**Assignment-15**

# Create Serverless Computing Service using AWS Lambda

📝 **Objective**

To create and deploy a simple AWS Lambda function that prints a custom welcome message — demonstrating serverless computing on AWS.

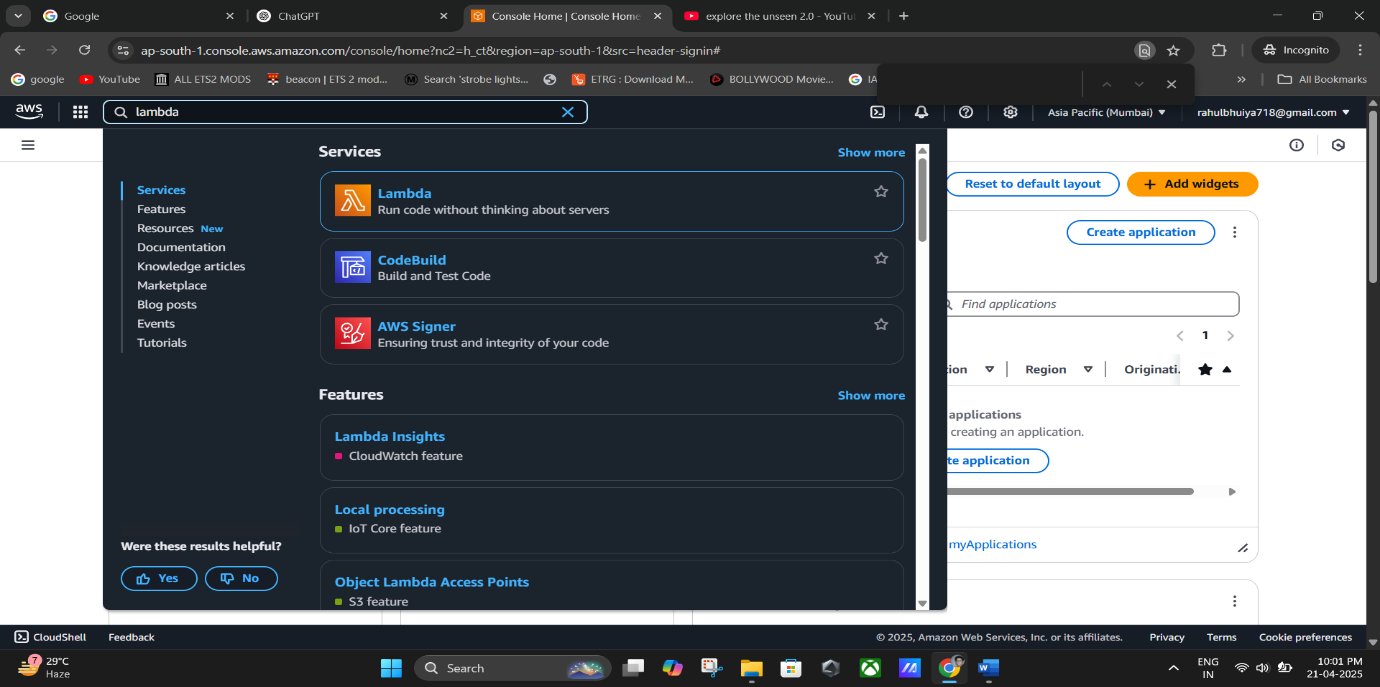
🚀 Part 1: Creating the Lambda Function

## ✅ Step 1: Open Lambda Service

1. Log in to your AWS Console: <https://aws.amazon.com/console/>
2. In the **Search bar**, type **Lambda** and click on it.

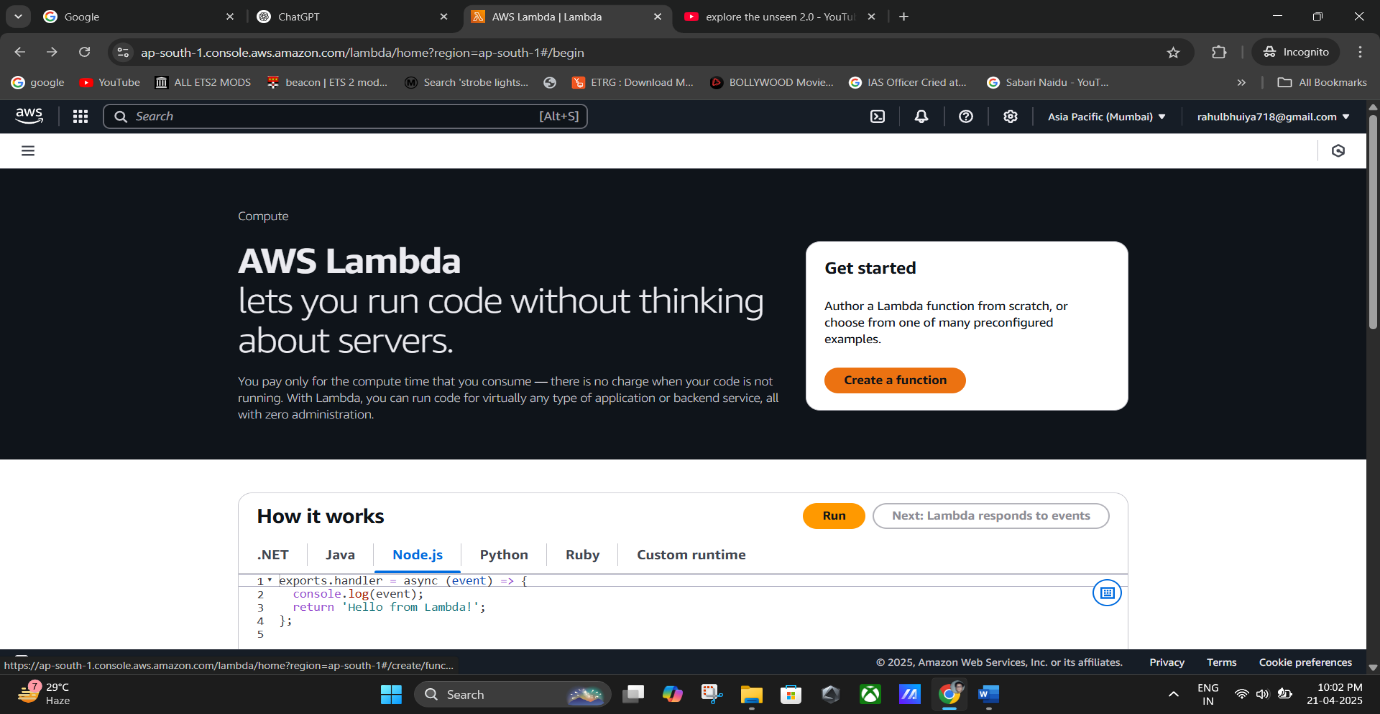
### Explanation:

