schedules.

**Benefits of Cloud Functions**

1) Cost Savings

2) Flexibility

3) Scalability

4) High Availability

5) Security

Cloud Functions to execute in response to various scenarios by specifying a trigger for your function.

**Triggers supported in Cloud Functions (2nd gen)**

Cloud Functions (2nd gen) supports the following types of triggers:

Triggers.

**1) HTTP triggers,** which react to HTTP(S) requests, and correspond to HTTP functions.

**2) Event triggers,** which react to events within your Google Cloud project, and correspond to event-driven functions.

* Pub/Sub triggers
* Cloud Storage triggers
* Cloud Firestore

**Pub/Sub:**

* Google Cloud Pub/Sub is a fully managed, scalable, global and secure messaging service that allows you to send and receive messages among applications and services.
* Pub/Sub is used for streaming analytics and data integration pipelines to load and distribute data.
* Pub/Sub lets you create systems of event producers and consumers, called publishers and subscribers. Publishers communicate with subscribers asynchronously by broadcasting events, rather than by synchronous remote procedure calls (RPCs).

**In Pub/Sub we have two subcategories:**

* **Topic:** In Pub/Subtopic is a named resource that represents a stream of messages.

You need to create a topic before you can publish or subscribe to it.

You can create a topic using **Google Cloud Console.**

* **Subscription:** Subscriptions in Pub/Sub are used to receive and process the messages that are published to a specific topic

**Dataset**

I took the weather data from **OpenWeather API .**

A screenshot of a computer

Description automatically generated

We used key to get the data from API.

**Approaches:**

We can perform this project by using following ways:

1. **Google Cloud Console**
2. Client Libraries (**Python**, C++, C#, Go)
3. Cloud shell

