PROJECT ON API Gateway-LAMBDA and DYNAMODB INTEGRATION

🌐 OVERVIEW

□💻 User Journey:

1. 📝 **User submits a form** on a website.
2. 📡 **Frontend sends a `POST` request** to **API Gateway**.
3. ⚡ **API Gateway triggers a Lambda function**.

□ Lambda Responsibilities:

- ✅ **Validates** input data.
- 🗄 **Saves** user data to **DynamoDB**.
- 📤 **Returns** a **success** or **error** message.

🔁 Final Step:

🔁 **API Gateway sends the Lambda response back** to the client (frontend).

🔧 Architecture Components

◦O☼ 1. AWS Lambda

□ *What it is:*
A compute service that lets you run backend code **without managing servers**.

⬥ **Write functions** in Node.js, Python, Java, etc.
⬥ **Triggered by events** such as:

- HTTP requests
- DynamoDB changes
- S3 uploads

🌐 2. Amazon API Gateway

□ *What it is:*
A fully managed service to **create, publish, and manage RESTful or WebSocket APIs**.

⬥ **Acts as a front door** for your application to access backend services.
⬥ **Converts HTTP requests into Lambda function invocations**.

🗄□ 3. Amazon DynamoDB

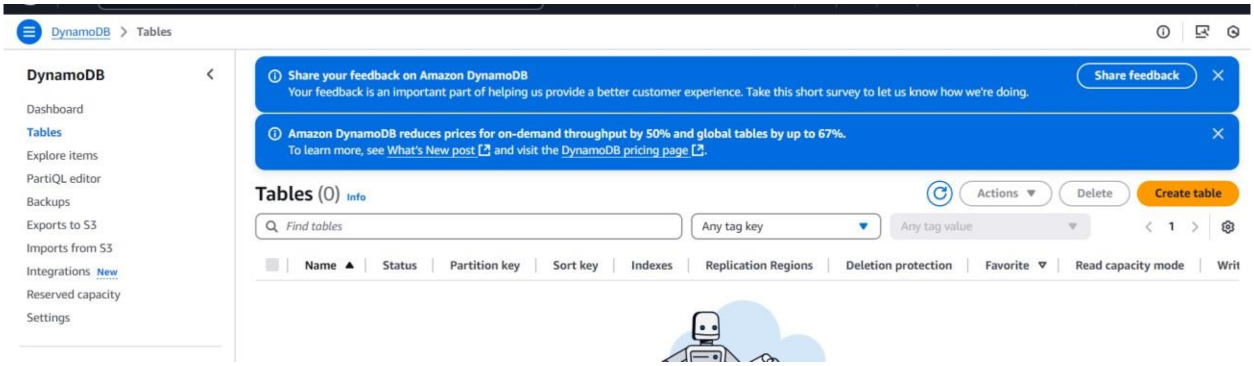
□ *What it is:*
A **fully managed NoSQL database** service.

⬥ **Fast**, **scalable**, and **serverless**
⬥ Ideal for **key-value** or **document-based** storage

-- process start --

- ■ Open Dynamodb

- Click on create table



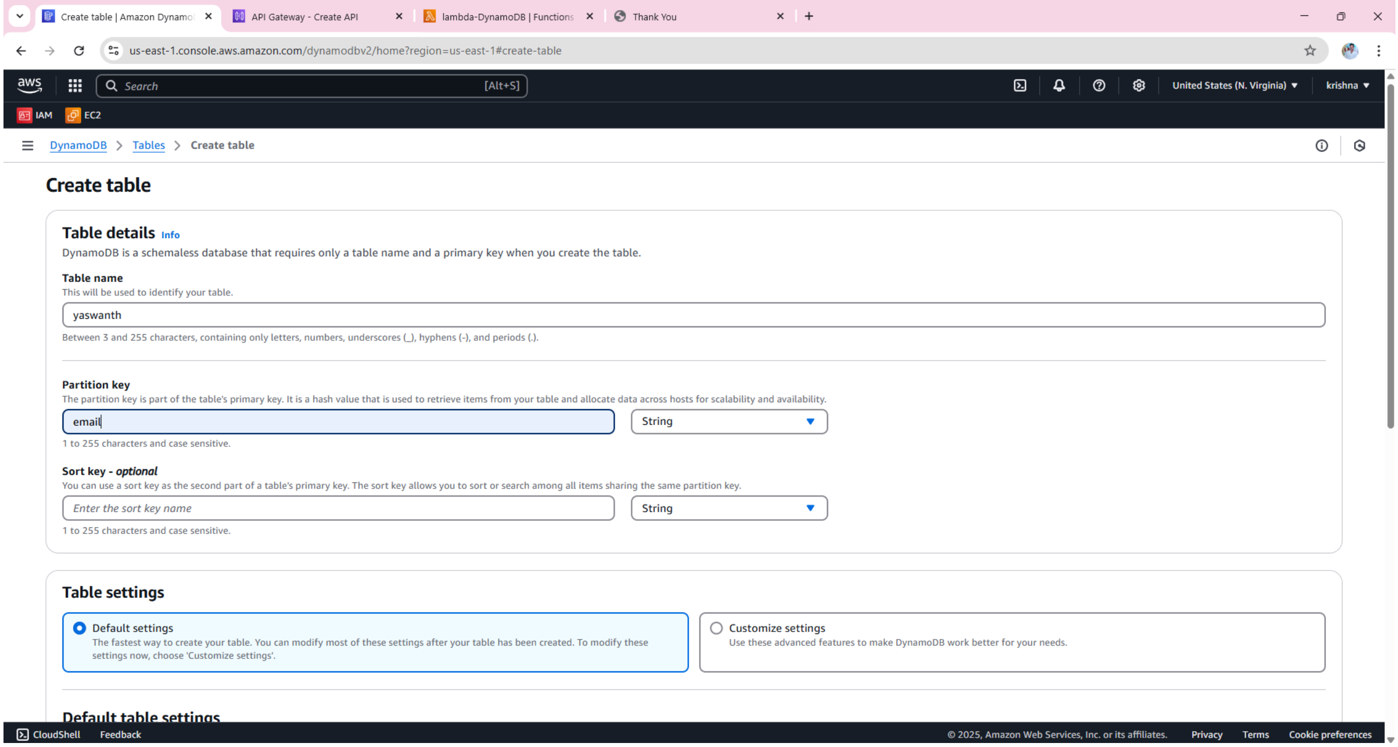
Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Amazon DynamoDB reduces prices for on-demand throughput by 50% and global tables by up to 67%.
To learn more, see What's New post and visit the DynamoDB pricing page.

Tables (0) Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Writ

- Give table name as yaswanth
- Partition key as a email
- If you change table name you need to change in lambda function


Create table | Amazon Dynamo
API Gateway - Create API
lambda-DynamoDB | Functions
Thank You
us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table
aws
Search
[Alt+S]
United States (N. Virginia)
krishna
IAM
EC2
DynamoDB > Tables > Create table
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create table

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.
yaswanth
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.
email
String
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name
String
1 to 255 characters and case sensitive.

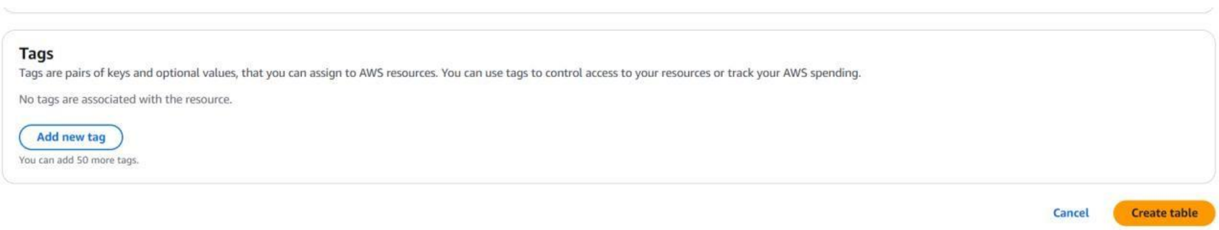
Table settings

● Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

○ Customize settings
Use these advanced features to make DynamoDB work better for your needs.

Default table settings

- Click on create

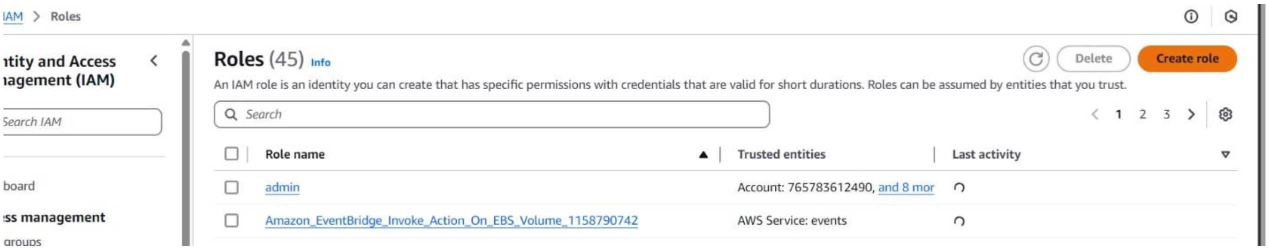


Tags
Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

You can add 50 more tags.

- Open IAM
- Click on create role



IAM > Roles

ntity and Access
nagement (IAM)

Search IAM

board

ss management

groups

Roles (45) 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.