AWS Lambda lets you run code without managing servers. You only focus on writing the function logic.

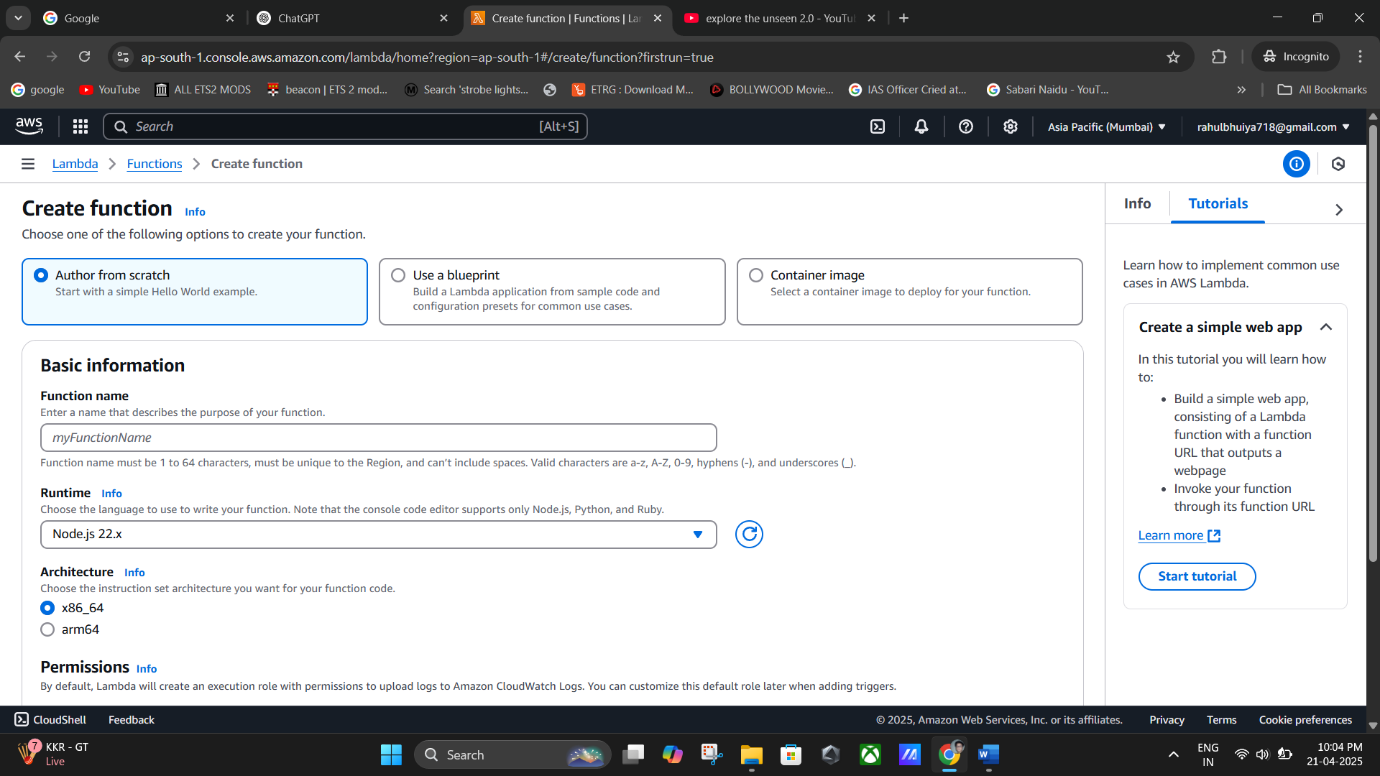


## ✅ Step 2: Create the Function

* 1. Click on the **“Create function”** button.x
  2. Select **“Author from scratch.”**



