BUILD A BASIC WEB APPLICATION-Using Lambda,Api gate way,Dynamo Db,Amplify

##### [INTRODUCTION](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/)

##### [CREATE WEB APP](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-one/)

##### [BUILD SERVERLESS FUNCTION](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-two/)

##### [LINK SERVERLESS FUNCTION TO WEB APP](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-three/)

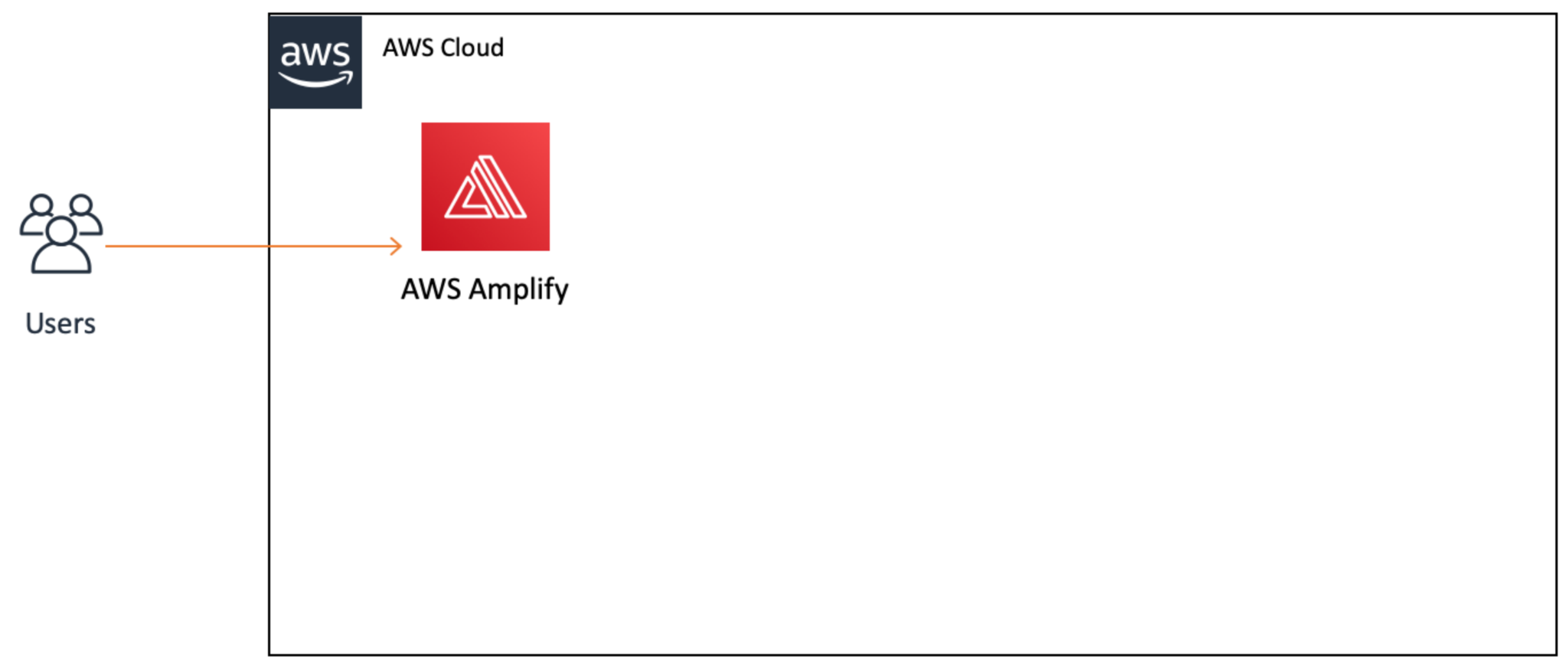
##### [CREATE DATA TABLE](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-four/)

