

# Create a Lambda function on AWS.

1

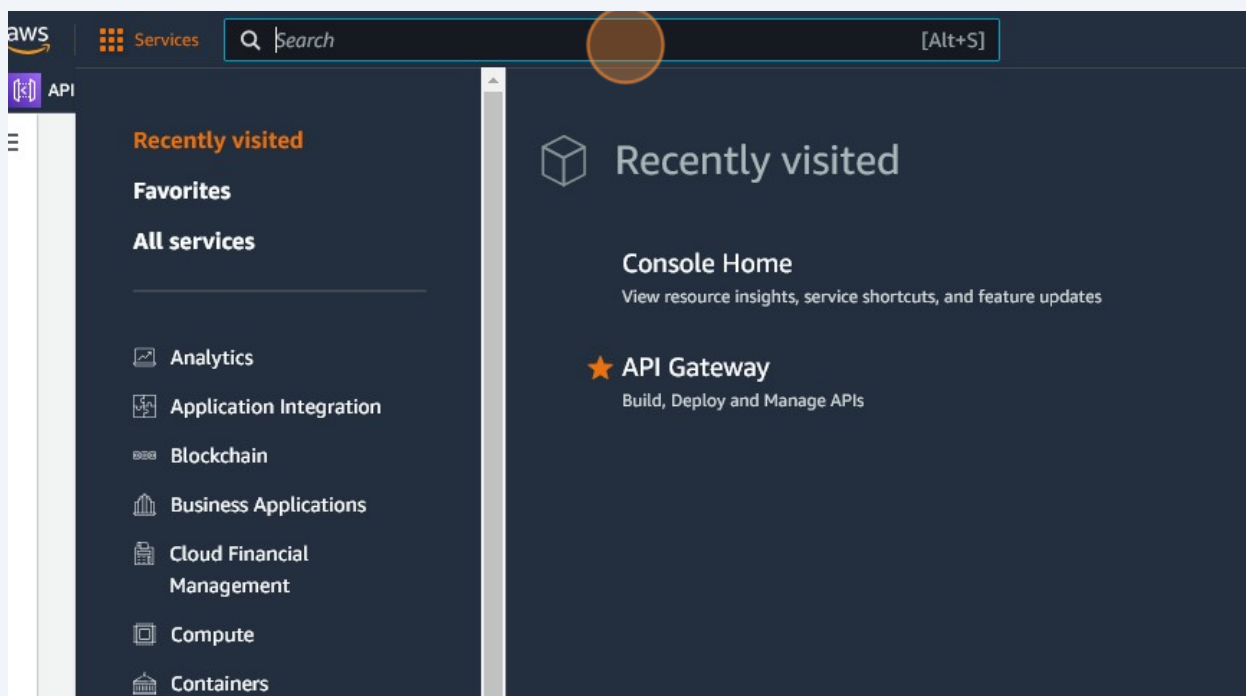
Navigate to <https://ap-southeast-2.console.aws.amazon.com/console/home?region=ap-southeast-2>

2

Type "l"