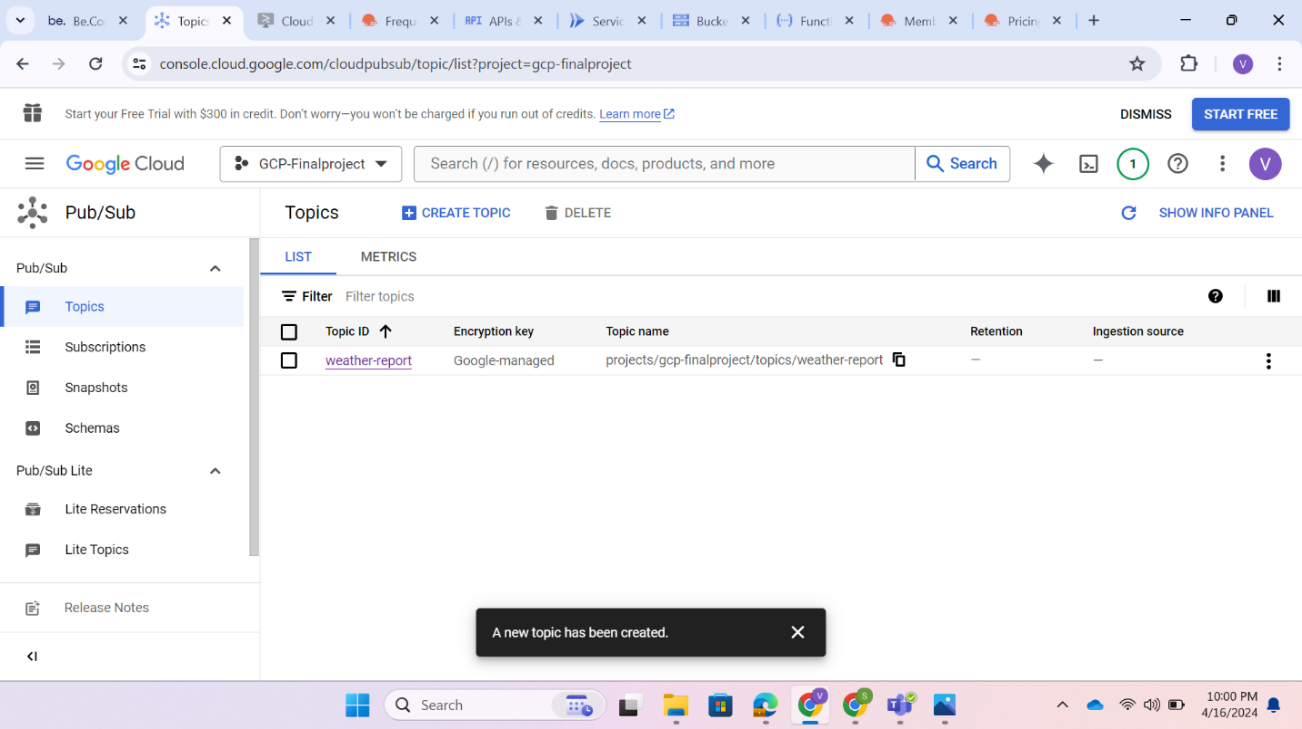
**Implementation Steps**

**Step1:** **Create a Topic in Pub/Sub**

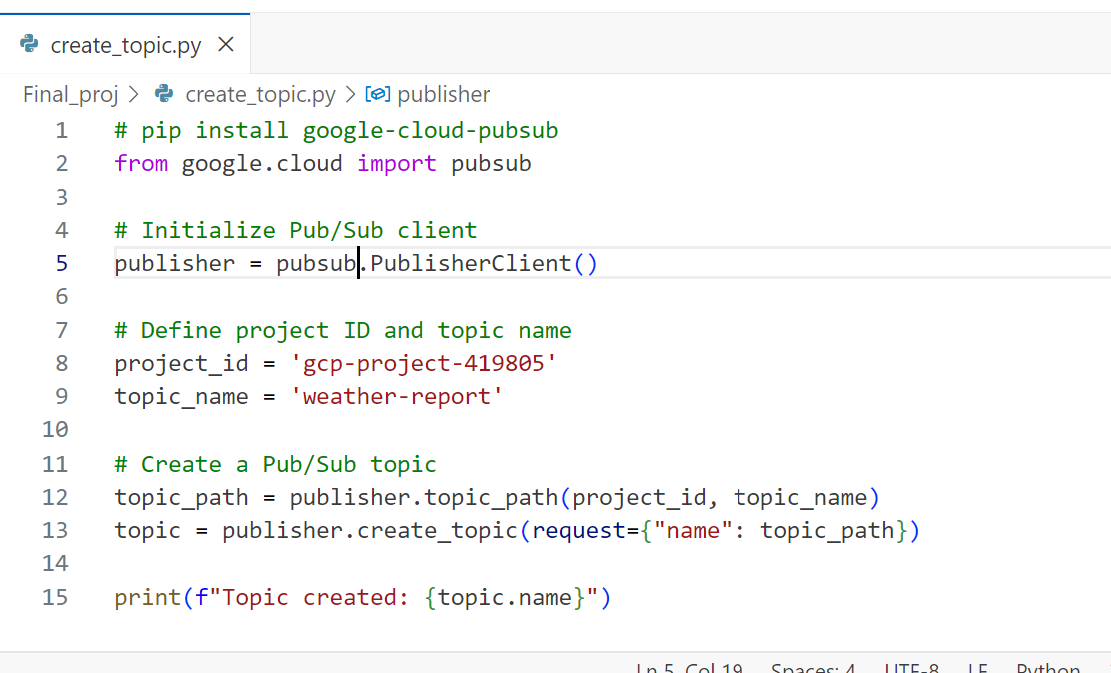
* Navigate to the **Pub/Sub** section in the Google Cloud Console.
* Click **Create a Topic.**