**(Click on the “Create function”).**



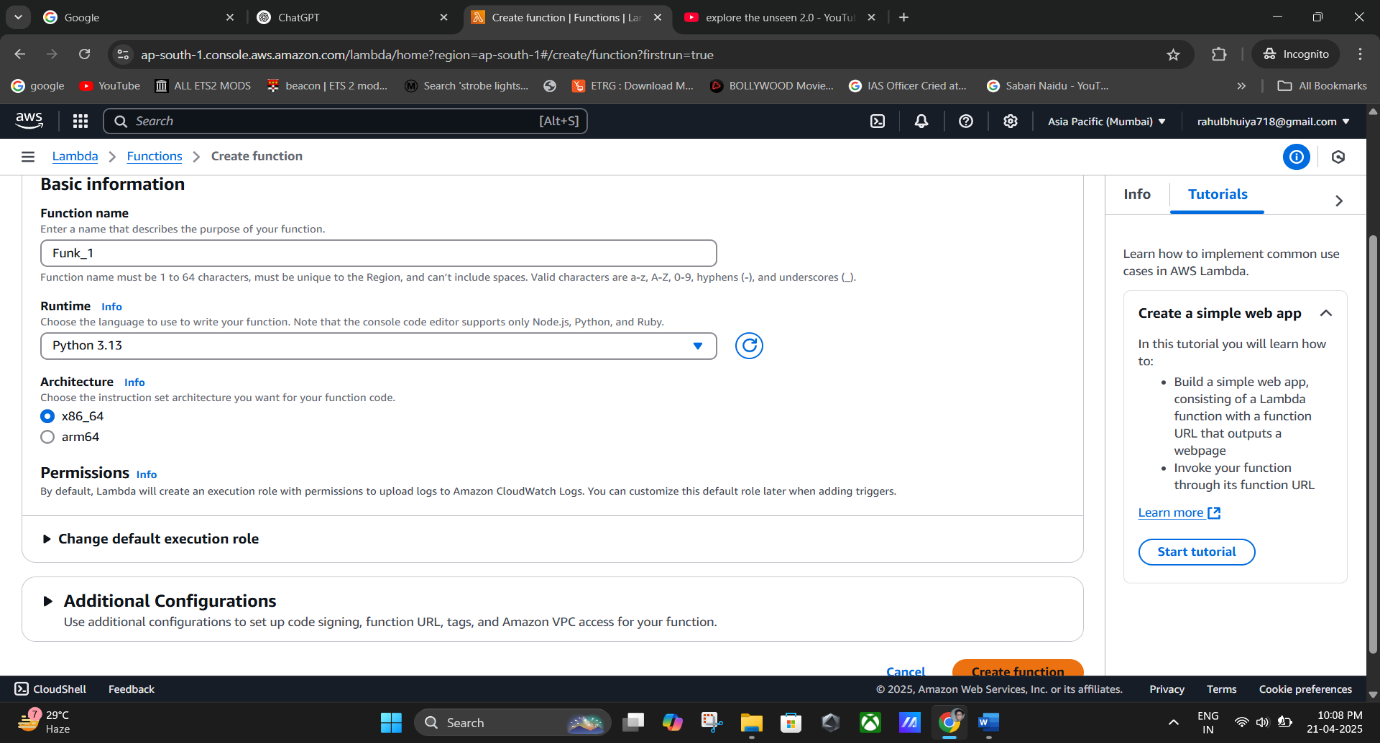
**(Select “Author from scratch).**

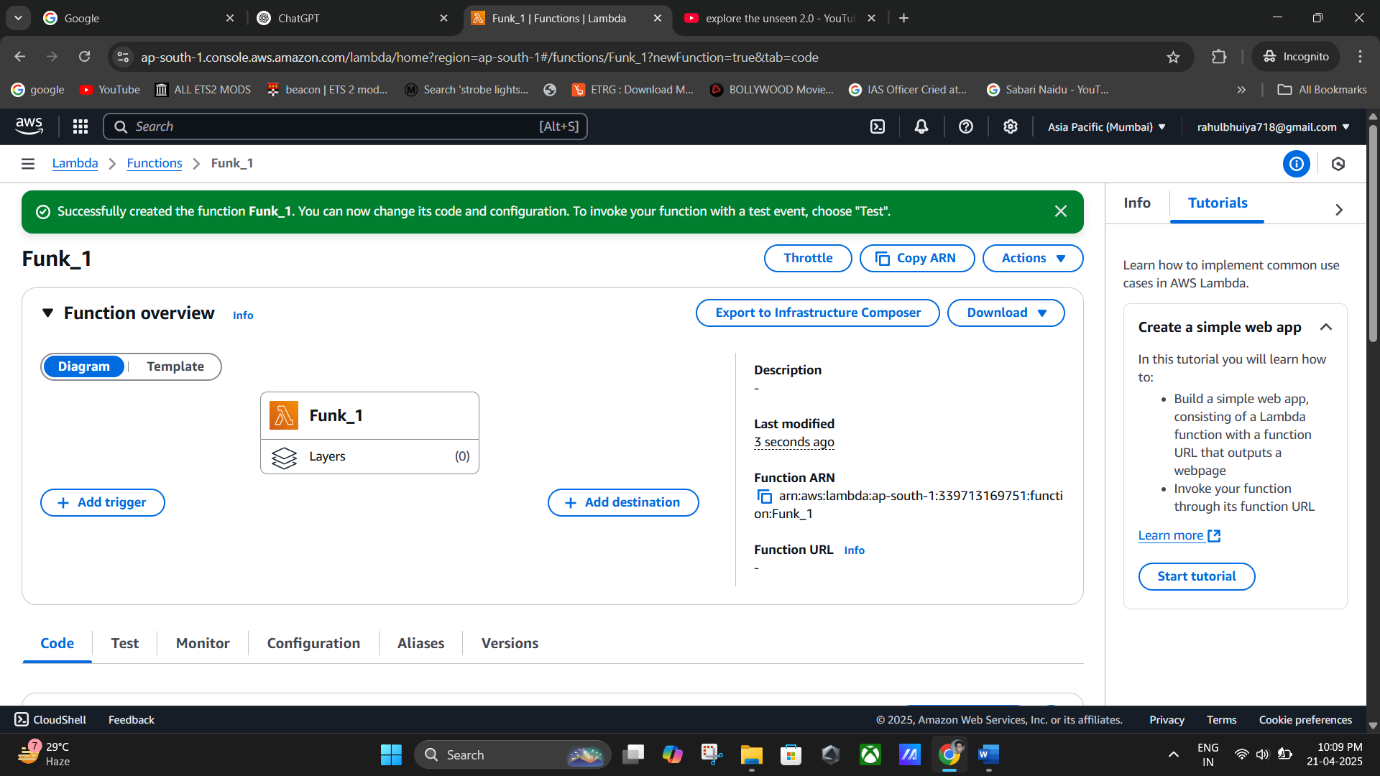
## ✅ Step 3: Set Function Details

* + - **Function name**: e.g., Funk\_1
    - **Runtime**: Choose **Python 3.13** or any preferred runtime (Node.js, etc.)

**Tip:** The runtime determines what programming language your Lambda function will use.

1. Scroll down and leave all other settings as **default**.
2. Click **Create function**.





