

Basic information

Function name

Enter a name that describes the purpose of your function.

prathameshFunction

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 role later when adding triggers.

► [Change default execution role](#)

[Lambda](#) > [Functions](#) > prathameshFunction

prathameshFunction

Throttle

Copy ARN

Actions ▼

▼ Function overview [Info](#)

Export to Application Composer

Download ▼

Diagram

Template



prathameshFunction



Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

1 minute ago

Function ARN

arn:aws:lambda:ap-south-1:992382535340: function:prathameshFunction

Function URL [Info](#)

-

CodeTestMonitorConfigurationAliasesVersions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Monitoring and operations tools

General configurationInfo

Edit

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStartInfo	
0 min 3 sec	None	

Ephemeral storageInfo

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512

MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStartInfo

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

None

Supported runtimes: Java 11, Java 17, Java 21.

Timeout

0

min

3

sec

Execution role

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

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

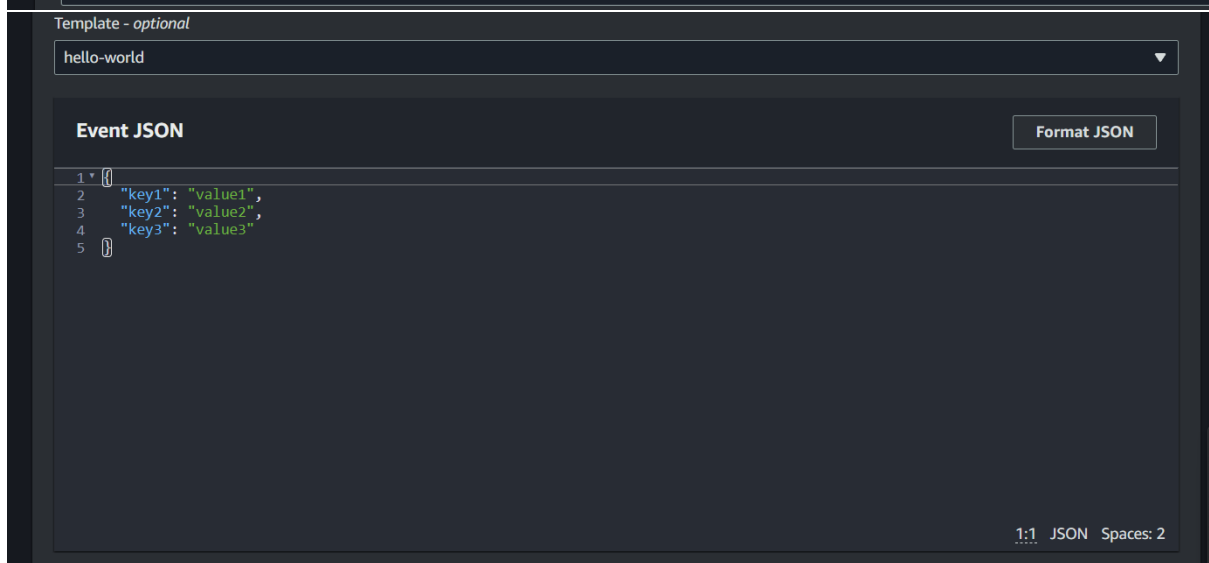
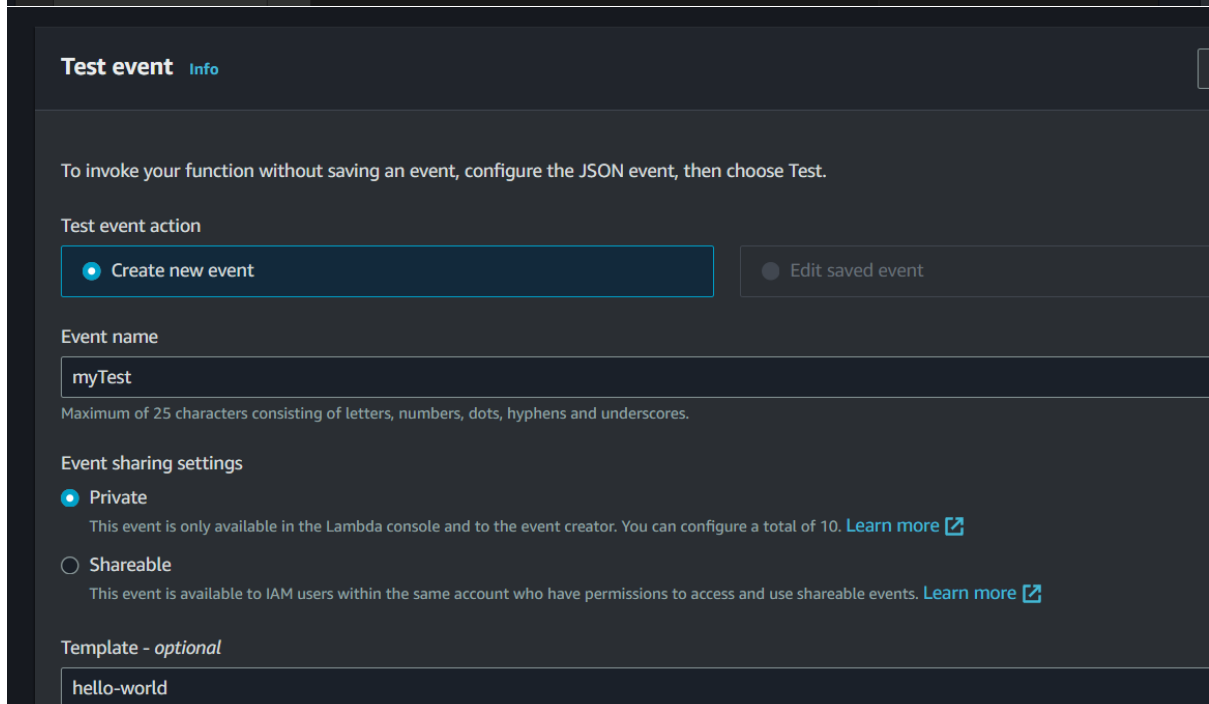
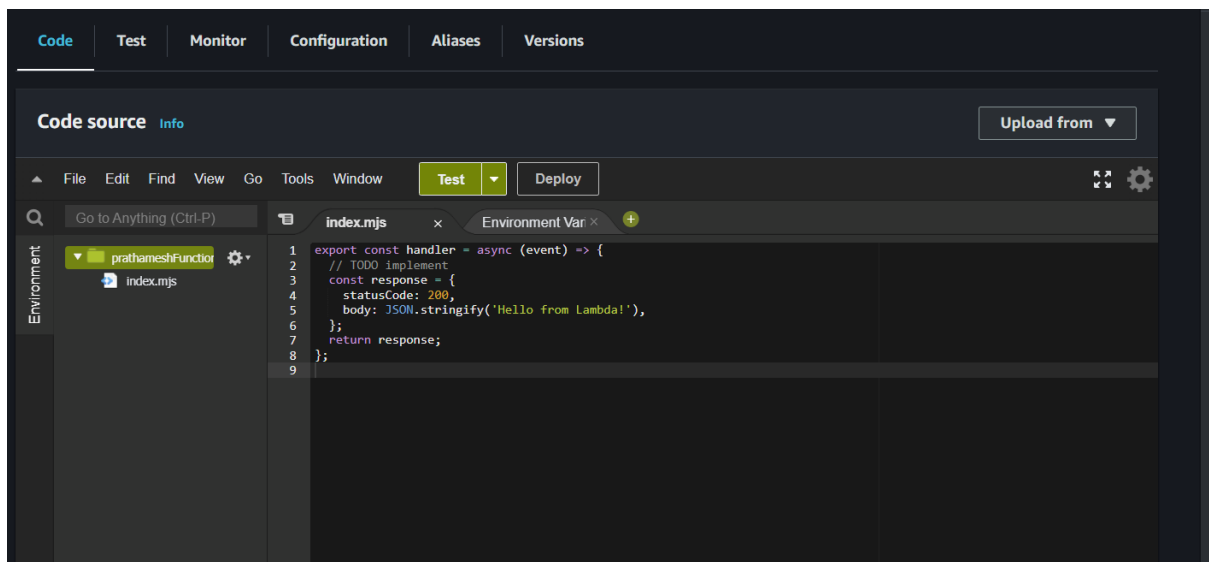
service-role/prathameshFunction-role-lhj6f62h

↻

[View the prathameshFunction-role-lhj6f62h role](#) on the IAM console.

Cancel

Save



Tools

Window

Test

▼

Deploy

Changes not deployed

index.mjs

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

Configure test event

Ctrl-Shift-C

Private saved events

myTest

Execution Results

Status: Successful

max memory used: 0.2 MB

Test Event Name

myTest

Response

{
 "statusCode": 200,
 "body": "\"Hello I am Prathamesh!\""
}

Function Logs

START RequestId: 68a17381-dd86-4ef9-b248-b4d6289dc664 Version: \$LATEST
END RequestId: 68a17381-dd86-4ef9-b248-b4d6289dc664
REPORT RequestId: 68a17381-dd86-4ef9-b248-b4d6289dc664 Duration: 9.79 ms Billed Duration: 10 ms Memory Size

Request ID

68a17381-dd86-4ef9-b248-b4d6289dc664