

Create API Gateway for Backend Operation

(LAB-M09-01)

Version Control	
Document	Create API Gateway for Backend Operation
Owner	Ahmad Majeed Zahoory
Version	3.1
Last Change	28 th May 2024
Description of Change	Code version updated

Lab duration: 60 minutes

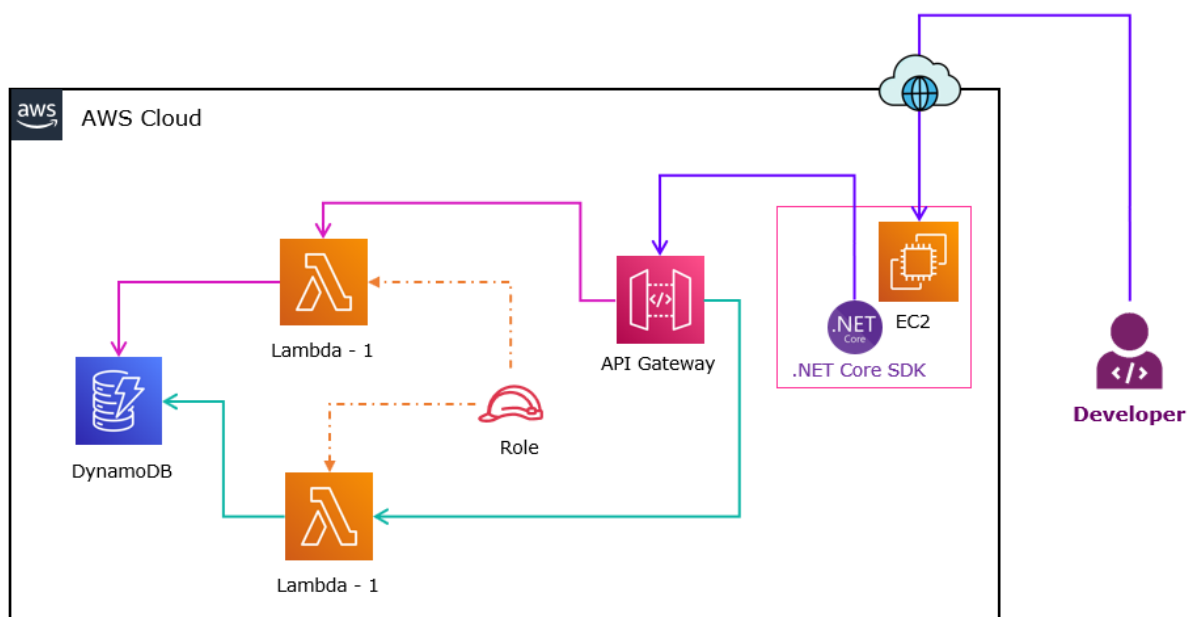
Lab scenario

In this lab, you will learn how to use AWS Lambda to trigger a Lambda function and update the DynamoDB. You will also integrate the Lambda function with API gateway and trigger a Lambda function via API Gateway.

Objectives

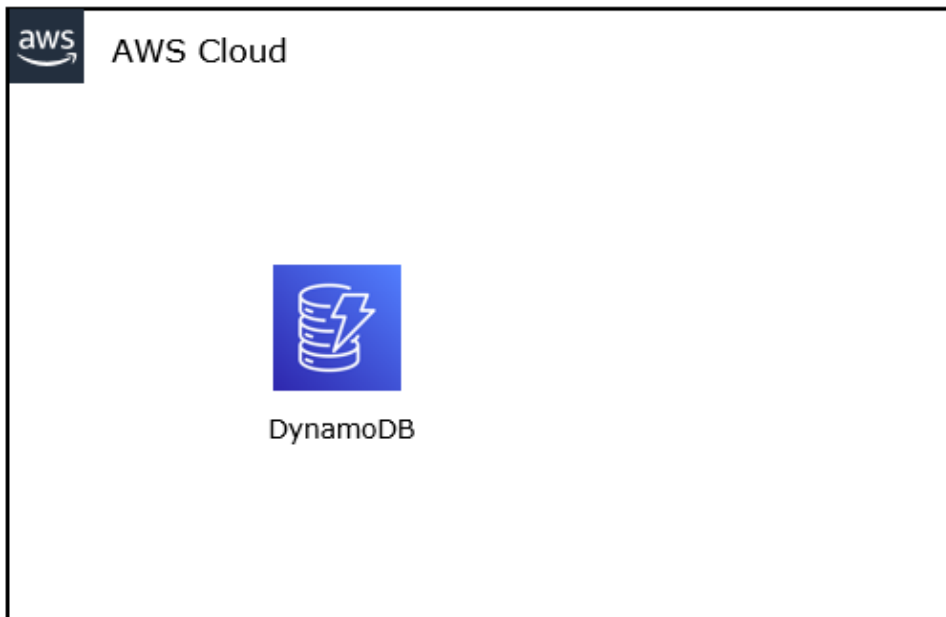
After you complete this lab, you will be able to:

- Create DynamoDB table.
- Create Lambda function.
- Add the data to the DynamoDB table.
- Integrate the Lambda function with API gateway.



Task 1: Create Database

In this task, you will create the DynamoDB table to host items.



Step 1: Create a DynamoDB Table

1. In the **AWS Management Console**, on the **Services** menu, search and select **DynamoDB**.
2. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
3. Choose **Create Table**.
 - a. In the **Create table** page:
 - i. In the **Table details** section:
 - a) **Table name**: Write **empdata**.
 - b) **Primary key**: Write **empid**.
 - 1) Set the data type to **String**.

Note: Write the **table name** and **primary key** in the **lower case** only.

Table name
This will be used to identify your table.

empdata

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 hosts for scalability and availability.

empid

String

Up to 255 characters and case sensitive.

Note: Leave the other details as default.

c) Select **Create table**.

Note: Wait till **DynamoDB table** gets **created**.

Step 2: Add Items into DynamoDB Table

4. From the **DynamoDB** console.

5. Select **Explore items**.

a. Select **empdata**.

i. Select **Create item**.

Tables (1)

Any tag key

Any tag value

Find tables by table name

empdata

empdata

Autopreview

View table details

► Scan or query items
Expand to query or scan items.

Items returned (0)

Create item

No items
No items to display.

Create item

a) In the **Create item** page:

1) **empid**: Write **001** (*in value field*).

I. Select **Add new attribute**.

A. Select **String**.

Create item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

Form JSON view

Attribute name	Value	Type
empid - Partition key	001	String

Add new attribute ▲

- String
- Number
- Boolean
- Binary
- Null
- String set
- Number set
- Binary set
- List
- Map

2) **Attribute name**: Write **empfirstname**.

I. **Value**: Write **Ajay**.

A. Select **Add new attribute**.

B. Select **String**.

3) **Attribute name**: Write **emplastname**.

I. **Value**: Write **Kaushik**.

A. Select **Add new attribute**.

B. Select **String**.

4) **Attribute name**: Write **empage**.

I. **Value**: Write **32**.

Note: Write the **empfirstname**, **emplastname** and **empage** in the **lower case** only.

Attribute name	Value
empid - Partition key	001
empfirstname	Ajay
emplastname	Kaushik
empage	32

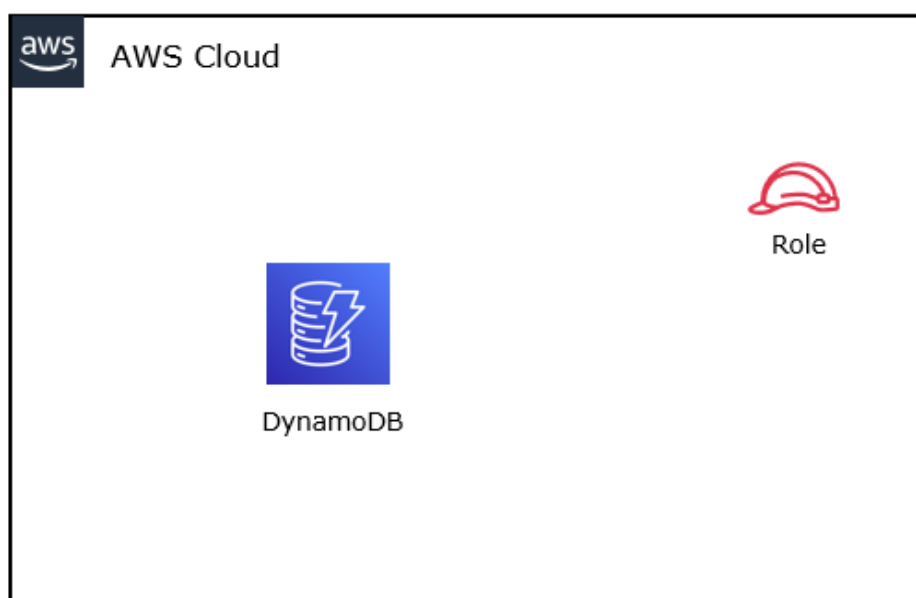
5) Select **Create item**.

