PROJECT ON API Gateway-LAMBDA and DYNAMODB INTEGRATION

OVERVIEW

- User submits a form on a website.
- The frontend sends a POST request to API Gateway.
- API Gateway triggers a Lambda function.
- Lambda:
  - Validates input.
  - Saves user data to DynamoDB.
  - Returns a success or error message.
- API Gateway sends that response back to the client.

⚙️ 1. AWS Lambda

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

- You write functions (e.g., in Node.js, Python, Java).
- Triggered by events (like HTTP requests, DynamoDB changes, S3 uploads, etc.).

🌐 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 apps to **securely call Lambda functions**.
- Converts **HTTP requests into Lambda invocations**.

🗃️ 3. Amazon DynamoDB

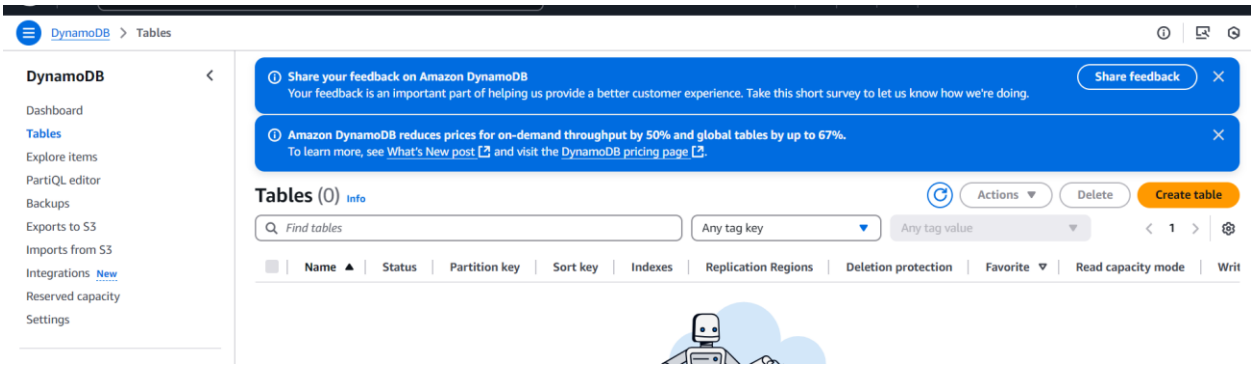
What it is: A fully managed **NoSQL database** service.

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

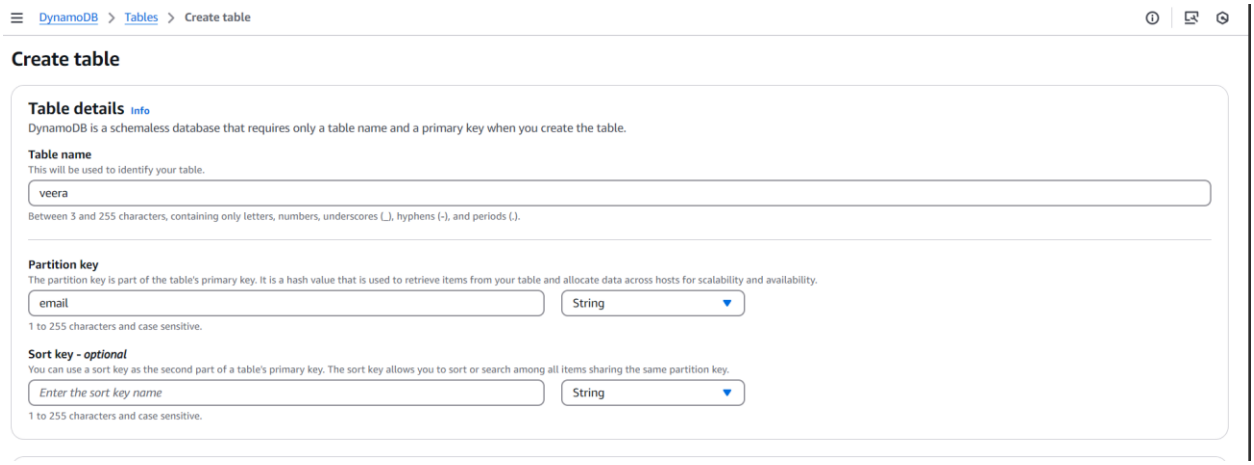
----------process start----------

- Open Dynamodb

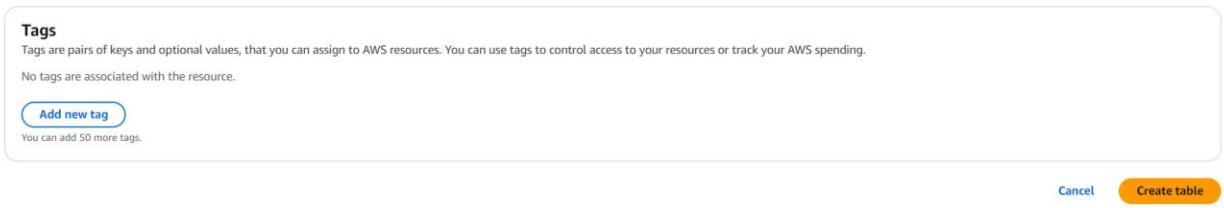
- Click on create table



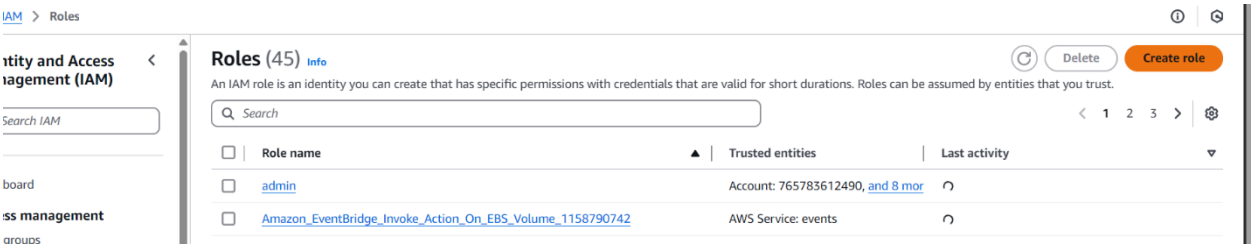
- Give table name as veera
- Partition key as a email
- If you change table name you need to change in lambda function