##### Click here[ADD INTERACTIVITY TO WEB APP](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-five/)

##### Click here[CLEAN UP RESOURCES](https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-six/)

##### 

## Module 1: Create a Web App



Step-1 Create web app application

HTML-CODE

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hello World</title>

</head>

<body>

Hello World

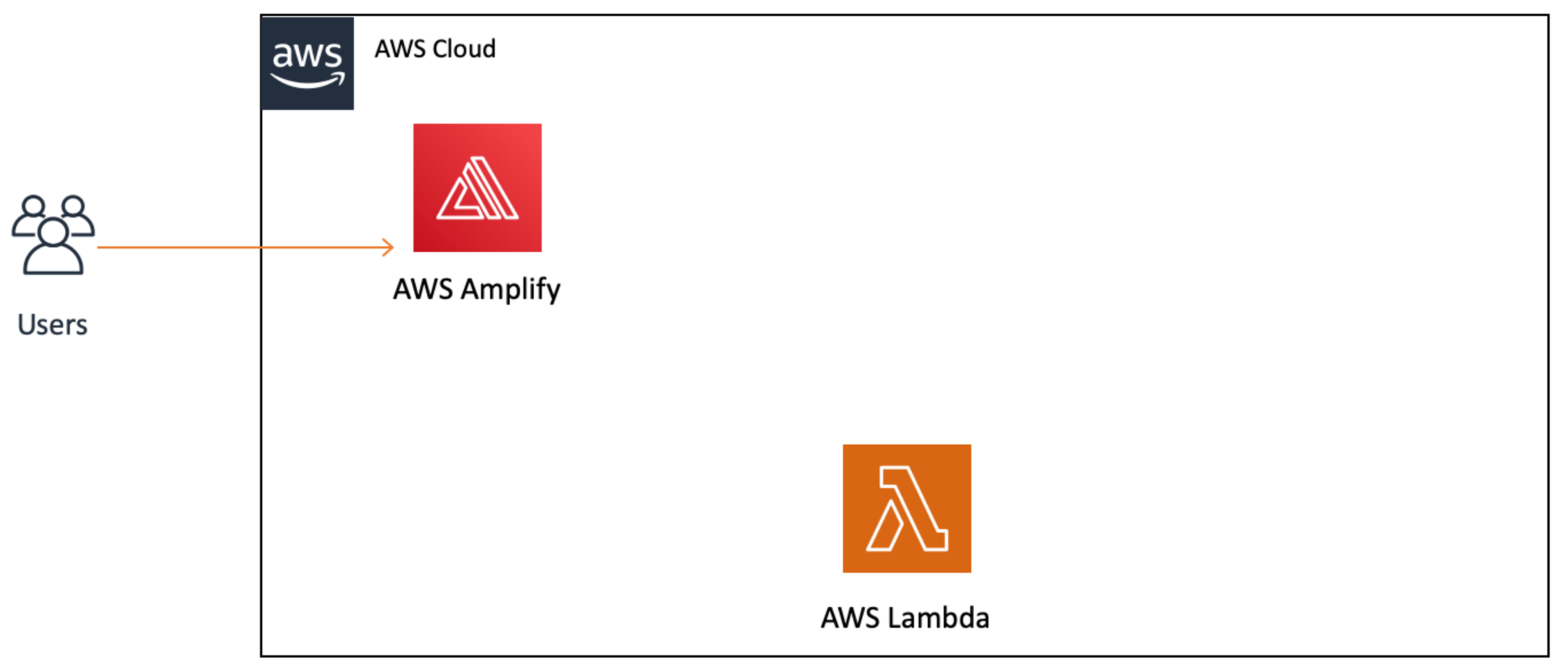
</body>

</html>

<https://us-west-2.console.aws.amazon.com/amplify/home>

Step-2– Test you web app

## Module 2: Build a Serverless Function



Step 3-Create and configure your lambda function

<https://console.aws.amazon.com/lambda/>

Python code-

# import the JSON utility package since we will be working with a JSON object

import json

# define the handler function that the Lambda service will use as an entry point

def lambda\_handler(event, context):

# extract values from the event object we got from the Lambda service

name = event['firstName'] +' '+ event['lastName']

# return a properly formatted JSON object

return {

'statusCode': 200,

'body': json.dumps('Hello from Lambda, ' + name)