Note: You can see the item details under Items.

Items returned (1)					Actions	Create item
<input type="text" value="Find items"/>						
<input type="checkbox"/>	empid	▼	empage	▼	empfirst...	▼
<input type="checkbox"/>	001		32		Ajay	Kaushik

Task 2: Create IAM Role

In this task, you will create the IAM role for Lambda.



Step 1: Create IAM Roles for AWS Lambda

6. In the **AWS Management Console**, on the **Services** menu, search and select **IAM**.
7. Select **Roles**.
 - a. Click on **Create role**.
 - i. In the **Select trusted entity** section.
 - a) **Trusted entity type**: Select **AWS service**.
 - b) **Use cases**: Select **Lambda**.
 - c) Select **Next**.
 - ii. In the **Add permissions** section.
 - a) In the **Search box**, write **AmazonDynamoDBFullAccess** and select **Enter Key**.
 - 1) Select **AmazonDynamoDBFullAccess**.
 - 2) Select **Clear search query**.
 - b) In the **Search box**, write **AWSLambdaBasicExecutionRole** and select **Enter Key**.
 - 1) **Filter by type**: Dropdown and select **AWS managed**.
 - 2) Select the **AWSLambdaBasicExecutionRole**.
 - c) Select **Next**.
 - iii. In the **Name, review, and create** section.
 - a) **Role name**: Write **Lambda-DynamoDB-Role-YOUR-ID**.

Role details

Role name

Enter a meaningful name to identify this role.

Lambda-DynamoDB-Role

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


Description

Add a short explanation for this role.

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @- _' characters.

Note: You can see the **AmazonDynamoDBFullAccess** and **AWSLambdaBasicExecutionRole** policy under the **Add permissions** section.

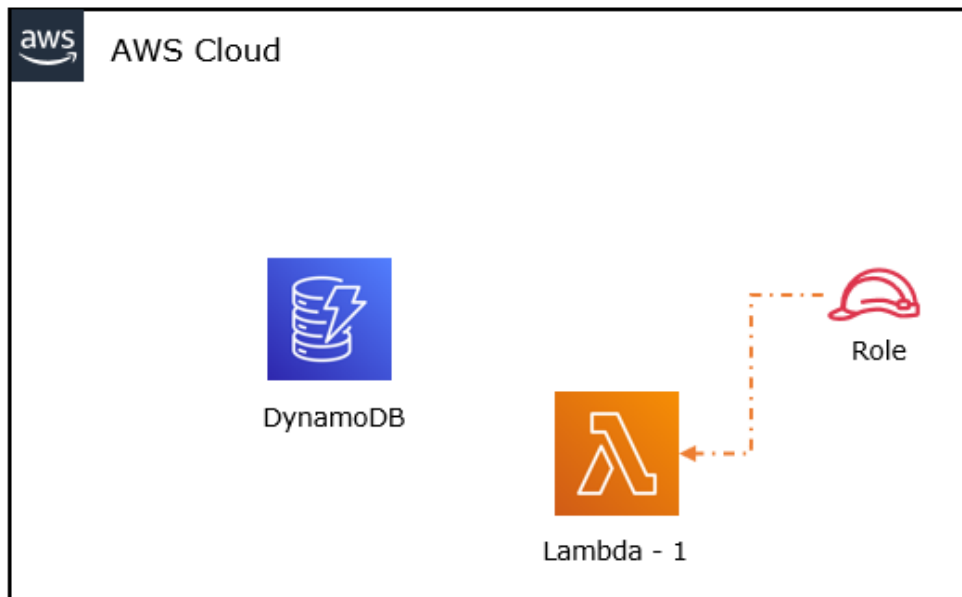
Permissions policy summary		
Policy name 	Type	Attached as
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy
AmazonDynamoDBFullAccess	AWS managed	Permissions policy

b) Click **Create role**.

Note: **Wait**, till you can see the **message "Role Lambda-DynamoDB-Role created"**.

Task 3: Create Lambda Function to Read the Items

In this task, you will create the Lambda function to read the items from the DynamoDB.



Step 1: Create Lambda Function to Read the Items

8. In the **AWS Management Console**, on the **Services** menu, search and select **Lambda**.
9. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
10. Select **Create a function**.
11. Select **Author from scratch**.
 - a. In the **Basic information** section:
 - i. **Name**: Write **ReadItems**.
 - ii. **Runtime**: Dropdown and Select **Node.js 20.x**.

Basic information

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

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.

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.

iii. **Expand** **Change default execution role.**

a) **Role:** Select **Use an existing role.**

1) **Existing role:** Dropdown and Select **Lambda-DynamoDB-Role.**

▼ **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 to Amazon CloudWatch Logs.

[View the Lambda-DynamoDB-Role role](#) on the IAM console.

iv. Select **Create function.**

Note: **Wait**, till you can see the **message "Successfully created the function ReadItems"**.

b. From the **ReadItems** Lambda function:

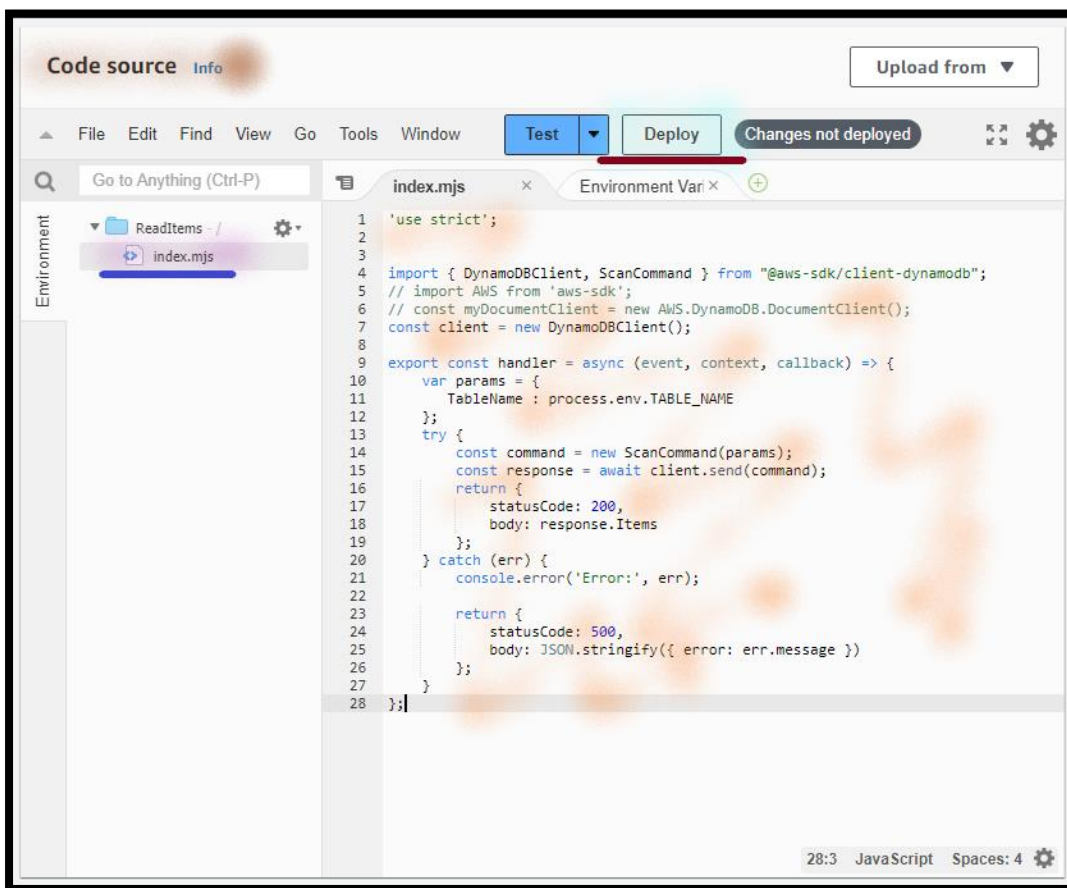
i. Select the **Code** section:

a) Click on **index.mjs**.