A screenshot of a computer

Description automatically generated



**Second Approach via Python API**

****

**Step 2:** **Create a Subscription in Pub/Sub**

A screenshot of a computer

Description automatically generated

* Click on **Create**

A screenshot of a computer

Description automatically generated

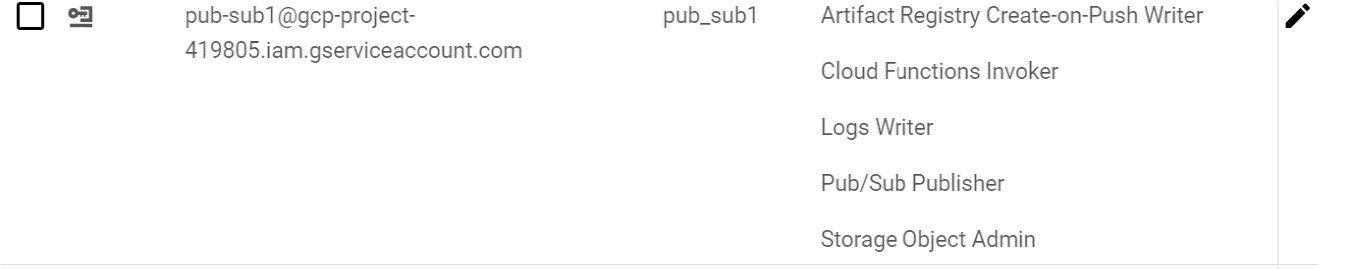
**Second Approach via Python API**



**Step 3: Create a service account.**

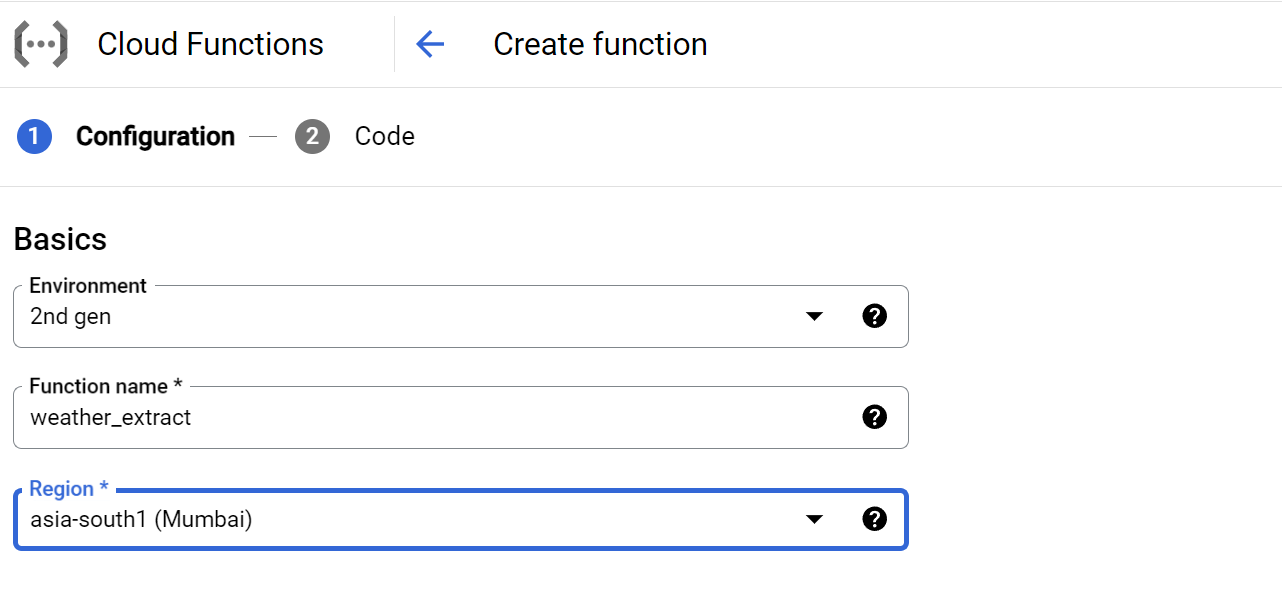
**Add roles:**

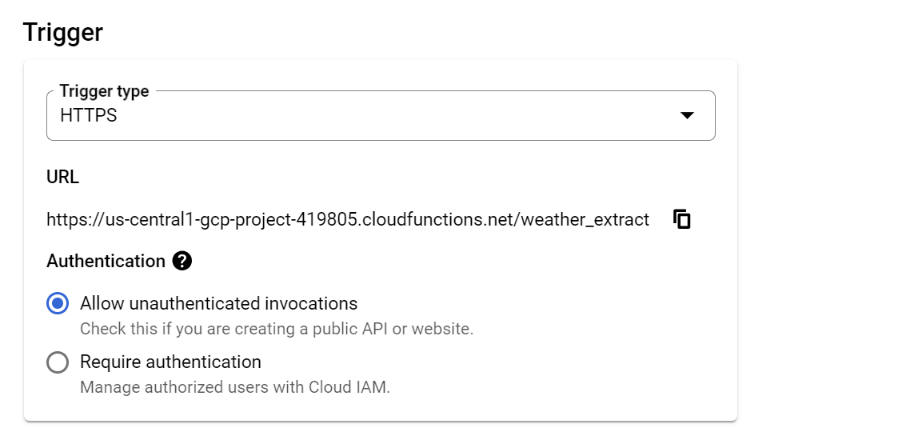
* **Pub/Sub publisher:** To publish the message from cloud function to Pub/Sub topic.
* **Artifact Registry create-on-Push Writer:** This role allows you to read and write artifacts and create gcr.io repositories in Artifact Registry.
* **Cloud Functions Invoker**