3

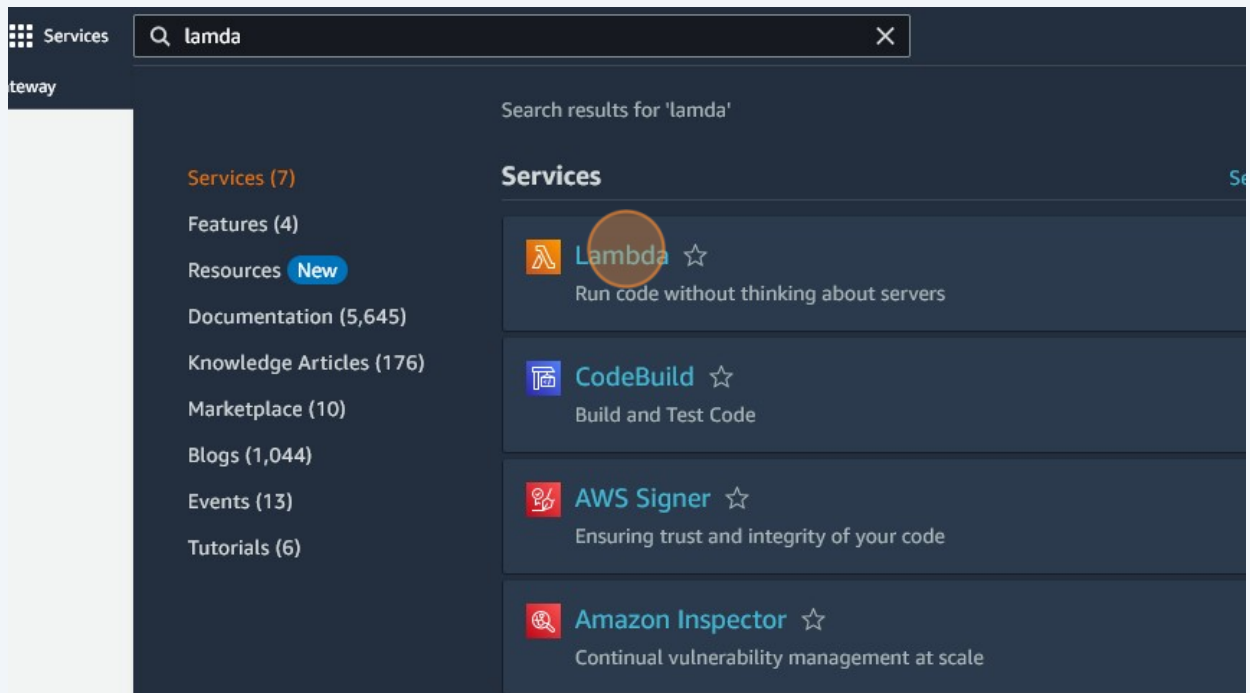
Click the "Search" field.



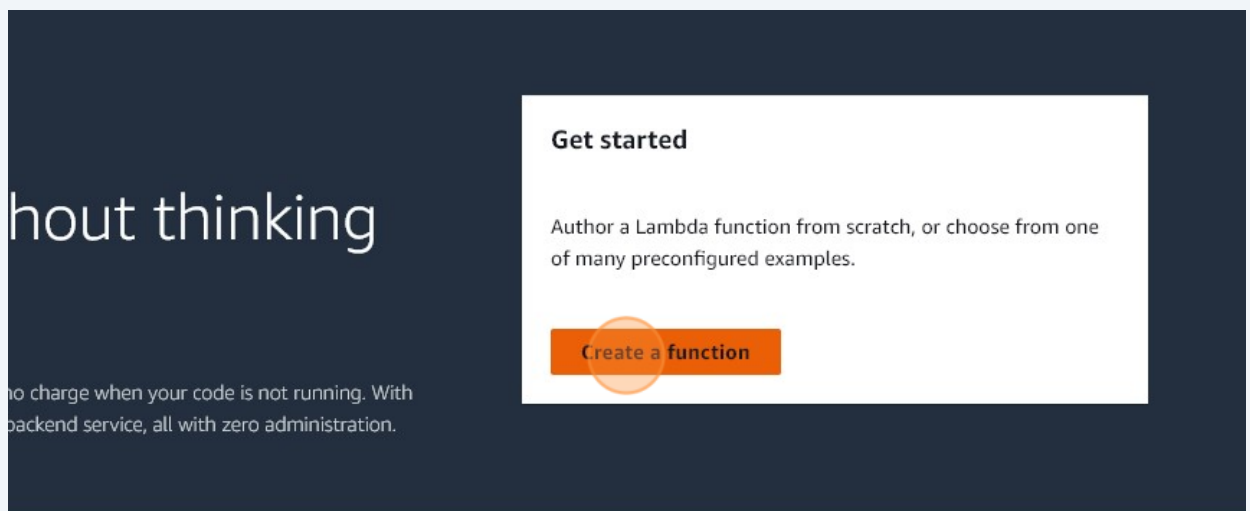
4

Type "lamda"