- 1) **Replace** the **existing code**.
- 2) **Copy** the **Code** from **read-function-code** file.

Note: **read-function-code.txt** is available with the **Lab manual**.

3) Select **Deploy**.



b) Select the **Configuration** section:

1) Select **Environment variables**.

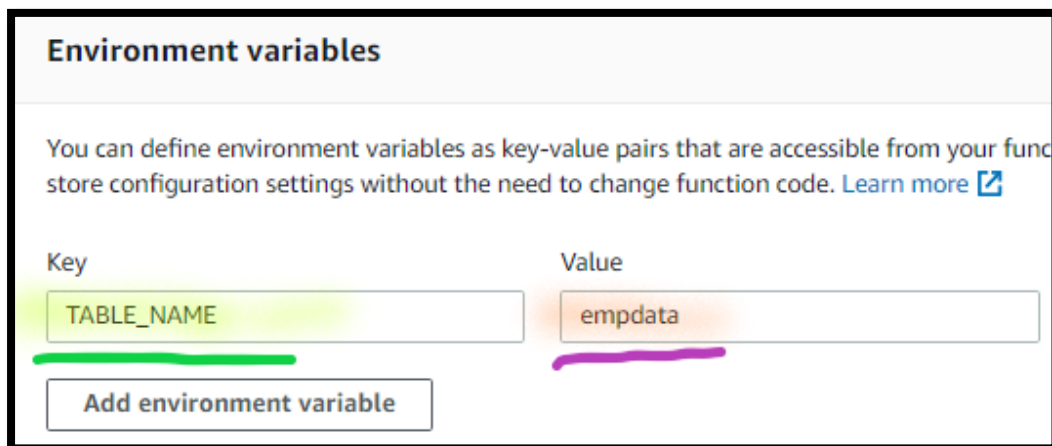
I. Select **Edit**.



II. Select **Add environment variables**.

A. **Key**: Write **TABLE_NAME**.

B. **Value**: Write **empdata**
(DynamoDB table name).



C. Select **Save**.

Step 2: Validate Your Implementation

12. From the **ReadItems** Lambda function:

a. Select the **Test** section:

i. In the **Test event** section:

a) **Event Name**: Write **TestReadItems**.

b) **Event JSON**:

1) **Remove** the **existing event**.

I. **Copy** the **below event**.

```
{  
  "empid": "*"   
}
```

Event name
TestReadItems
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 configure a total of 10. [Learn more](#)
☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional
hello-world

Event JSON
Format JSON

```
1 {  
2   "empid": "*"   
3 }
```

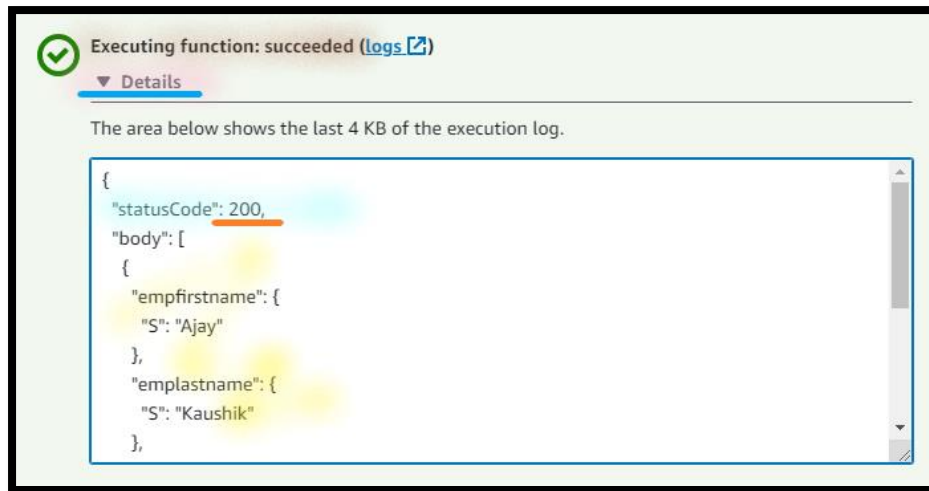
Note: Leave the other details as default.

c) Select **Test**.

Note: If Test executed successfully, you can see the **Execution result** as **succeeded**.

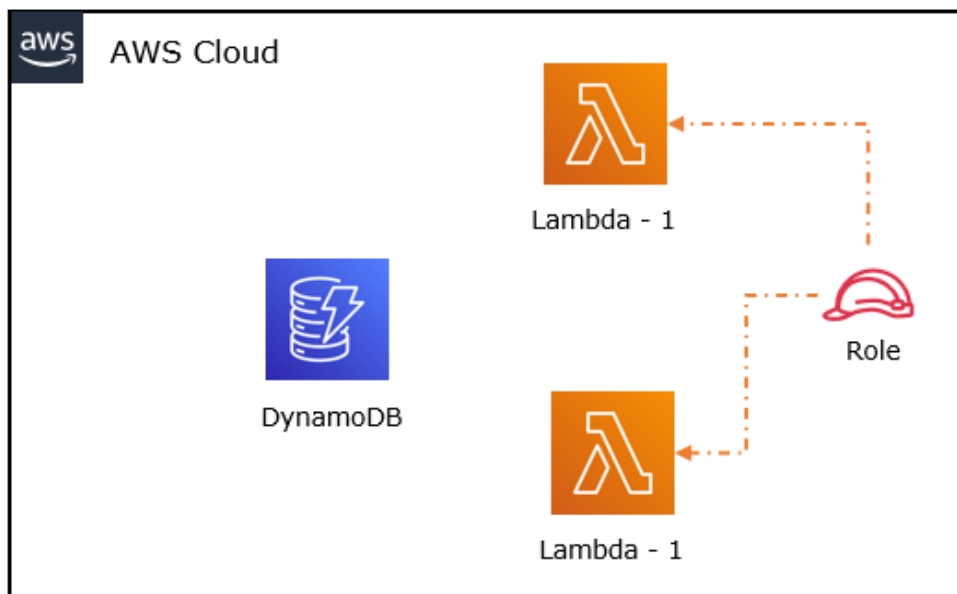
ii. **Expand** the **Details** section of the **execution result** section.

Note: You can see the **Items**, which you have **added** in the **DynamoDB** in the Previous step.



Task 4: Create Lambda Function to Write the Items

In this task, you will create the Lambda function to write the items to the DynamoDB.



Step 1: Create Lambda Function to Write the Items

13. From the **Lambda** Function console.
14. Select **Function**.
15. Select **Create a function**.
16. Select **Author from scratch**.

- a. In the **Basic information** section:
 - i. **Name:** Write **WriteItems**.
 - ii. **Runtime:** Dropdown and Select **Node.js 20.x**.
 - iii. **Expand Change default execution role.**
 - a) **Role:** Select **Use an existing role**.
 - 1) **Existing role:** Dropdown and select **Lambda-DynamoDB-Role**.
 - iv. Select **Create function**.

Note: **Wait**, till you can see the **message** "**Successfully created the function WriteItems**".

- b. From the **WriteItems** Lambda function:
 - i. Select the **Code** section.
 - a) Click on **index.js**.
 - 1) **Replace** the **existing code**.
 - 2) **Copy** the **Code** from **write-function-code** file.

Note: **write-function-code.txt** is available with the **Lab manual**.

- b) Select **Deploy**.
- ii. Select the **Configuration** section:
 - a) Select **Environment variables**.
 - 1) Select **Edit**.
 - 2) Select **Add environment variables**.
 - I. **Key:** Write **TABLE_NAME**.
 - II. **Value:** Write **empdata** (*DynamoDB table name*).
 - III. Select **Save**.