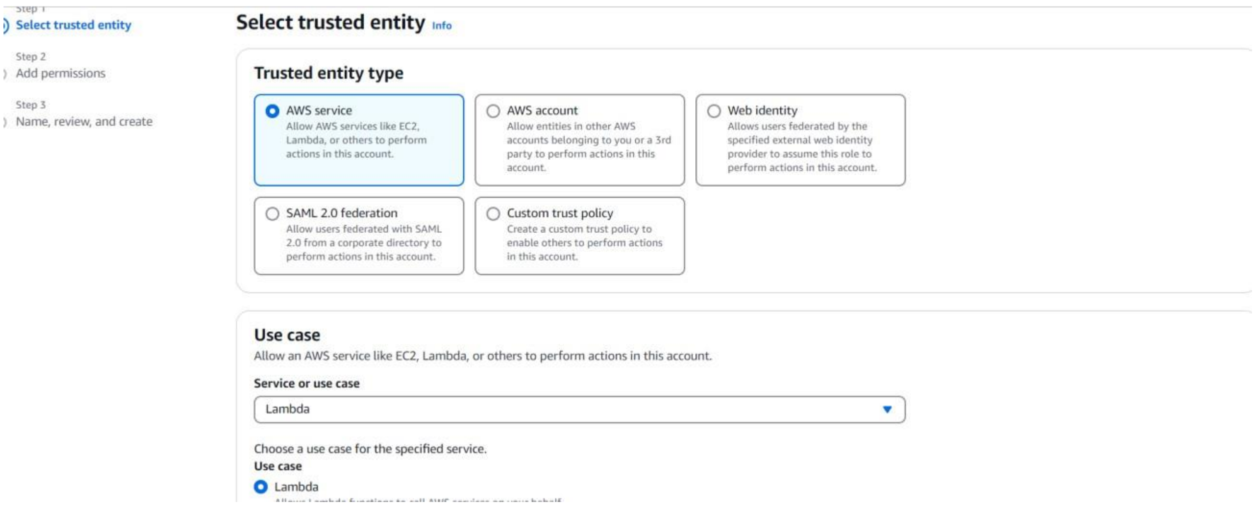
Search

< 1 2 3 >

Delete Create role

□	Role name	Trusted entities	Last activity
□	admin	Account: 765783612490, and 8 mor	
□	Amazon_EventBridge_Invoke_Action_On_EBS_Volume_1158790742	AWS Service: events	

- Service is select lambda



Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

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

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role 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.

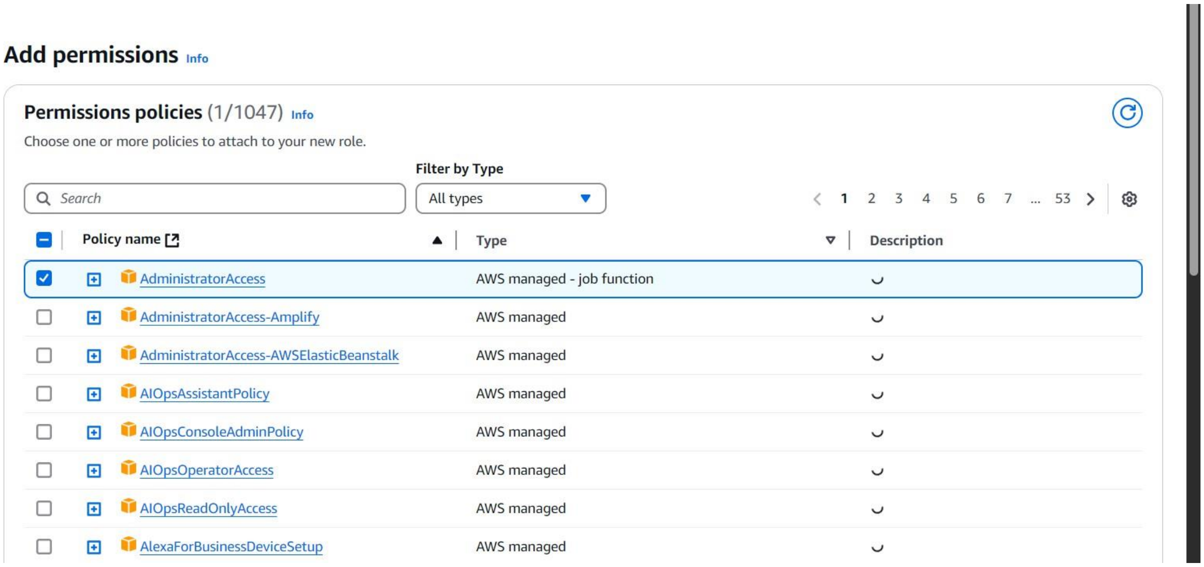
○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

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

Service or use case
Lambda

Choose a use case for the specified service.
Use case
● Lambda
Allows Lambda functions to call AWS services on your behalf.

- Add admin access
- Click on next



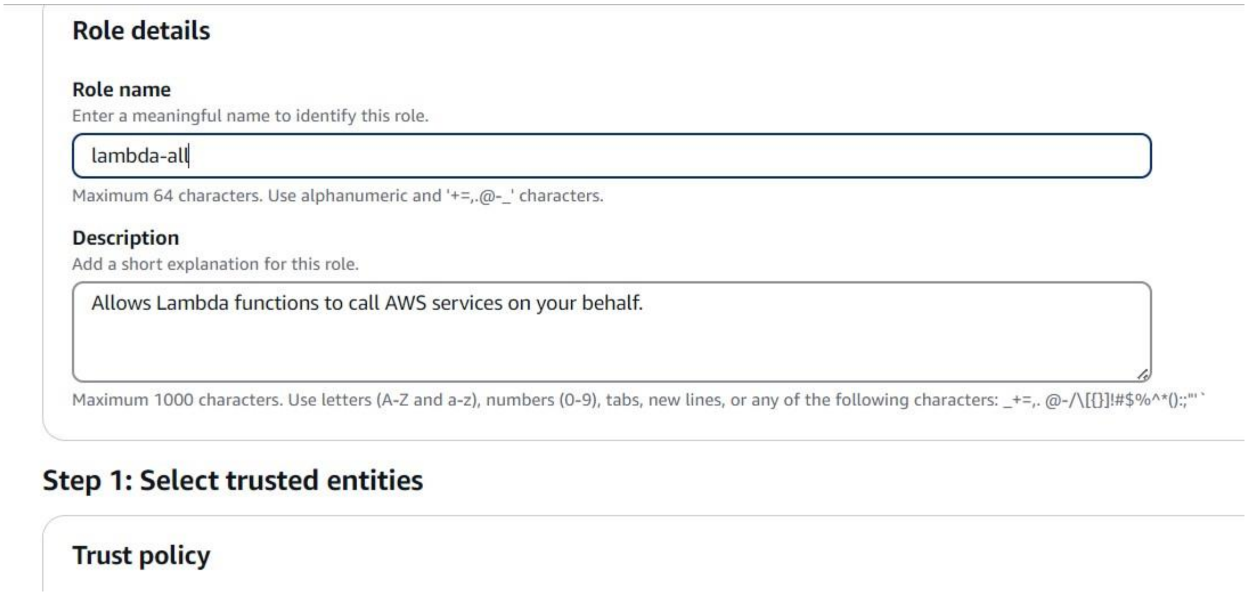
Add permissions Info

Permissions policies (1/1047) Info
Choose one or more policies to attach to your new role.

Search | Filter by Type: All types | < 1 2 3 4 5 6 7 ... 53 >

	Policy name	Type	Description
☑	AdministratorAccess	AWS managed - job function	
□	AdministratorAccess-Amplify	AWS managed	
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	
□	AIOpsAssistantPolicy	AWS managed	
□	AIOpsConsoleAdminPolicy	AWS managed	
□	AIOpsOperatorAccess	AWS managed	
□	AIOpsReadOnlyAccess	AWS managed	
□	AlexaForBusinessDeviceSetup	AWS managed	

- Give the name to role
- Click on create



Role details

Role name

Enter a meaningful name to identify this role.

lambda-all

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

Description

Add a short explanation for this role.

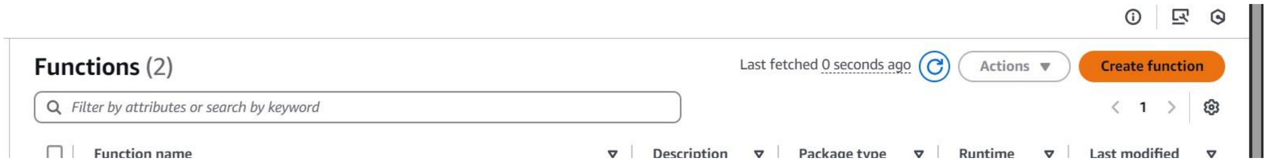
Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*():;"'`