**(Lambda Function Created Successfully).**

## ✅ Step 4: Modify the Code

1. Wait for the function page to load. You’ll be taken to the function dashboard.
2. Under the **Code** tab, locate and open the index.mjs or main file (for Python, it might be lambda\_function.py).
3. Replace any occurrence of the word **“lambda”** with **“Rahul”** in the sample code.

### ✍️ Example (Node.js):

export const handler = async (event) => { const response = {

statusCode: 200,

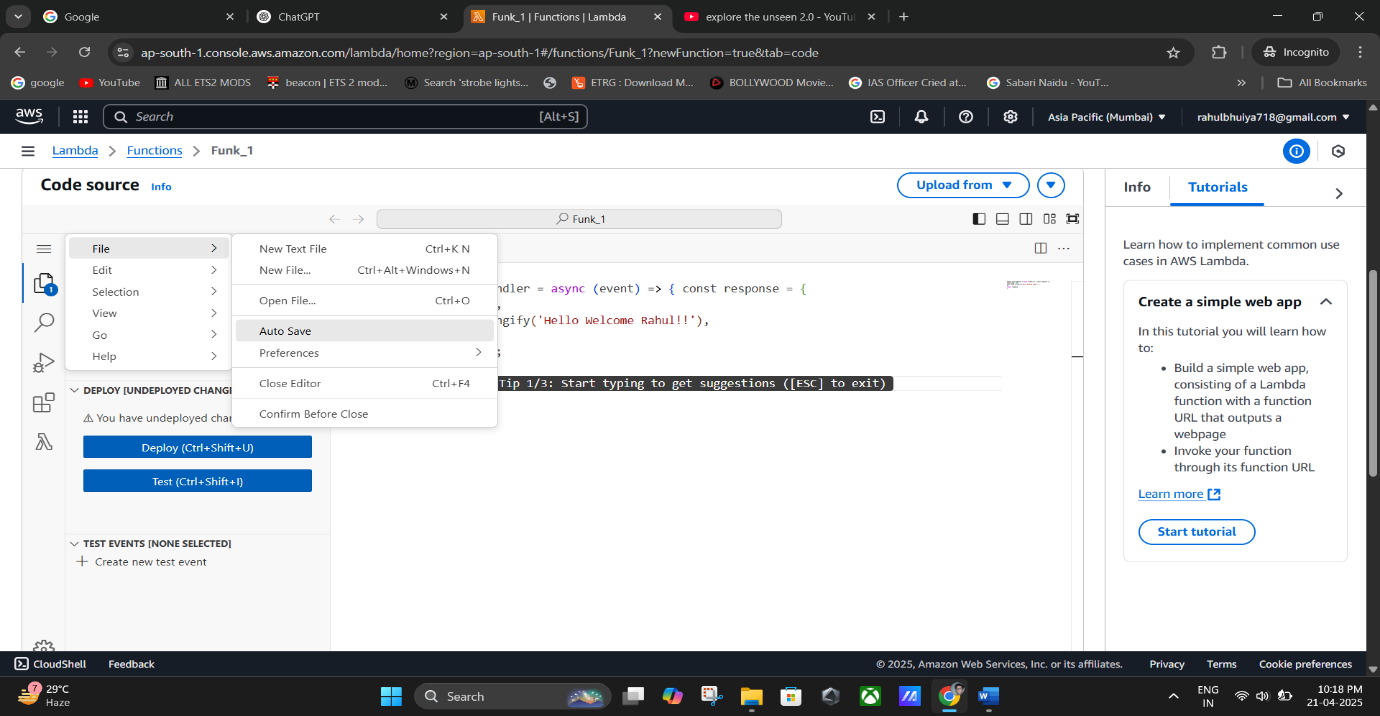
body: JSON.stringify('Hello Welcome Rahul!!'),

};

return response;

};

