**Serverless URL Shortener Using AWS Lambda, Api Gateway and DynamoDB**

**INTRO:**

* A SERVERLESS URL SHORTENER THAT CONVERTS LONG URLS INTO A SHORT CODES.

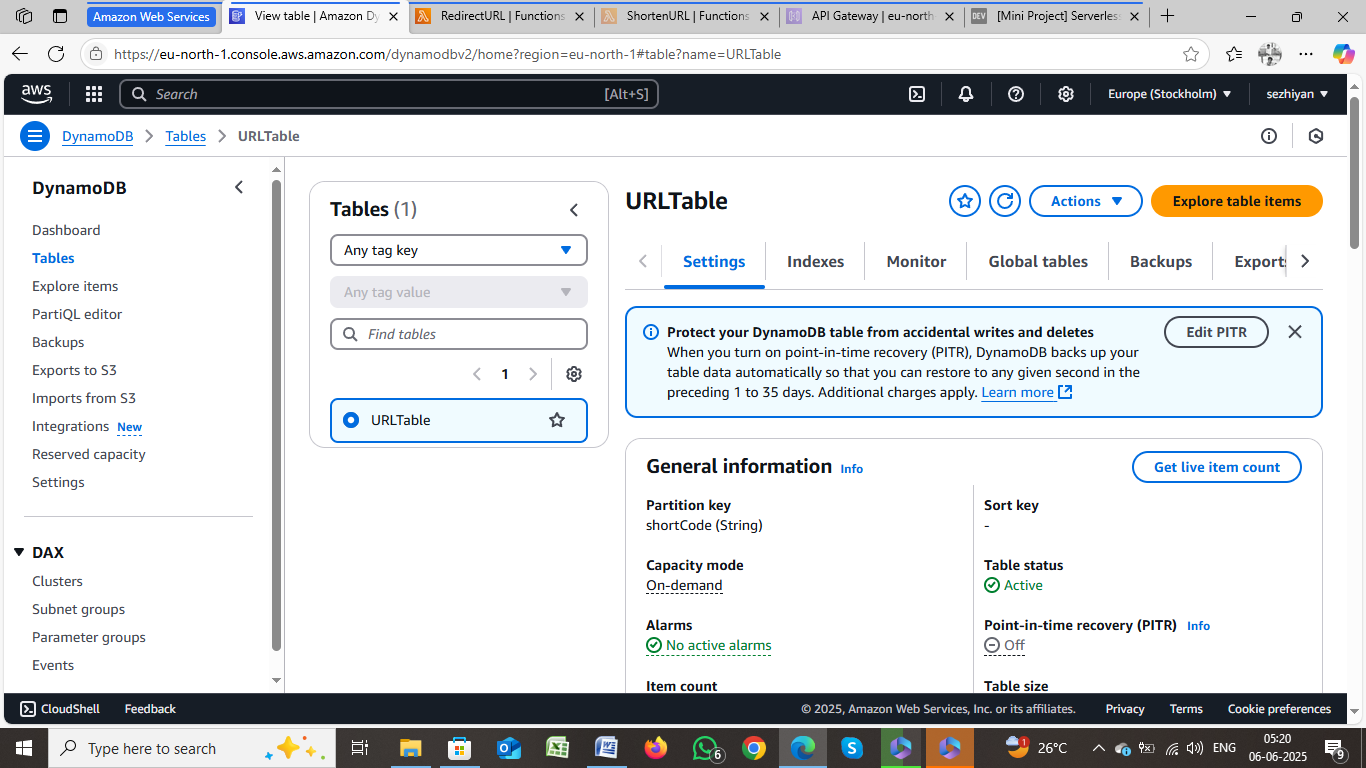
(eg:https://google.com/).

**TOOLS:**

* AWS LAMBDA – HOST LOGIC FOR SHORTENING URLS AND REDIRECTING URLS.
* AMAZON DYNAMODB – STORES SHORT CODES AND CLICK ANALYTICS.
* API GATEWAY – CREATING HTTP ENDPOINTS FOR URL SHORTENER.

