ELEVATE LABS INTERN

Abinaya K – abinaya26kannan@gmail.com

Date: 24.10.2025

Task 4: AWS Lambda Function

Objective:

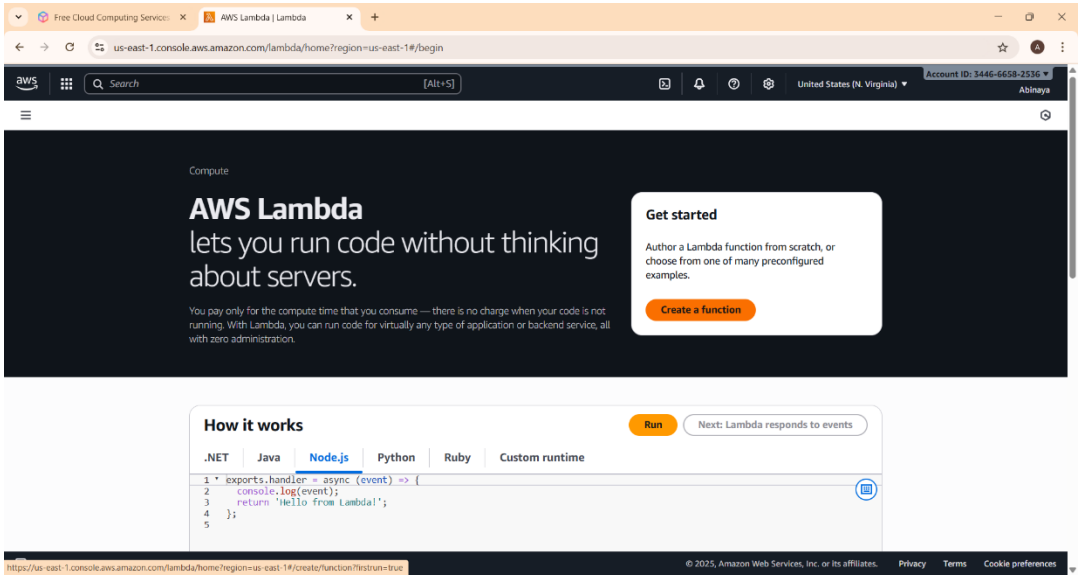
We are going to create a small cloud function using AWS Lambda that runs automatically when someone calls it (via a web URL). You don't have to manage any servers—AWS handles infrastructure, scaling, and billing. When the function runs, it will return a simple message, like "Hello from my first cloud function!"

In short: write code → deploy to AWS Lambda → trigger via HTTP → get a response.

Steps Performed

1. **Sign in to AWS Console**

- Open https://console.aws.amazon.com and sign in.
- In Services search, type Lambda → open AWS Lambda.

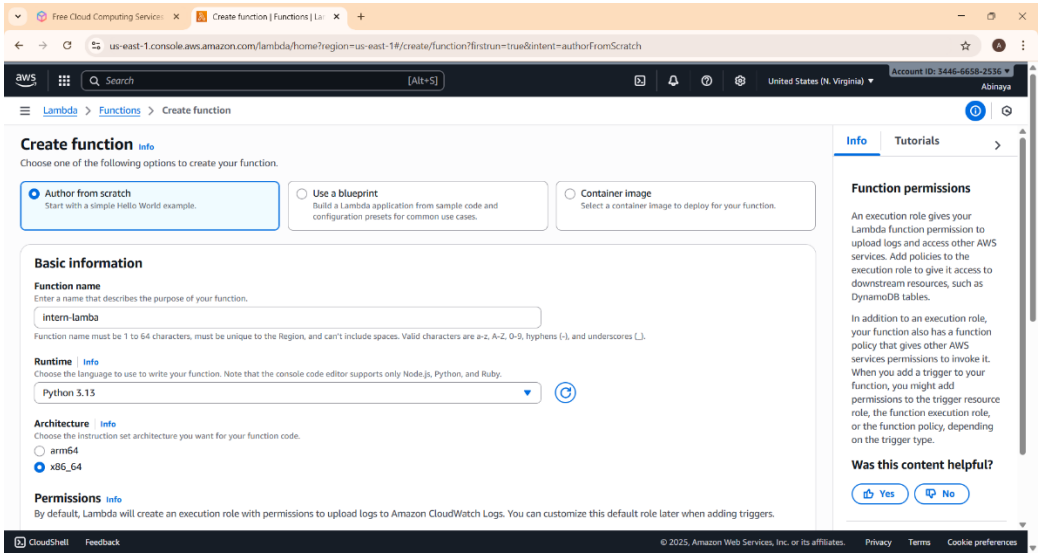


2. **Create the Lambda function**

- Click Create function → Author from scratch.

- Name: intern-lambda
- Runtime: Python 3.11
- Permissions: Create a new role from AWS policy templates → lambda-basic execution (gives CloudWatch Logs).

Click Create function → wait for the function page to load.