Step 2: Validate Your Implementation

17. From the **WriteItems** Lambda function:

a. Select the **Test**.

i. In the **Test event** section:

a) **Event Name:** Write **TestWriteItems**.

b) **Event JSON:**

1) **Remove** the **existing events**

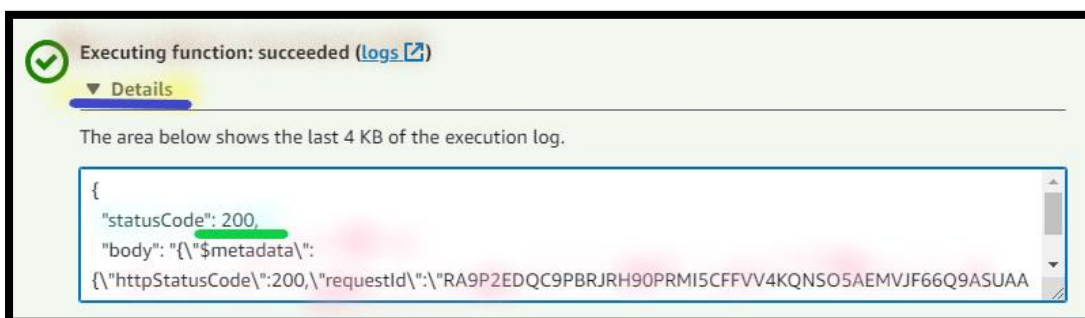
2) **Copy** the **below event**.

```
{
  "empid": "002",
  "empfirstname": "Sana",
  "emplastname": "Yusuf",
  "empage": "21"
}
```

Note: Leave the other details as default.

ii. Select **Test**.

Note: If Test executed successfully, you can see the **execution result** as **succeeded**.



Step 3: View the DynamoDB Data

18. In the **AWS Management Console**, on the **Services** menu, search and select **DynamoDB**.

19. Select **Explore items**.

a. Select **empdata**.

i. Select **Refresh**.

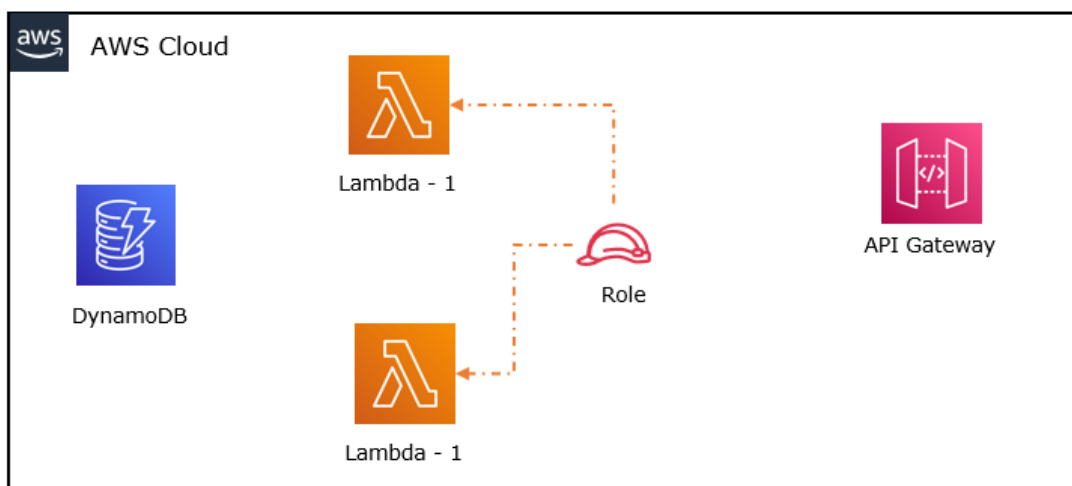
Note: You can see the **Added Items**, which you have added in the DynamoDB via the **Lambda function** and **Console**.



	empid (String)	empage	empfirstname	emplastname
<input type="checkbox"/>	001	32	Ajay	Kaushik
<input type="checkbox"/>	002	21	Sana	Yusuf

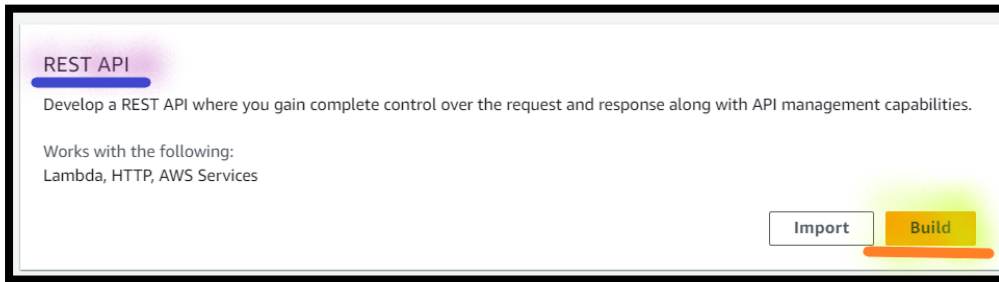
Task 5: Create a RESTful API

In this task, you will create the API gateway.

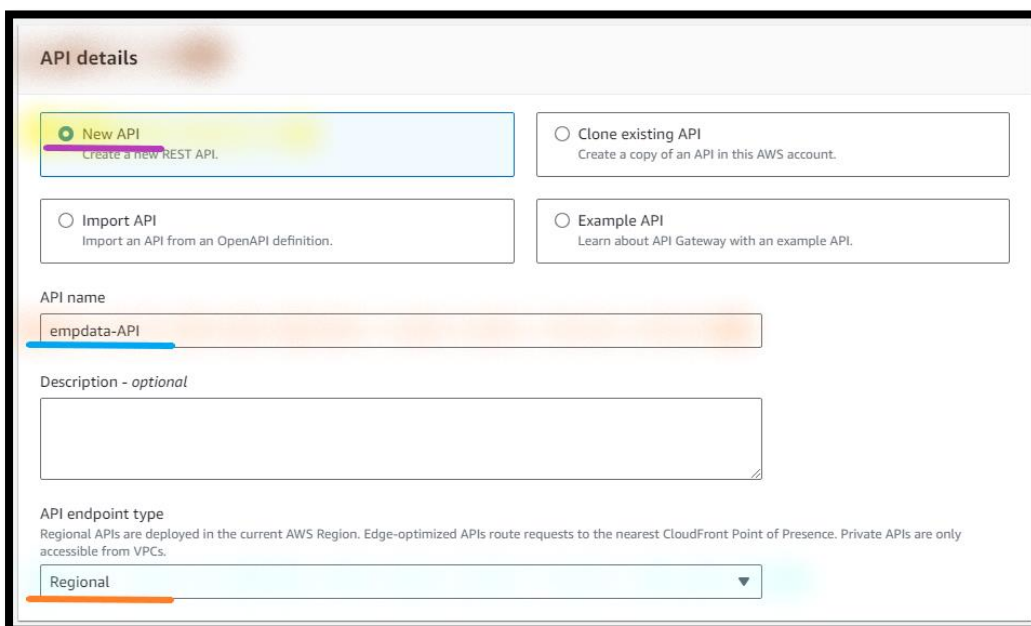


Step 1: Create REST API

20. In the **AWS Management Console**, on the **Services** menu, search and select **API Gateway**.
21. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
22. Select **Rest API** (*Don't select Rest API private*).
 - a. Select **Build**.



- i. From the **Create REST API** page:
 - a) In the **API details** section:
 - 1) Select **New API**.
 - 2) **API name**: Write **empdata-API**.
 - 3) **API Endpoint type**: Dropdown and select **Regional**.



- ii. Select **Create API**.

Task 6: Create API Method to Read and Write the Data

In this task, you will configure the API gateway to fetch the items from the DynamoDB using the Lambda function.