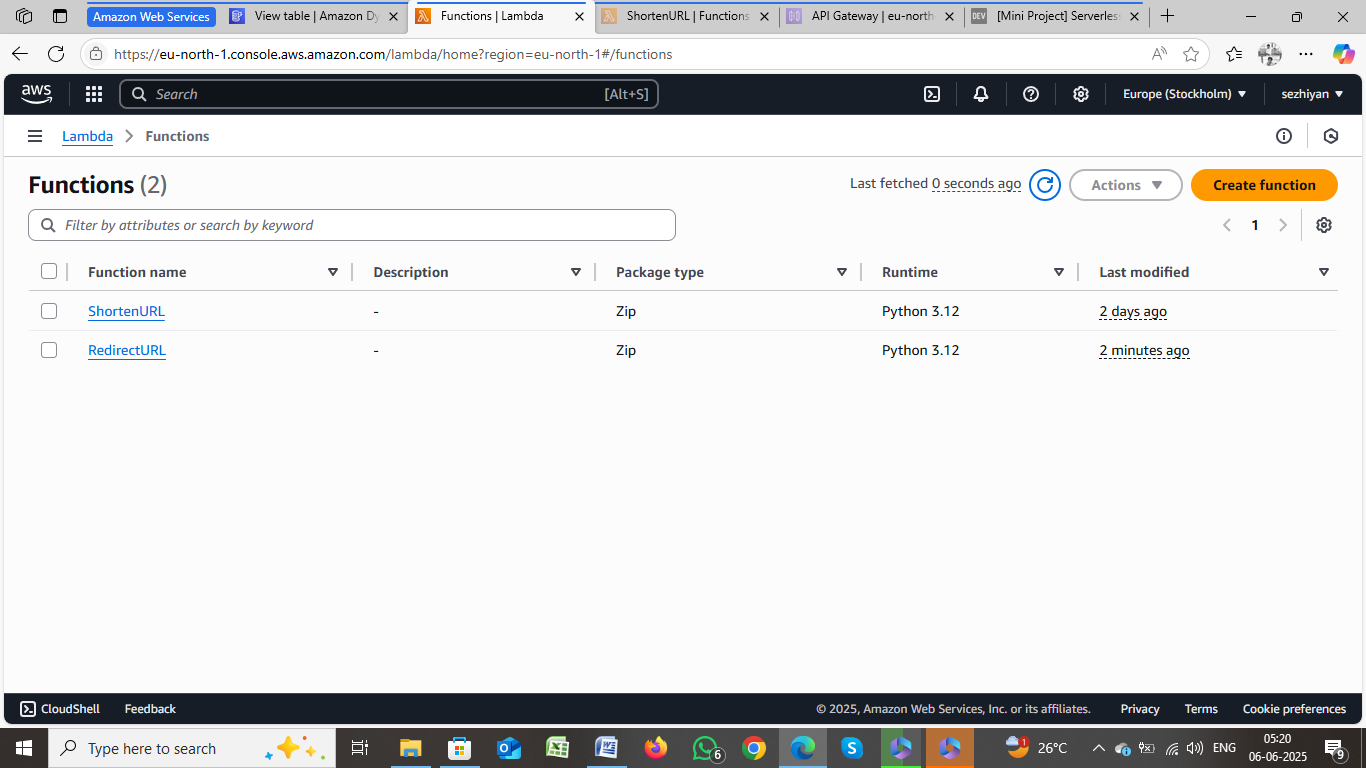
**DYNAMODB**:

1. Open the **AWS Management Console** and navigate to **DynamoDB**.
2. Click **Create Table**.
3. Set the **Table name** **as URLTable**.
4. Set the **Partition key** to **shortcode**(Type: String).
5. Click **Create**.