Step 1: Select trusted entities

Trust policy

- Open lambda
- Click on create function

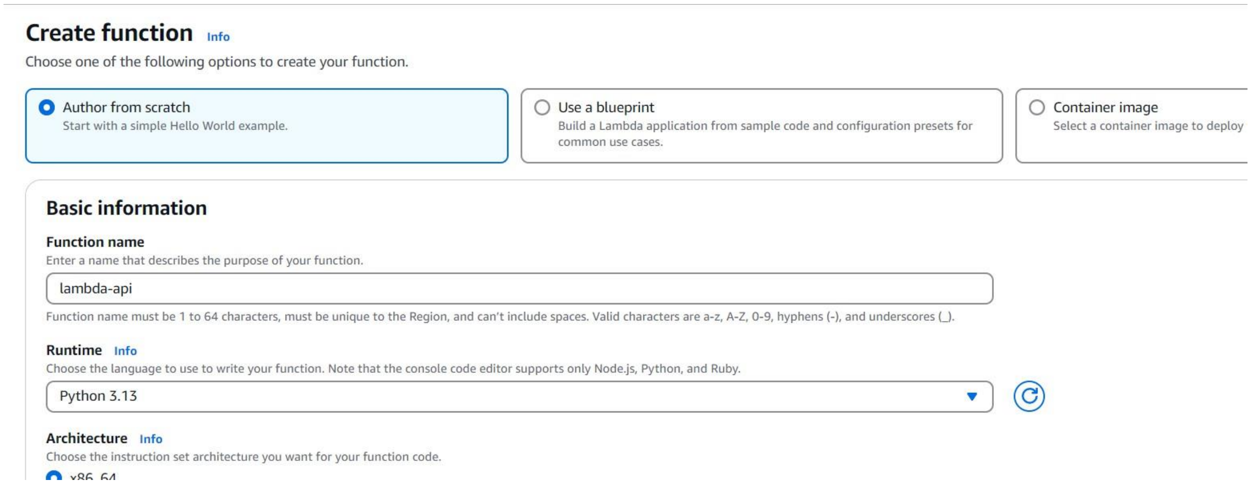


Functions (2)

Last fetched 0 seconds ago

Actions ▼
Create function
Filter by attributes or search by keyword
< 1 >
□ Function name ▽ Description ▽ Package type ▽ Runtime ▽ Last modified ▽

- Give name
- Select the run time as python



Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy

Basic information

Function name

Enter a name that describes the purpose of your function.

lambda-api

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 Info

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

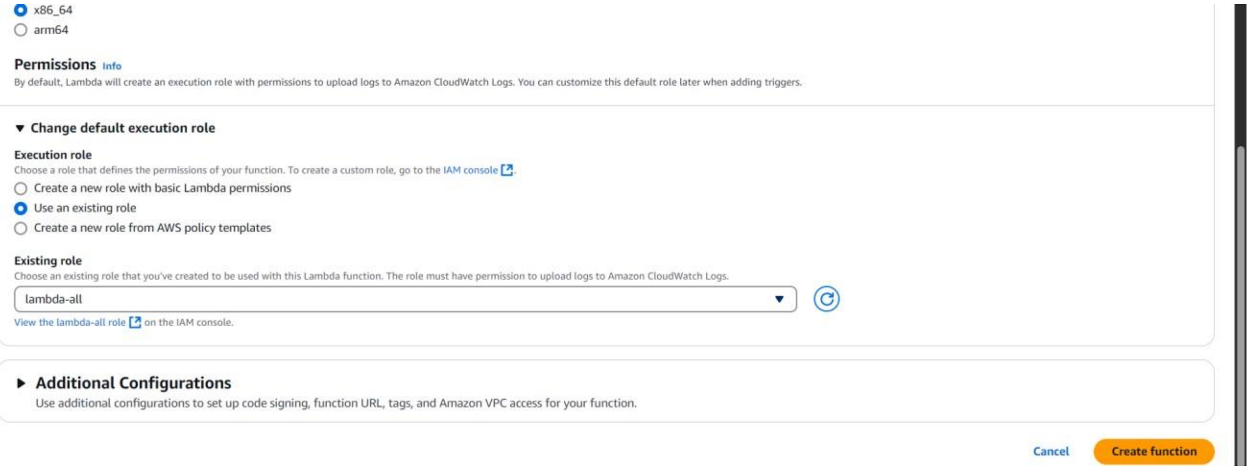
Python 3.13

Architecture Info

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

● x86_64

- Click on change default execution role
- Select use an existing role
- Select previously created role
- Click on create function


● 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.

▼ 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.

lambda-all

View the lambda-all role on the IAM console.

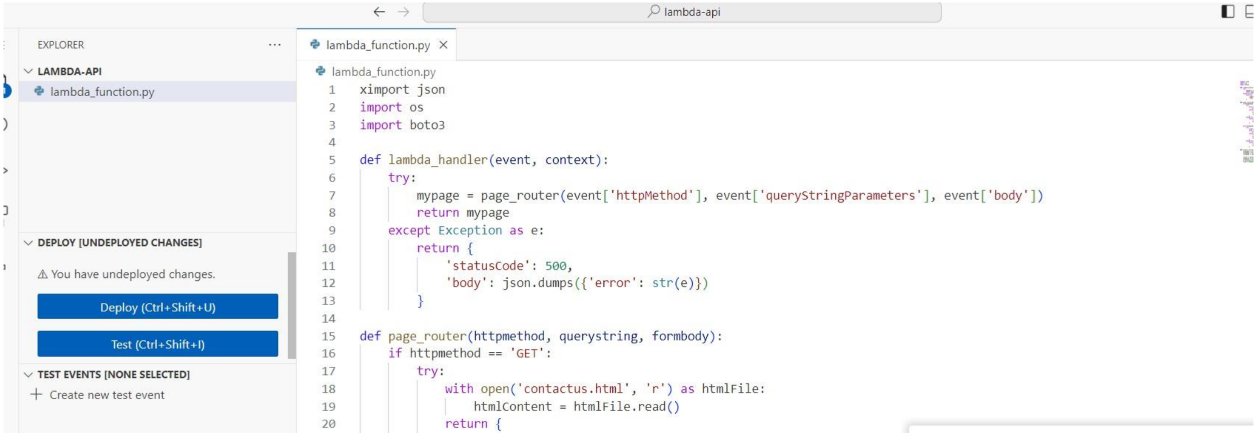
▶ **Additional Configurations**

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

- In lambda_function.py file paste below code

  https://github.com/arumullayaswanth/AWS-Lambda-boto3-automation-project/blob/master/6.lambda-dynamodb-api-gateway/lambda_function.py

- Code will be on above git link copy and paste it

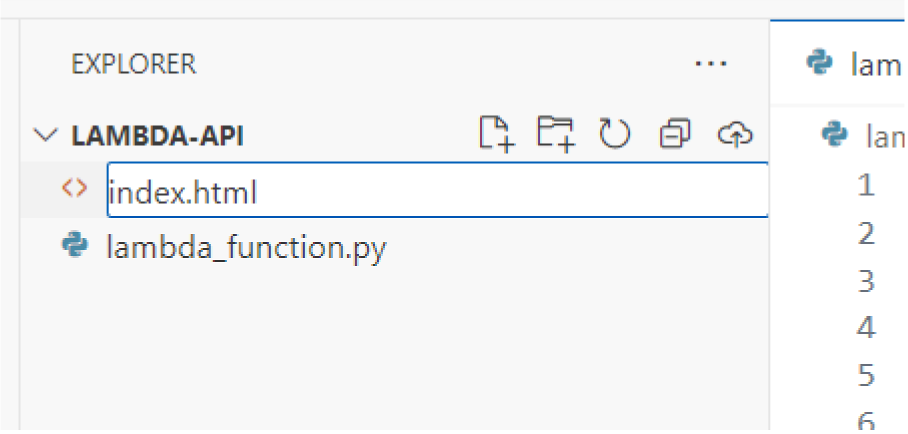


```
ximport json
import os
import boto3

def lambda_handler(event, context):
    try:
        mypage = page_router(event['httpMethod'], event['queryStringParameters'], event['body'])
        return mypage
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({'error': str(e)})
        }

def page_router(httpmethod, querystring, formbody):
    if httpmethod == 'GET':
        try:
            with open('contactus.html', 'r') as htmlFile:
                htmlContent = htmlFile.read()
            return {
```

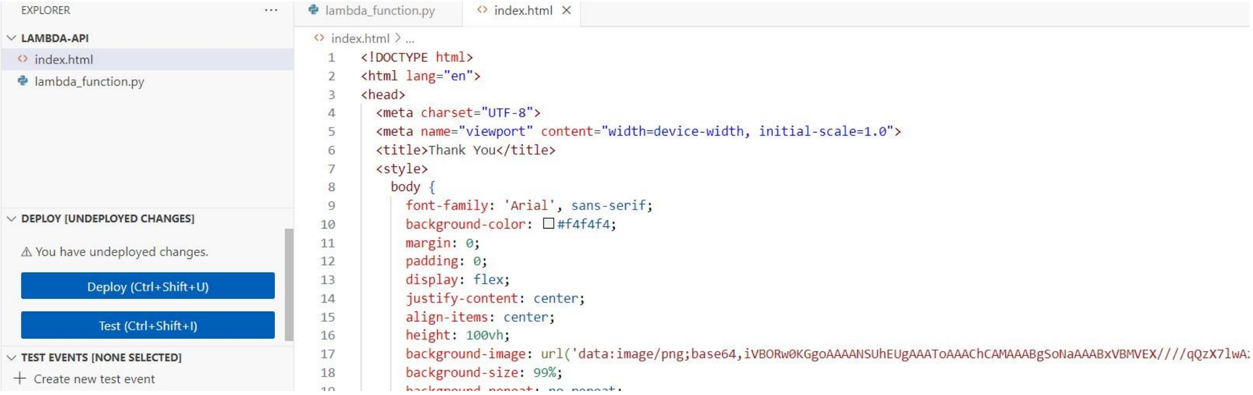
- Create a new file name as index.html
- Paste the below code

  https://github.com/arumullayaswanth/AWS-Lambda-boto3-automation-project/blob/master/6.lambda-dynamodb-api-gateway/index.html