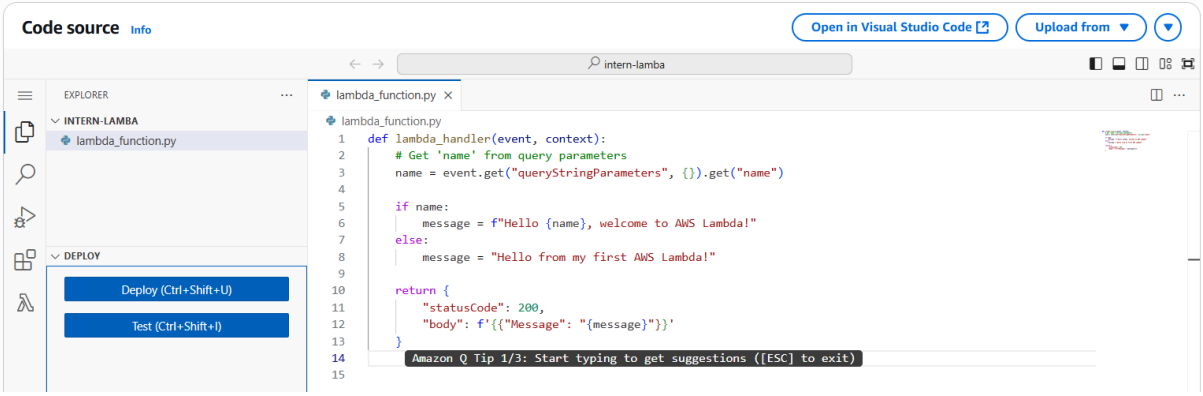
3. **Paste the Function Code**

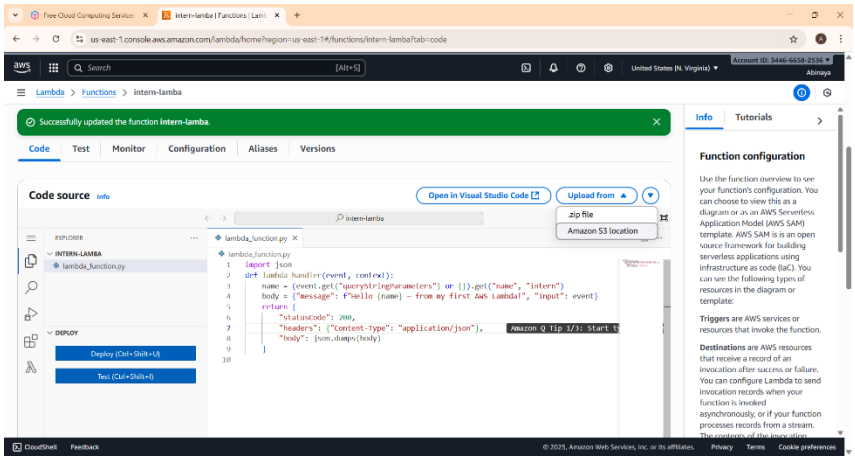
   Python (main.py)

```
import json

def lambda_handler(event, context):
    name = (event.get("queryStringParameters") or {}).get("name", "intern")
    body = {"message": f"Hello {name} — from my first AWS Lambda!", "input": event}
    return {
        "statusCode": 200,
        "headers": {"Content-Type": "application/json"},
        "body": json.dumps(body)
    }
```

   Paste code in the **inline editor → Deploy**.



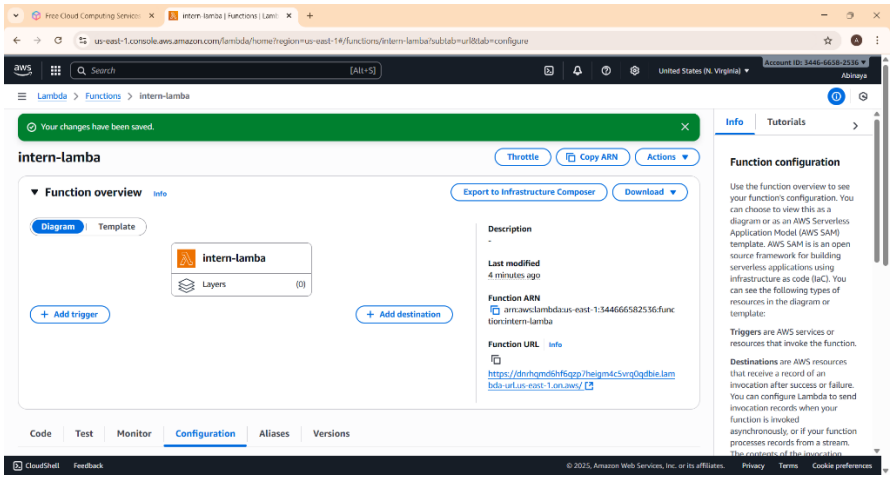


Create a Function URL (public HTTP endpoint)

- On the Lambda function page, open **C**onfiguration tab (or Configuration → Function URL).
- Click Create function URL.

    a. Auth Type: choose NONE for simple public test (no auth). For secure access choose `AWS_IAM`.
    b. CORS: enable if you want to call from browser frontends.