## 5 Click "Lambda"



## 6 Click "Create a function"



## 7 Click "Author from scratch"

The screenshot shows the AWS Lambda console's 'Create function' page. The breadcrumb navigation is 'Lambda > Functions > Create function'. The page title is 'Create function' with an 'Info' link. Below the title, it says 'Choose one of the following options to create your function.' There are two radio button options: 'Author from scratch' (selected and highlighted with an orange circle) and 'Use a blueprint'. The 'Author from scratch' option has a subtext 'Start with a simple Hello World example.' Below these options is a section titled 'Basic information'. Under 'Function name', there is a text input field containing 'myFunctionName' and a note: 'Enter a name that describes the purpose of your function. Use only letters, numbers, hyphens, or underscores with no spaces.'

## 8 Click the "Function name" field.

This screenshot shows the same 'Create function' page, but with more details visible. The 'Author from scratch' option remains selected. In the 'Basic information' section, the 'Function name' field is highlighted with an orange circle. Below it, the 'Runtime' section is expanded, showing 'Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.' The 'Runtime' dropdown is set to 'Node.js 20.x'. The 'Architecture' section is also expanded, showing 'Choose the instruction set architecture you want for your function code.' The 'Architecture' dropdown is set to 'x86\_64', with 'arm64' also visible as an option.

9 Type "TestingAWSLambdaFunction"

10 Click "Node.js 20.x"

**Basic information**

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

TestingAWSLambdaFunction

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.

Node.js 20.x

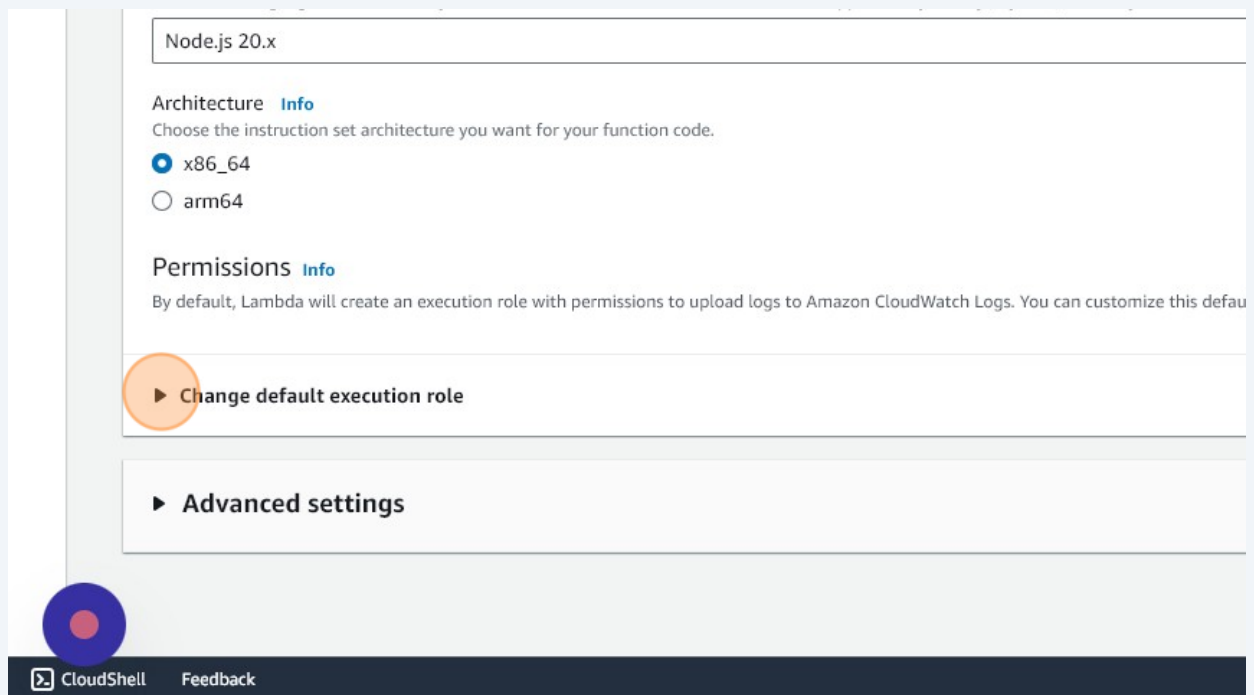
**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.

