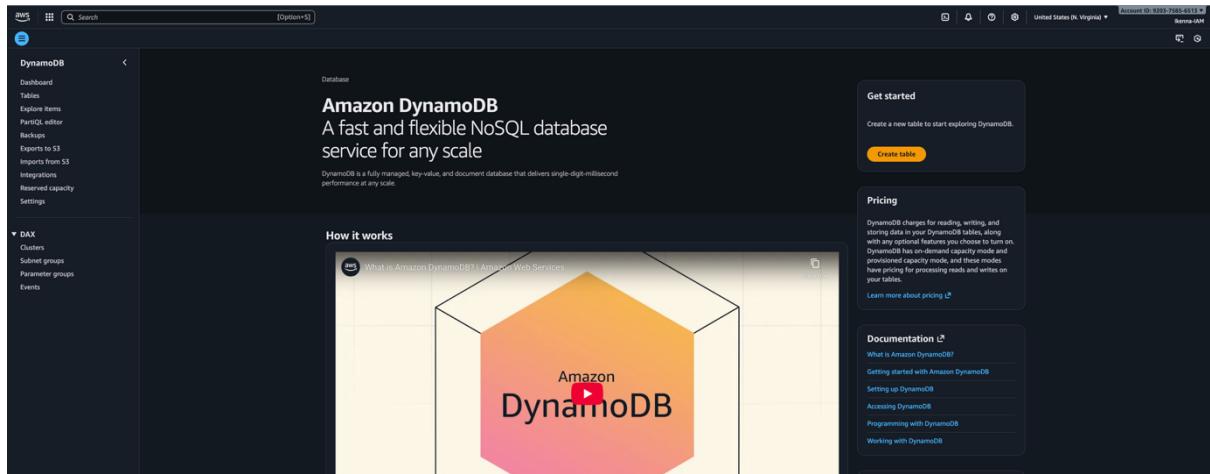


ANASIEZE IKENNA – CLOUD ENGINEER.
SERVERLESS: SOLVING PERFORMANCE ISSUES WITH API GATEWAY, LAMBDA AND DYNAMODB

TASK 1 - CREATE THE DYNAMODB TABLE



TASK 2 - CONFIGURE THE TABLE

- ⇒ **TABLE NAME: RENTAL_APP**
- ⇒ **PARTITION KEY: RECORD_TYPE**
- ⇒ **SORT KEY: ID**

A screenshot of the Amazon DynamoDB Tables page. At the top, a green success message says "The rental_app table was created successfully." Below this, the "Tables (1)" section is shown. The table "rental_app" is listed with the following details:

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite
rental_app	Active	record_type (\$)	id (\$)	0	0	Off	☆

At the bottom of the page, there are links for CloudShell, Feedback, Copyright notice (© 2025, Amazon Web Services, Inc. or its affiliates.), Privacy, Terms, and Cookie preferences.

TASK 3 - CREATE THE LAMBDA FUNCTION (PYTHON), CONFIGURE FUNCTION

BASICS

- ⇒ **AUTHOR FROM SCRATCH**

⇒ **FUNCTION NAME: LAMBDAFUNCTION**

⇒ **RUNTIME: PYTHON 3.14**

⇒ **ARCHI: x86 64**

⇒ **PERMISSIONS**

Lambda > Functions > lambdafunction

lambdafunction

Function overview Info

Diagram | Template

lambdafunction
Layers (0)

+ Add trigger | + Add destination

Description
-
Last modified 28 seconds ago
Function ARN arn:aws:lambda:us-east-1:154280210964:function:lambdafunction
Function URL | Info

Code Test Monitor Configuration Aliases Versions

Code source Info

CloudShell Feedback

lambda_function.py

```
1 # AWS SDK for Python
2
3 import json
4 import os
5 import boto3
6 import uuid
7 from datetime import datetime
8
9 # Initialize DynamoDB client
10 dynamodb = boto3.resource('dynamodb') Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)
11 table = dynamodb.Table(os.environ['TABLE_NAME'])
12
13 def lambda_handler(event, context):
14     """
15     Lambda handler for processing API Gateway requests
16     """
17     http_method = event['httpMethod']
18     path = event['path']
19
20     # Route the request based on path and method
21     if path == '/vehicles':
22         if http_method == 'GET':
23             return list_vehicles()
24         elif http_method == 'POST':
25             return create_vehicle(json.loads(event['body'])) if 'body' in event else {}
26         elif http_method == 'PUT':
27             return update_vehicle(json.loads(event['body'])) if 'body' in event else {}
28     elif path == '/locations':
29         if http_method == 'GET':
30             return list_locations()
31         elif http_method == 'POST':
32             return create_location(json.loads(event['body'])) if 'body' in event else {}
33     elif http_method == 'PUT':
```

Open in Visual Studio Code | Upload from

EXPLORER LAMBDAFUNCTION lambda_function.py

DEPLOY Current Deploy (F5) Test (Shift+F5)

TEST EVENTS [NONE SELECTED] Create new test event

ENVIRONMENT VARIABLES

✓ Lambda Deployed 0 △ 0 ▷ Amazon Q

Ln 10, Col 38 Spaces: 4 UTF-8 LF Python Lambda Layout: U.S.