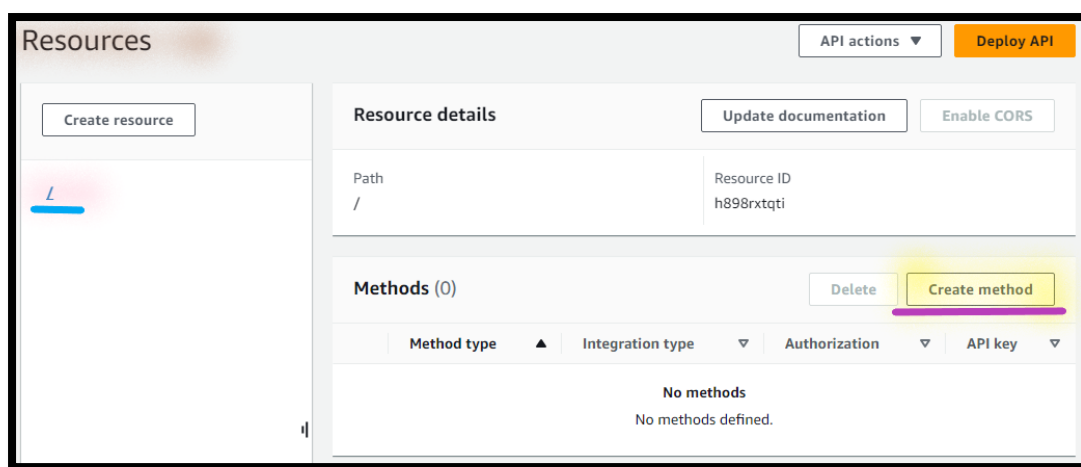
Step 1: Create Method to Read the Items

23. From the **empdata-API** gateway:

24. Select **Resources**.

a. Select **/** (resource path).

i. Select **Create method**.



a) From the **Create method** page:

1) In the **Method details** section:

I. **Method type**: Dropdown and select **Get**.

II. **Integration type**: Select **Lambda function**.

III. **Lambda function**:

A. **Region**: Dropdown and select the **YOUR ALLOCATED REGION-ID**.

Note: Replace the **region-identifier**.

Refer the **link** to know your **respective region region identifier**

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

- B. **Choose a lambda function:** In the **Search** section, click and select **ReadItems** (lambda function you created in the previous step).

Method details

Method type
GET

Integration type

- ☒ **Lambda function**
Integrate your API with a Lambda function.
- ☐ HTTP
Integrate with an existing HTTP endpoint.
- ☐ Mock
Generate a response based on API Gateway mappings and transformations.
- ☐ AWS service
Integrate with an AWS Service.
- ☐ VPC link
Integrate with a resource that isn't accessible over the public internet.

☐ Lambda proxy integration
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

☒ Default timeout
The default timeout is 29 seconds.

Note: Leave the other details as default.

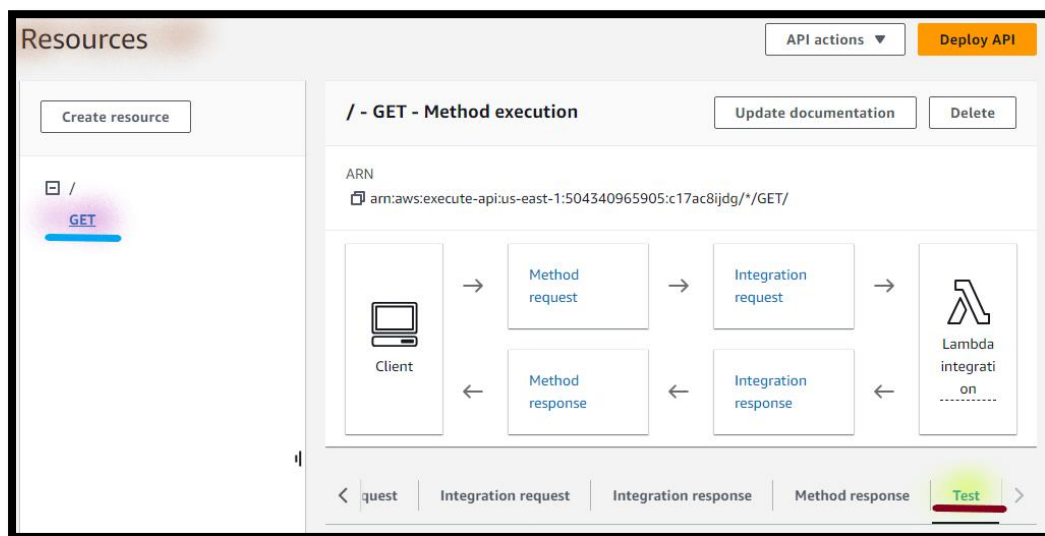
2) Select **Create method**.

Note: **Wait**, till the **Get - Method execution** gets created.

Step 2: Test the API Gateway to Read the Items

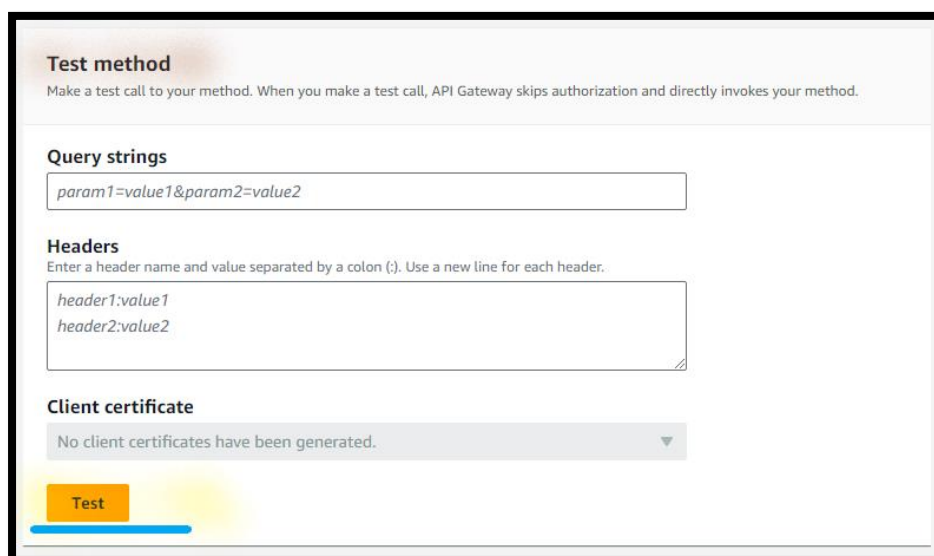
25. From the **empdata-API** gateway:

- a. Select **Resources**.
 - i. Select **/** (*resource path*).
 - a) Select **Get**.
 - 1) Select **Test**.



Note: Leave all the details as default.

- I. Select **Test**.



Note: If request **executed succesfully**, you can see the request **Status** as **200**.

Note: In the **Response body**, you can see the **Items**, which you have added in the DynamoDB in the previous steps.



Step 3: Create Method to Write the Items

26. From the **empdata-API** gateway:

27. Select **Resources**.

a. Select **/** (resource path).

i. Select **Create method**.

a) From the **Create method** page:

1) In the **Method details** section:

I. **Method type:** Dropdown and select **Post**.

II. **Integration type:** Select **Lambda function**.

III. Lambda function:

- A. **Region:** Dropdown and select the **YOUR ALLOCATED REGION-ID**.

Note: Replace the **region-identifier**.

Refer the **link** to know your **respective region region identifier**

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

- B. **Choose a lambda function:** In the **Search** section, click and select **WriteItems** (lambda function you created in the previous step).

Note: Leave other details as default.

The screenshot shows the 'Method details' configuration page in the AWS API Gateway console. The 'Method type' is set to 'POST'. Under 'Integration type', the 'Lambda function' option is selected. Below this, there are five other integration options: 'HTTP', 'Mock', 'AWS service', 'VPC link', and 'Lambda proxy integration'. The 'Lambda function' section is expanded, showing a dropdown for the region (set to 'us-east-1') and a search bar containing the ARN 'arn:aws:lambda:us-east-1:504340965905:function:WriteItems'. A note below the search bar states: 'Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.' At the bottom, the 'Default timeout' is set to 29 seconds.

2) Select **Create method**.

Note: **Wait**, till the **Post - Method execution** gets created.

Step 4: Test the API Gateway to Write the Items

28. **From** the **empdata-API** gateway:

a. Select **Resources**.

i. Select **/** (resource path).

a) Select **POST**.