☒ x86\_64

☐ arm64

**Permissions** [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default

## 11 Click here.



Node.js 20.x

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

☒ x86\_64  
☐ arm64

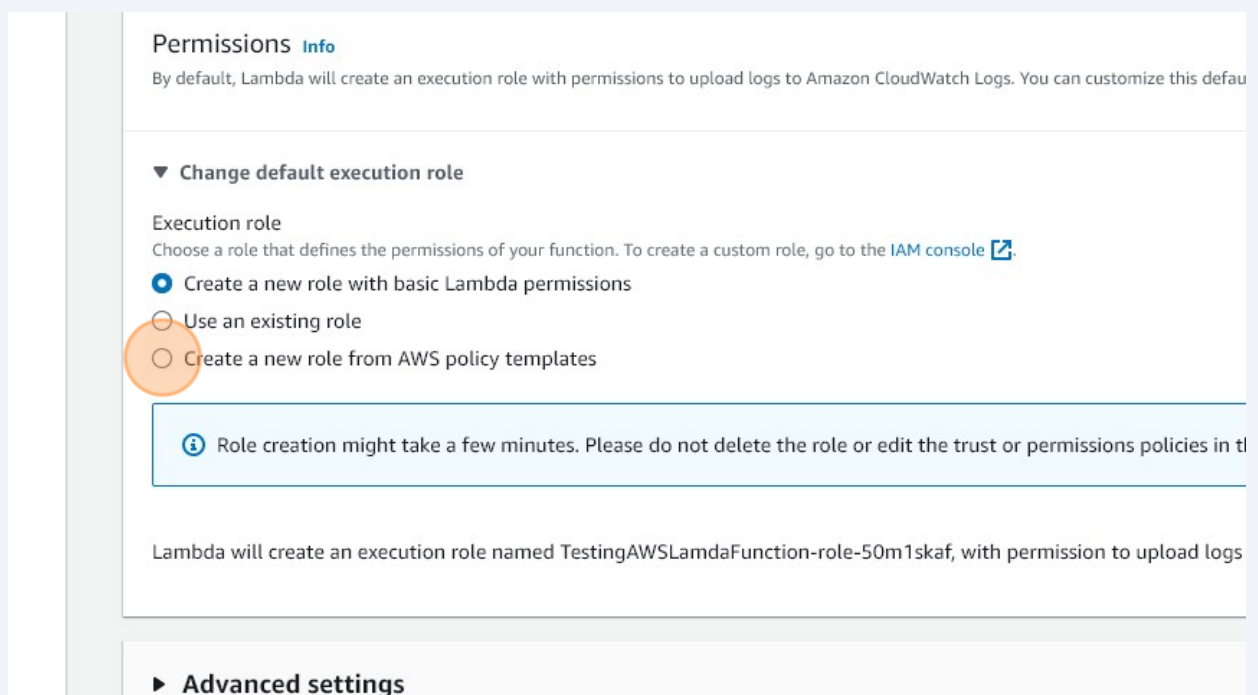
Permissions [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default

► Change default execution role

► Advanced settings

CloudShell Feedback

## 12 Click this radio button.



Permissions [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default

▼ 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 the IAM console.

Lambda will create an execution role named TestingAWSLambdaFunction-role-50m1skaf, with permission to upload logs

► Advanced settings

### 13 Click the "Role name" field.

☐ Create a new role with basic Lambda permissions

☐ Use an existing role

☒ Create a new role from AWS policy templates

**Role name**  
Enter a name for your new role.

myRoleName

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

**Policy templates - optional** [Info](#)  
Choose one or more policy templates.

**Advanced settings**

### 14 Type "AWS\_LAMDA\_FUNCTION\_DEFAULT"

## 15 Click "Role name"

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 the console.

**Role name**

Enter a name for your new role.

AWS\_LAMDA\_FUNCTION\_DEFAULT

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

Policy templates - optional **Info**

Choose one or more policy templates.

