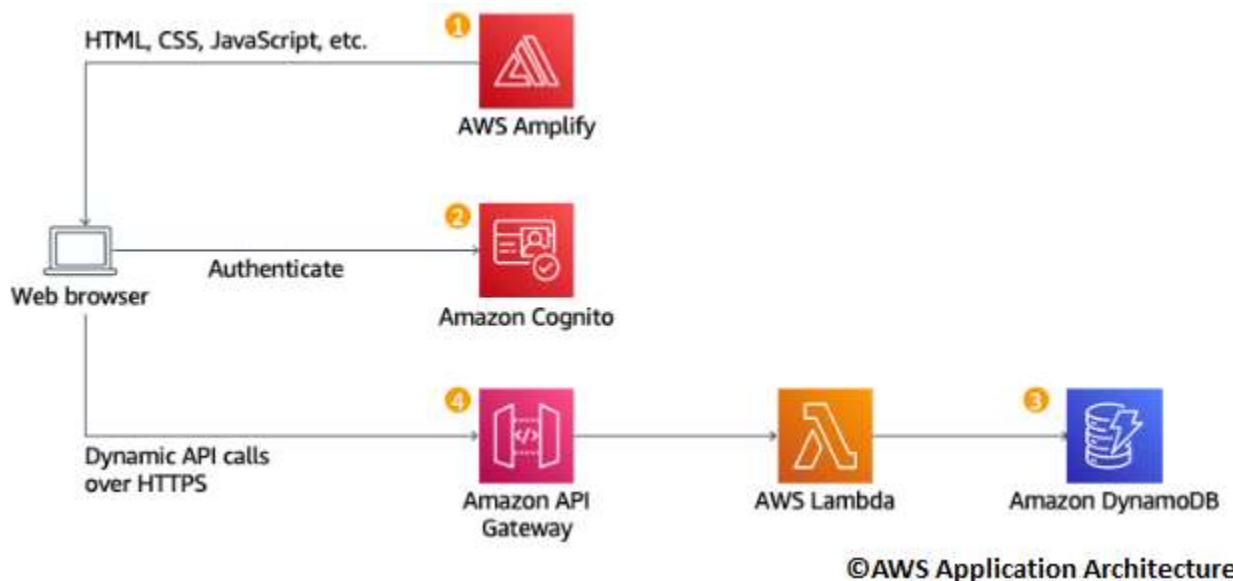


What is AWS Lambda?

It is a serverless compute service provided by Amazon Web Services (AWS). It allows you to run your code without provisioning or managing servers. With AWS Lambda, you can focus on writing your application code and let AWS handle the underlying infrastructure.

Creating a Serverless Website using AWS Lambda!

This is AWS Application Architecture:



1. Create a Lambda Function

- Provide Function name, select the appropriate Runtime & Architecture for your function. Leave the remaining default settings and then create a function.

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ Author from scratch
Start with a simple Hello World example.

Basic information

Function name
Enter a name that describes the purpose of your function.

myTestFunction

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node

Python 3.10

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

- Once the function is created, this would be the default JSON code in the lambda function:

Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl-P)

Environment

myTestFunction - /
lambda_function.py

1 import json
2
3 def lambda_handler(event, context):
4 # TODO implement
5 return {
6 'statusCode': 200,
7 'body': json.dumps('Hello from Lambda!')
8 }
9


- The above JSON code will basically not display much except “Hello from Lambda!”.

Add Trigger

- Click on Add Trigger: For the trigger configuration, select API Gateway.
- Select Create a new API and HTTP API for API type
- Select Open for security.
- Keep all the defaults.

Add trigger

Trigger configuration [Info](#)

 **API Gateway**
api application-services aws serverless

Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. [Learn more](#)

Intent
Use an existing api or have us create one for you.
☒ **Create a new API**
☐ Use existing API

API type

☒ **HTTP API**
Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

☐ **REST API**
Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Security
Configure the security mechanism for your API endpoint.