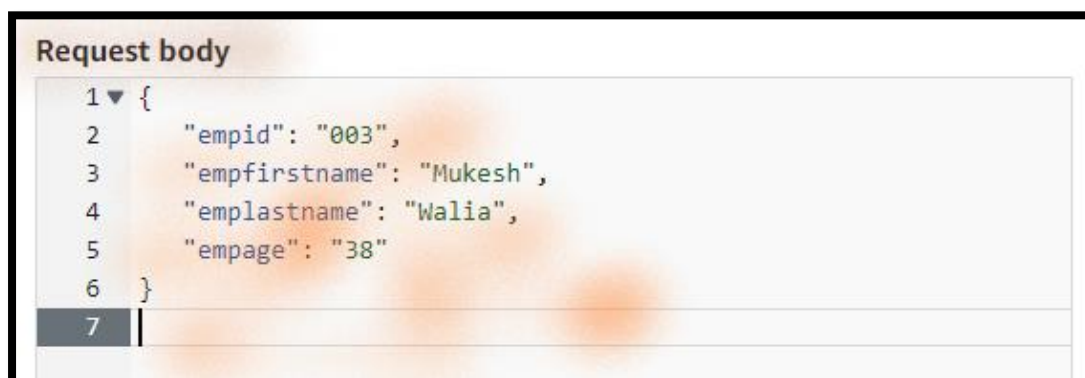
1) Select **Test**.

I. **Request body:** Write the **below event**:

```
{
  "empid": "003",
  "empfirstname": "Mukesh",
  "emplastname": "Walia",
  "empage": "38"
}
```

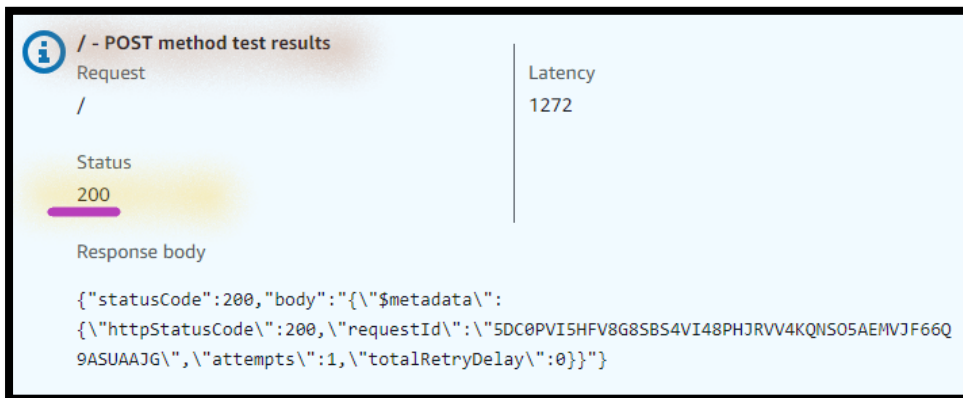
Note: Leave all the details as default.

Note: You can now **add** the **new items** via **API Gateway**.



A. Select **Test**.

Note: If request **executed successfully**, you can see the request **Status** as **200**.



Step 5: View the DynamoDB Data

29. In the **AWS Management Console**, on the **Services** menu, search and select **DynamoDB**.

30. Select **Explore items**.

a. Select **empdata**.

i. Select **Refresh**.

Note: You can view the **Added Items**, which you have **added in the DynamoDB** via the **API gateway**.

Items returned (3)

Actions Create item

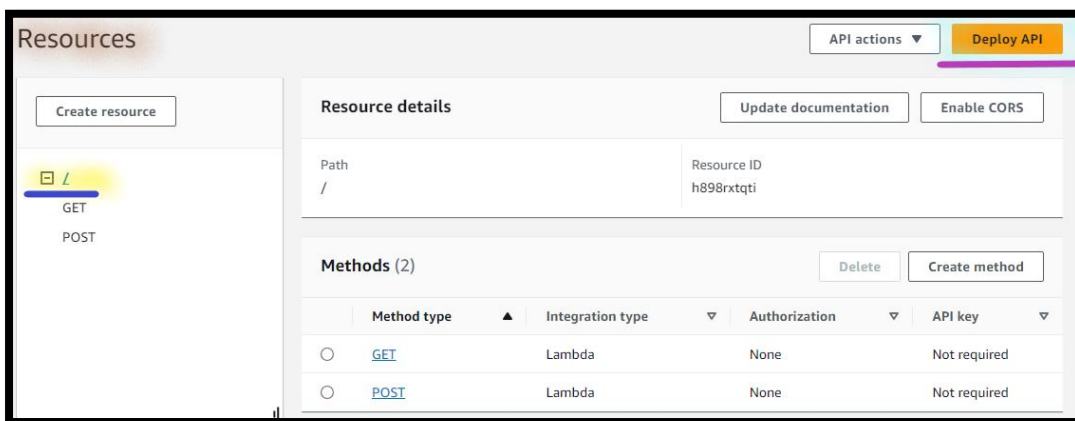
<input type="checkbox"/>	empid (String)	empage	empfirstname	emplastname
<input type="checkbox"/>	001	32	Ajay	Kaushik
<input type="checkbox"/>	002	21	Sana	Yusuf
<input type="checkbox"/>	003	38	Mukesh	Walia

Task 7: Deploy API

In this task, you will publish the API to access via tools.

Step 1: Deploy API

31. In the **AWS Management Console**, on the **Services** menu, search and select **API Gateway**.
32. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
33. Open the **empdata-API** API.
 - a. Select **Resources**.
 - i. Select **/** (resource path).
 - a) Select **Deploy API**.



- 1) **Stage:** Dropdown and select ***New Stage***.
- 2) **Stage name:** Write **ReadWrite-API**.

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Stage

New stage

Stage name

ReadWrite-API

Note: Leave other details as default.

b) Select **Deploy**.

1) **Copy** the **Invoke URL** in the **Notepad**.

Stage details

Stage name: ReadWrite-API

Rate: 10000

Web ACL: -

API cache: Inactive

Burst: 5000

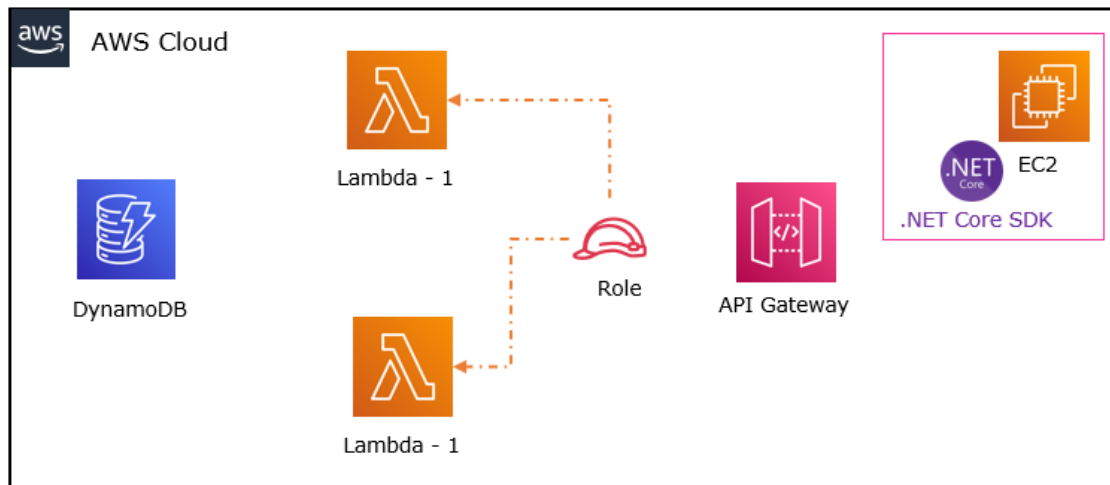
Client certificate: -

Invoke URL: <https://c17ac8jdg.execute-api.us-east-1.amazonaws.com/ReadWrite-API>

Active deployment: 6bw35h on September 24, 2023, 20:50 (UTC+05:30)

Task 8: Build the Environment

In this task, you will build the environment to test the API using the tool.



Step 1: Create EC2 Instance

34. In the **AWS Management Console**, on the **Services** menu, search and select **EC2**.
35. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
36. Select **Instances**.
37. Select **Launch Instances**.
 - a. In the **Name and tags** section:
 - i. **Name**: Write **Dev-API-Server**.
 - b. In the **Application and OS Images** section:
 - i. In the **Search box**:
 - a) Type **Microsoft Windows Server 2022 Base**.
 - b) Press **Enter** key.

