

HW #5 Microservice using Lambda and API Gateway

Shashank Reddy Gopireddy

David Tang

Haixiang Yu

Venkata Kalyan Ponduri

To create an API

1. Sign in to the AWS Management Console and open the AWS Lambda console.
2. Choose **Create Lambda function**.
3. Choose **Blueprint**.
4. Enter **microservice** in the search bar. Choose the **microservice-http-endpoint** blueprint and then choose **Configure**.
5. Configure 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 a new API**.
 - **Security** – **Open**.

Choose **Create function**.

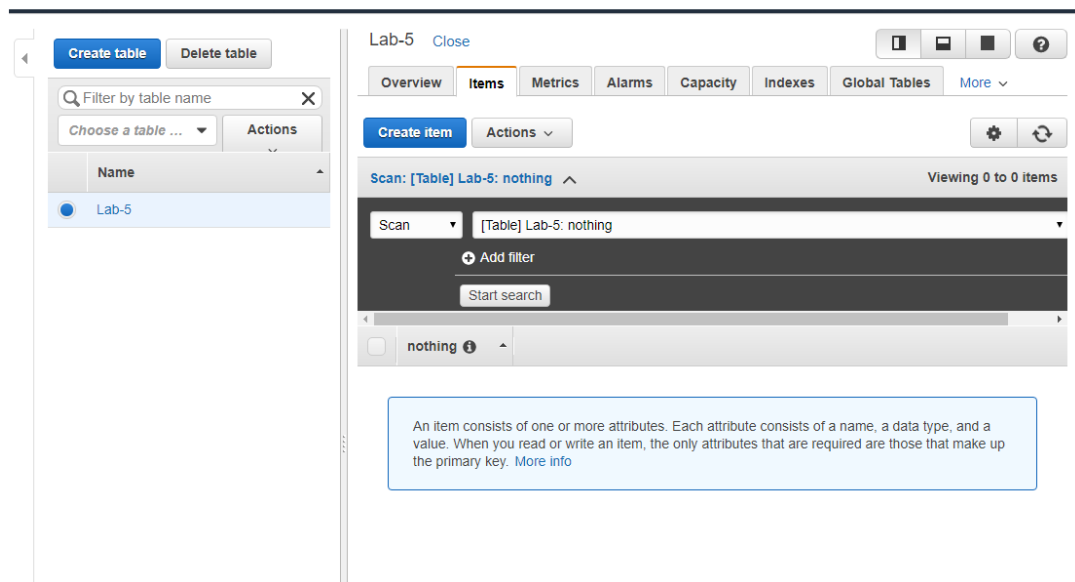
To create a DynamoDB table

1. Open the [DynamoDB console](#).
2. Choose **Create table**.
3. Create a table with the following settings.

- **Table name** – Lab-5
 - **Primary key** – nothing (string)
4. Choose **Create**.

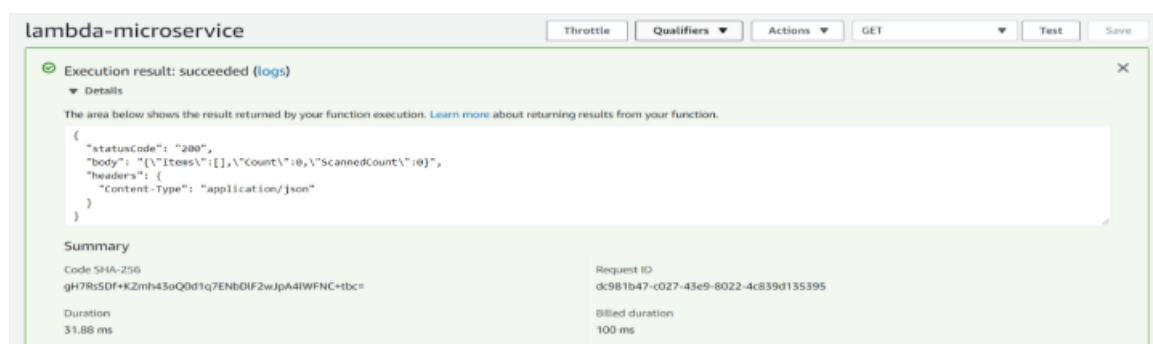
To enable streams

1. Open the [DynamoDB console](#).
2. Choose **Tables**.
3. Choose the **lambda-dynamodb-stream** table.
4. Under **Overview**, choose **Manage stream**.
5. Choose **Enable**.