- Code will be on above git link copy and paste it



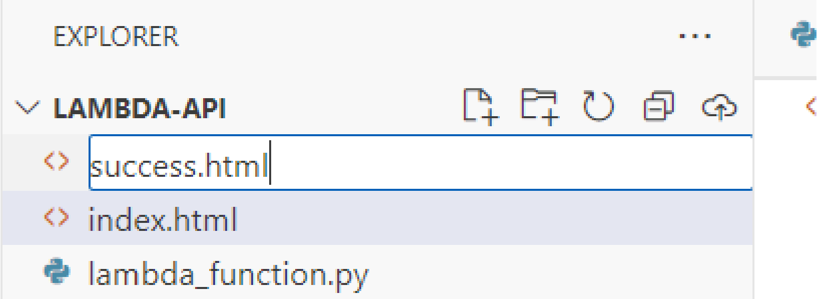
- Like this



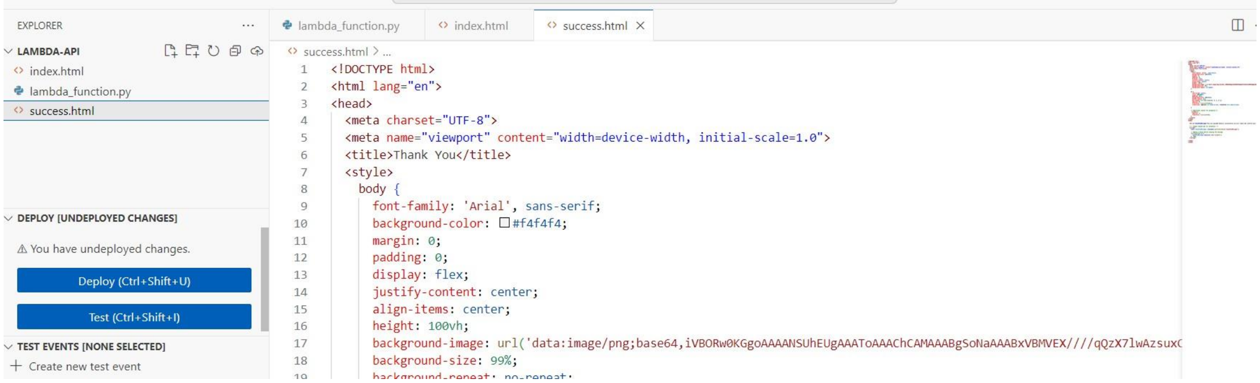
⚠ You have undeployed changes.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Thank You</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      background-color: □#f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-image: url('data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAToAAAChCAMAAABgSoNaAAABxVBMVEX////qQzX7lwA
      background-size: 99%;
      background-repeat: no-repeat;
```

- Create a new file name as success.html
- Paste the below code https://github.com/arumullayaswanth/AWS-Lambda-boto3-automation-project/blob/master/6.lambda-dynamodb-api-gateway/success.html
- Code will be on above git link copy and paste it



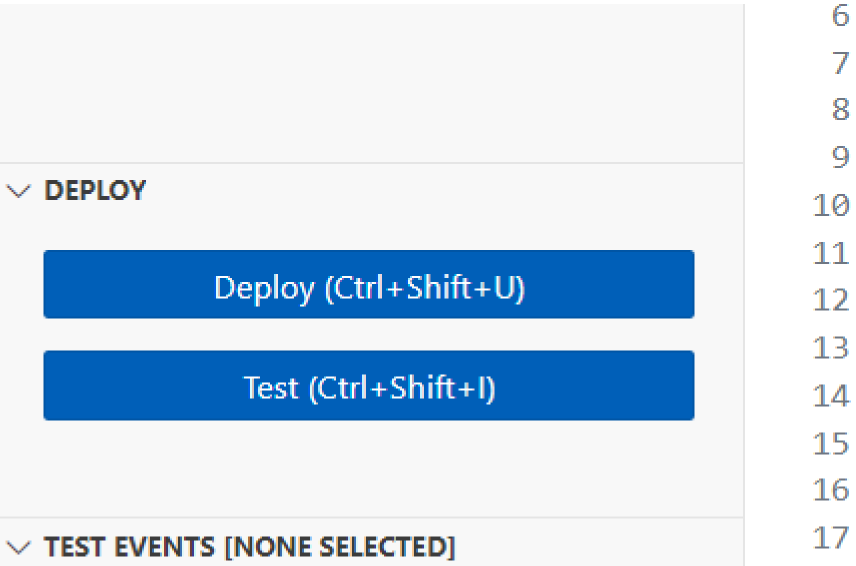
- Like this



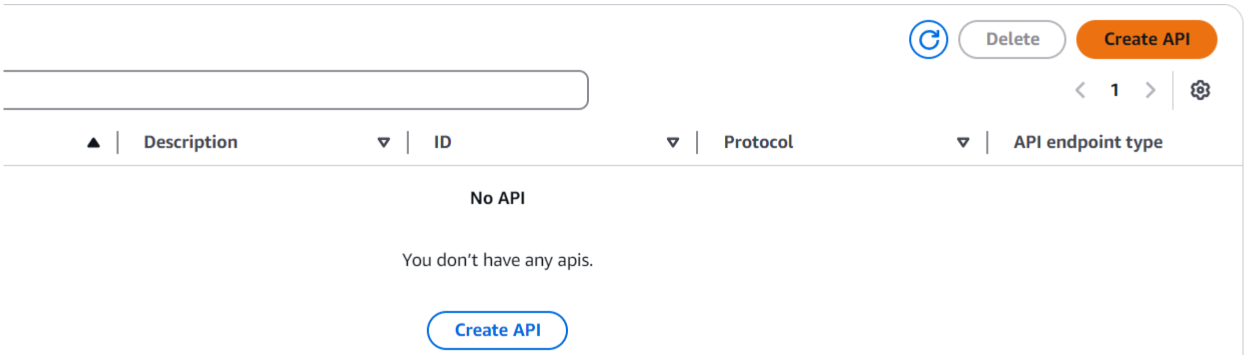
⚠ You have undeployed changes.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Thank You</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      background-color: □#f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-image: url('data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAToAAAChCAMAAABgSoNaAAABxVBMVEX////qQzX7lwAzsux
      background-size: 99%;
      background-repeat: no-repeat;
```

- Click on deploy



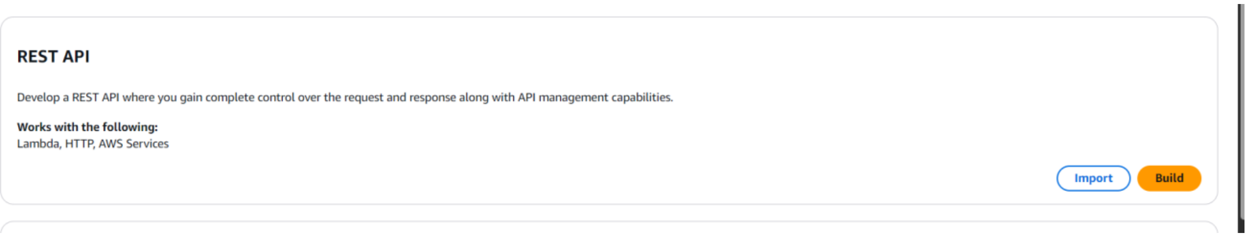
- Open api gateway page
- Click on create api



No API

You don't have any apis.

- In apis list build the rest api
- Click build on rest api

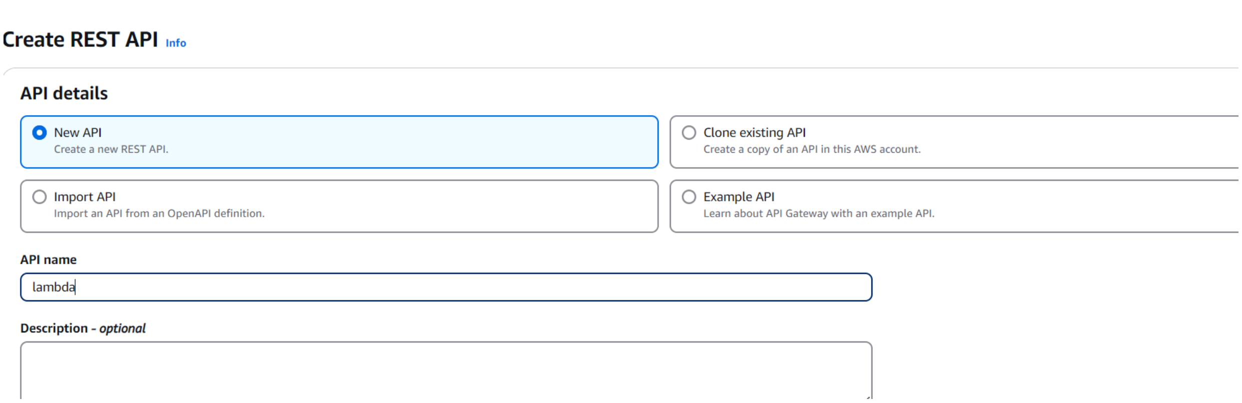


REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:
Lambda, HTTP, AWS Services

- Give the api name



Create REST API Info

API details

● New API
Create a new REST API.

○ Clone existing API
Create a copy of an API in this AWS account.