}

JSON code for test–

{

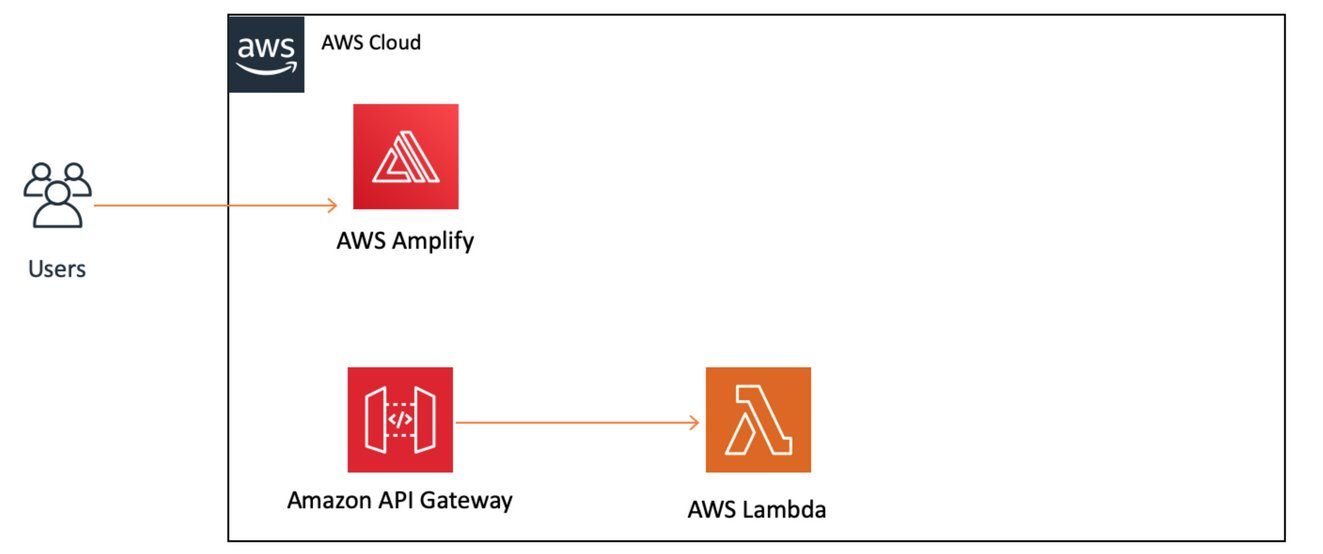
"firstName": "Ada",

"lastName": "Lovelace"

}

Step 4–Test your lambda function

## Module 3: Link a Serverless Function to a Web App



Step 5- create new rest api

Step 6 -create new resource method

Step 7- Deploy API

Step 8-Validate API

Java script code to validate api

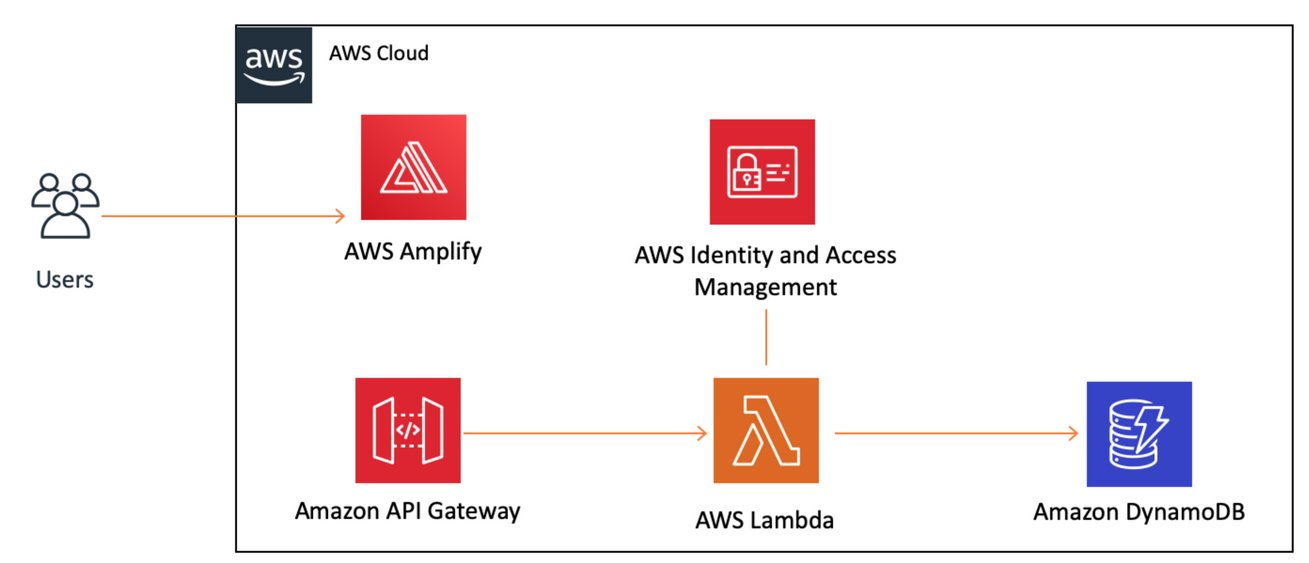
{

"firstName":"Grace",

"lastName":"Hopper"