## Lambda Function:

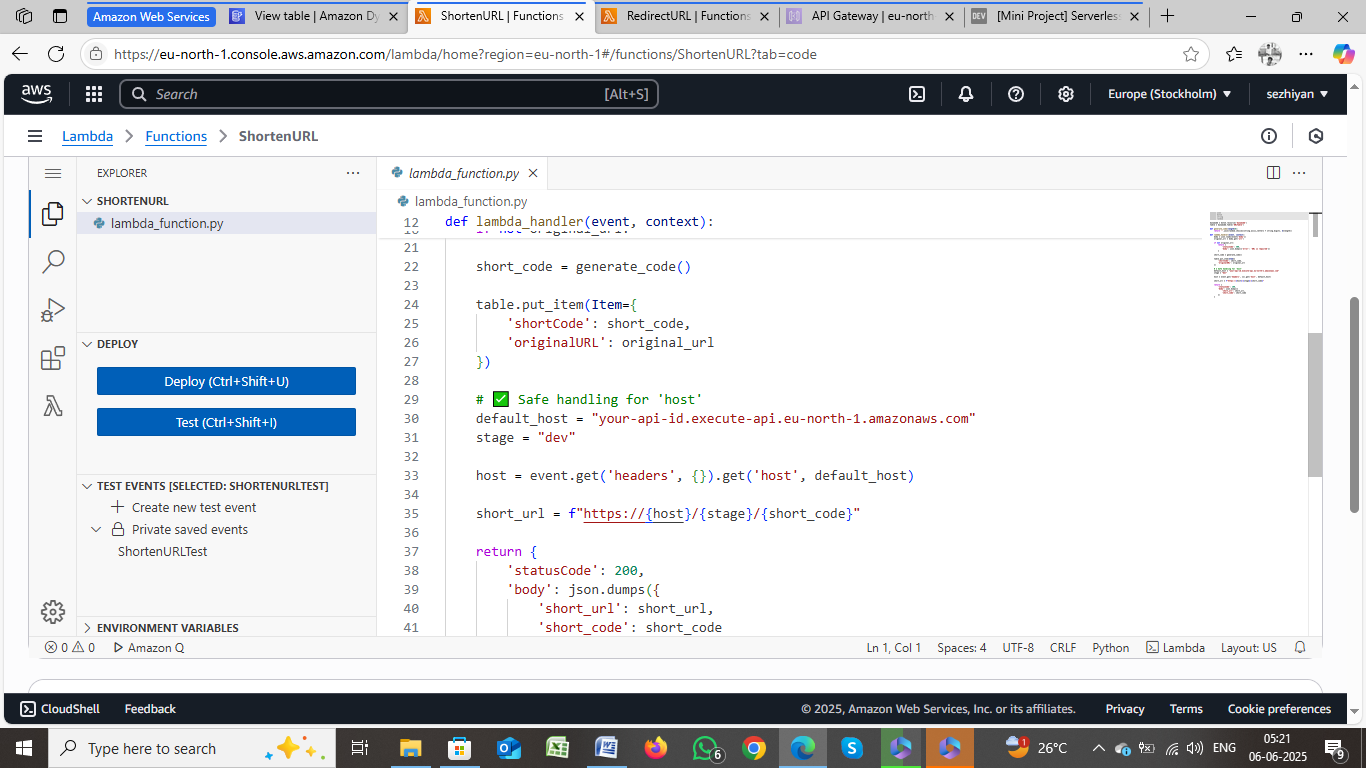
1. Navigate to **AWS Lambda** in the AWS Management Console.
2. Click **Create Function**.Select **Author from Scratch**.
   * Set the **Function name** as **(ShortenURL AND RedirectURL)**
   * Set the**Runtime** to Python 3.12
   * In the **Permissions** section, create a new role with basic Lambda permissions (AWS will automatically create the role).
3. Click **Create Function**.



Add Code to the Lambda Function ShortenURL

1. import json
2. import boto3
3. import string
4. import random
5. dynamodb = boto3.resource('dynamodb')
6. table = dynamodb.Table('URLTable')
7. def generate\_code(length=6):
8. return ''.join(random.choices(string.ascii\_letters + string.digits, k=length))
9. def lambda\_handler(event, context):
10. body = json.loads(event['body'])
11. original\_url = body.get('url')
12. if not original\_url:
13. return {
14. 'statusCode': 400,
15. 'body': json.dumps({'error': 'URL is required'})
16. }
17. short\_code = generate\_code()
19. table.put\_item(Item={
20. 'shortCode': short\_code,
21. 'originalURL': original\_url
22. })
23. # ✅ Safe handling for 'host'
24. default\_host = "your-api-id.execute-api.eu-north-1.amazonaws.com"
25. stage = "dev"
26. host = event.get('headers', {}).get('host', default\_host)
28. short\_url = f"https://{host}/{stage}/{short\_code}"
29. return {
30. 'statusCode': 200,
31. 'body': json.dumps({
32. 'short\_url': short\_url,
33. 'short\_code': short\_code
34. })
35. }