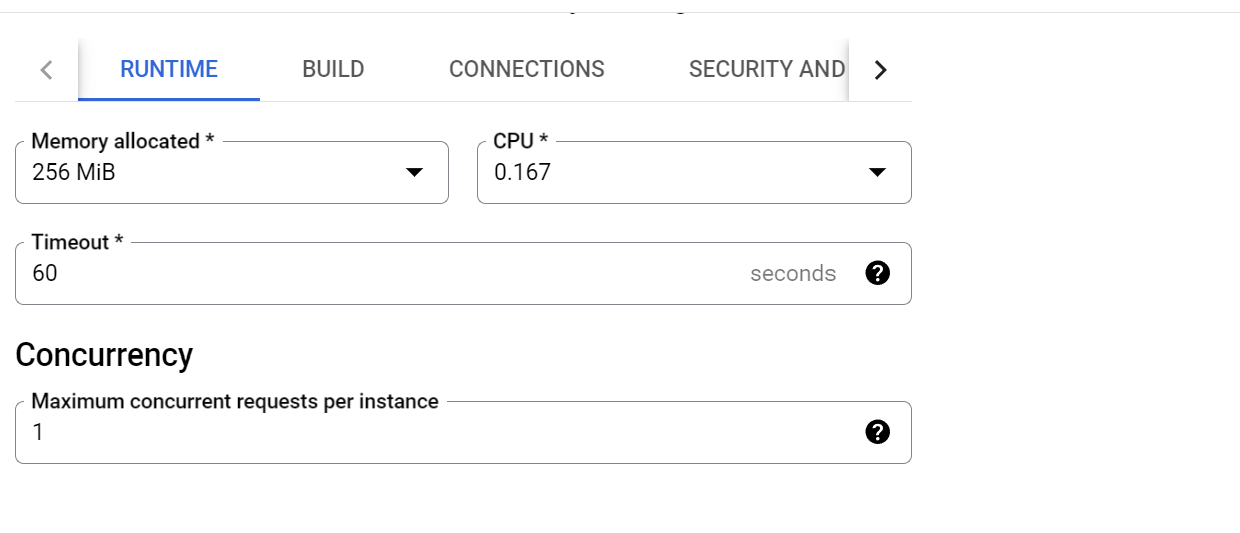
****

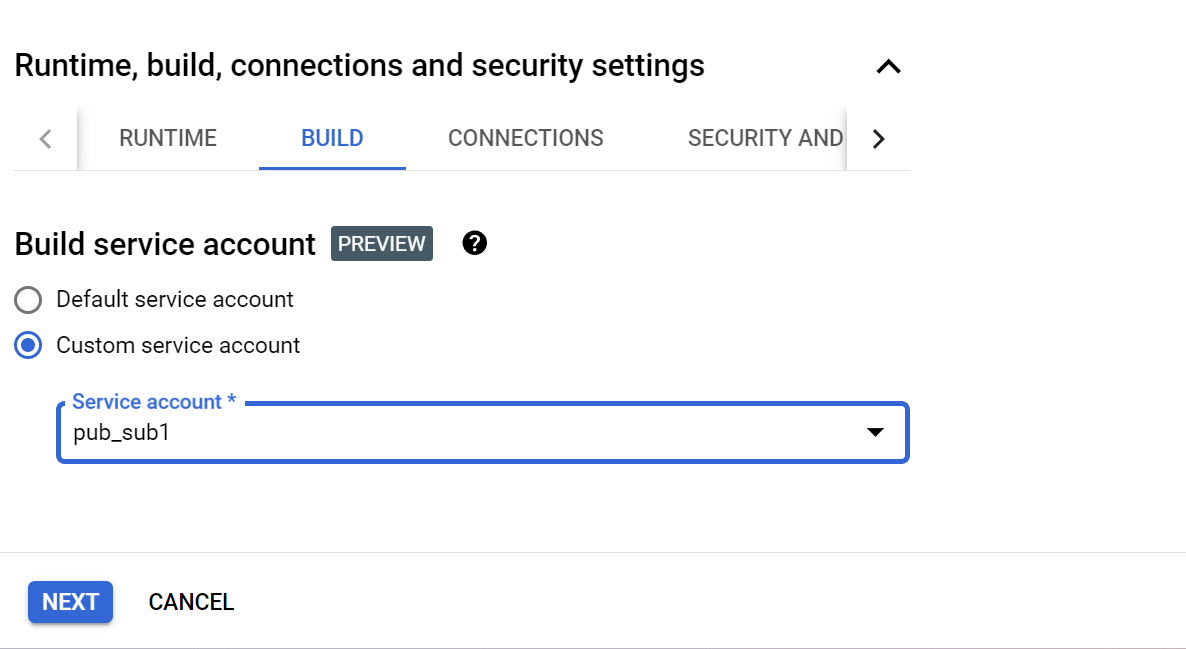
**Step 4: Create a Cloud Function.**

* Navigate to the **Cloud Functions** section in the Google Cloud Console.
* Click on **Create Function.**

****

****

****

****

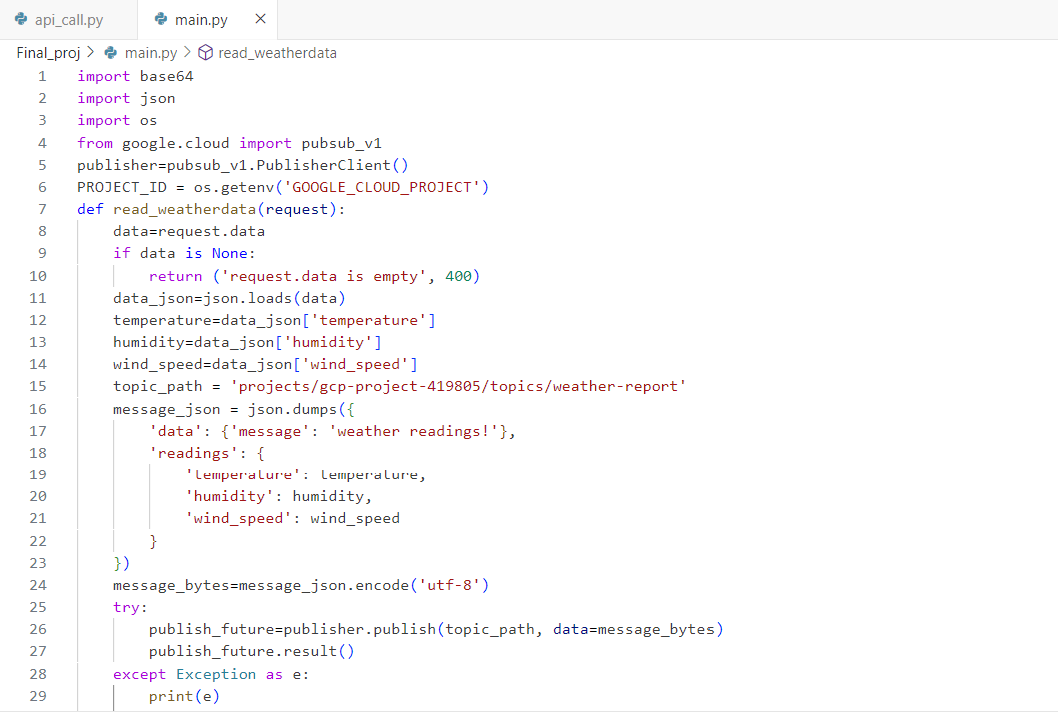
* Click on **NEXT.**

