

## AWS Lambda

**AWS Lambda** is a serverless computing service that lets you run code without managing servers. You write and deploy code, and Lambda handles the *execution, scaling, and infrastructure management*. You only pay for the compute time used.

### Advantages of Lambda

1. Serverless Architecture: No need to manage or provision servers.
2. Cost-Effective: Pay only for the compute time your code runs.
3. Scalability: Automatically scales to handle changes in workload.
4. Event-Driven: Can be triggered by various AWS services or HTTP requests.
5. Flexibility: Supports multiple programming languages.
6. Easy Integration: Seamlessly integrates with other AWS services.
7. Reduced Administrative Overhead: No server maintenance or patching required.

## Lab Steps

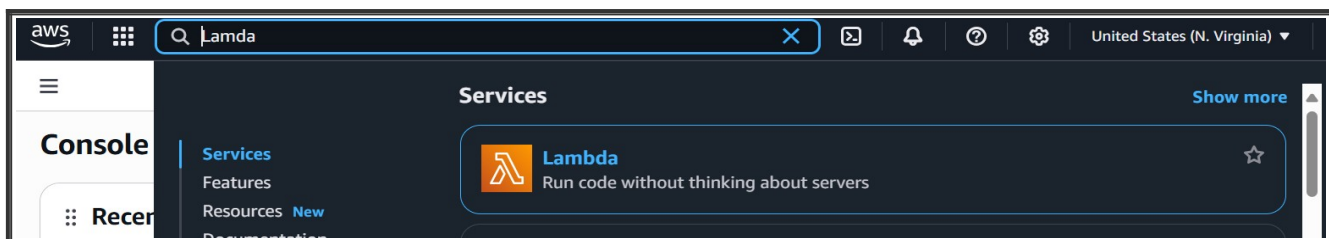
### Task 1 :- Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
  - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
  - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

**Note:-** If you face any issues, please go through [FAQs and Troubleshooting for Labs](#).

### Task 2 :- Create a function

1. Search for Lambda.



2. Click on [Lambda](#).
3. Click on [Create function](#).
4. Choose :- **Author from scratch**
5. Write your function name.  
**Example:** DemoFunction
6. Choose the language to use to write your function in **Runtime**.  
**Example :-** we are going for **Python** here

## Create function [Info](#)

Choose one of the following options to create your function.

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

☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**  
Select a container image to deploy for your function.


---

### Basic information

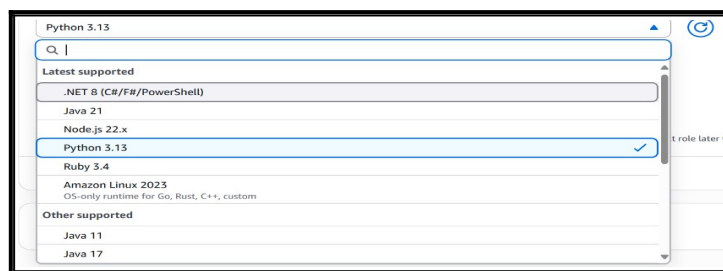
**Function name**  
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

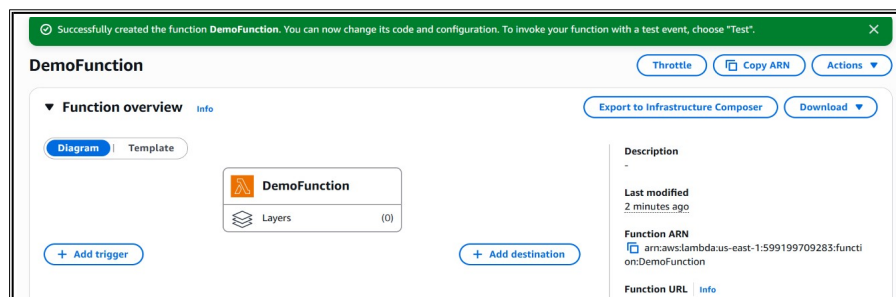
**Runtime** [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.