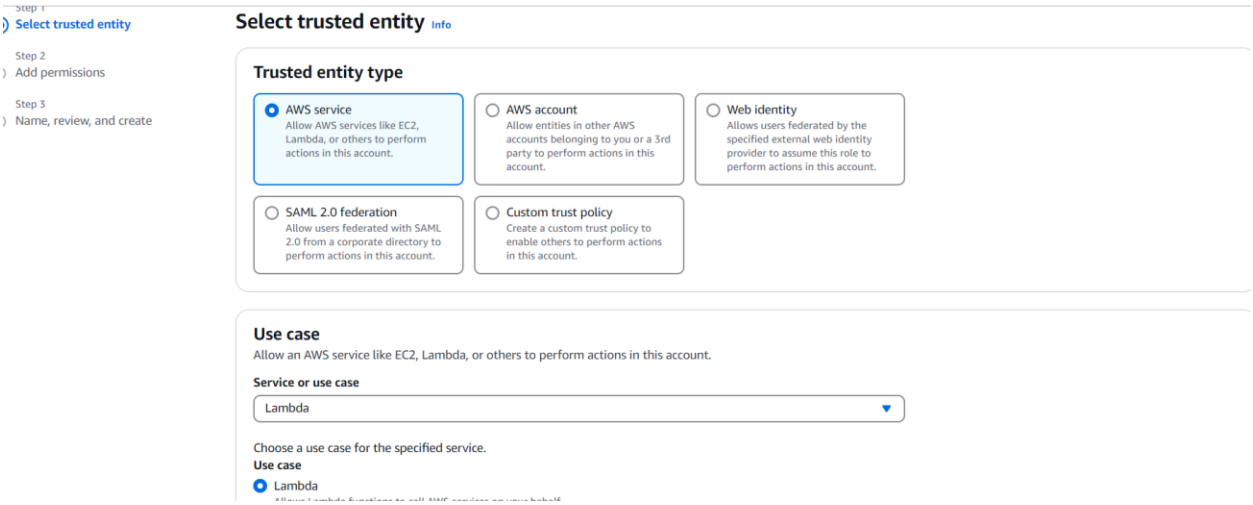
- Click on create



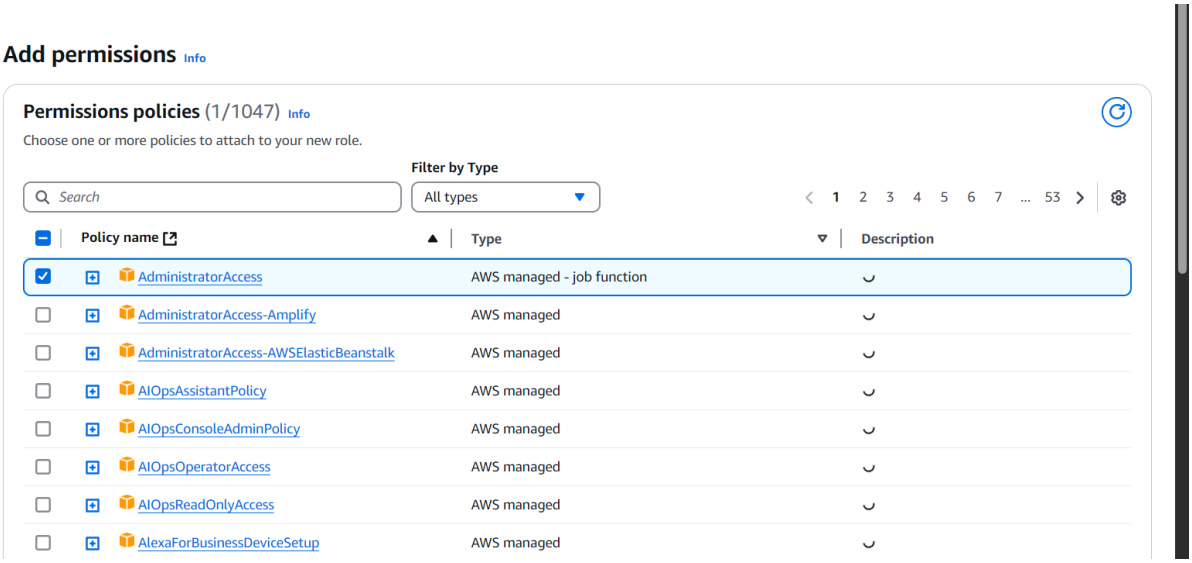
- Open IAM
- Click on create role



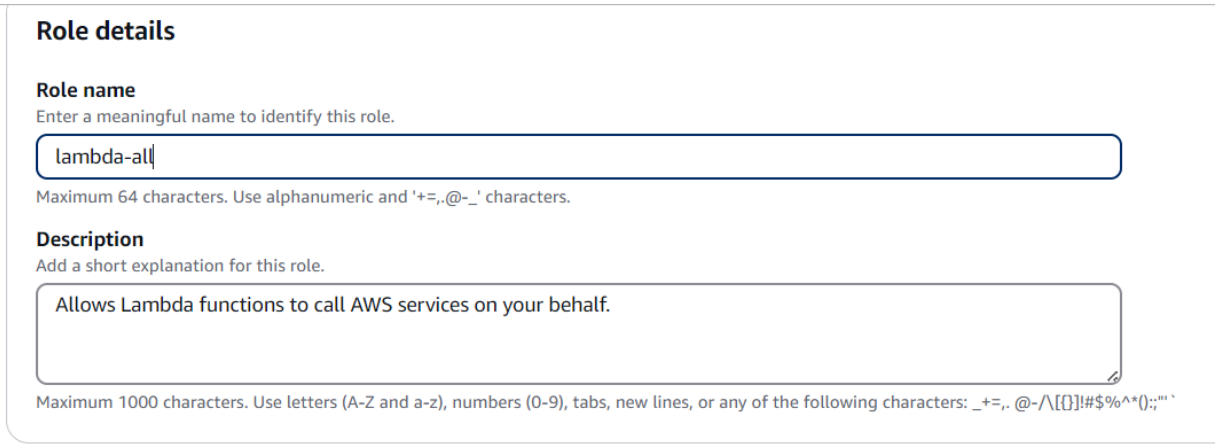
- Service is select lambda



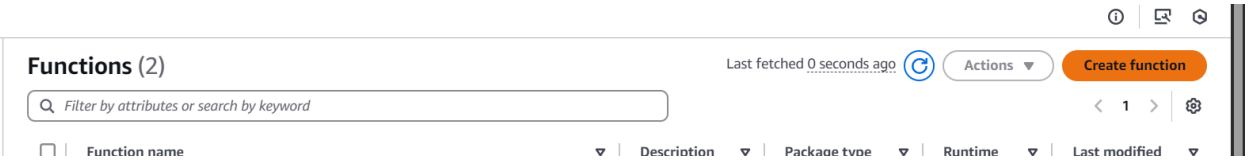
- Add admin access
- Click on next



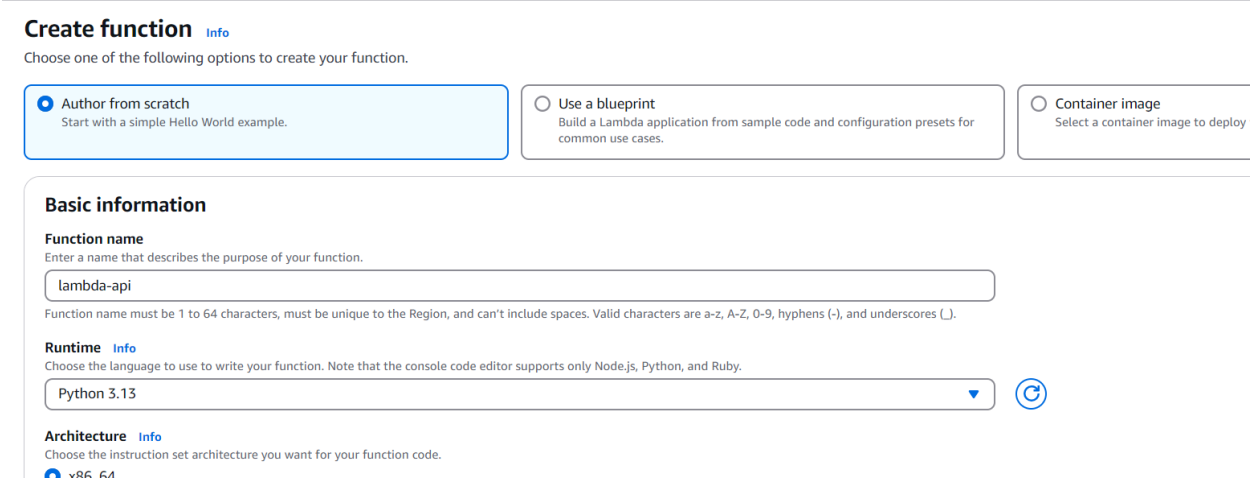
- Give the name to role
- Click on create



