

Q10) To create Lambda function using Python to display "Hello\_StudentName".

Create a new Role:

**Roles (5)** [Info](#)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

☐ **Role name**

☐ [AWSServiceRoleForResourceExplorer](#)

☐ [AWSServiceRoleForSupport](#)

☐ [AWSServiceRoleForTrustedAdvisor](#)

☐ [ObjectAdded-role-ceagmly3](#)

☐ [ObjectAdded-role-kr32i7cc](#)

**Trusted entities**

AWS Service: resource-explorer-2 (Service-Linker)

AWS Service: support (Service-Linker)

AWS Service: trustedadvisor (Service-Linker)

AWS Service: lambda

AWS Service: lambda

**Last activity**

C

C

C

C

C

**Select trusted entity** [Info](#)

**Trusted entity type**

☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

**Use case**

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Service or use case**

**Choose a use case for the specified service.**

**Use case**

☒ **Lambda**  
Allows Lambda functions to call AWS services on your behalf.

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+', '=', '@', and '-' characters.

**Description**

**Permissions policy summary**

Policy name <a href="#">i</a>	Type	Attached as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSLambda_FullAccess</a>	AWS managed	Permissions policy
<a href="#">CloudWatchFullAccess</a>	AWS managed	Permissions policy

☒ **Role RishabhRole created.**

Now , we will create the Lambda Function

**Create function** [Info](#)

Choose one of the following options to create your function.

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

## Basic information

### Function name

Enter a name that describes the purpose of your function.

PythonHelloFunction

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

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.13

### Architecture

Choose the instruction set architecture you want for your function code.

☐ arm64

☒ x86\_64

### ▼ 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

#### 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

RishabhRole

[View the RishabhRole role](#) on the IAM console.

✓ Successfully created the function **PythonHelloFunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

## PythonHelloFunction

Now, move to the code section and change the code to "Hello StudentName"

CodeTestMonitorConfigurationAliasesVersions

Code sourceInfo

EXPLORER

PYTHONHELLOFUNCTION

lambda\_function.py

lambda\_function.py

2

3def lambda\_handler(event, context):

4 # TODO implement

5 return {

6 'statusCode': 200,

7 'body': json.dumps('Hello Rishabh')}

8 }

9

Now create a test event

### Create new test event

Event Name

Maximum of 25 characters consisting of letters, n

Event sharing settings

☒ Private  
This event is only available in the Lambda Console and t  
[more](#)

☐ Shareable  
This event is available to IAM users within the same acc  
[Learn more](#)

Template - optional

Event JSON

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

DEPLOY [UNDEPLOYED CHANGES]

You have undeployed changes.

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

Create new test event

Successfully created the function **PythonHelloFunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

The test event "testevent" was successfully saved.

Now, we will click deploy and then click Test

```
Status: Succeeded  
Test Event Name: testevent  
  
Response:  
{  
  "statusCode": 200,  
  "body": "\"Hello Rishabh\""  
}
```

Now we will delete the roles and functions that we have created