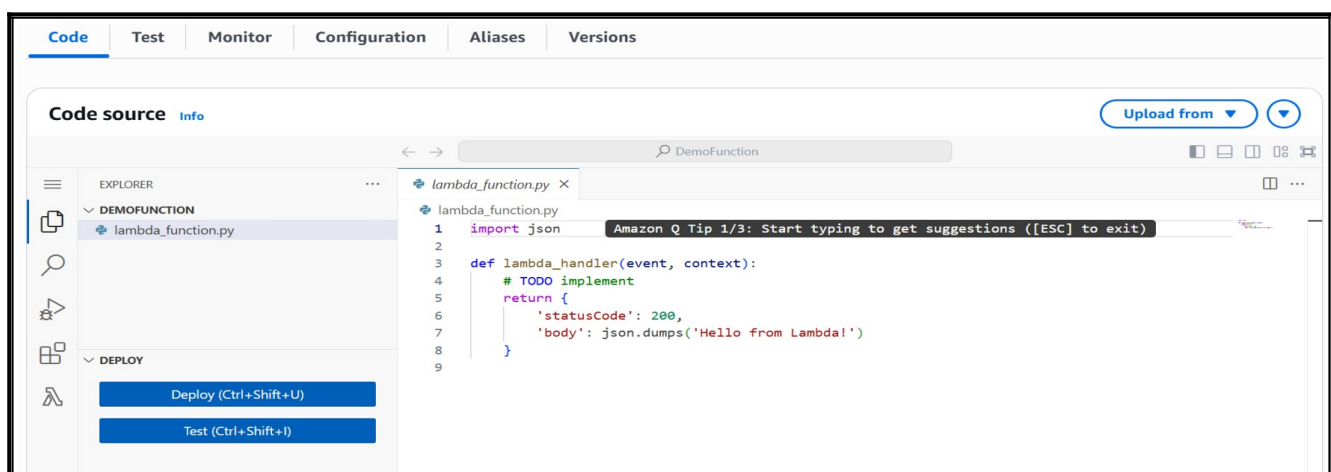
**Note:** There are many language options available here, so you can choose any of them as per your convenience.



- Choose **x84\_64** in **Architecture**
- In **Execution role**:- Create a new role with basic Lambda permissions
- Leave the rest as **default** and click on **create function**.
- Function is successfully created.**



- Go to **code** tab.
- Here the default code will be written which will print **Hello from lambda!**