}

## Module 4: Create a Data Table



Step-9 Create a dynamo db table

<https://console.aws.amazon.com/dynamodb/home>

Step-10 create and add IAM policy to Lambda function

<https://console.aws.amazon.com/lambda/home>

JSON code for IAM policy—

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"dynamodb:PutItem",

"dynamodb:DeleteItem",

"dynamodb:GetItem",

"dynamodb:Scan",

"dynamodb:Query",

"dynamodb:UpdateItem"

],

"Resource": "YOUR-TABLE-ARN"

}

]

}

Step 11- Modify Lambda function to write dynamo db table

Python code-

# import the json utility package since we will be working with a JSON object

import json

# import the AWS SDK (for Python the package name is boto3)

import boto3

# import time

import time

# import two packages to help us with dates and date formatting

# create a DynamoDB object using the AWS SDK

dynamodb = boto3.resource('dynamodb')

# use the DynamoDB object to select our table

table = dynamodb.Table('HelloWorldDatabase')

# define the handler function that the Lambda service will use as an entry point

def lambda\_handler(event, context):

# Get the current GMT time

gmt\_time = time.gmtime()

# store the current time in a human readable format in a variable

# Format the GMT time string

now = time.strftime('%a, %d %b %Y %H:%M:%S +0000', gmt\_time)

# extract values from the event object we got from the Lambda service and store in a variable

name = event['firstName'] +' '+ event['lastName']

# write name and time to the DynamoDB table using the object we instantiated and save response in a variable

response = table.put\_item(

Item={

'ID': name,

'LatestGreetingTime':now

})

# return a properly formatted JSON object

return {

'statusCode': 200,

'body': json.dumps('Hello from Lambda, ' + name)

}

Step 12- Test Changes

<https://console.aws.amazon.com/dynamodb/home>

## Module 5: Add Interactivity to Your Web App

Step 13 - Update web app with amplify console

Replace index.html code–

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hello World</title>

<!-- Add some CSS to change client UI -->

<style>

body {

background-color: #232F3E;

}

label, button {

color: #FF9900;

font-family: Arial, Helvetica, sans-serif;

font-size: 20px;

margin-left: 40px;

}

input {

color: #232F3E;

font-family: Arial, Helvetica, sans-serif;

font-size: 20px;

margin-left: 20px;

}

</style>

<script>

// define the callAPI function that takes a first name and last name as parameters

var callAPI = (firstName,lastName)=>{

// instantiate a headers object

var myHeaders = new Headers();

// add content type header to object

myHeaders.append("Content-Type", "application/json");

// using built in JSON utility package turn object to string and store in a variable

var raw = JSON.stringify({"firstName":firstName,"lastName":lastName});

// create a JSON object with parameters for API call and store in a variable

var requestOptions = {

method: 'POST',

headers: myHeaders,

body: raw,

redirect: 'follow'

};

// make API call with parameters and use promises to get response

fetch("YOUR-API-INVOKE-URL", requestOptions)

.then(response => response.text())

.then(result => alert(JSON.parse(result).body))

.catch(error => console.log('error', error));

}

</script>

</head>

<body>

<form>

<label>First Name :</label>

<input type="text" id="fName">

<label>Last Name :</label>

<input type="text" id="lName">

<!-- set button onClick method to call function we defined passing input values as parameters -->

<button type="button" onclick="callAPI(document.getElementById('fName').value,document.getElementById('lName').value)">Call API</button>

</form>

</body>

</html>

Step 14- Test updated app

## Module 6: Clean Up Resources

1. Go to the Dynamo DB console and select the [HelloWorldDatabase](https://us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#table?name=HelloWorldDatabase) table and choose the Delete button.

2. Go to the API Gateway console to select the [HelloWorldAPI](https://us-west-2.console.aws.amazon.com/apigateway/home?region=us-west-2#/apis/0tyvkp46w1/resources) and choose Delete from the Actions dropdown menu.

3. Go to the Lambda console. Select [HelloWorldFunction](https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/HelloWorldFunction) and then select Delete from the Actions dropdown menu.

4. Go to the Amplify console to select GettingStarted to delete the app. Select Delete app from the Actions dropdown menu.