AND CLICK DIPLOY



Add Code to the Lambda Function IN **RedirectURL**

import json

import boto3

dynamodb = boto3.resource('dynamodb')

table = dynamodb.Table('URLTable')

def lambda\_handler(event, context):

    short\_code = event['pathParameters']['code']

    response = table.get\_item(Key={'shortCode': short\_code})

    if 'Item' not in response:

        return {

            'statusCode': 404,

            'body': json.dumps({'error': 'Short URL not found'})

        }

    original\_url = response['Item']['originalURL']

    return {

        'statusCode': 302,

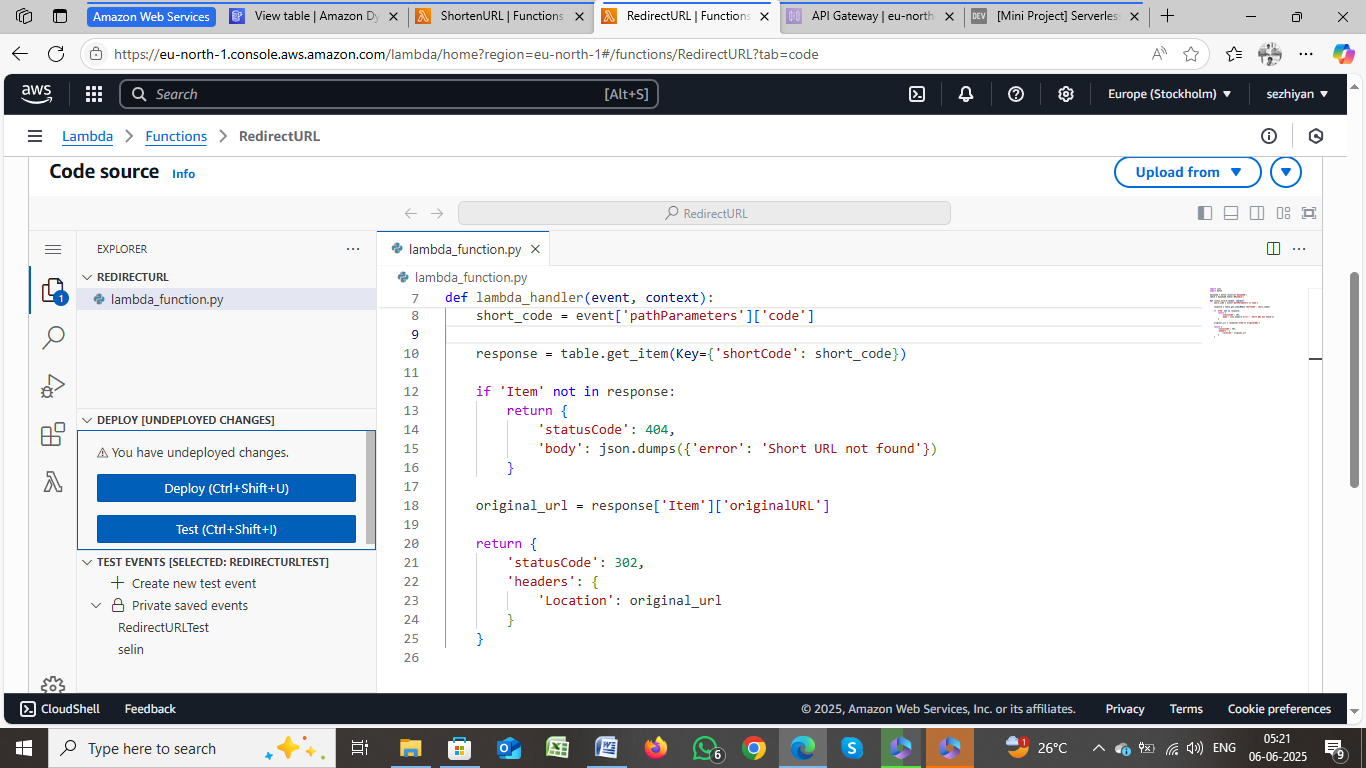
        'headers': {

            'Location': original\_url

        }

    }

* + Click Deploy to save the changes.

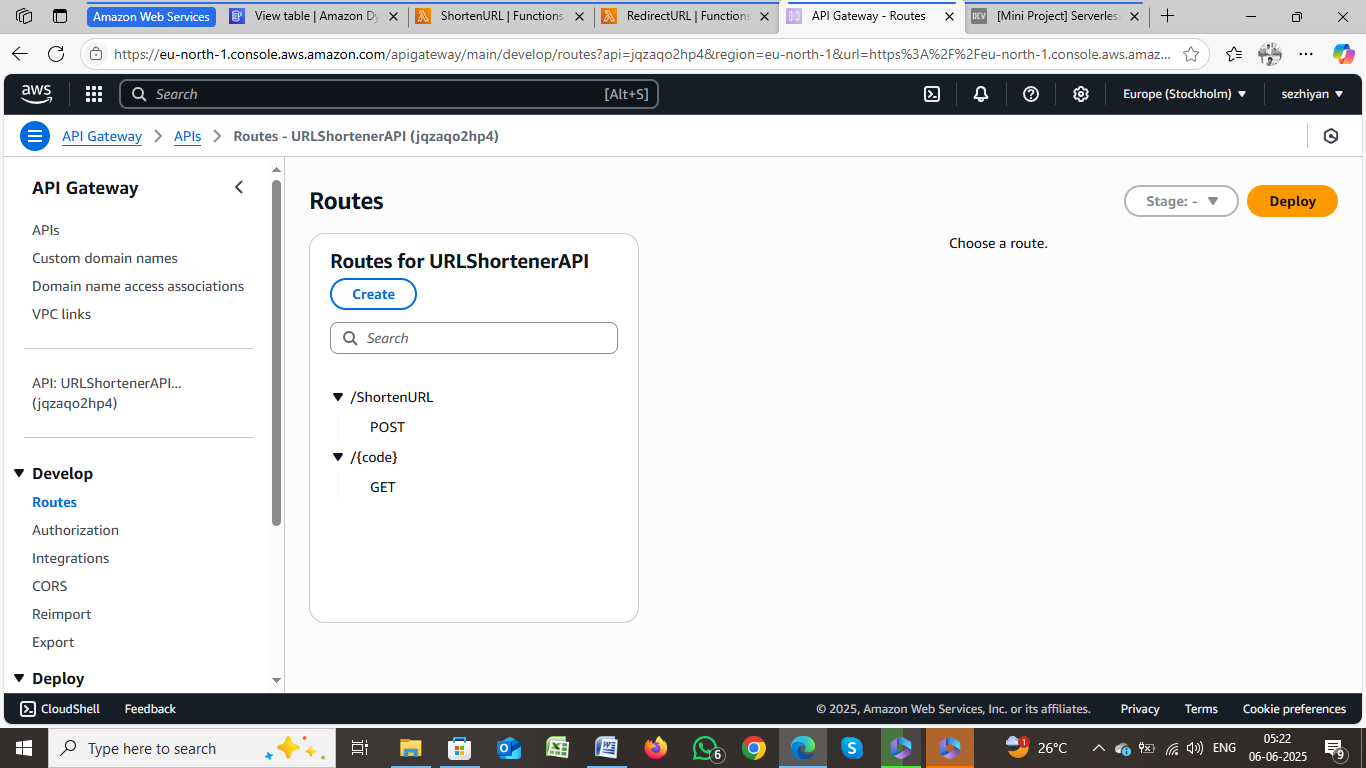


**API Gateway:**

Click on **Create API**. Choose **HTTP API**, and then click **Build** to begin creating a new API. **POST /shorten**.**GET /redirect**.