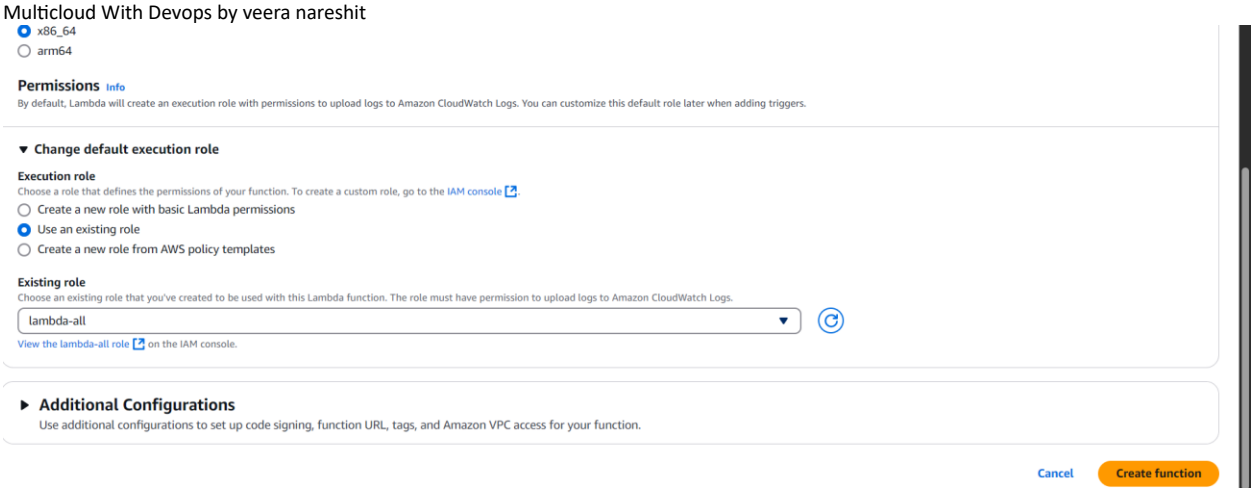
- Open lambda
- Click on create function



- Give name
- Select the run time as python



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



- In lambda_function.py file paste below code

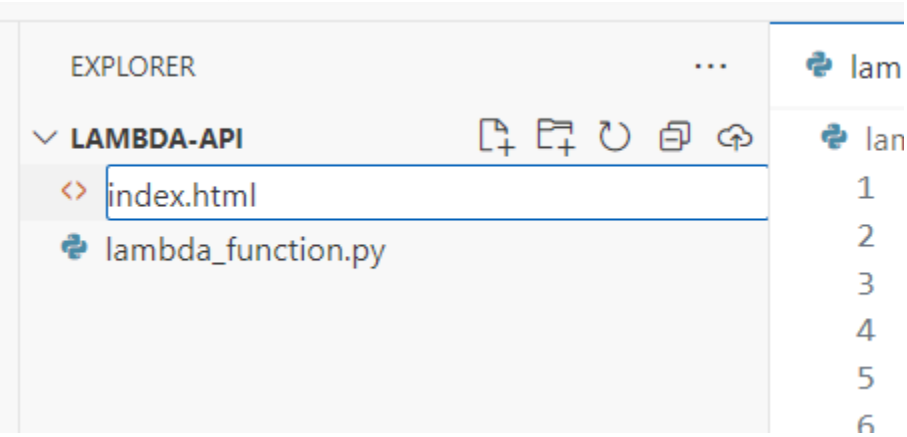
https://github.com/CloudTechDevOps/project-api-lambda-dynamodb-intigration/blob/main/lambda_function.py

