

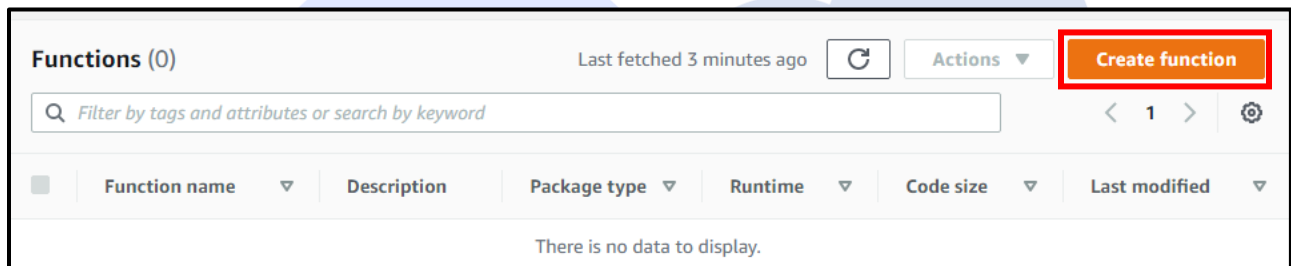
Tutorial to create a Microservice using AWS Lambda & Amazon API Gateway

Tutorial Objectives:

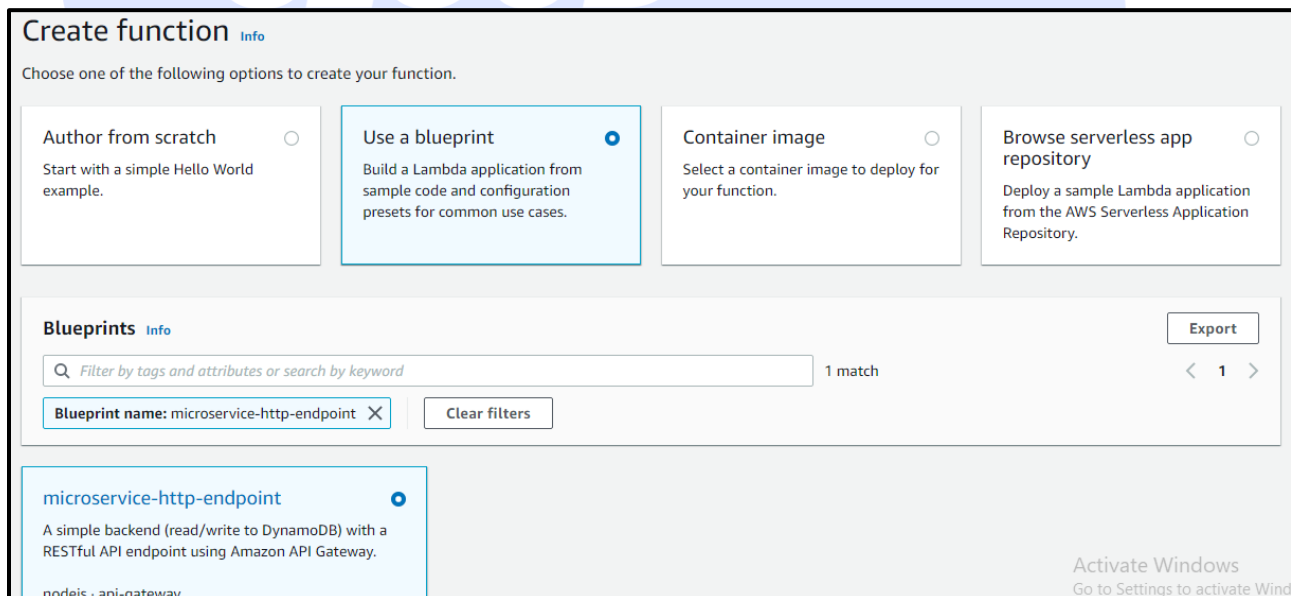
1. Learn to create a REST API using Amazon API Gateway.
2. Learn to create AWS Lambda function using AWS CLI

Step 1: Log on to your AWS Management Console and open AWS Lambda console.

Click on **Create Function**.



Step 2: Creating an API



Choose **Use a blueprint**.

Enter 'microservice' in the search bar. Choose the microservice-http-endpoint blueprint. Click on **Configure**.

Cloud Plus Plus Services



Configure your function with the following settings.

- Name – **lambda-microservice**
- Role – **Create a new role from AWS policy templates**
- Role name – **lambda-apigateway-role**
- Policy templates – **Simple microservice permissions**
- API – **Create an API**
- API Type – **HTTP API**
- Security – **Open**

Choose **Create function**.