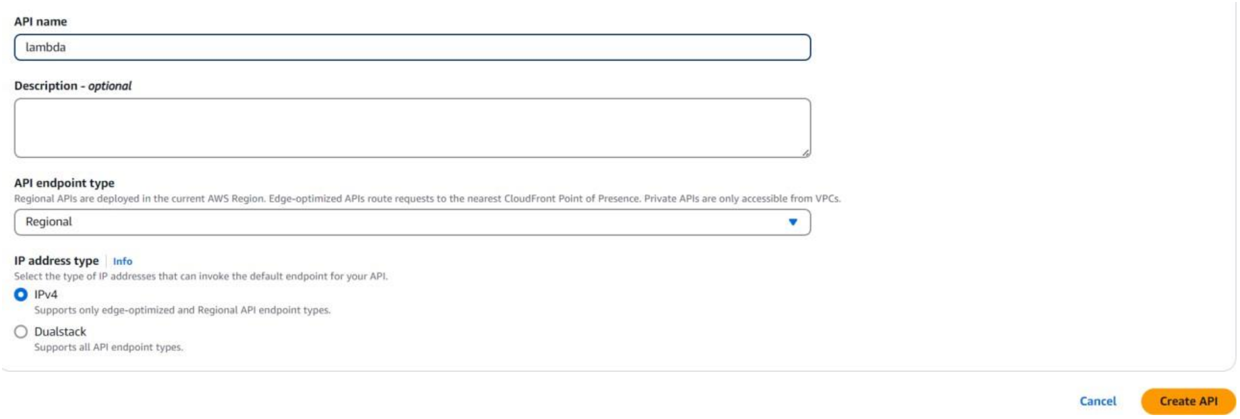
○ Import API
Import an API from an OpenAPI definition.

○ Example API
Learn about API Gateway with an example API.

API name
lambda

Description - *optional*

- Api endpoint type as regional
- Click on create API



API name
lambda

Description - *optional*

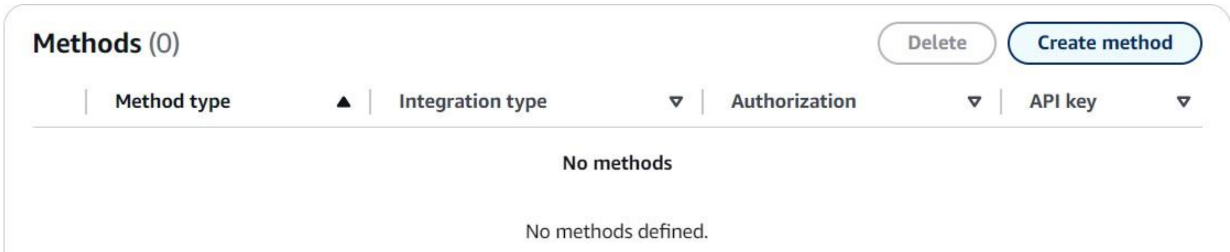
API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Regional

IP address type Info
Select the type of IP addresses that can invoke the default endpoint for your API.

● IPv4
Supports only edge-optimized and Regional API endpoint types.

○ Dualstack
Supports all API endpoint types.

- After that open your api
- Click on create method



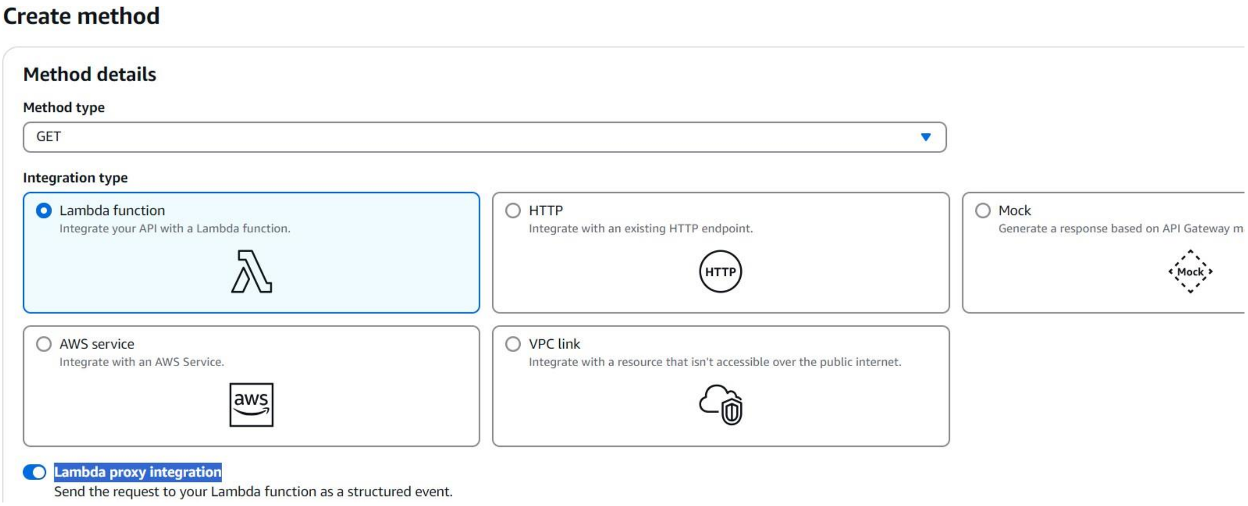
Methods (0)

Method type	Integration type	Authorization	API key

No methods

No methods defined.

- Method type select GET
- Turn on the lambda proxy integration



Create method

Method details

Method type

GET

Integration type

● Lambda function
Integrate your API with a Lambda function.

○ HTTP
Integrate with an existing HTTP endpoint.

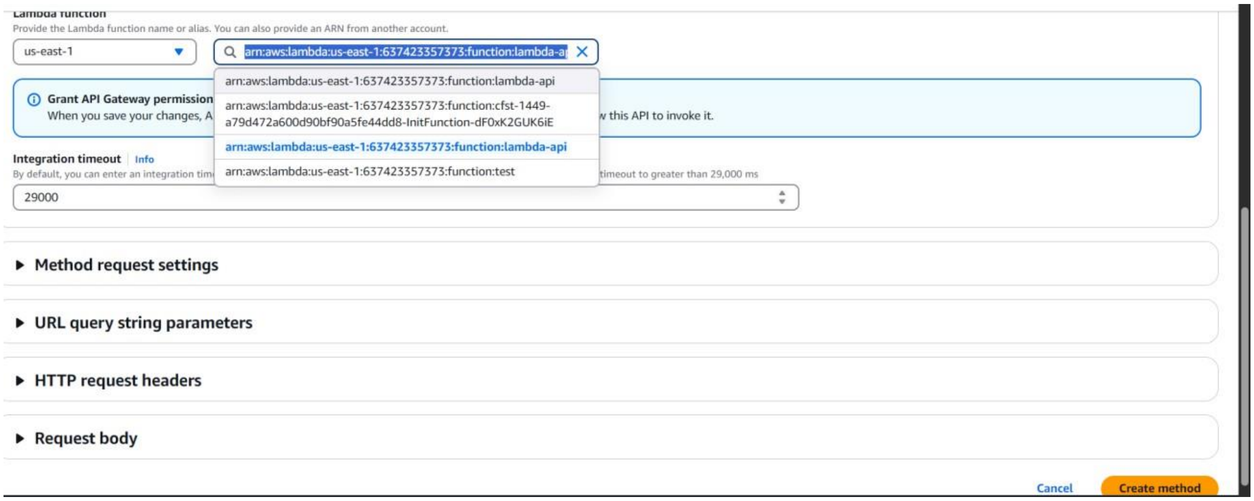
○ Mock
Generate a response based on API Gateway m

○ 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.

- Select your lambda function
- Click on create method



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

us-east-1

arn:aws:lambda:us-east-1:637423357373:function:lambda-ap

arn:aws:lambda:us-east-1:637423357373:function:lambda-api
arn:aws:lambda:us-east-1:637423357373:function:cfst-1449-a79d472a600d90bf90a5fe44dd8-InitFunction-dF0xK2GUK6iE
arn:aws:lambda:us-east-1:637423357373:function:lambda-api
arn:aws:lambda:us-east-1:637423357373:function:test

Grant API Gateway permission
When you save your changes, A w this API to invoke it.

Integration timeout Info
By default, you can enter an integration tim timeout to greater than 29,000 ms

29000

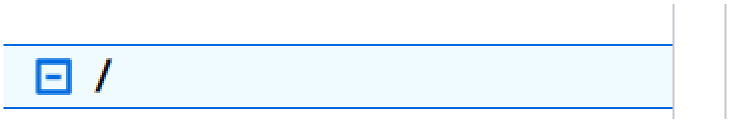
▸ **Method request settings**

▸ **URL query string parameters**

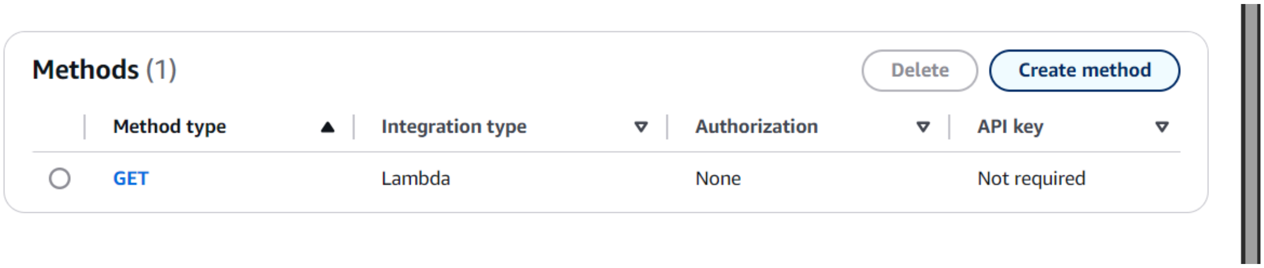
▸ **HTTP request headers**

▸ **Request body**

- One method is created we need to create one more method
- Click on the slash to create one more method



