

## AWS LAMBDA

AWS Lambda is an Amazon Web Services server less computing service (AWS). AWS Lambda users write functions, which are self-contained applications written in one of the supported languages and runtimes, and upload them to AWS Lambda, which then executes them in a fast and versatile manner.

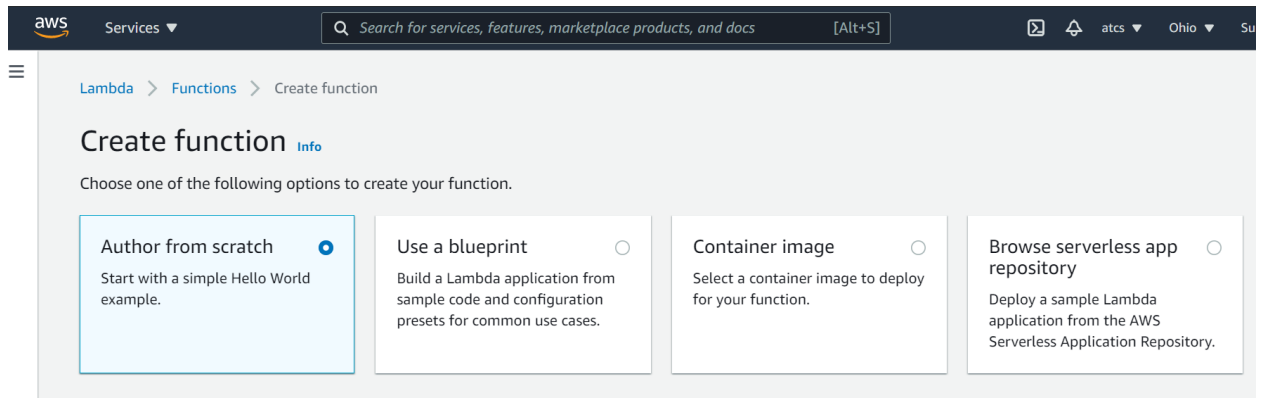
Lambda functions can be used to perform a wide range of tasks, including serving web pages and processing data.

Each Lambda function has its own container to run in. When a function is developed, Lambda bundles it into a new container, which it then runs on an AWS-managed multi-tenant cluster of machines. Each function's container is given the required RAM and CPU power before it begins to run. When the functions are finished, the RAM allocated at the start is multiplied by the length of time the function took to complete. Customers are then compensated depending on the amount of allocated memory and the length of time it took for the function to complete.

Customers may not get much visibility into how the system operates, but they also don't need to worry about updating the underlying machines and infrastructure. This allows AWS users to focus solely on the code and their application.

To create the AWS lambda function for this assignment, we first logged into AWS and go to AWS Lambda function.

### 1. We chose Author from Scratch



2. We chose *847Lambda* as the function name. We chose Python 3.8 and pressed Create Function

**Basic information**

Function name

Enter a name that describes the purpose of your function.

847lambda

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.js, Python, and Ruby.

Python 3.8

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☒ Create a new role with basic Lambda permissions

☐ Use an existing role

☐ Create a new role from AWS policy templates

[i](#) Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

3. It should open code source. Our code is shown below.

**Code source** [Info](#)

Upload from ▼

File Edit Find View Go Tools Window Test Deploy Changes deployed

Go to Anything (Ctrl-P)

Environment

847lambda - /

lambda\_function.py

```
1 import json
2 print('Loading function')
3
4 def lambda_handler(event, context):
5     # TODO implement
6     firstName = event['first']
7     lastName = event['last']
8     return {
9         'statusCode': 200,
10        'body': firstName + ' ' + lastName
11    }
12
```

4. Pressed *Deploy*. It showed *Changes Deployed* in green. Then pressed *Test*.

5. It should open a Configure test window. We wrote *847testLambdaFunction* as event name. Code for the test and input is shown below. Pressed Create

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

Create new test event

Edit saved test events

Event template

hello-world

Event name

847testLamdafunction

1 {

2 "first": "Jane",

3 "last": "Doe"

4 }

Cancel

Format JSON

Create

6. The Response is shown below. The code has run successfully and returns what we want it to return.

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy Changes deployed

Go to Anything (Ctrl-P)

Execution results x lambda\_function x

Environment

847lambda /  
lambda\_function.py

Execution results

Status: Succeeded Max memory used: 51 MB Time: 1.49 ms

Response

{  
 "statusCode": 200,  
 "body": "Jane Doe"  
}

Function Logs

START RequestId: 9258ae72-ab9a-481f-9e1b-1e83794169a0 Version: \$LATEST  
END RequestId: 9258ae72-ab9a-481f-9e1b-1e83794169a0  
REPORT RequestId: 9258ae72-ab9a-481f-9e1b-1e83794169a0 Duration: 1.49 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 51 MB Init Duration: 128.06 ms

Request ID

9258ae72-ab9a-481f-9e1b-1e83794169a0