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

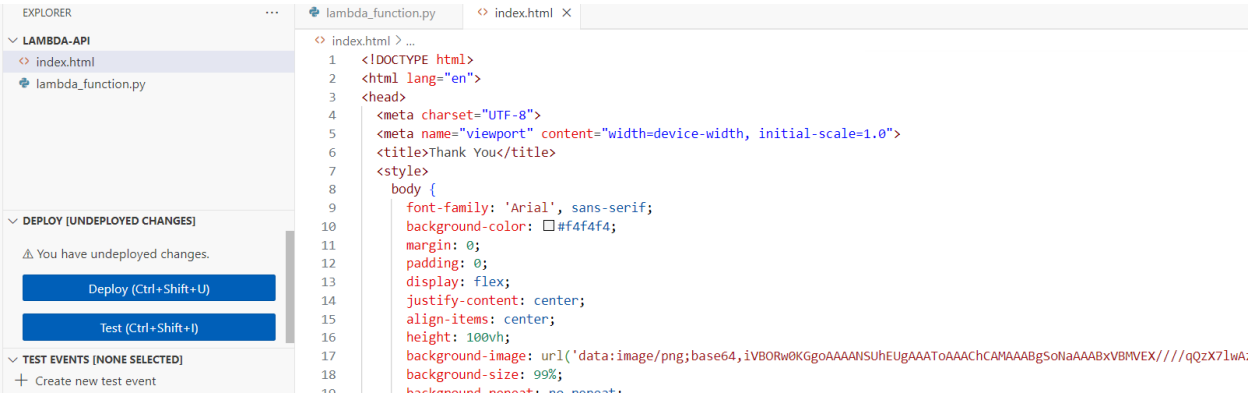
- Create a new file name as index.html
- Paste the below code

https://github.com/CloudTechDevOps/project-api-lambda-dynamodb-intigration/blob/main/index.html

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



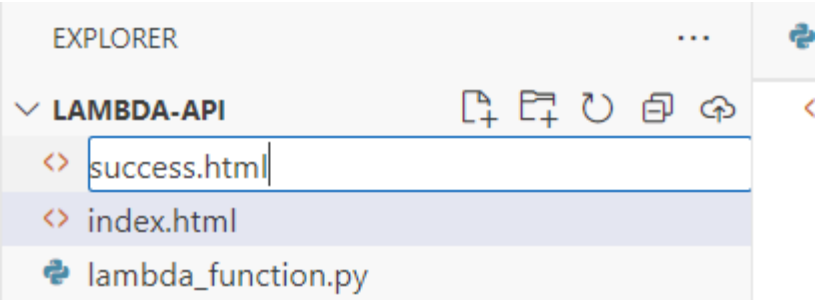
- Like this



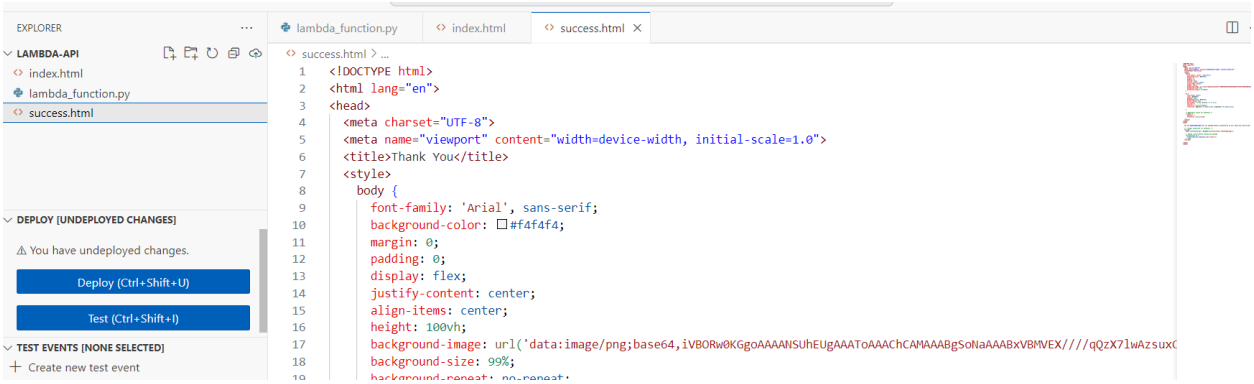
- Create a new file name as success.html
- Paste the below code

https://github.com/CloudTechDevOps/project-api-lambda-dynamodb-intigration/blob/main/success.html

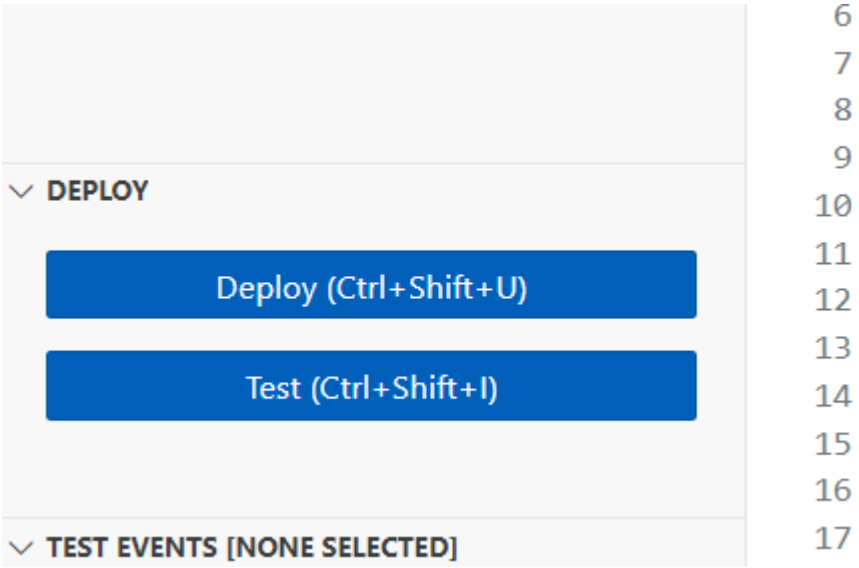
- Code will be on above git link copy and paste it