## 16 Click here.

**i** Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in the console.

**Role name**

Enter a name for your new role.


AWS\_LAMDA\_FUNCTION\_DEFAULT

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

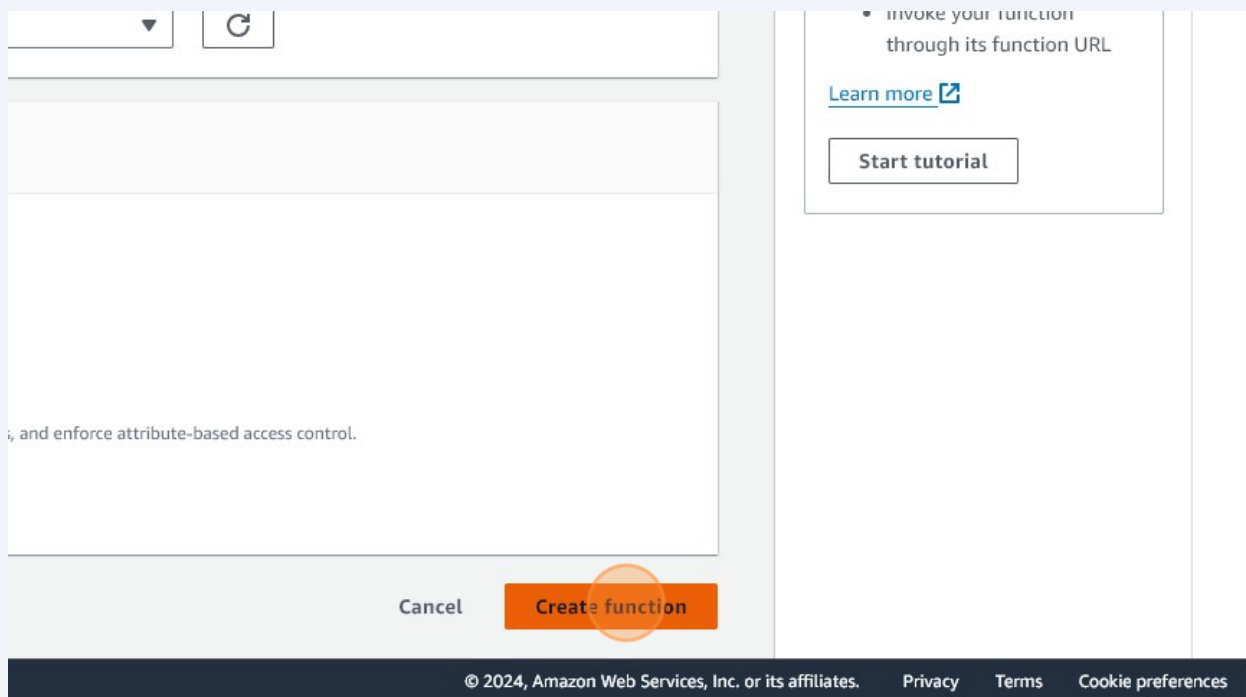
Policy templates - optional **Info**

Choose one or more policy templates.