1. Click **File > Save** to save the code.

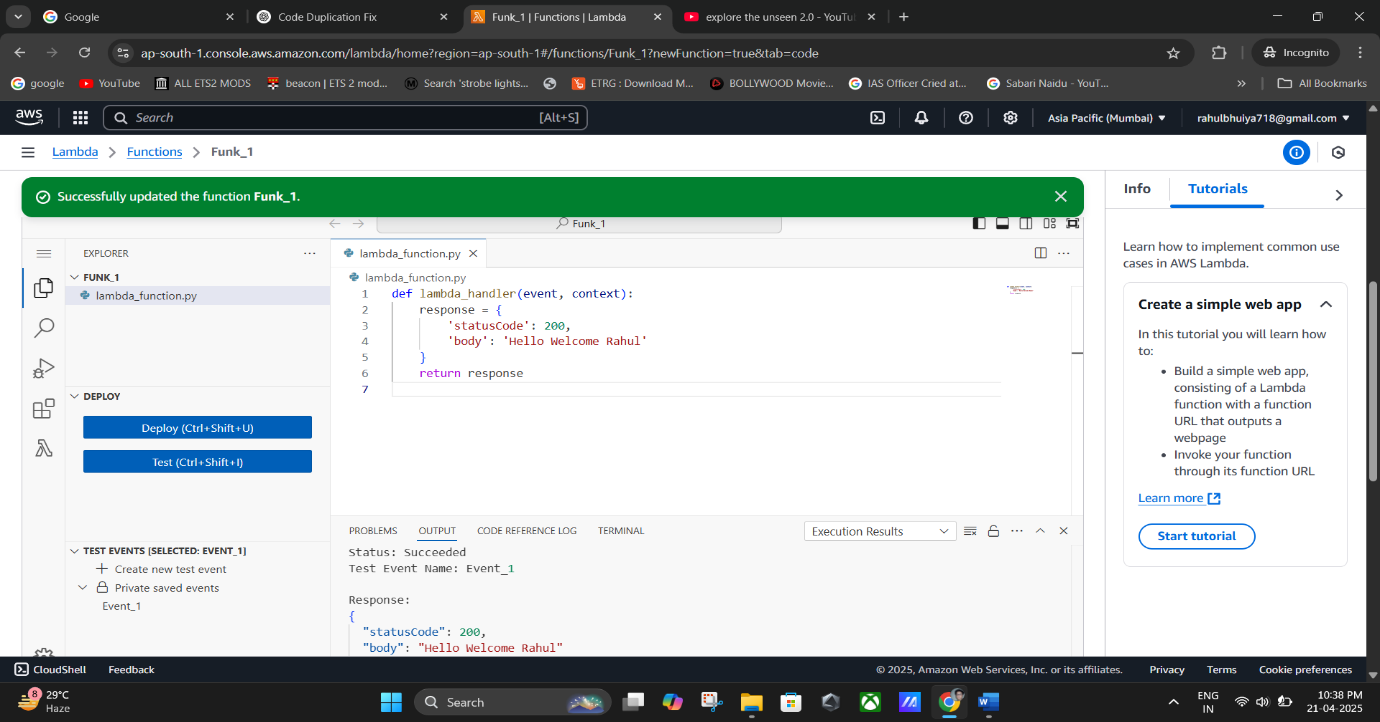


**(Saved The Code).**

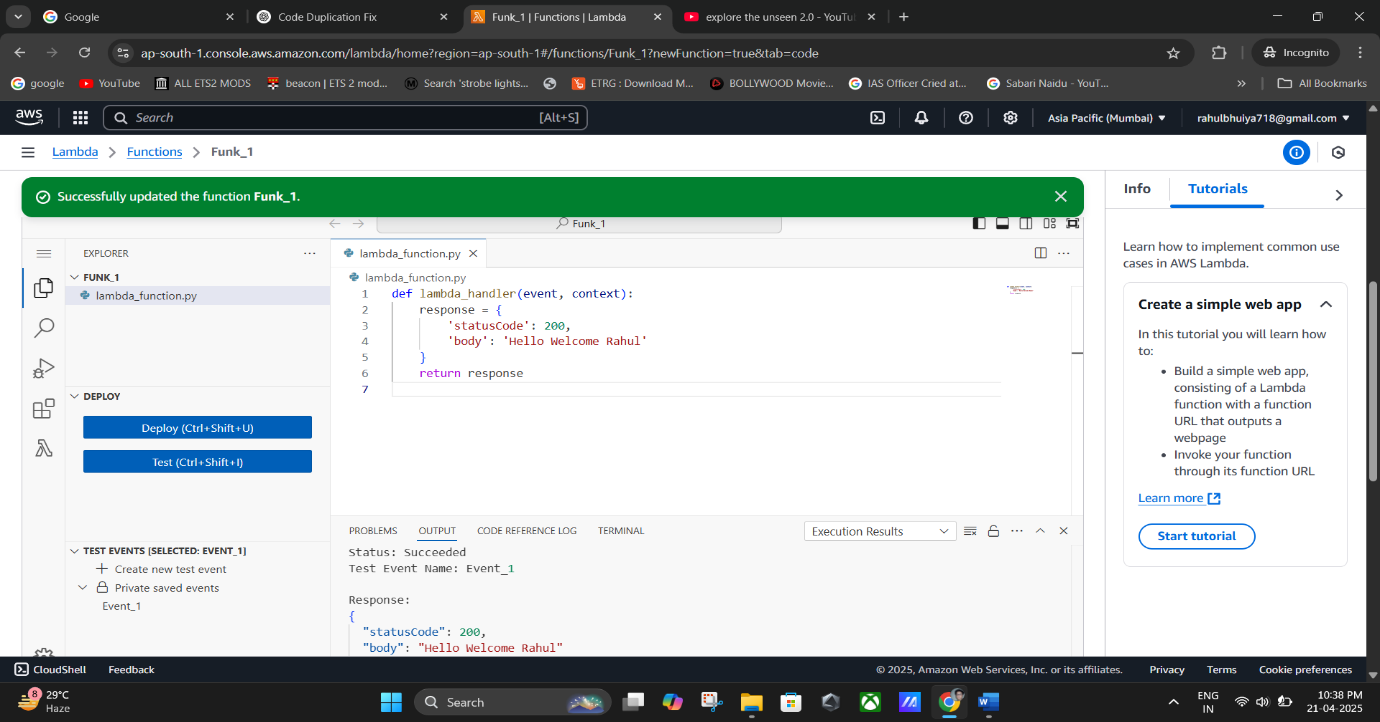
* Part 2: Test the Lambda Function

## ✅ Step 5: Create and Run a Test Event

* 1. Click on the **Test** button (top-right).
  2. Select **“Create new test event.”**
  3. Give it an **Event name**, e.g., Event\_1.
  4. Leave the default JSON data as is (you don’t need to change anything).
  5. Click **Save**.
  6. Now click **Test** to execute the Lambda function.
* **Note**: If you don’t see your message change (e.g., “Hello Welcome Rahul”), it means you haven’t deployed the latest code yet.