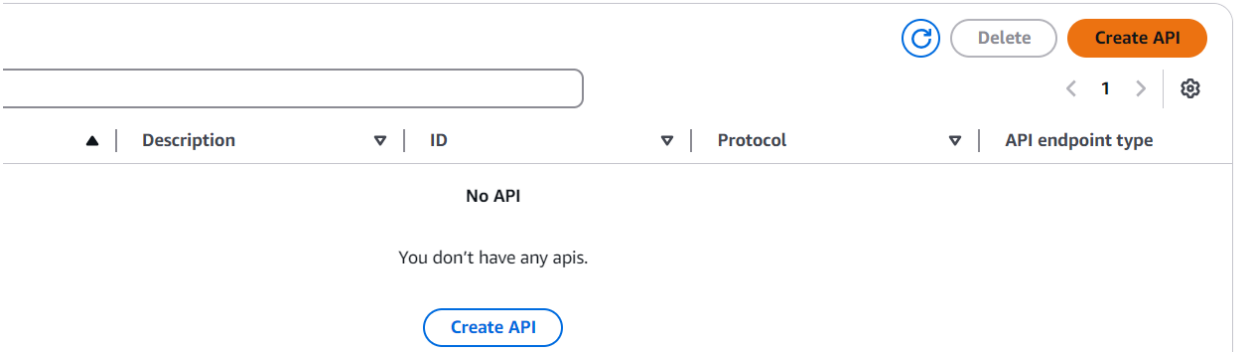
- Like this



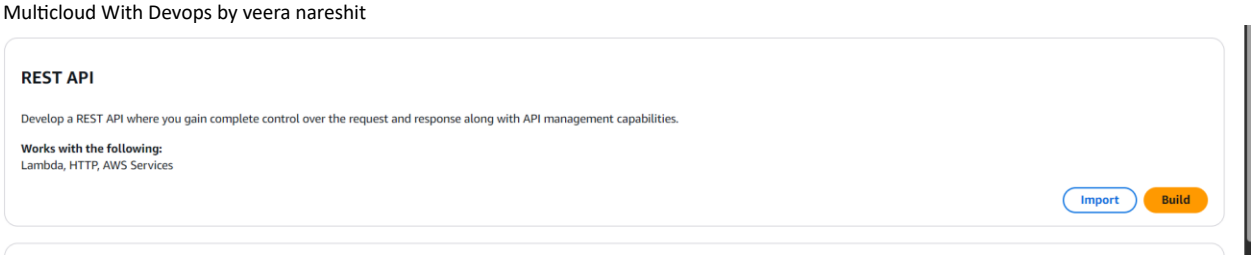
- Click on deploy



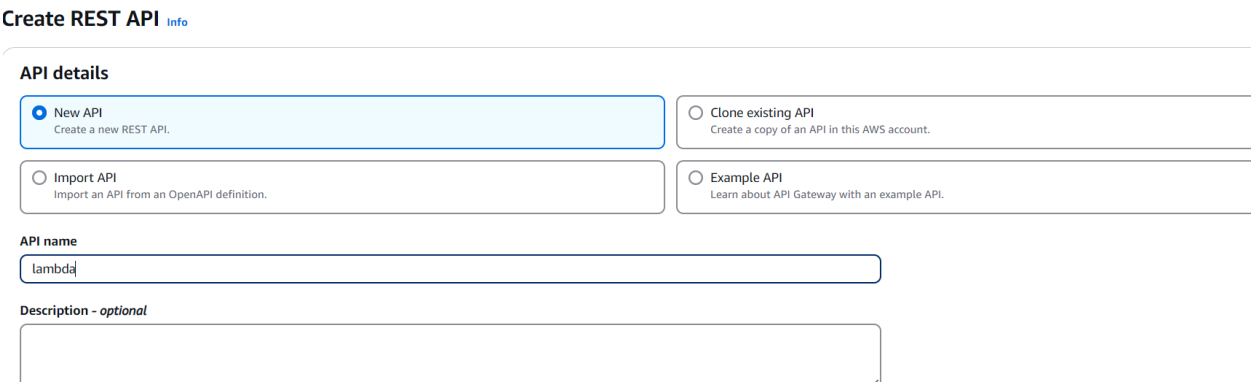
- Open api gateway page
- Click on create api



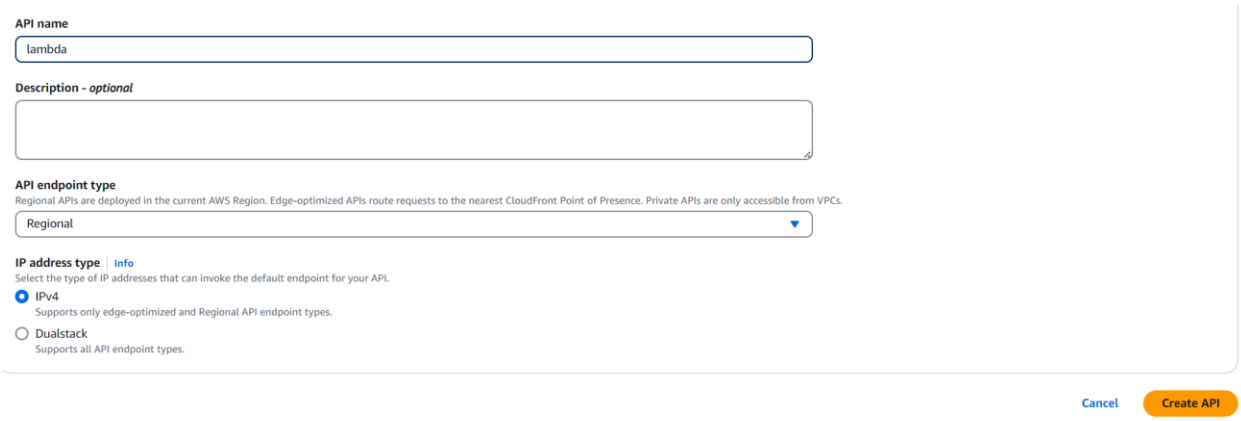
- In apis list build the rest api
- Click build on rest api