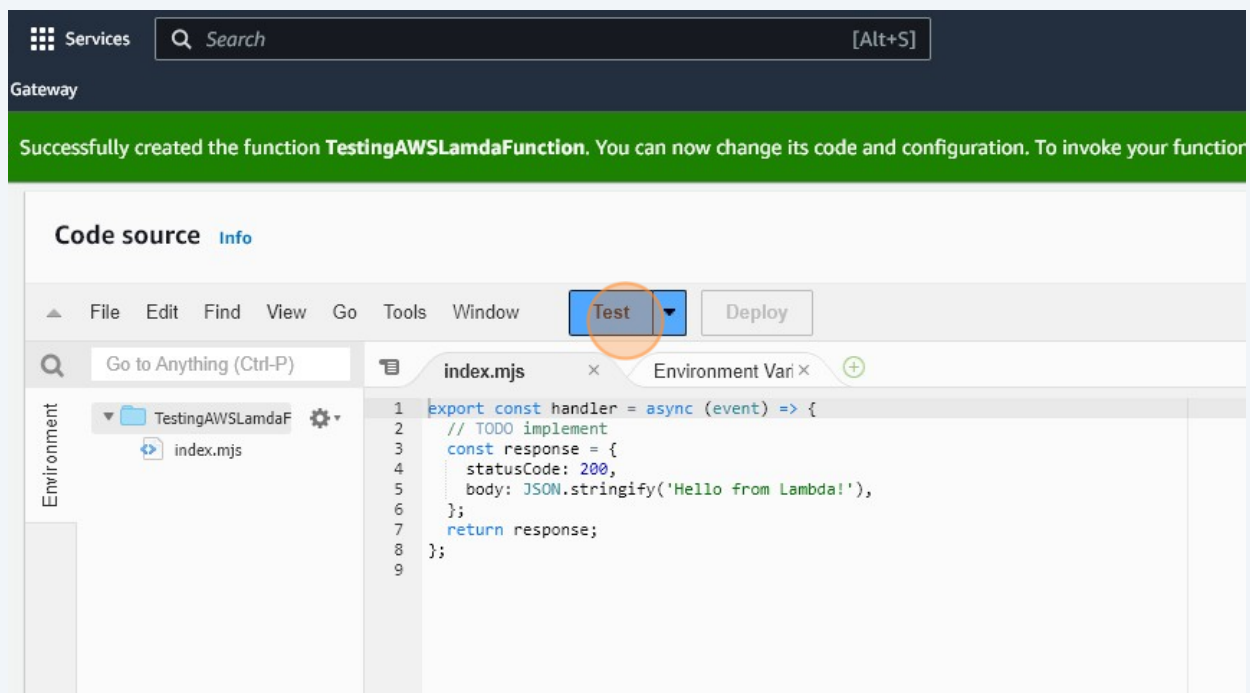
**▶ Advanced settings**

 CloudShell [Feedback](#)

## 17 Click "Create function"



## 18 Click "Test"





## 19 Click here. and remove all key

☐ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable e

Template - *optional*

hello-world

Event JSON

```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3",  
5 }
```

Cancel

## 20 Type

## 21 Click the "Event name" field. and give an event name

ed the function `TestingAWSLambdaFunction`. You can n

e Info

Find View Go Tools Window Test

thing (Ctrl-P)

index.mjs

```
1 export const handler = async
2 // TODO implement
3 const response = {
4   statusCode: 200,
5   body: JSON.stringify('Ha
6 };
7 return response;
8 };
9
```

ngAWSLambdaF

Test event action

☒ Create new event ☐ Edit save

Event name

MyEventName

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

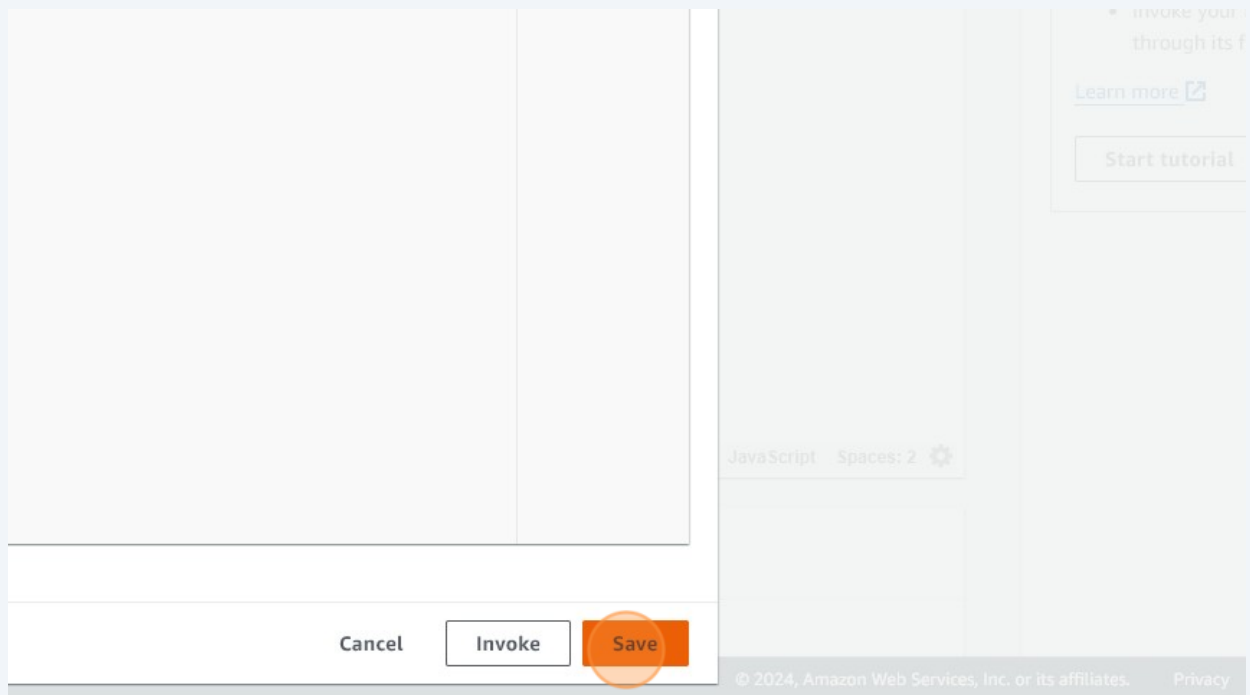
☒ Private  
This event is only available in the Lambda console and to the event creator. You can co