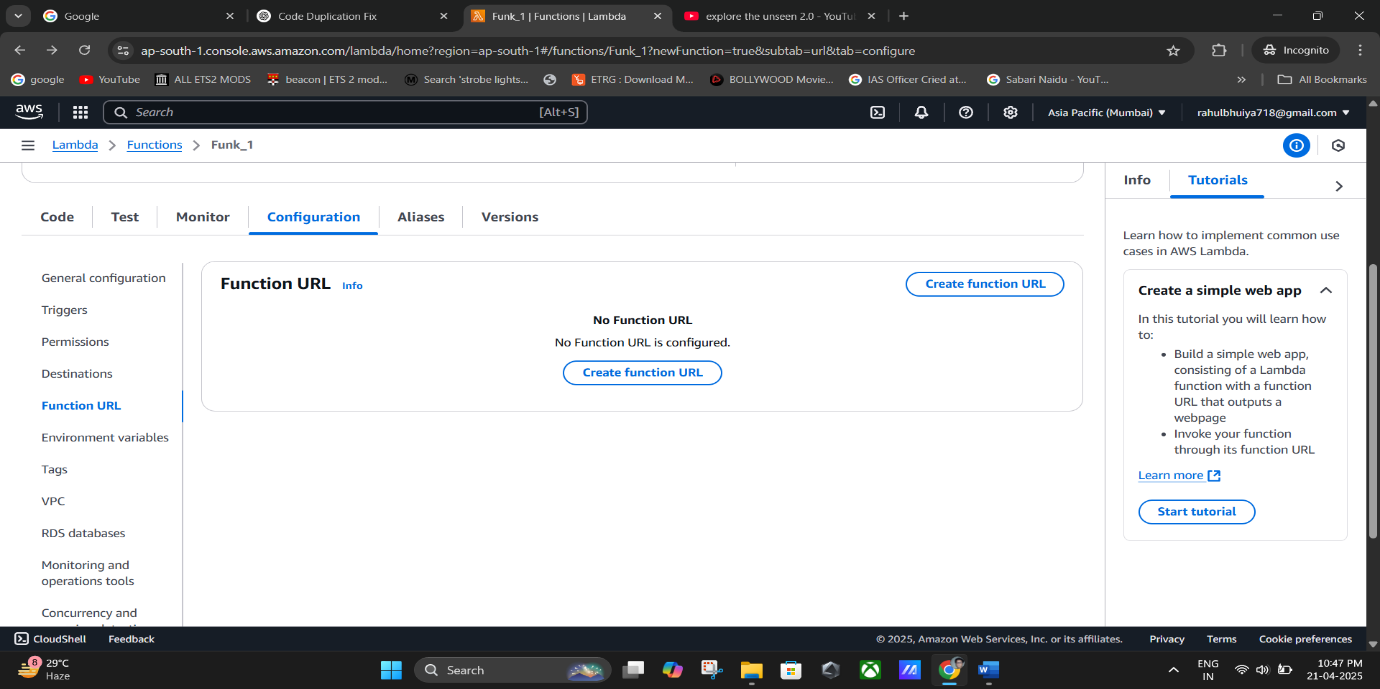
## ✅ Step 6: Deploy and Re-Test

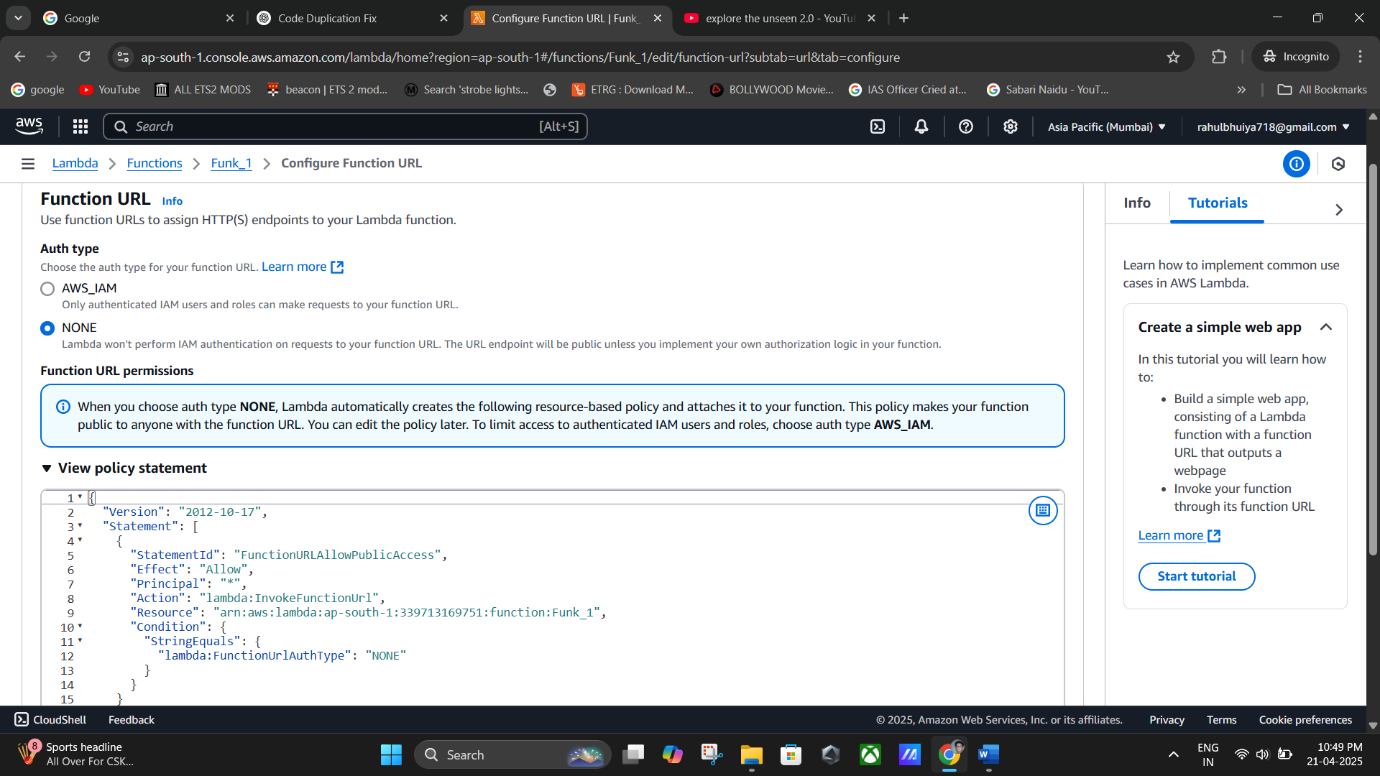
* 1. Click the **Deploy** button to apply your code changes.
  2. Click **Test** again to see the updated result.

🌐 Part 3: Expose Your Lambda Function via URL

## ✅ Step 7: Create Function URL

1. Go to the **Configuration** tab.
2. Under the left-side menu, click **Function URL**.
3. Click **Create function URL**.
4. For **Auth type**, choose **None**.
5. Click **Save**.