Open

► Additional settings

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel

Add

Up on clicking Add, the successful message will be displayed:

myTestFunction



The trigger myTestFunction-API was successfully added to function myTestFunction. The function is now

Note – if you copy and paste the below **API endpoint URL** into a web browser and you will see the **"Hello from Lambda!"** text displayed.

Triggers (1) [Info](#)



Trigger



API Gateway: myTestFunction-API

arn:aws:execute-api:us-east-1:361030023395:h4aucy9xj7/*/*/myTestFunction



API endpoint: <https://h4aucy9xj7.execute-api.us-east-1.amazonaws.com/default/myTestFunction>

► Details



<https://h4aucy9xj7.execute-api.us-east-1.amazonaws.com/default/myTestFunction>



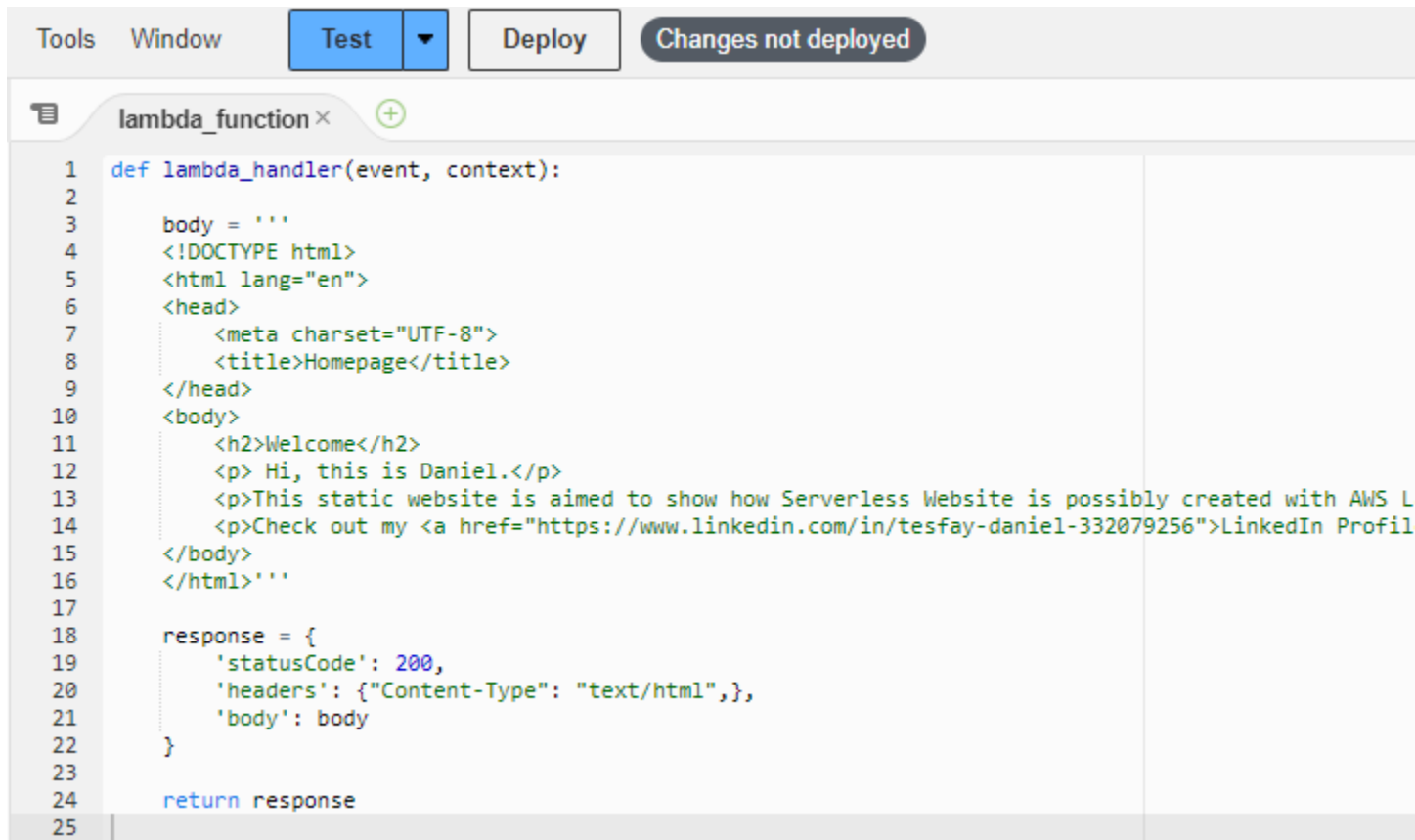
"Hello from Lambda!"