- Click on create method

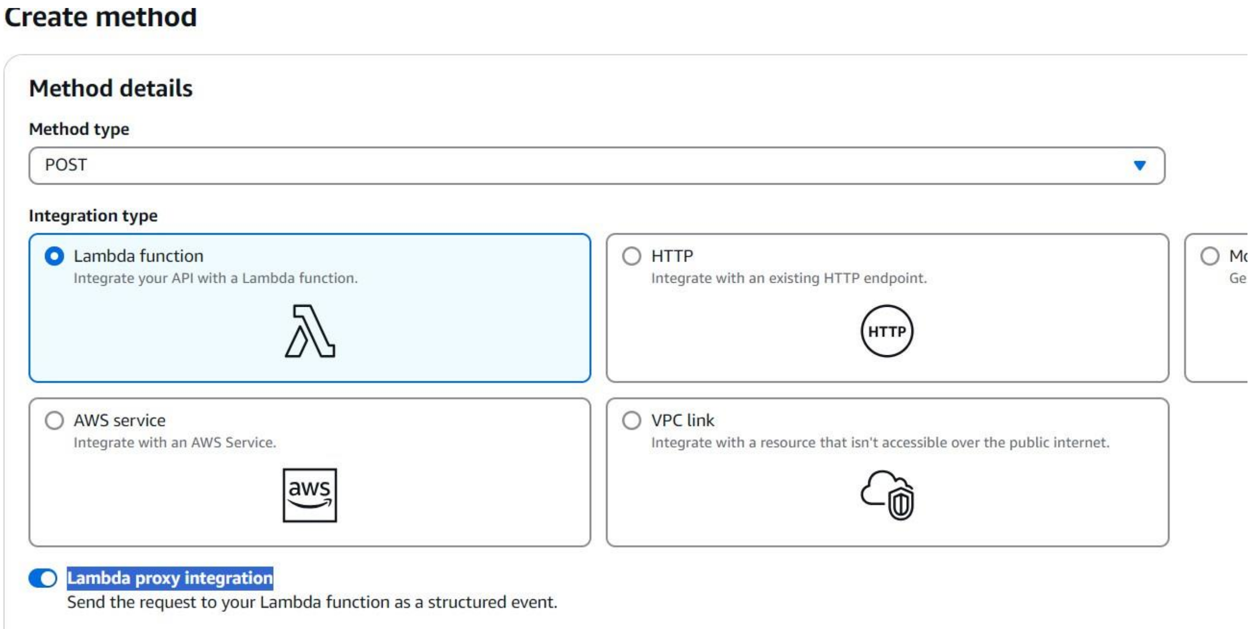


Methods (1)

	Method type	Integration type	Authorization	API key
○	GET	Lambda	None	Not required

- Method type select POST

- Turn on the lambda proxy integration



Create method

Method details

Method type

POST

Integration type

● Lambda function
Integrate your API with a Lambda function.

○ HTTP
Integrate with an existing HTTP endpoint.

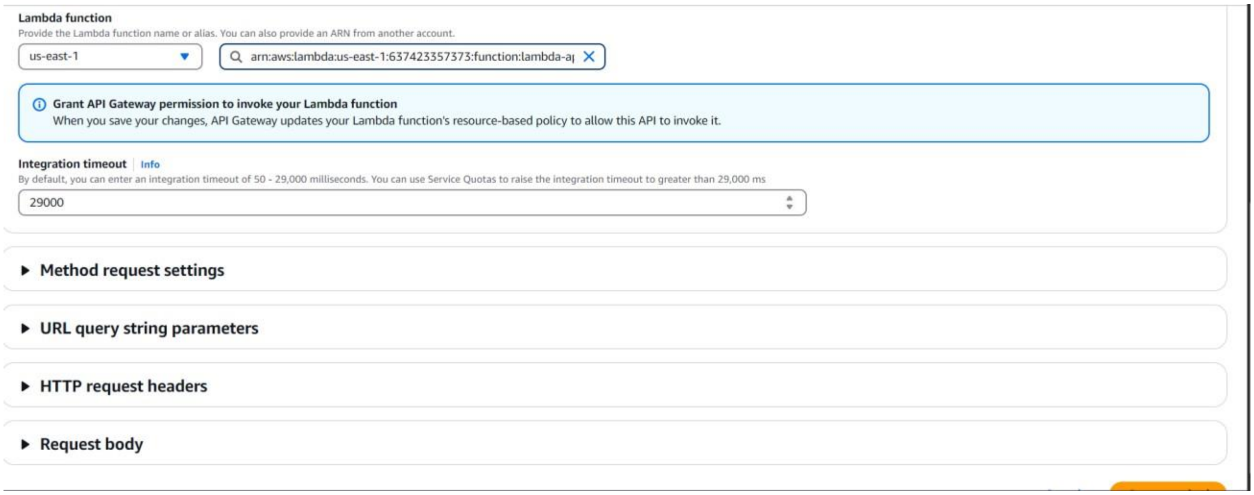
○ Mo
Ge

○ 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.

- Select your lambda function
- Click on create method



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

us-east-1

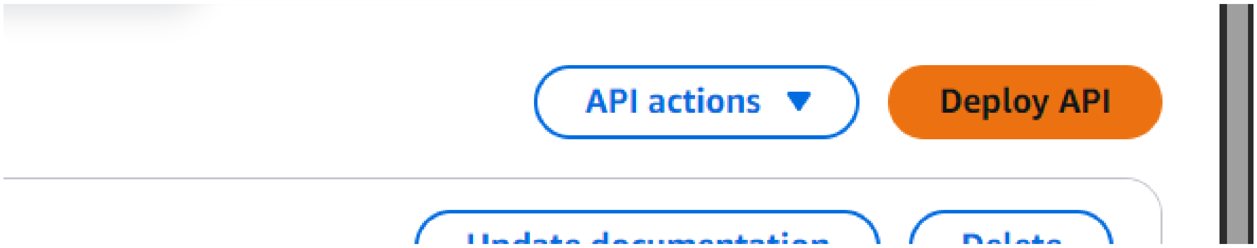
arn:aws:lambda:us-east-1:637423357373:function:lambda-a|

Grant API Gateway permission to invoke your Lambda function
When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.

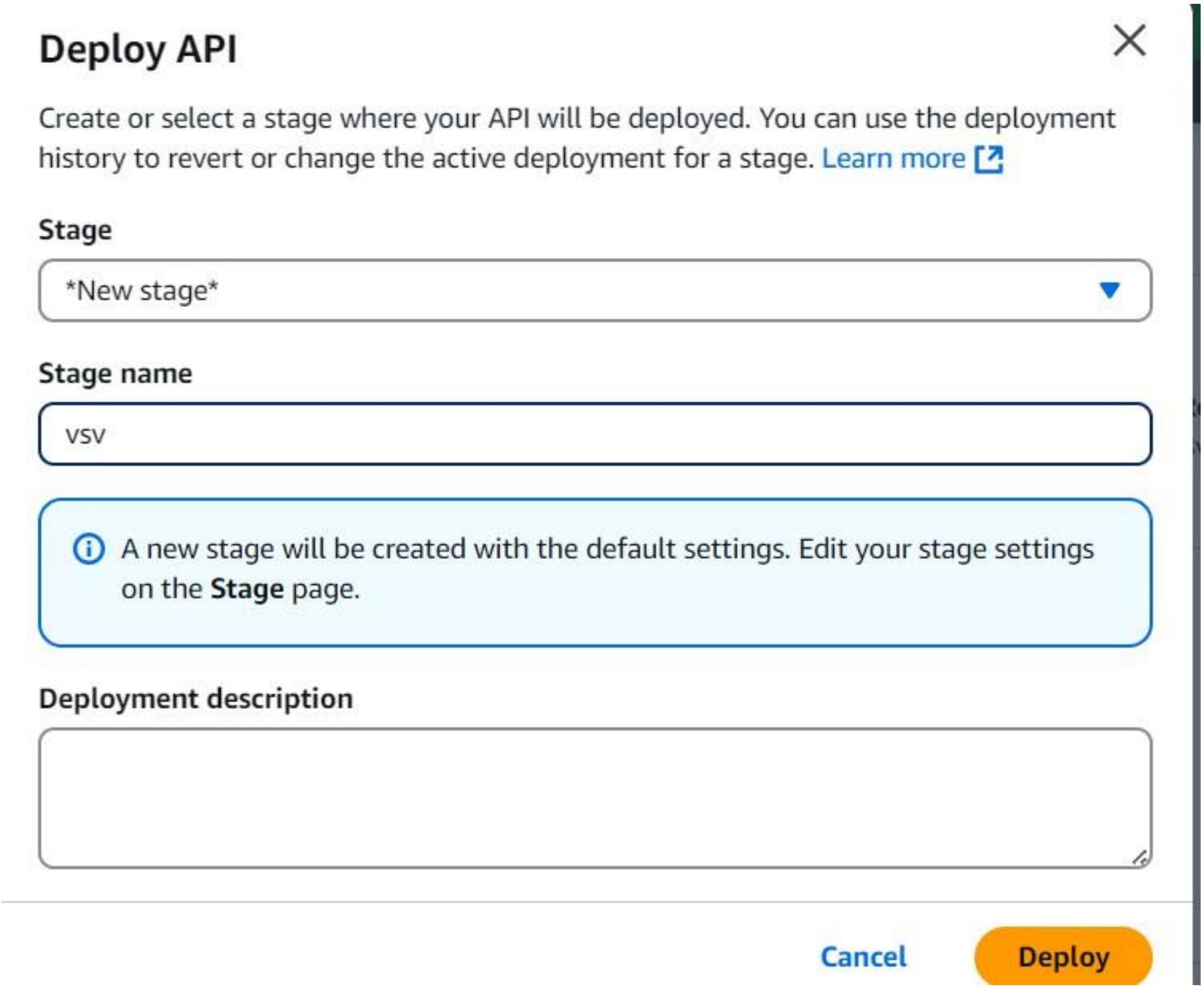
Integration timeout | Info
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms

