Lab 11: Serverless App with AWS Lambda & API Gateway

Objective

Build a serverless application using AWS Lambda and expose it via API Gateway. Students will create a Lambda function, set up an API trigger, test the endpoint, and implement logging.

Theory

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. You simply upload your function code, and Lambda takes care of everything required to run and scale the function.

Amazon API Gateway is a fully managed service that allows developers to create, publish, maintain, monitor, and secure REST, HTTP, and WebSocket APIs at any scale. It serves as a front door to access data, business logic, or functionality from your backend services such as Lambda functions.

Together, Lambda and API Gateway form the backbone of a serverless architecture — where compute resources are allocated dynamically and automatically based on requests. This allows for highly scalable and cost-efficient applications.

Prerequisites

- AWS Free Tier Account
- IAM role with the following permissions:
  - AWSLambdaBasicExecutionRole
  - AmazonAPIGatewayInvokeFullAccess
  - CloudWatchLogsFullAccess

Setup

Use the AWS Console or AWS CloudShell to create the Lambda function and API Gateway resources. All resources will be created in a selected AWS region.

Lab Steps

Step 1: Write a Lambda Function

Create a Lambda function using Python 3.10 runtime.
Here is the sample function that returns a personalized HTML response:

```
def lambda_handler(event, context):

  name = "Guest"

  if event.get("queryStringParameters") and "name" in
event["queryStringParameters"]:

    name = event["queryStringParameters"]["name"]

  html_response = f"""

  <html>

    <head><title>Lambda Response</title></head>

    <body style='font-family: Arial; background-color: #f4f4f4;
padding: 20px;'>

      <h1 style='color: #333;'>Hello, {name}!</h1>

      <p>This response is served from an AWS Lambda function
using API Gateway.</p>

    </body>

  </html>
```

```
    """

    return {
        'statusCode': 200,
        'headers': {
            'Content-Type': 'text/html'  # Important!
        },
        'body': html_response
    }
```

Step 2: Create an API Gateway Trigger

- Go to API Gateway in AWS Console
- Choose “Create API” > HTTP API or REST API
- Add a new route (e.g. /greet) and attach the Lambda function as integration target
- Deploy the API and note the invoke URL

Step 3: Test the Endpoint

Use a browser or terminal to test the endpoint:
- Default: https://<your-api-id>.execute-api.<region>.amazonaws.com/greet
- With query: https://<your-api-id>.execute-api.<region>.amazonaws.com/greet?name=Alice

Step 4: View Logs in CloudWatch

- Open CloudWatch > Logs > Log groups > /aws/lambda/<function-name>
- Check for logs from the test events or actual calls from API Gateway

Student Assignment

1. Modify the Lambda function to accept and validate query parameters like “email”.
2. Return a personalized message including the user input.
3. Log the input parameters to CloudWatch using print().
4. Test and redeploy the function and validate using curl or browser.

Cleanup (Optional)

Delete the Lambda function and API Gateway resources to avoid unnecessary charges.

Summary

This lab demonstrated how to use AWS Lambda and API Gateway to build a serverless function. You created and deployed a function, linked it with an HTTP endpoint, validated input, and reviewed logs in CloudWatch.