    - Click Create. You'll get a URL like `https://<id>.lambda-url.<region>.on.aws`. Copy this URL.

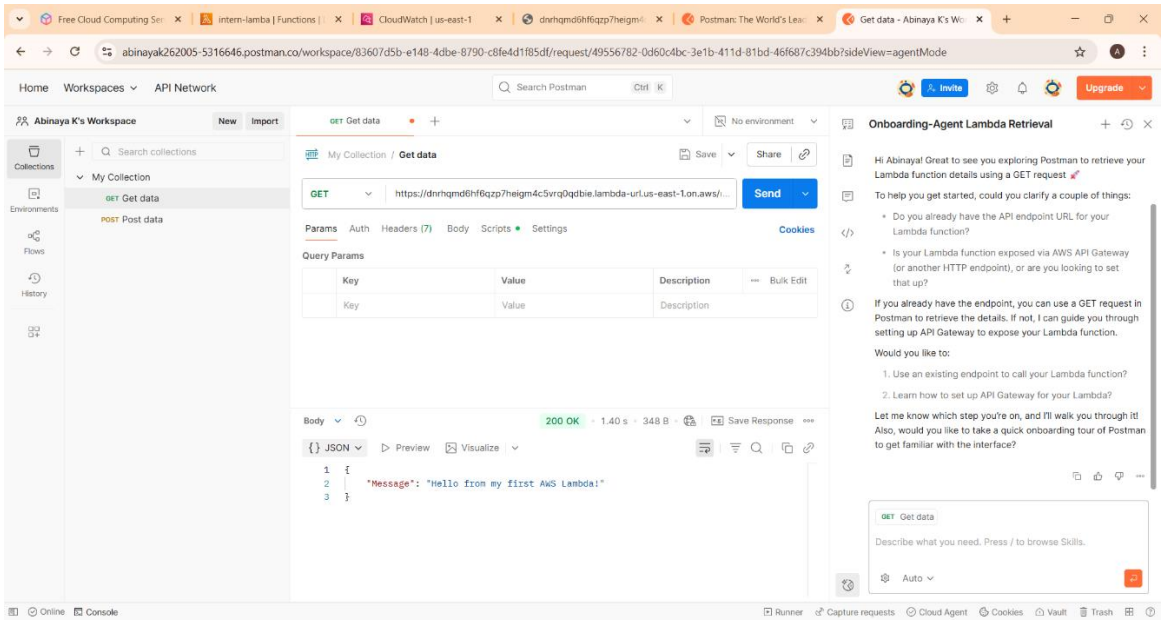


Test the function (browser / curl / Postman)

- **Browser**: paste the Function URL into the address bar → add ?name=Abinaya to test query params. You should see JSON response.

ii) **Using Postman**

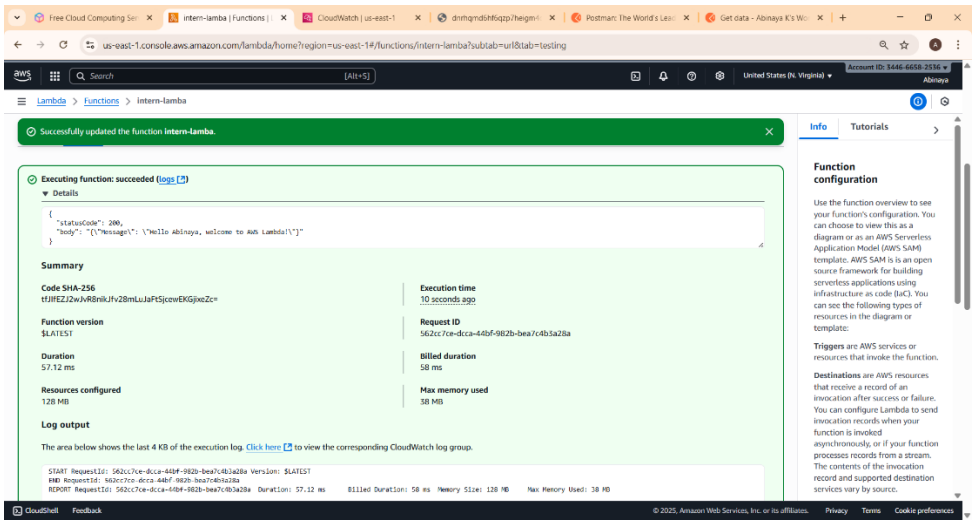
1. Open Postman (free app).
2. Choose GET request.
3. Paste your Function URL into the request bar.
4. Click Send.
5. You'll see the same JSON response in the output window.



iii) Using the AWS Console's built-in Test

1. Open your Lambda function in the AWS Console.
2. At the top, click Test → Configure test event.
3. Choose Create new test event → name it "demoEvent".
4. Leave the default JSON, or replace with:

{ "queryStringParameters": { "name": "Abinaya" } }



Conclusion:

we successfully created and deployed a serverless AWS Lambda function that executes automatically in response to HTTP requests. Through this task, we learned how serverless computing (FaaS) allows code to run without managing servers, enabling event-driven execution, efficient resource usage, and cost-effective deployment.