29000

- Click on Deploy API



- Select the stage as new stage
- Give the stage name as dev
- Click on deploy



Deploy API

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. Learn more

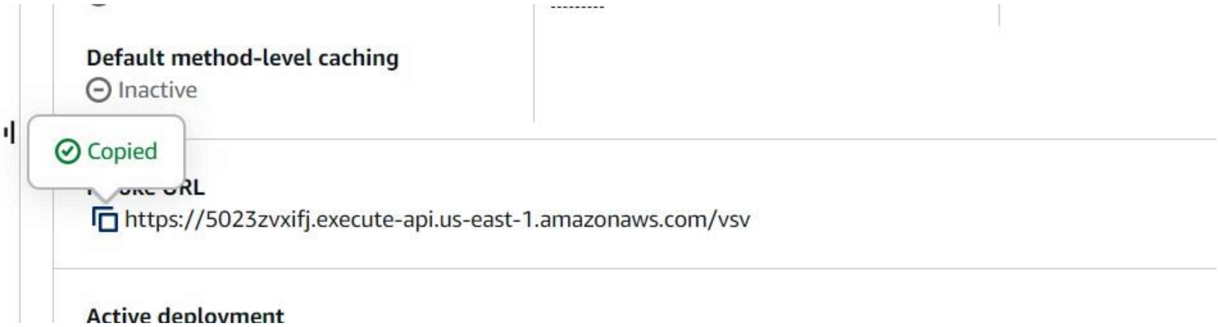
Stage: *New stage*

Stage name: vsv

ⓘ A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description

- Copy the stage invoke url

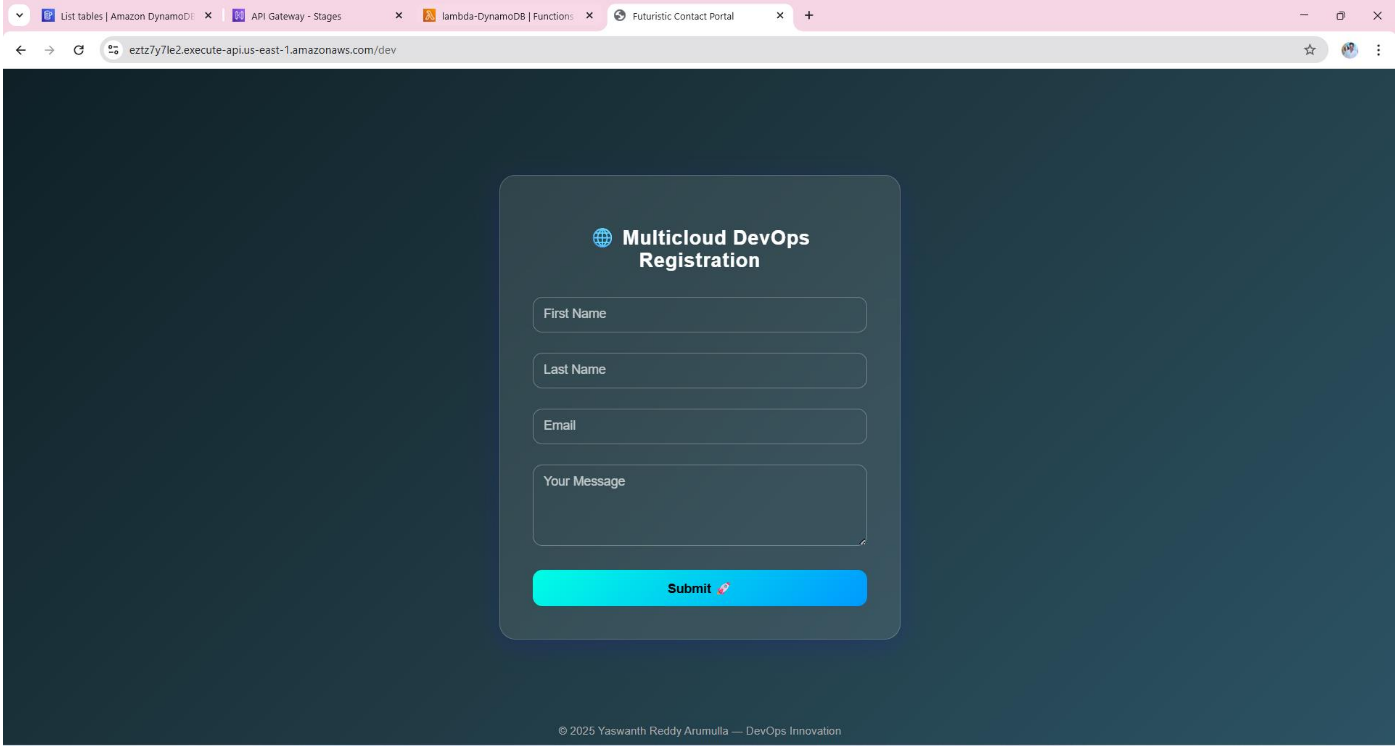


Default method-level caching: Inactive

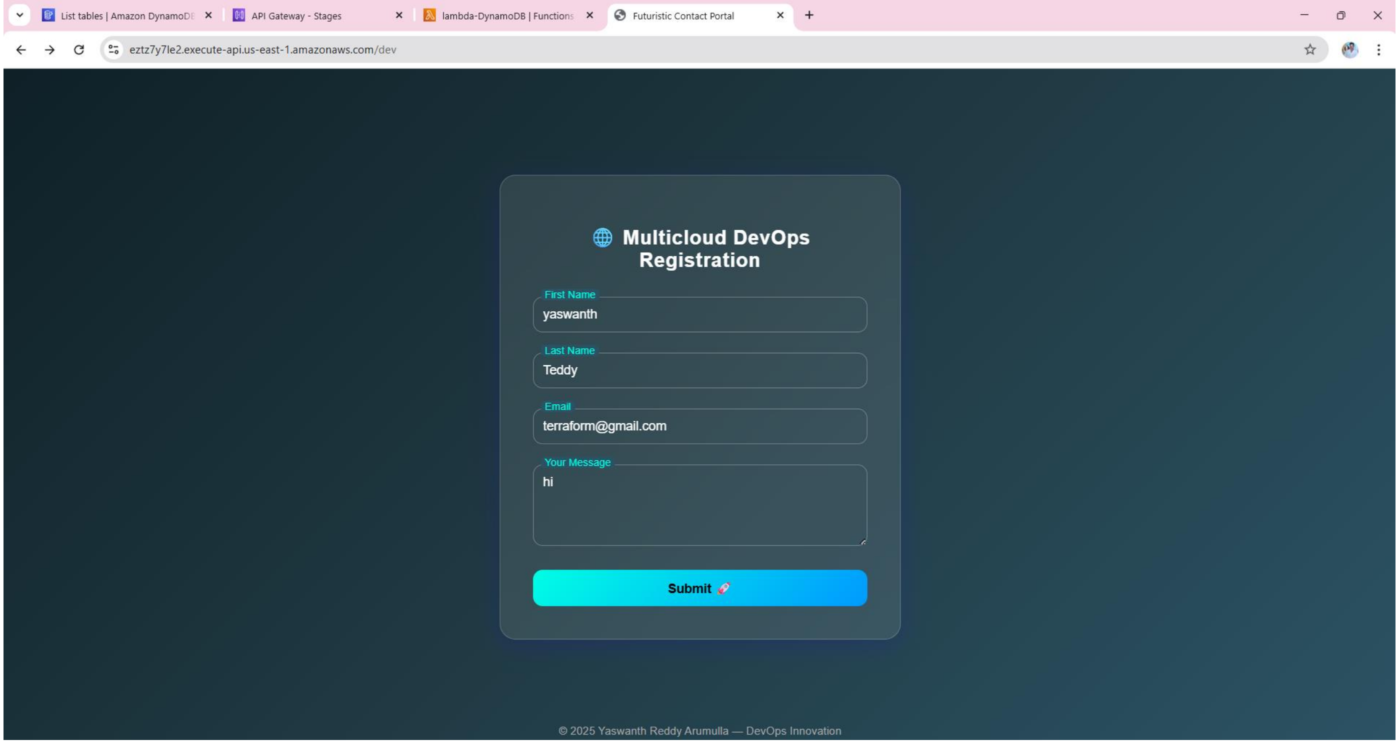
Invoke URL: https://5023zvxifj.execute-api.us-east-1.amazonaws.com/vsv

Active deployment

- Search it on web
- You will get the response


List tables | Amazon DynamoDB
API Gateway - Stages
lambda-DynamoDB | Functions
Futuristic Contact Portal
eztz7y7le2.execute-api.us-east-1.amazonaws.com/dev
🌐 Multicloud DevOps Registration
First Name
Last Name
Email
Your Message
Submit
© 2025 Yaswanth Reddy Arumulla — DevOps Innovation

- Add the details
- Click on submit


List tables | Amazon DynamoDB
API Gateway - Stages
lambda-DynamoDB | Functions
Futuristic Contact Portal
eztz7y7le2.execute-api.us-east-1.amazonaws.com/dev
🌐 Multicloud DevOps Registration
Submit
© 2025 Yaswanth Reddy Arumulla — DevOps Innovation