2. Edit Lambda Function with HTML Code

Edit the lambda function with the sample HTML code below. This would replace the JSON string information with new HTML displayed.

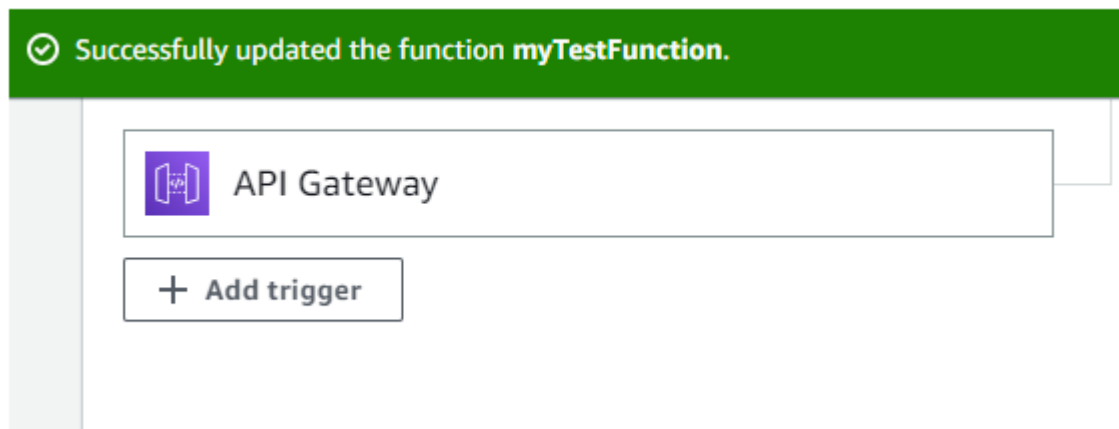
- Go to the Function (**myTestFunction**)



The screenshot shows the AWS Lambda console's code editor interface. At the top, there are tabs for 'Tools' and 'Window', and buttons for 'Test' (highlighted in blue), 'Deploy', and a status indicator 'Changes not deployed'. Below the tabs, the editor window is titled 'lambda_function' and contains the following Python code:

```
1 def lambda_handler(event, context):
2
3     body = '''
4     <!DOCTYPE html>
5     <html lang="en">
6     <head>
7         <meta charset="UTF-8">
8         <title>Homepage</title>
9     </head>
10    <body>
11        <h2>Welcome</h2>
12        <p>Hi, this is Daniel.</p>
13        <p>This static website is aimed to show how Serverless Website is possibly created with AWS L
14        <p>Check out my <a href="https://www.linkedin.com/in/tesfay-daniel-332079256">LinkedIn Profil
15    </body>
16    </html>'''
17
18    response = {
19        'statusCode': 200,
20        'headers': {"Content-Type": "text/html",},
21        'body': body
22    }
23
24    return response
25
```

- Once editing complete, Deploy the code, then successful notification will be shown at the top of your Amazon Console like shown below:



Navigate to the same browser with the JSON string "**Hello from Lambda!**" displayed, refresh the page and you will see the new HTML information displayed as coded above.