Step 3: Testing the API

With your lambda-microservice function still open in the Lambda console, in the bottom, choose the **Test** tab.

Choose **New event**.

Choose the **Hello World** template.

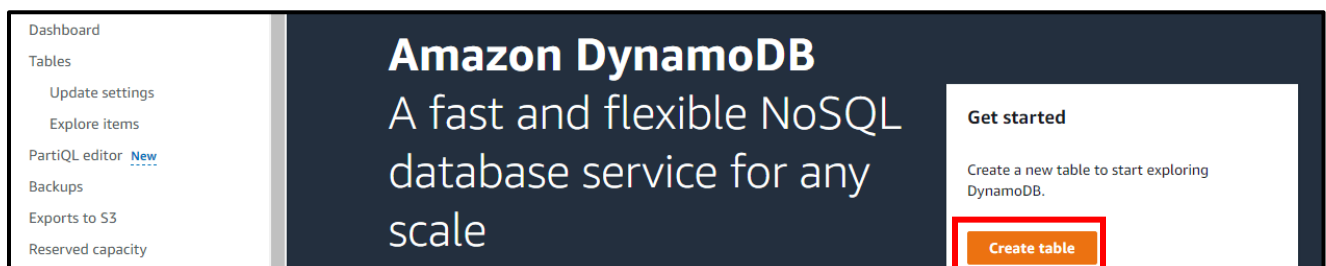
In Name, enter as the name as **TestEvent**.

In the text entry panel, **replace the existing text** with the following and click on save changes.

```
{
  "httpMethod": "GET",
  "queryStringParameters": {
    "TableName": "MyTable"
  }
}
```

(This GET command scans your DynamoDB table and returns all items found.)

Step 4: Once this is done, open AWS Management Console on another tab and open the DynamoDB console. Let us now create a table.



Cloud Plus Plus Services



Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

Make sure to **name** the table as '**MyTable**' and set the **partition key** as **Name**.
Keeping the rest as default, click on **Create Table**.

Tables (1) [Info](#)

Actions

Delete

Create table

Find tables by table name

Any table tag

< 1 >

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity
<input type="checkbox"/>	MyTable	<div><div></div>Active</div>	Name (Stri...	-	0	Provisioned with a...	Provisioned

Once the table is active, get back to the Lambda tab.

Now click on **Test**

[The test event TestEvent was successfully saved.](#)

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Test event [Delete](#) [Format](#) [Save changes](#) [Test](#)

Invoke your function with a test event. Choose a template that matches the service that triggers your function, or enter your event document in JSON.

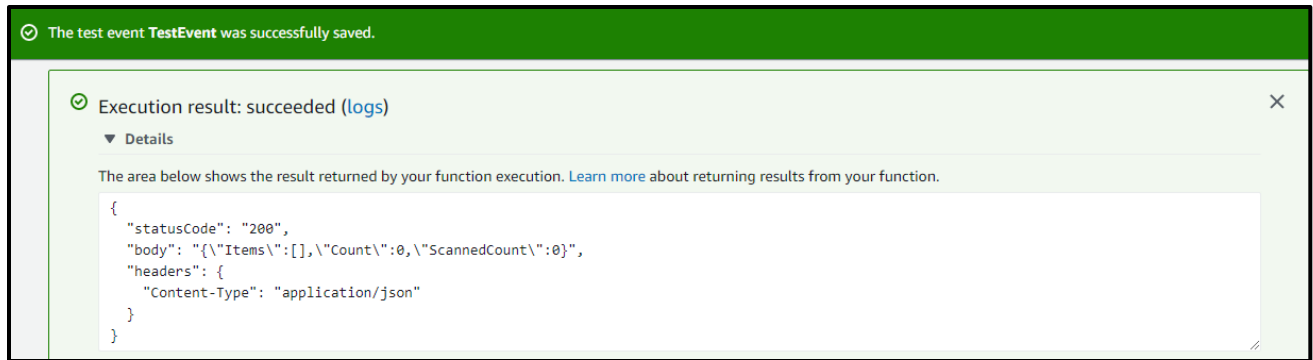
☐ New event
☒ Saved event

Saved event

```
1 {  
2   "httpMethod": "GET",  
3   "queryStringParameters": {  
4     "TableName": "MyTable"  
5   }  
6 }
```

Once you click on Test, the following details will be visible.

Cloud Plus Plus Services



Note: Now, if you no longer need the resources, delete the DynamoDB table and the Lambda function that was created.



Your trusted partner for
cloud enablement

Document Created by	Version
Bavyaa R	04-Feb-2022