Now under “**Runtime**”, select a Python version. We have selected “**Python 3.9**”. Now here comes the main part, the SOURCE CODE of the function.

**A screenshot of a computer

Description automatically generated**

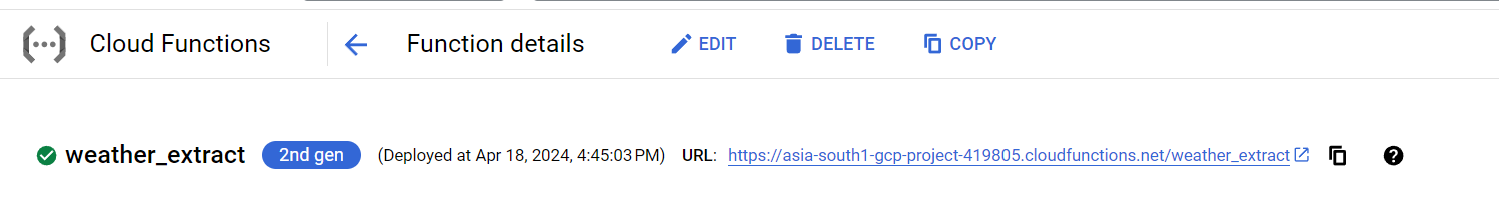
**Note:** The “Entry point” beside the “Runtime” shows the method which will be invoked when a request has been received.