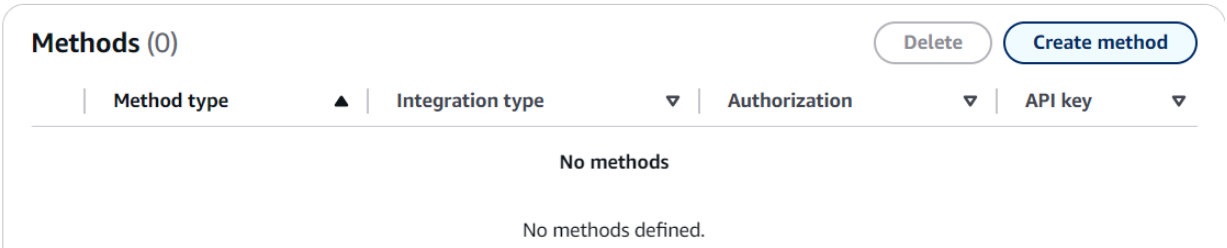
- Give the api name



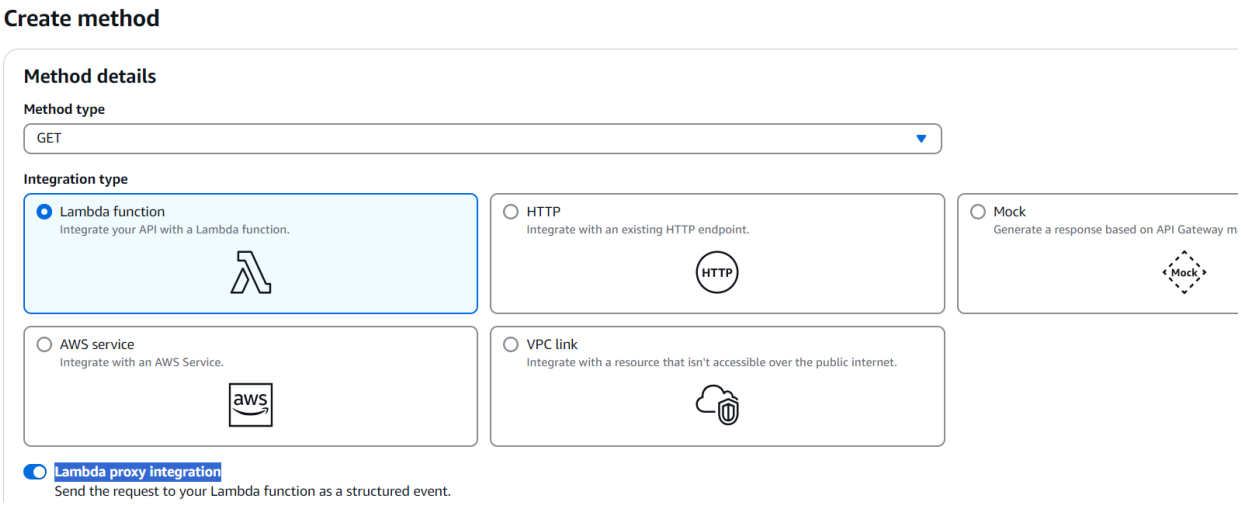
- Api endpoint type as regional
- Click on create API



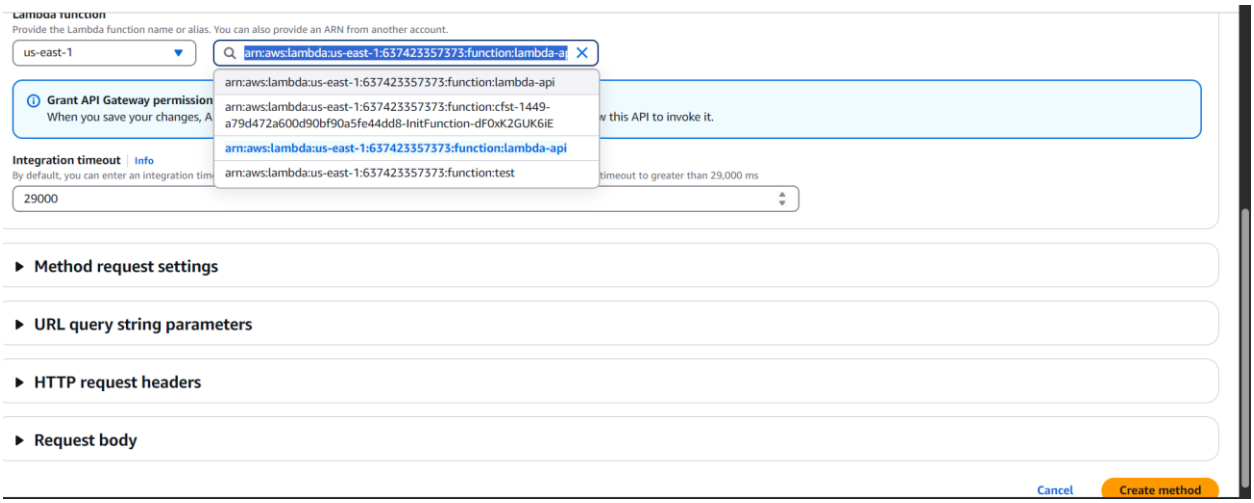
- After that open your api
- Click on create method



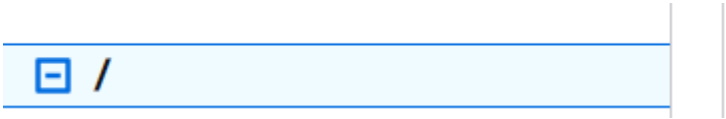
- Method type select GET
- Turn on the lambda proxy integration



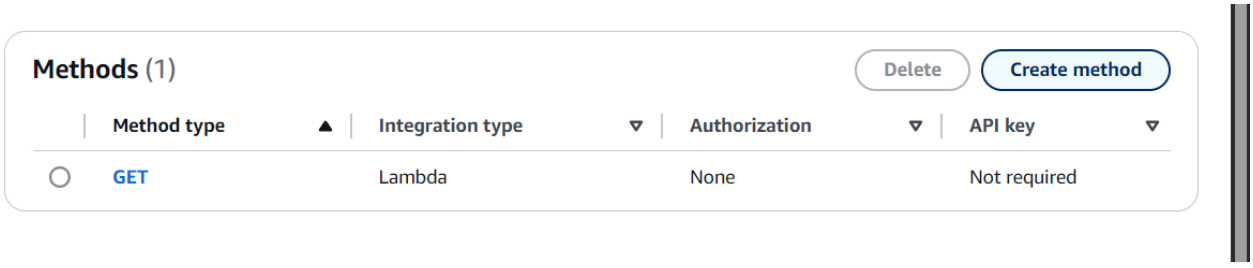
- Select your lambda function
- Click on create method