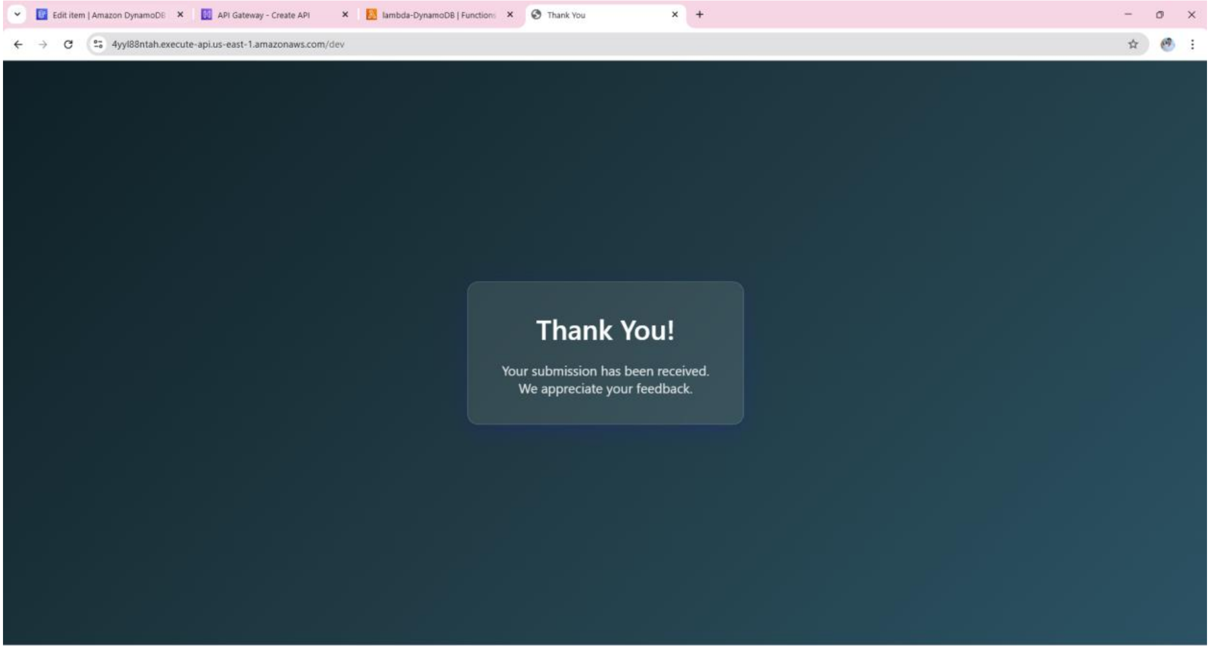
First Name: yaswanth

Last Name: Teddy

Email: terraform@gmail.com

Your Message: hi

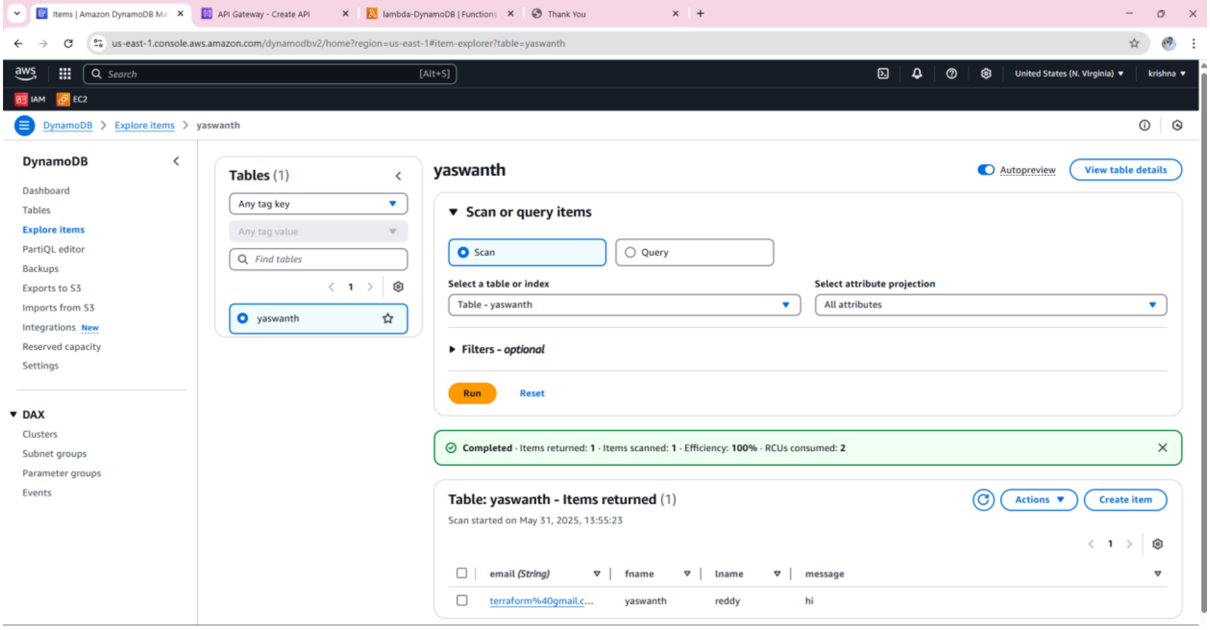
- This is successful submit output


Edit item | Amazon DynamoDB
API Gateway - Create API
lambda-DynamoDB | Functions
Thank You
4yyl88ntah.execute-api.us-east-1.amazonaws.com/dev

Thank You!

Your submission has been received.
We appreciate your feedback.

- Open your dynamodb table
- Click on explore items
- In right side corner you Are able to see the records



yaswanth

Scan or query items

● Scan ○ Query

Select a table or index: Table - yaswanth

Select attribute projection: All attributes

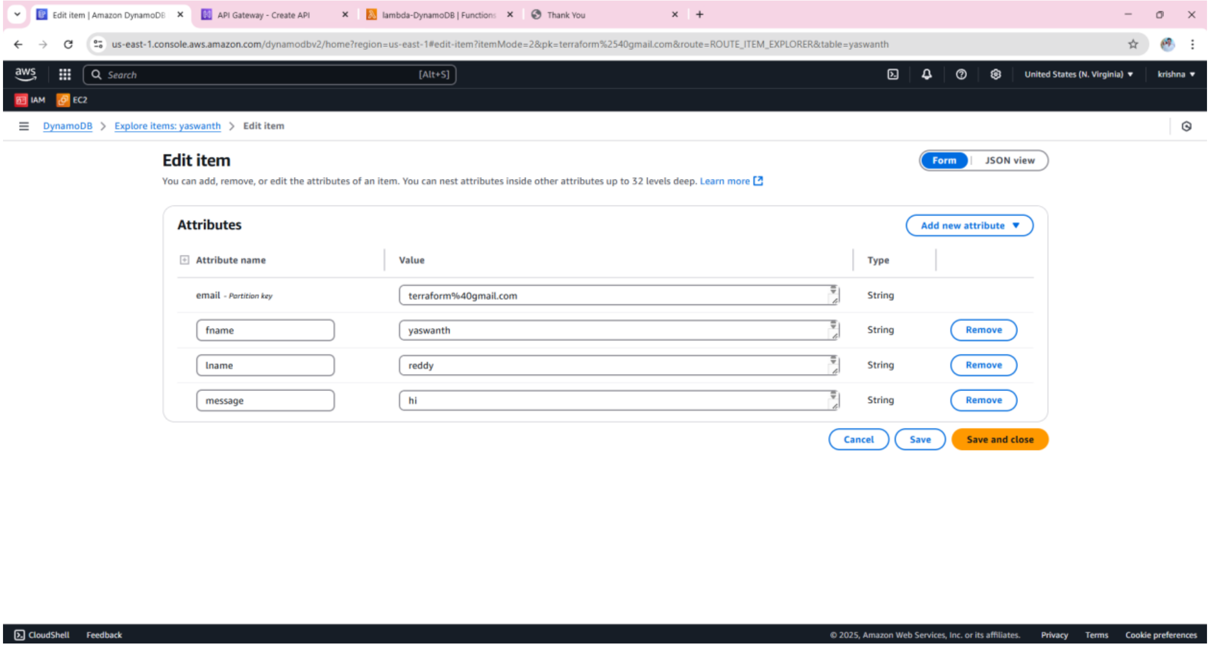
Filters - *optional*

Completed · Items returned: **1** · Items scanned: **1** · Efficiency: **100%** · RCUs consumed: **2**

Table: yaswanth - Items returned (1)

Scan started on May 31, 2025, 13:55:23

□	email (*String*)	fname	lname	message
□	terraform%40gmail.c...	yaswanth	reddy	hi



us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#edit-item?itemMode=2&pk=terraform%2540gmail.com&route=ROUTE_ITEM_EXPLORER&table=yaswanth

DynamoDB > Explore items: yaswanth > Edit item

Edit 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

Attributes

Attribute name	Value	Type	
email - *Partition key*	terraform%40gmail.com	String	
fname	yaswanth	String	Remove
lname	reddy	String	Remove
message	hi	String	Remove

THANK YOU

✅ Benefits of This Architecture

- **Serverless**: No server management needed
- **Scalable**: Handles traffic spikes automatically
- **Cost-effective**: Pay only for what you use
- **Decoupled**: Each service (API Gateway, Lambda, DynamoDB) has its own responsibility