☐ Shareable  
This event is available to IAM users within the same account who have permissions to a

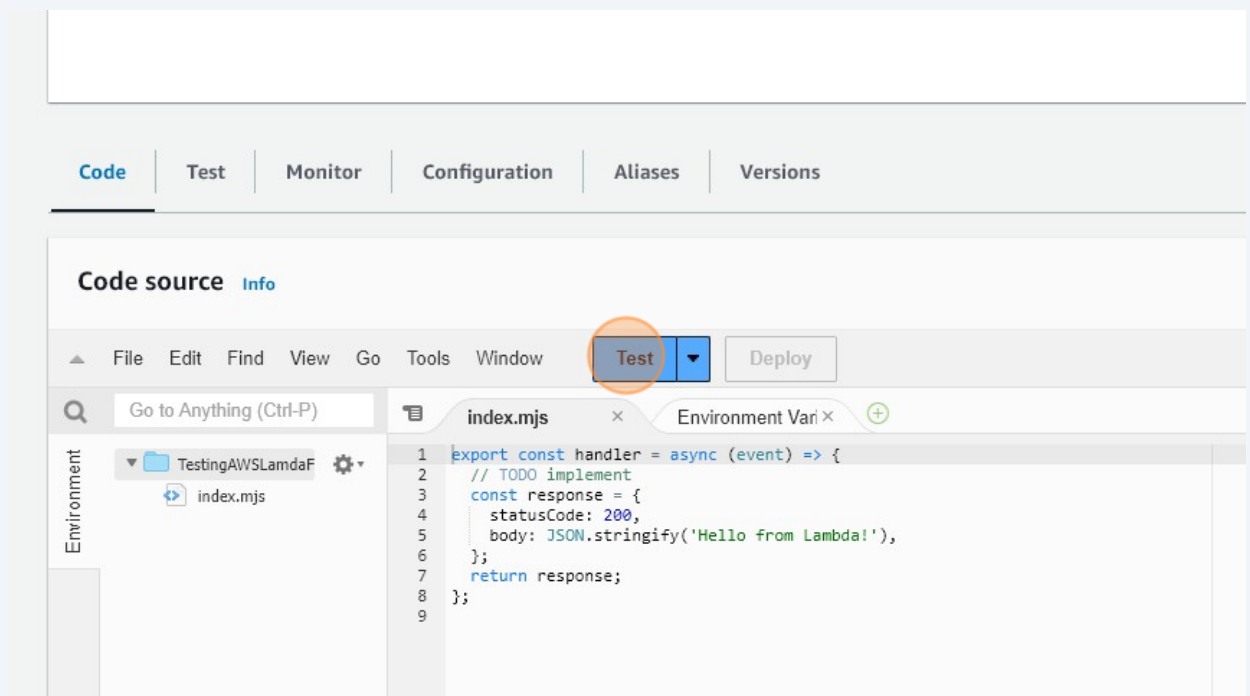
Template - optional

## 22 Type "AWS\_LAMDA\_NO\_DATA\_EVENT"

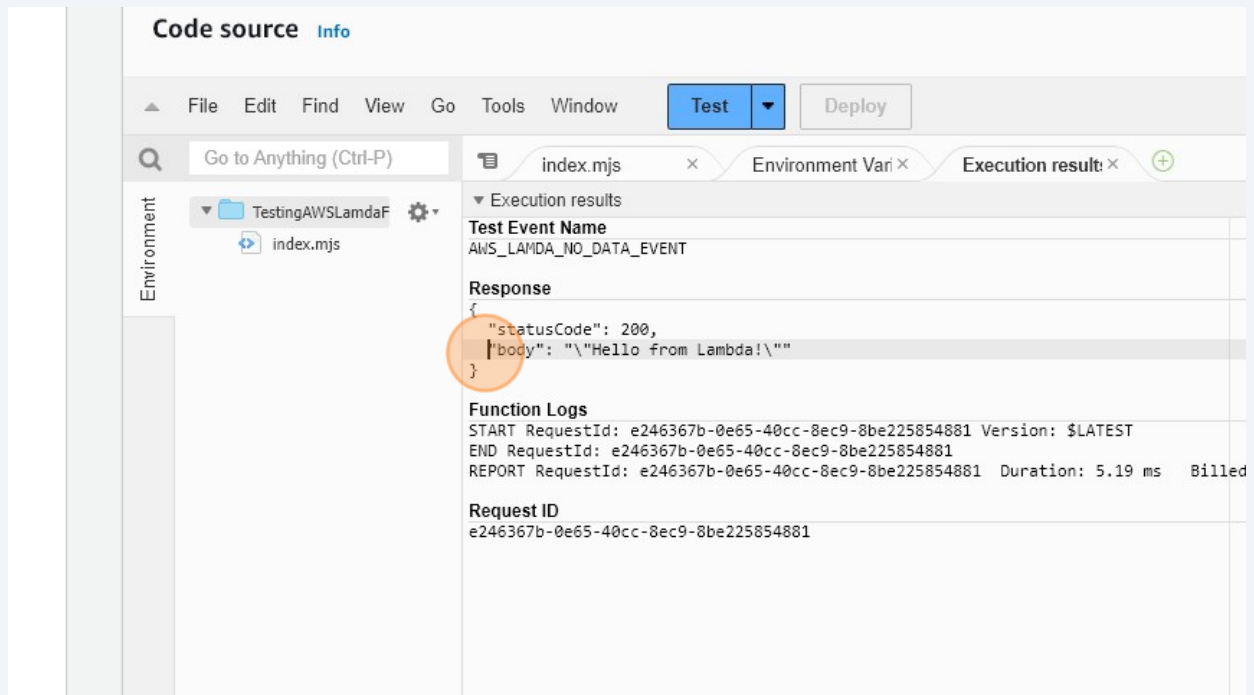
## 23 Click "Save"



## 24 Click "Test"



## 25 Verify the text is coming



The screenshot displays the AWS Lambda console interface. On the left, the 'Environment' sidebar shows a folder named 'TestingAWSLambdaF' containing a file 'index.mjs'. The main panel is titled 'Code source' and includes tabs for 'index.mjs', 'Environment Vari', and 'Execution results'. The 'Execution results' tab is active, showing the following details:

- Test Event Name:** AWS\_LAMDA\_NO\_DATA\_EVENT
- Response:** A JSON object with 'statusCode': 200 and 'body': '"Hello from Lambda!"'. The 'body' value is highlighted with an orange circle.
- Function Logs:** A log entry showing the start, end, and report of the function execution. The report includes the RequestId, Duration (5.19 ms), and Billed amount.
- Request ID:** e246367b-0e65-40cc-8ec9-8be225854881