Note: You can see the **Choose an Amazon Machine Image** page.

c) From the **Choose an Amazon Machine Image** page:

- 1) Select **Microsoft Windows Server 2022 Base**.

Note: You can see the **Launch an Instance** page.

c. In the **Instance Type** section:

i. **Instance type:** Dropdown and in the **Search box**:

a) Type and select **t2.micro**.

d. In the **Key pair (login)** section:

i. **Key pair name:** Dropdown and select **Proceed without a key pair**.

e. In the **Network setting** section:

Note: You can see "**Allow RDP traffic**" is already **enabled** from "**Anywhere**".

Note: Leave the other details as default.

f. In the **Advance details** section:

i. **User data:** **Copy** the **script** from **script-api-server**.

Note: **script-api-server.txt** script is provided with the **Lab manual**.

Note: User Data **script** performing the **following** tasks:

1. Set the **Administrator's Password**.

- g. In the **Summary** section:
 - i. Select **Launch Instances**.

Note: **Wait**, till you can see the **message "Successfully initiated launch of instance"**.

- h. Select **View all instances**.

Note: **Wait**, till you can see the **Dev API Server** Instance **State** is **Running**.

Note: **Wait**, till you can see the **Dev API Server** Instance **Status check** is **2/2 check passed**.

Note: **Refresh** your **screen** unless you can see the **2/2 check passed**.

Step 2: Copy the IP Address

- 38. **From** the **EC2** console.
- 39. Select the **Dev API Server**.
 - a. Select the **Details**.

Note: **Copy** the **Public IP address** of **Dev API Server** in the **Notepad**.

Step 3: Connect to Instance

- 40. From the **Local Desktop/ Laptop** (*Windows desktop*), right click on **Start** & **Run**.
 - a. In the **Open**, write **mstsc**.
 - b. Select **Ok**.
 - i. **From** the **Remote Desktop Connection**:
 - a) **Computer**: Write the **Public IP Address** of the **Dev API Server**.
 - b) Select **Connect**.

Note: You can **get the prompt** to enter the **Username** and **Password**.

- 1) **Username:** Write **Administrator**.
- 2) **Password:** Write the **lab-password@123**.
- 3) Select **Ok**.

Step 4: Install the Dot Net Core SDK

41. **From** the **Dev API Server**.

- a. **Download** and **install** the **.Net Core SDK** for **Windows x64**.

Note: Use the below **URL** to download the **.Net Core SDK 7.0**.

<https://download.visualstudio.microsoft.com/download/pr/e3f91c3f-dbcc-44cb-a319-9cb15c9b61b9/6c87d96b2294afed74ccf414e7747b5a/dotnet-sdk-7.0.400-win-x64.exe>

Note: **Wait**, till **.NET Core SDK 7.0** install **sucessfully**.

Step 5: Check the .NET Core SDK Version

42. **From** the **Dev API Server**, right click on **Start** & **Run**.

- a. In the **Open**, write **cmd**.
- b. Select **Ok**.
 - i. From the **Command line interpreter**:
 - a) **Execute** the **below command** to **Check** the **dotnet version**.

```
dotnet --version
```

Note: You can see the **Dotnet** installed **version**.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>dotnet --version
7.0.400

C:\Users\Administrator>
```

Step 6: Install the HTTPREPL

43. From the **Dev API Server**, right click on **Start** & **Run**.

a. In the **Open**, write **cmd**.

b. Select **Ok**.

i. From the **Command line interpreter**:

a) **Execute** the **below command** to **Install** the **HTTPREPL**.

```
dotnet tool install -g Microsoft.dotnet-httprepl
```

Note: Wait, till **httprepl** install **successfully**.

```
C:\Users\azureadmin>dotnet tool install -g Microsoft.dotnet-httprepl

Welcome to .NET Core 3.1!
-----
SDK Version: 3.1.301

Telemetry
-----
The .NET Core tools collect usage data in order to help us improve your experience. The data is anonymous. It is coll
ed by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOU
nvironment variable to '1' or 'true' using your favorite shell.

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-docs
Write your first app: https://aka.ms/first-net-core-app
-----
Since you just installed the .NET Core SDK, you will need to reopen the Command Prompt window before running the tool
u installed.
You can invoke the tool using the following command: httprepl
Tool 'microsoft.dotnet-httprepl' (version '3.0.47301') was successfully installed.
```

Info: The HTTP Read-Eval-Print Loop (REPL) is A lightweight, cross-platform command-line tool, used for making HTTP requests to test web APIs and view their results.

- b) **Execute** the **below command** to **Close** the **Command line interpreter**.

exit

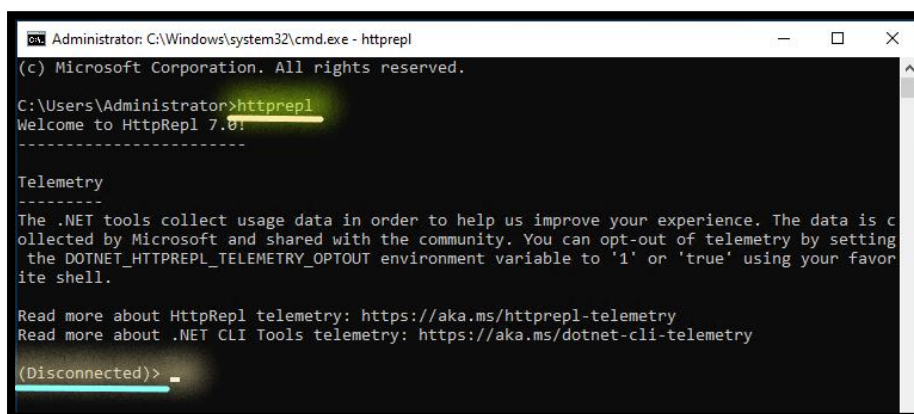
Step 7: Verify the HTTPREPL

44. From the **Dev API Server**, right click on **Start** & **Run**.

- a. In the **Open**, write **cmd**.
- b. Select **Ok**.
 - i. From the **command line interpreter**:
 - a) **Execute** the **below command** to **Initiate** the **HTTPREPL**.

httprepl

Note: You can see the **disconnected**.



```
Administrator: C:\Windows\system32\cmd.exe - httprepl
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>httprepl
Welcome to HttpRepl 7.0!
-----
Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. The data is c
ollected by Microsoft and shared with the community. You can opt-out of telemetry by setting
the DOTNET_HTTPREPL_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favor
ite shell.

Read more about HttpRepl telemetry: https://aka.ms/httprepl-telemetry
Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

(Disconnected)> _
```

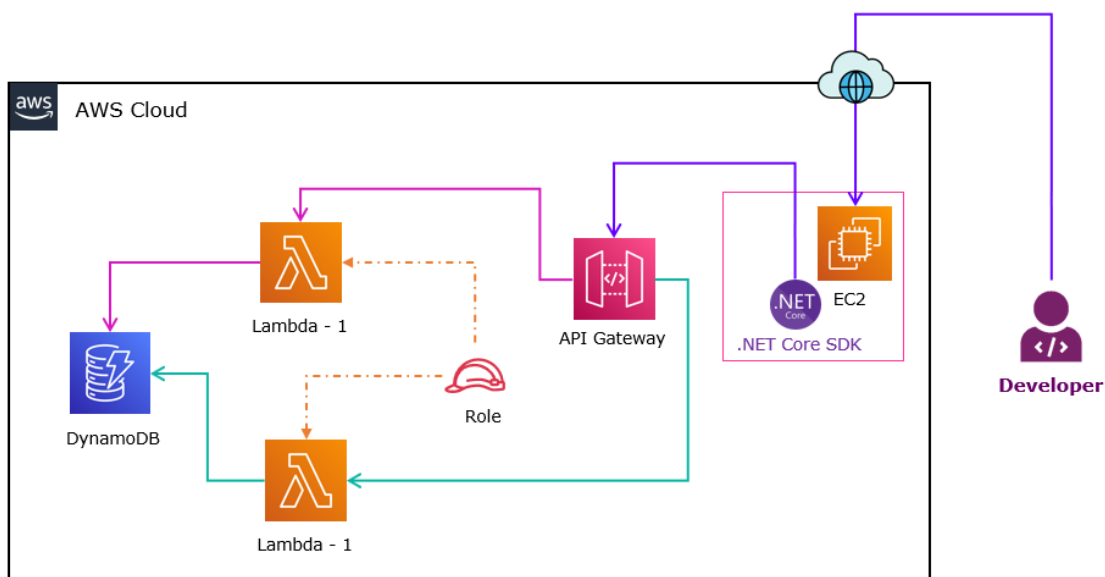

b) From the **httprepl**:

- 1) **Execute** the **below command** to **Close** the **httprepl**.