13. Now go to the **Test** tab to see the output of the written code.

14. Click on **Test**.

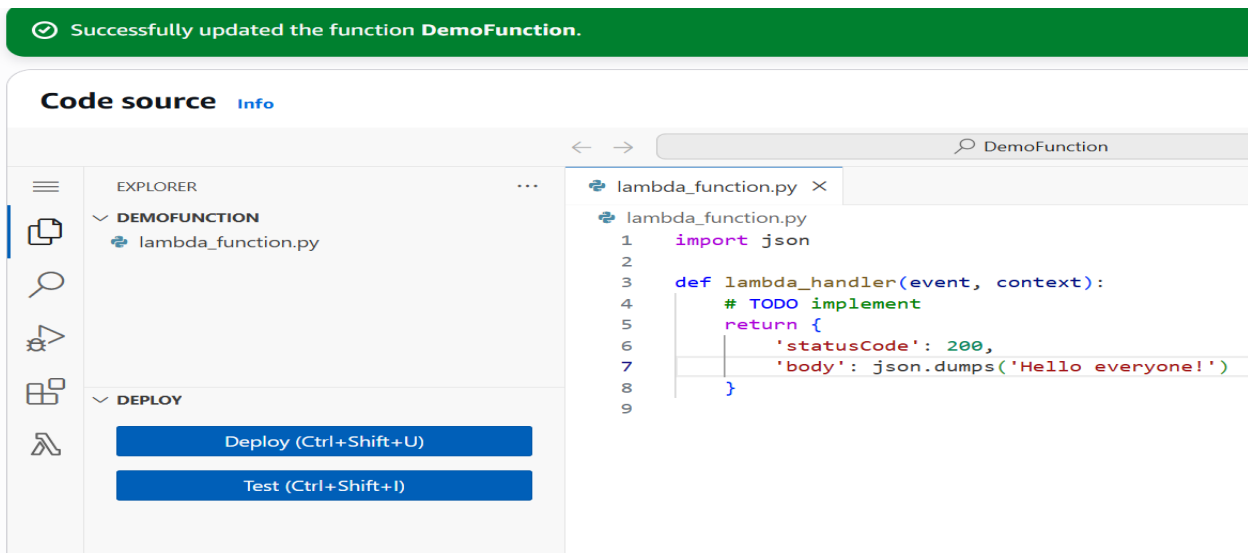


## What if we need to return something other than “Hello from Lambda!” ?

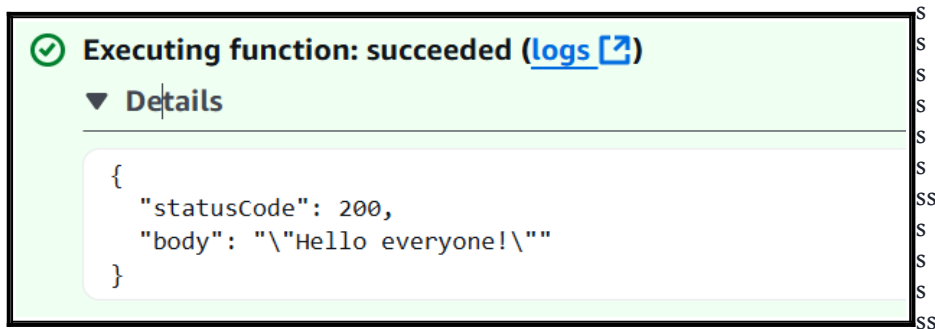
1. Change Hello from Lambda! To Hello everyone! .

2. Doing this will not give any output, we will have to **deploy** it after this.

3. Click on **Deploy**.



4. This will be output :-



5. Now we will modify this code further

```
lambda_function.py X
lambda_function.py
1  import json
2
3  def lambda_handler(event, context):
4      # TODO implement
5      print("Hello Everyone!")
6      return {
7          'statusCode': 200,
8          'body': json.dumps('Hello from Lambda!')
9      }
10
```

- But there will be no output of print.

For that we need to click on [click here](#) option after clicking on **test**.

**Details**

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

**Summary**

<b>Code SHA-256</b> HAPq9EReJVEC5gLavtc/gyd5vZtd9eiUGF932t0jBxY=	<b>Execution time</b> 21 seconds ago
<b>Function version</b> \$LATEST	<b>Request ID</b> 2b44fb20-a13a-48d8-a5bf-63ac4145b76e
<b>Duration</b> 2.19 ms	<b>Billed duration</b> 3 ms
<b>Resources configured</b> 128 MB	<b>Max memory used</b> 35 MB
<b>Init duration</b> 101.89 ms	

**Log output**

The area below shows the last 4 KB of the execution log. [Click here](#) to view the corresponding CloudWatch log group.

6. We will go to **CloudWatch** where we will see our output.

▶	2025-06-11T14:20:10.156Z	INIT_START Runtime Version: python:3.13.v43 Runtime Version ARN: arn:aws:lambda:us-east-1::runt...
▶	2025-06-11T14:20:10.261Z	START RequestId: 2b44fb20-a13a-48d8-a5bf-63ac4145b76e Version: \$LATEST
▶	2025-06-11T14:20:10.262Z	Hello Everyone! ←
▶	2025-06-11T14:20:10.266Z	END RequestId: 2b44fb20-a13a-48d8-a5bf-63ac4145b76e
▶	2025-06-11T14:20:10.267Z	REPORT RequestId: 2b44fb20-a13a-48d8-a5bf-63ac4145b76e Duration: 2.19 ms Billed Duration: 3 ms ...