Excution results for GET

“200” means succeeded



Exculution results for POST

The screenshot shows the AWS Lambda console for the function 'lambda-microservice'. The 'POST' method is selected. The execution result is 'succeeded (logs)'. The response body is a JSON object:

```
{  "statusCode": "200",  "body": "{}",  "headers": {    "Content-Type": "application/json"  }}
```

. The summary shows the function was executed successfully with a duration of 50.52 ms and a billed duration of 100 ms.

“200” means succeeded and no error

And an item has been created in table.

The screenshot shows the AWS DynamoDB console for the table 'Lab-5'. The 'Items' tab is selected. The table is empty, showing 'nothing' as the scan result. The 'Create item' button is visible. The table name 'Lab-5' is also visible in the left sidebar.

Acting GET again to verify

The screenshot shows the AWS Lambda console for the function 'lambda-microservice'. The 'GET' method is selected. The execution result is 'succeeded (logs)'. The response body is a JSON object:

```
{  "statusCode": "200",  "body": "{"items":[{"id":"1","count":1,"scannedCount":1}],"count":1,"scannedCount":1}",  "headers": {    "Content-Type": "application/json"  }}
```

. The summary shows the function was executed successfully with a duration of 208.63 ms and a billed duration of 500 ms.

Acting PUT to change the item number to 2

The screenshot shows the AWS Lambda console for a function named 'lambda-microservice'. The 'PUT' button is highlighted in the top navigation bar. The 'Execution result: succeeded' section is expanded, showing a JSON response:

```
{  "statusCode": "200",  "body": "[]",  "headers": {    "Content-Type": "application/json"  }}
```

. Below this, the 'Summary' section provides details: Code SHA-256 (gh7R5dF+KZmh43oQd1q7ENbDF2w/pA4lWFNC+tb=), Request ID (d5db5a72-3517-42b2-8b8d-727827b75b07), Duration (50.50 ms), Billed duration (100 ms), Resources configured (512 MB), and Max memory used (75 MB).

The screenshot shows the AWS DynamoDB console for a table named 'Lab-5'. The 'Items' tab is selected, displaying a list of items. The table has a 'Scan' button and a 'Start search' button. The 'Items' list shows one item with a key of 'nothing' and a value of '2'. The 'Create item' button is visible at the top of the items list.

Acting DELETE to delete item 2

Lambda > Functions > lambda-microservice ARN - arn:aws:lambda:us-east-1:364571415246:function:lambda-microservice

lambda-microservice Throttle Qualifiers Actions DELETE Test Save

Execution result: succeeded (logs) Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "statusCode": "200",
  "body": "{}",
  "headers": {
    "Content-Type": "application/json"
  }
}
```

Summary

Code SHA-256	Request ID
gh7RsSDF+KZmh43oQd1q7ENbDf2wjpA4lWfNC+tbC=	9b62022c-5588-489d-b849-fda85a69b838
Duration	Billed duration
213.28 ms	300 ms
Resources configured	Max memory used
512 MB	74 MB Init Duration: 204.01 ms

Lab-5 Close

Create table Delete table

Filter by table name Choose a table ... Actions

Name

- Lab-5

Overview **Items** Metrics Alarms Capacity Indexes Global Tables More

Create item Actions

Scan: [Table] Lab-5: nothing Viewing 0 to 0 items

Scan [Table] Lab-5: nothing

Add filter

Start search

nothing

An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. [More info](#)

Github:

<https://github.com/Kwn1247/CMPE-207/tree/master/Assignments/Lab%205>