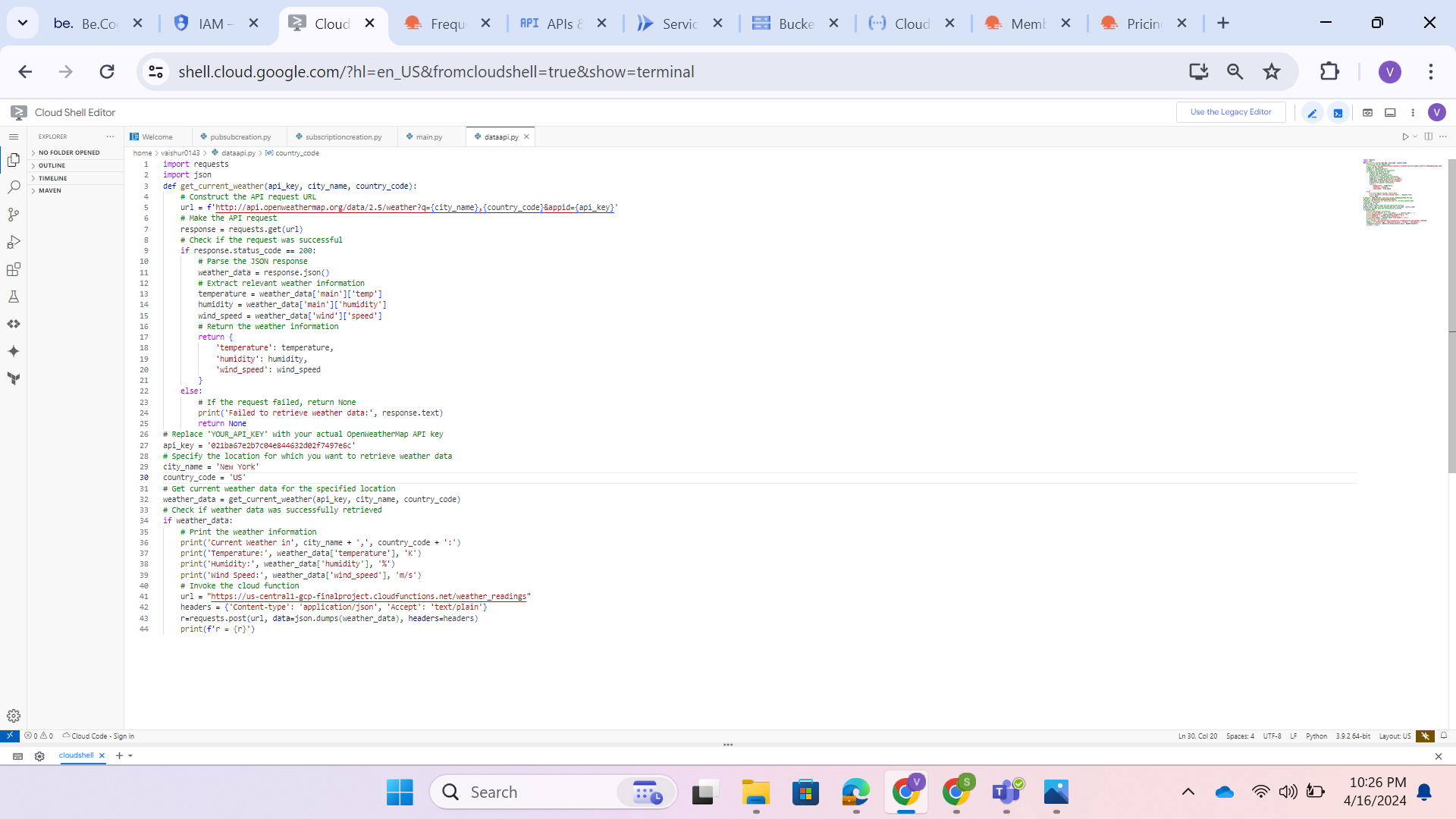
Under the source code section, you will also see the requirements.txt file. This

file is used to specify additional dependencies for your Cloud function.