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

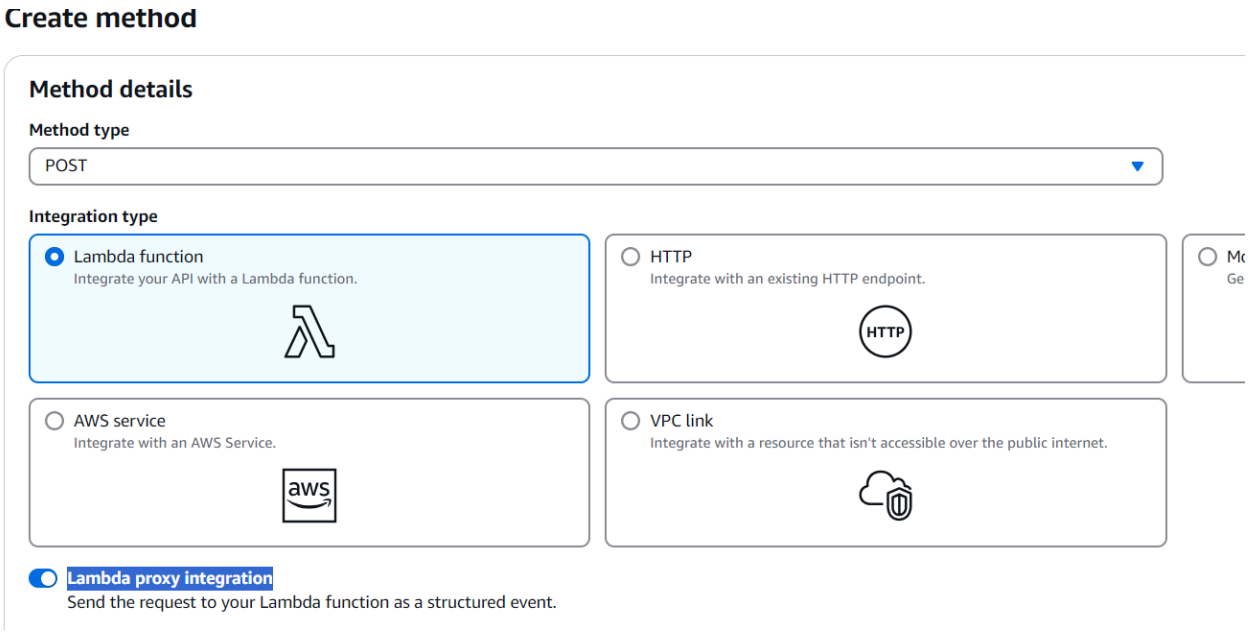


- Click on create method

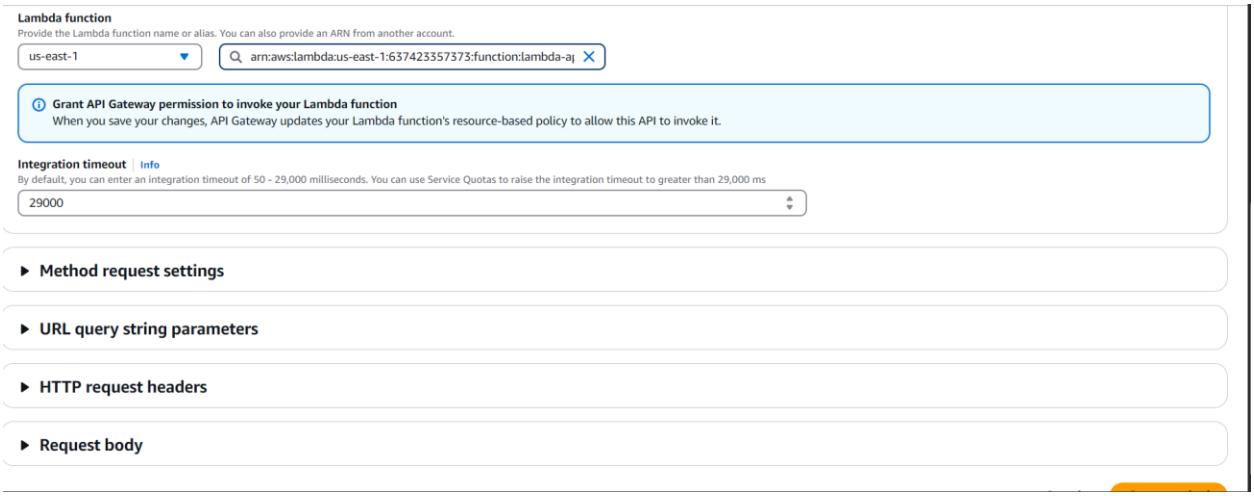


- Method type select POST

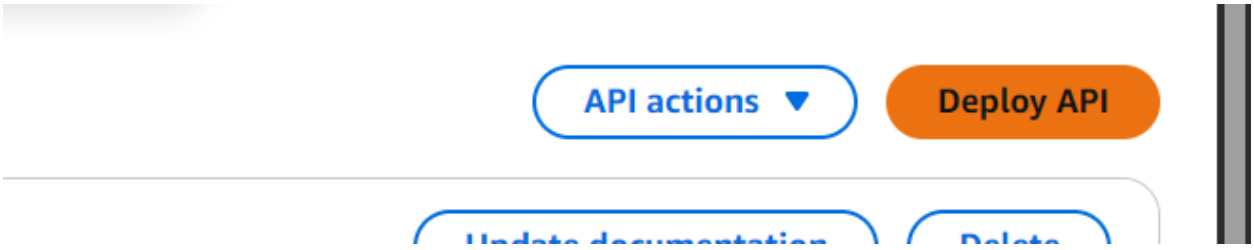
- Turn on the lambda proxy integration