```
exit
```

Task 9: Validate the Solution using Httprepl

In this task, you will access the API using the tools.



Step 1: Test API by using Httprepl

45. From the **Dev API Server**, right click on **Start** & **Run**.

- a. In the **Open**, write **cmd**.

- b. Select **Ok**.

- i. From the **Command line interpreter**:

Note: **Set** the base **Uniform Resource Identifier** (URI) to the value of the Request URL for the API operation.

- a) **Execute** the **below command** to **Invoke** the **API**.

```
httprepl API-INVOKE-URL
```

Note: **Replace** the **API-INVOKE-URL**, with the **empdata-API API Invoke URL** which you have copied in the previous step.

```
C:\Users\Administrator>httprepl https://ddllbetmbf.execute-api.us-east-1.amazonaws.com/ReadWrite-API
(Disconnected)> connect https://ddllbetmbf.execute-api.us-east-1.amazonaws.com/ReadWrite-API
Using a base address of https://ddllbetmbf.execute-api.us-east-1.amazonaws.com/ReadWrite-API/
Unable to find an OpenAPI description
For detailed tool info, see https://aka.ms/http-repl-doc
https://ddllbetmbf.execute-api.us-east-1.amazonaws.com/ReadWrite-API/> _
```

- b) From the **httprepl**:

- 1) **Execute** the **below command** to **Get** the **Items**.

```
get
```

Note: **Observe** the **JSON response** content.

Note: In the **Response body**, you can see the **Items**, which you have added in the DynamoDB table.

```
https://ddllbetmbf.execute-api.us-east-1.amazonaws.com/ReadWrite-API/> get
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 227
Content-Type: application/json
Date: Sat, 06 Feb 2021 18:57:02 GMT
x-amz-apigw-id: aVjzvFWvoAMFaw=
x-amzn-RequestId: 77b2387c-03e5-487b-8ed6-b22412831afe
X-Amzn-Trace-Id: Root=1-601ee67e-30764ac444b6ffdb34051a3d;Sampled=0

[
  {
    "empfirstname": "Ajay",
    "emplastname": "Kaushik",
    "empage": "32",
    "empid": "001"
  },
  {
    "empfirstname": "Mukesh",
    "emplastname": "Walia",
    "empage": "38",
    "empid": "003"
  },
  {
    "empfirstname": "Sana",
    "emplastname": "Yusuf",
    "empage": "21",
    "empid": "002"
  }
]
```

- 2) **Execute** the **below command** to **Set** the **Default text editor**.

```
pref set editor.command.default C:\Windows\system32\notepad.exe
```

Info: By default, the HttpRepl has no text editor configured for use.

To test web API methods requiring an HTTP request body, a default text editor must be set.

The HttpRepl tool launches the configured text editor for the sole purpose of composing the request body.

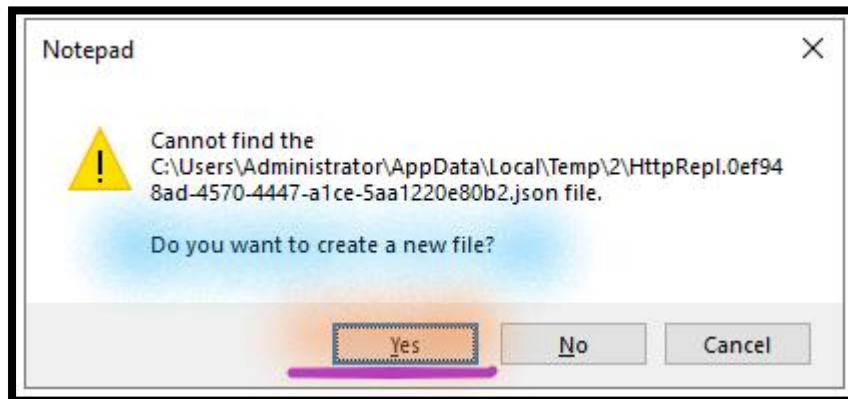
```
ReadWrite-API/> pref set editor.command.default C:\Windows\system32\notepad.exe  
ReadWrite-API/> _
```

- 3) **Execute** the **below command** to **Set** the **Media type (JSON)**.

```
post -h Content-Type=application/json
```

Info: In the preceding command, the HTTP request header is set to indicate a request body media type of JSON. The default text editor opens a *.tmp* file.

- I. Select **Yes**, when you **get prompt** to **Create new file**.



- II. In the **Body**, **Copy** the **below details**:

```
{  
  "empid": "004",  
  "empfirstname": "Aisha",  
  "emplastname": "Khan",  
  "empage": "45"  
}
```



- A. **From** the **Notepad**:
- Select **File** and select **Save**.
 - Select **File** and select **Exit**.

Note: You can see the **200 OK** message.

```
https://d1llbetmbf.execute-api.us-east-1.amazonaws.com/ppp/> post -h Content-Type=application/json
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 2
Content-Type: application/json
Date: Sat, 06 Feb 2021 13:44:58 GMT
x-amz-apigw-id: aU2F-E7QIAMF_cQ=
x-amzn-RequestId: 30809f30-b9ce-4789-a129-ca68393a8055
X-Amzn-Trace-Id: Root=1-601e9d59-5e48077a2852591727e45671;Sampled=0

{
}

https://d1llbetmbf.execute-api.us-east-1.amazonaws.com/ppp/>
```

- 4) **Execute** the **below command** to **Get** the **Items**.

```
get
```

Note: Observe the **JSON response** content.

Note: In the **Response body**, you can see the **New Item**, which you have added in the Previous step.

- 5) **Execute** the **below command** to **Exit** the **httprepl**.

```
exit
```

Step 2: View the DynamoDB Data

46. In the **AWS Management Console**, on the **Services** menu, search and select **DynamoDB**.
47. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
48. Select **Explore items**.
 - a. Select **empdata**.
 - i. Select **Refresh**.

Note: You can view the **Added Items**, which you have added in the **DynamoDB** via the **Invoke URL**.

Items returned (4)

Actions ▾

Create item

Find items

< 1 >

<input type="checkbox"/>	empid ▾	empage ▾	empfirst... ▾	emplastname ▾
<input type="checkbox"/>	001	32	Ajay	Kaushik
<input type="checkbox"/>	003	38	Mukesh	Walia
<input type="checkbox"/>	002	21	Sana	Yusuf
<input type="checkbox"/>	004	45	Aisha	Khan

Task 10: Delete the Environment

Step 1: Delete the DynamoDB Table

49. In the **AWS Management Console**, on the **Services** menu, search and select **DynamoDB**.
50. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
51. Select the **Tables**.
 - a. Select the **emdpdata**.
 - i. Select **Delete**.
 - a) When you **get prompt**, write **confirm**.
 - b) Select **Delete**.

Step 2: Delete Lambda Function

52. In the **AWS Management Console**, on the **Services** menu, search and select **Lambda**.
53. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
54. Select **Functions**.
 - a. Select **ReadItems**.
 - b. Select **WriteItems**.

- i. Select **Actions**.
 - a) Select **Delete**.
 - 1) When you **get prompt**, write **delete**.
 - 2) Select **Delete**.
 - b) Select **Close**.

Step 3: Delete the API Gateway

- 55. In the **AWS Management Console**, on the **Services** menu, search and select **API Gateway**.
- 56. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
- 57. Select the **APIs**.
- 58. Select the **empdata-API**.
 - a. Select **Delete**
 - i. When you **get prompt**, write **confirm**.
 - ii. Select **Delete**.

Step 4: Terminate EC2 Instances

- 59. In the **AWS Management Console**, on the **Services** menu, search and select **EC2**.
- 60. Choose the **YOUR ALLOCATED REGION** list to the right of your account information on the navigation bar.
- 61. Select **Instances**.
 - a. Select **Dev API Server**.
 - i. Select **Instance state**.
 - a) Select **Terminate instance**.
 - 1) Select **Terminate**.