TASK 4 – TEST LAMBDA LOGIC

The screenshot shows the AWS Lambda Test console interface. On the left, the EXPLORER panel displays a file named `lambda_function.py`. In the center, the code editor shows the function's logic, including a `lambda_handler` function that handles various HTTP methods for vehicle and location management. Below the code editor are tabs for PROBLEMS, OUTPUT, CODE REFERENCE LOG, and TERMINAL. The OUTPUT tab shows a successful test event named `create_location` with a response object containing a status code of 201 and a body representing a new location entry. The CODE REFERENCE LOG tab shows detailed logs of the function's execution, including Request and Response IDs and memory usage. The right side of the interface features a "Create new test event" dialog box where a test event name like `create_location` can be specified.

The screenshot shows the AWS DynamoDB Explore items interface. On the left, the sidebar includes sections for **DynamoDB** (Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, Settings) and **DAX** (Clusters, Subnet groups, Parameter groups, Events). The main area displays a table named `rental_app` with one item. The table schema is shown in a table header:

	record_type (String)	id (String)	createdAt (String)	name (String)	vehicles_available (Number)
location	c589a878-f	2025-12-1...	ikenna	4	

TASK 5 – CREATE AND CONFIGURE API GATEWAY (HTTP API)

The screenshot shows the AWS API Gateway console. On the left, the navigation bar includes 'API Gateway > APIs > Resources - rental_app (afwoskczf3)'. The main area displays a test result for a GET request to '/locations'. The response details are as follows:

- Request:** /locations
- Latency ms:** 276
- Status:** 200
- Response body:**

```
[{"createdAt": "2025-12-12T19:45:13.730582", "vehicles_available": "4", "name": "Ikenna", "record_type": "location", "id": "40fb3bb7-2"}]
```
- Response headers:**

```
{
  "X-Amzn-Trace-Id": "Root=1-693c7136-438fdce83133f0a57da27ad;Parent=2e5e89034890ab26;Sampled=0;Lineage=1:50c46fc5:0"
}
```
- Logs:**

```
Execution log for request cd95fd8f-cc08-4418-a02d-edfe321416a8
Fri Dec 12 19:47:02 UTC 2025 : Starting execution for request: cd95fd8f-cc08-4418-a02d-edfe321416a8
Fri Dec 12 19:47:02 UTC 2025 : HTTP Method: GET, Resource Path: /locations
Fri Dec 12 19:47:02 UTC 2025 : Method request path: {}
Fri Dec 12 19:47:02 UTC 2025 : Method request query string: {}
Fri Dec 12 19:47:02 UTC 2025 : Method request headers: {Content-Type=application/json}
Fri Dec 12 19:47:02 UTC 2025 : Method request body before transformations:
Fri Dec 12 19:47:02 UTC 2025 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:154280210964:function:lambdafunction:invocations
Fri Dec 12 19:47:02 UTC 2025 : Endpoint request headers: {X-Amz-Date=20251212T194702Z, X-Amzn-ApiGateway-api-id=afwoskczf3, Accept=application/json, User-Agent=AmazonAPIGateway_afwoskczf3, Host=lambda.us-east-1.amazonaws.com, X-Amz-Content-Sha256=e8cf8c05f821f654d331b25cf5a747f6cca7b44a10836136cd9058048332022da, X-Amzn-
```

The screenshot shows the AWS API Gateway console. The left sidebar includes 'APIs', 'Custom domain names', 'Domain name access associations', 'VPC links', 'AgentCore targets', and sections for 'API: rental_app' with 'Resources', 'Stages', 'Authorizers', 'Gateway responses', 'Models', 'Resource policy', 'Documentation', 'Dashboard', and 'API settings'. The main area displays a test result for a POST request to '/locations'. The response details are as follows:

- Request:** /locations
- Latency ms:** 1255
- Status:** 201
- Response body:**

```
{"id": "ab4f3069-4", "record_type": "location", "createdAt": "2025-12-12T19:53:11.552171", "name": "Mercedes Benz", "vehicles_available": "6"}
```
- Response headers:**

```
{
  "X-Amzn-Trace-Id": "Root=1-693c72a6-8981ca3af5499441c136da55;Parent=52560ed98c9ecb1b;Sampled=0;Lineage=1:50c46fc5:0"
}
```
- Logs:**

```
Execution log for request 9f4d33d4-d742-4f52-85ee-b346451e1a99
Fri Dec 12 19:53:10 UTC 2025 : Starting execution for request: 9f4d33d4-d742-4f52-85ee-b346451e1a99
Fri Dec 12 19:53:10 UTC 2025 : HTTP Method: POST, Resource Path: /locations
Fri Dec 12 19:53:10 UTC 2025 : Method request path: {}
Fri Dec 12 19:53:10 UTC 2025 : Method request query string: {}
Fri Dec 12 19:53:10 UTC 2025 : Method request headers: {Content-Type=application/json}
Fri Dec 12 19:53:10 UTC 2025 : Method request body before transformations: {
```

TASK 6 - VERIFY DYNAMODB DATA

DynamoDB > Explore items > rental_app

Filter by tag value: Any tag value

Scan Query

Select a table or index: Table - rental_app

Select attribute projection: All attributes

Filters - optional: rental_app

Run Reset

Completed · Items returned: 2 · Items scanned: 2 · Efficiency: 100% · RCU consumed: 2

Table: rental_app - Items returned (2)

Scan started on December 12, 2025, 16:23:44

record_type	id	createdAt	name	vehicles_available
location	40fb3bb7-2	2025-12-12T19:45:13.730...	Ikenna	4
location	ab4f3069-4	2025-12-12T19:53:11.552...	Mercedes Benz	6

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

TASK 7 - MONITOR LOGS AND METRICS