**Integrate Routes with Lambda.**Click on **Deploy** to create a new stage.

Name the stage, such as DEV



GO TO A LAMBDA FUNCTION IN **ShortenURL**

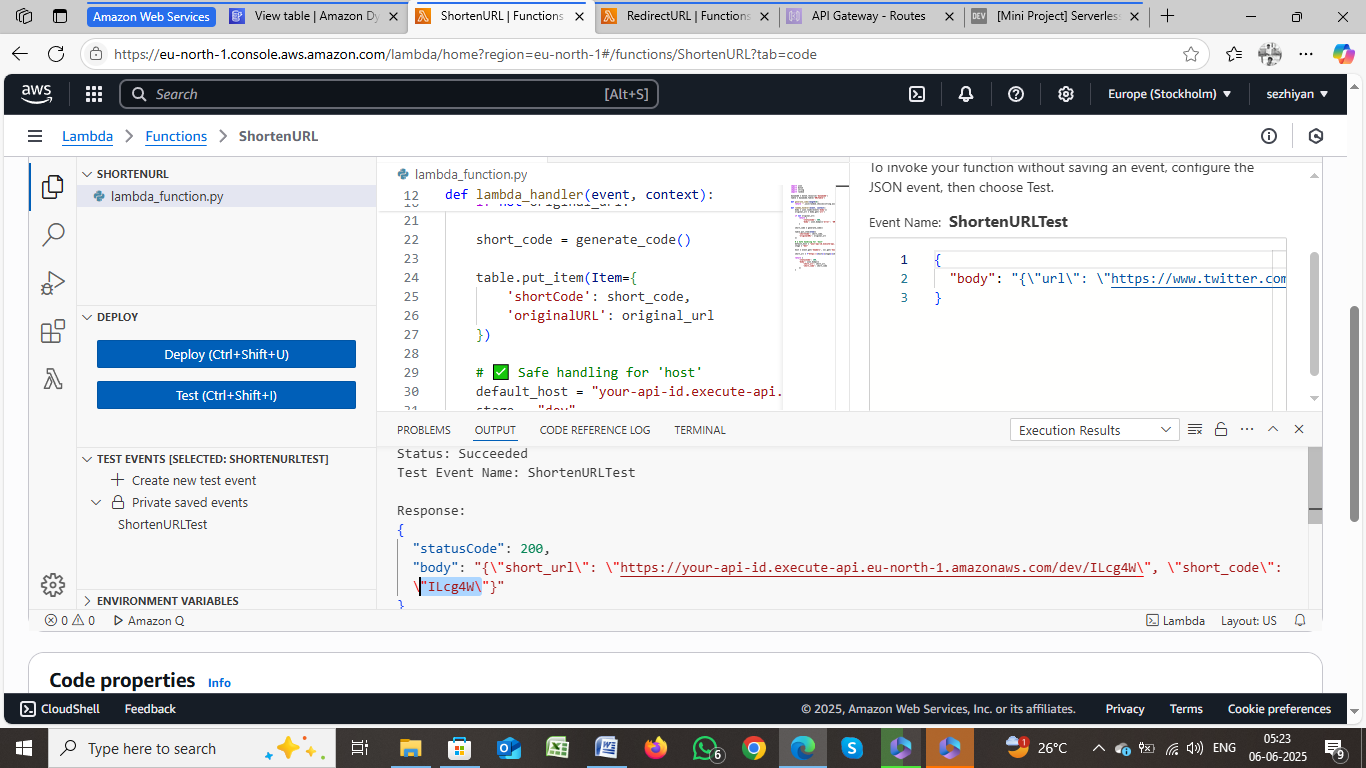
* + CLICK TEST AND EDIT THE EVENT AND SAVE
  + ADD JSON CODE

{

  "body": "{\"url\": \"https://www.twitter.com\"}"