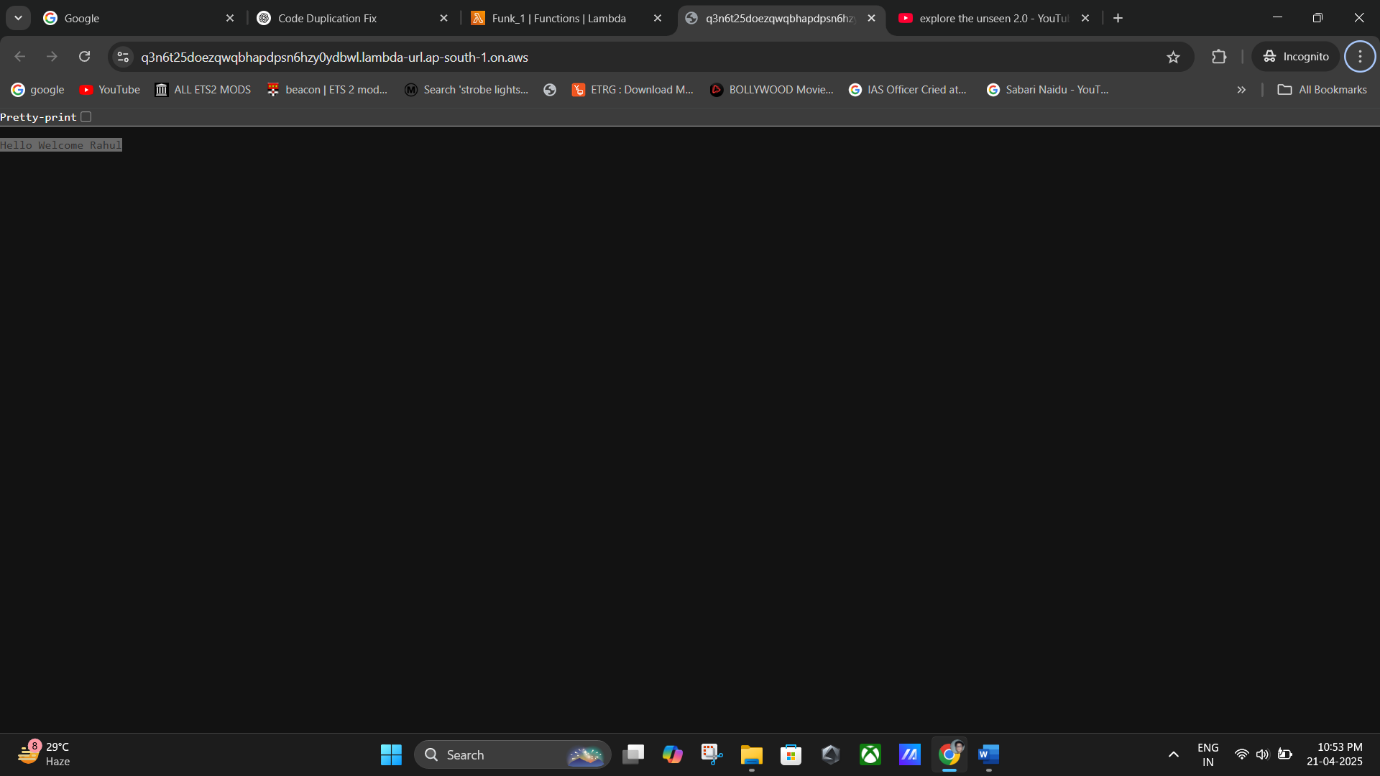


(For **Auth type**-> choose **None** -> Click **Save)**.

## ✅ Step 8: Test the URL

1. Once the Function URL is created, click on it.
2. A new browser tab opens, showing your Lambda function output (e.g., "Hello Welcome Rahul").

⚠️ If you see an error, ensure your function code returns a valid HTTP response.

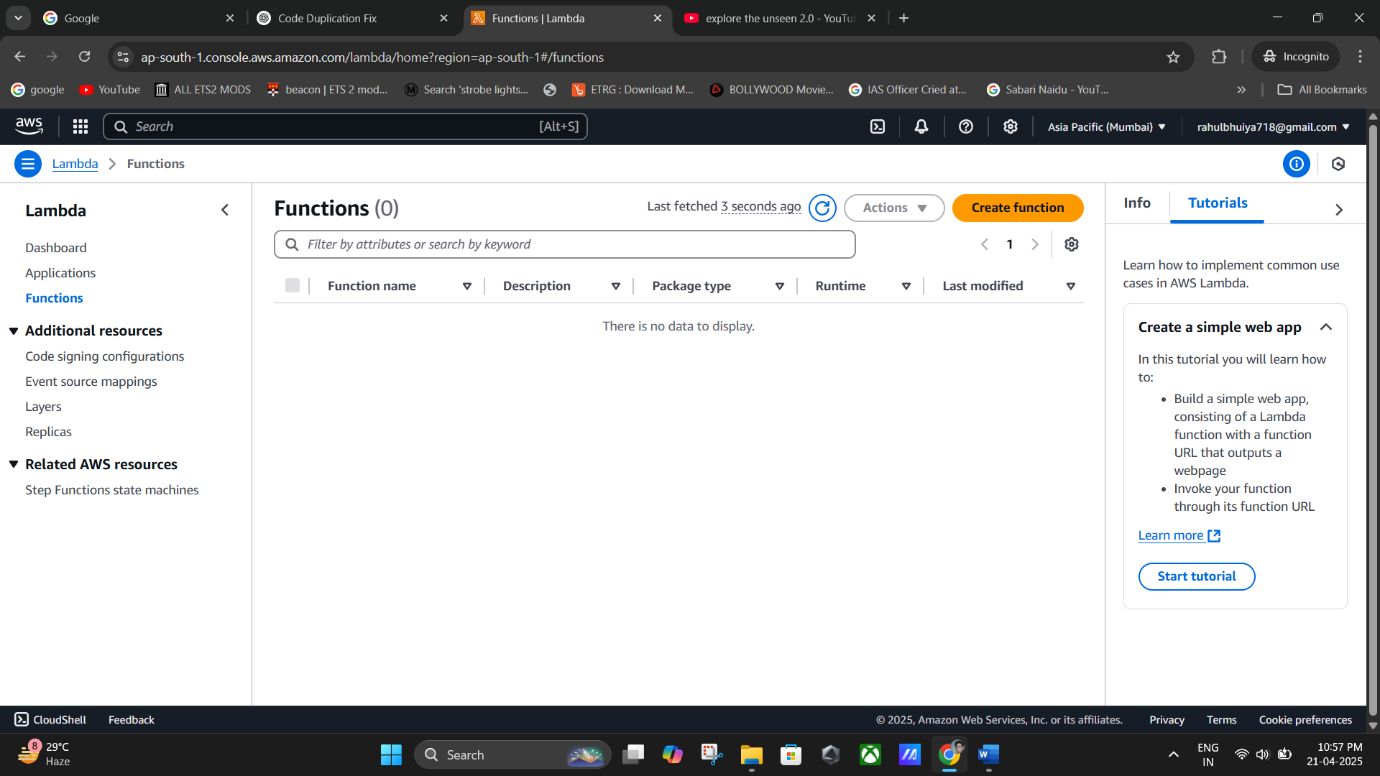


* Part 4: Clean Up Resources

To avoid unnecessary AWS charges:

## ✅ Step 9: Delete Resources

1. Go back to **Configuration > Function URL** and **delete the URL**.
2. Then return to the **Lambda dashboard**, select your function, and click **Delete**.



**(Lambda Function Successfully Deleted).**

🎯 Expected Output

* + After deployment and testing, your function should return: " Hello Welcome Rahul "
  + You should be able to view this output via the Test button and directly from the Function URL.