* Click on **DEPLOY** to deploy the function.

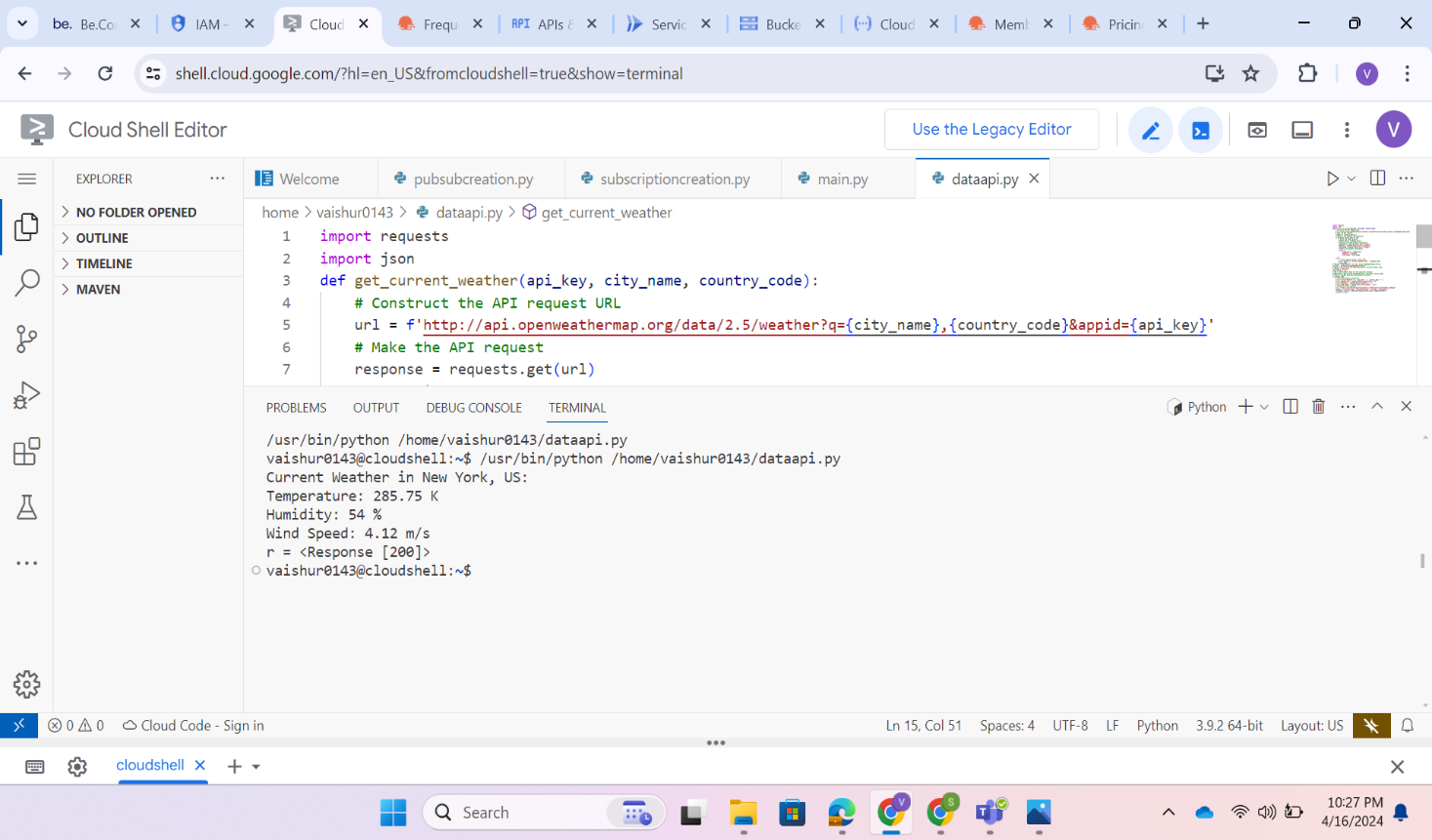


**Step 6: Once Cloud Function gets deployed, invoke the data from API to Cloud Function**

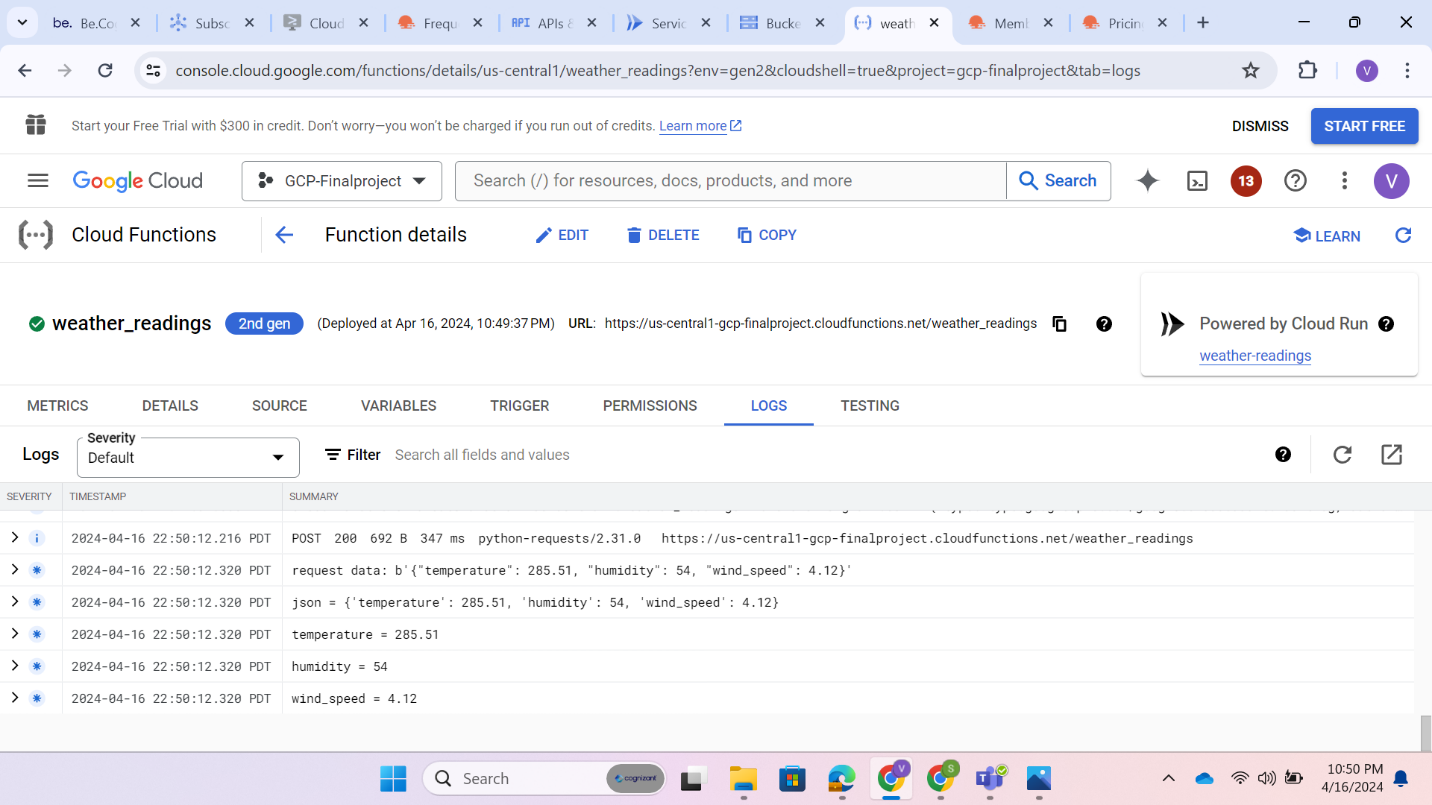


The**weather\_data = response.json()** extracts the JSON object sent by the client.

Then click on **RUN**, output can be display on the terminal.



In the next window, select the **“LOGS”** tab to see the logs of your function.



**Step 8:** **Check the content is publish to Pub/Sub or not.**

* Navigate to the **Pub/Sub** section in the Google Cloud Console.
* Click on **subscriptions**.
* Under Subscriptions click on **Messages** at the bottom.
* Click **PULL.**
* Once We pull it , the message gets displayed in Message body.

A screenshot of a computer

Description automatically generated

**To check the content is published or not.**

A screenshot of a computer

Description automatically generated