}

* + CLICK TEST AND RECEIVE THE SHORT URL



GO TO A LAMBDA FUNCTION IN **RedirectURL**

* + CLICK TEST AND EDIT THE EVENT AND SAVE
  + ADD JSON CODE

{

  "resource": "/{code}",

  "path": "/abc123",

  "httpMethod": "GET",

  "pathParameters": {

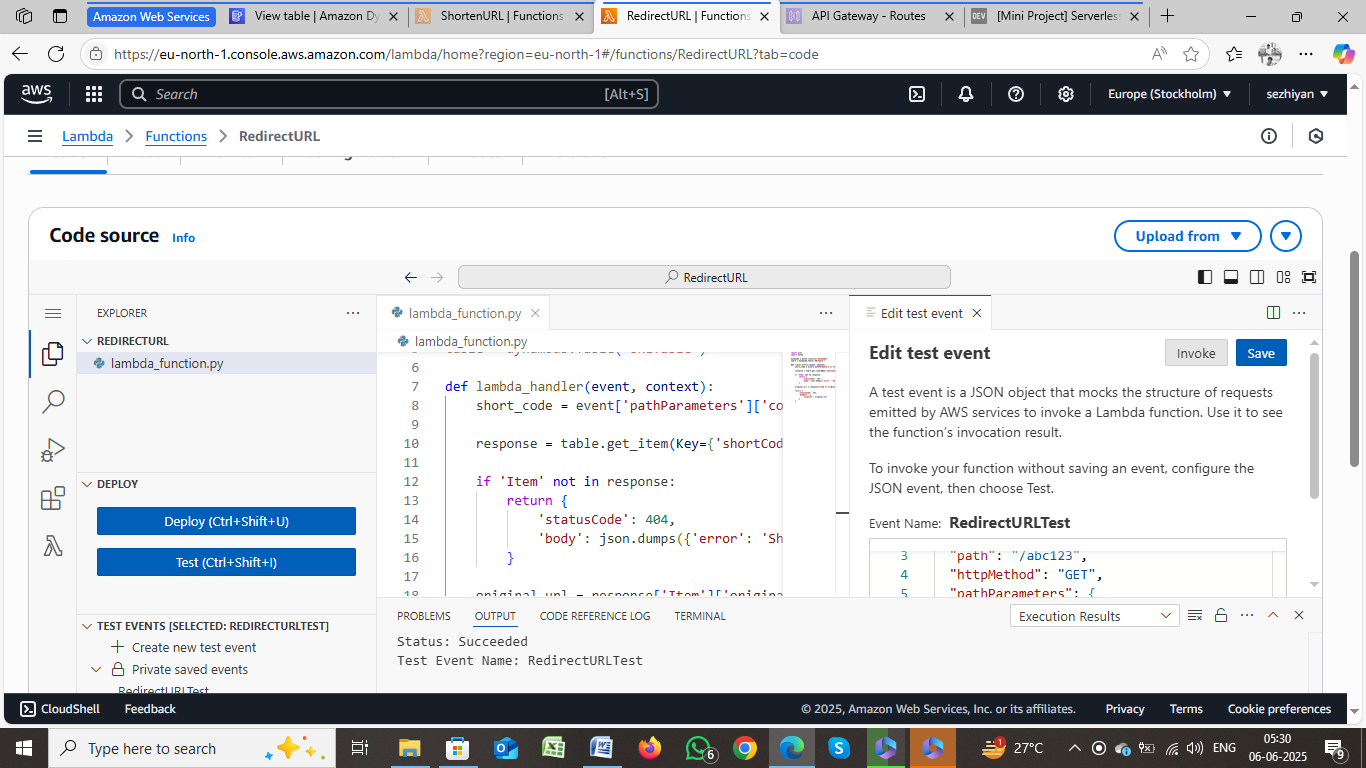
    "code": "abc123"

  },

  "isBase64Encoded": false

}

* + CLICK TEST



* + OPEN THE POSTMAN AND SELECT THE POST AND PAST THE POST URL AND SELECT BODY-RAW-JSON FORMAT AND PASTE THE LONG URL
  + CLICK SEND

long url ---{“url” :<https://www.youtube.com>} Short code---vS06E1