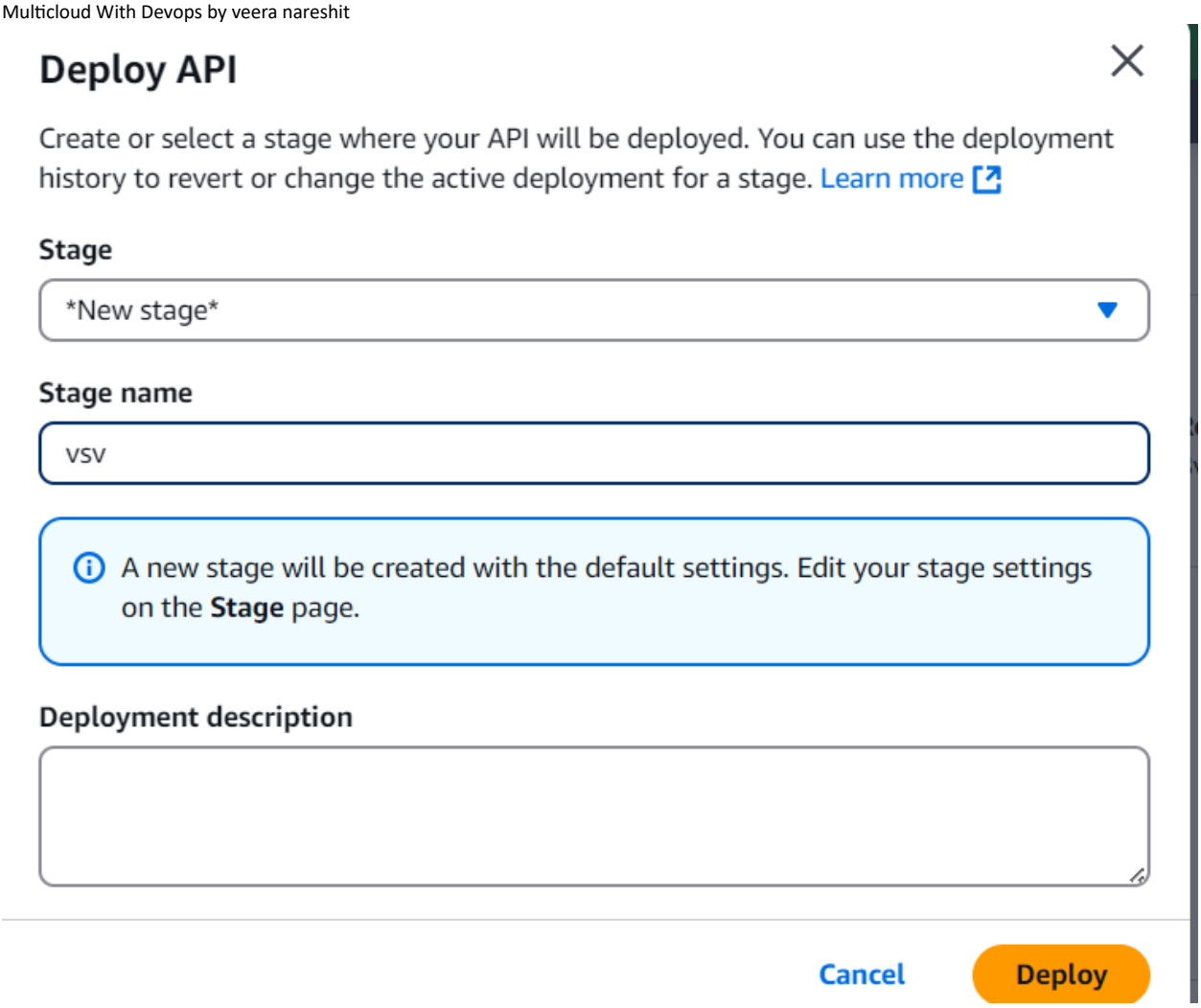
- Select your lambda function
- Click on create method



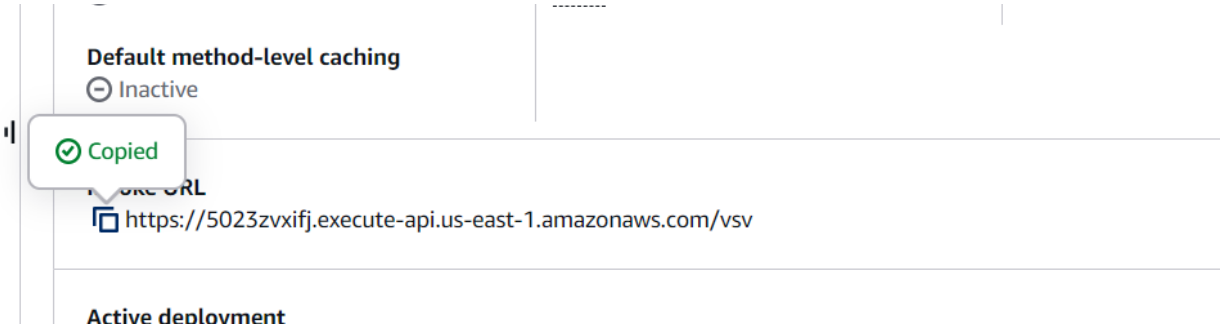
- Click on Deploy API



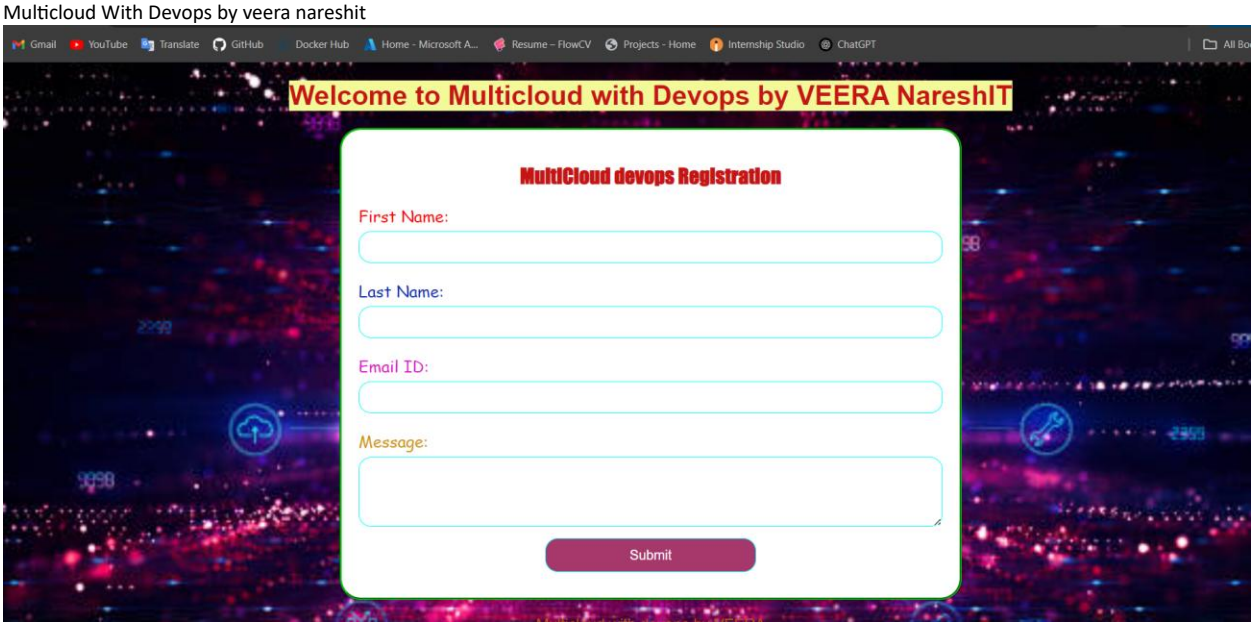
- Select the stage as new stage
- Give the stage name as dev
- Click on deploy



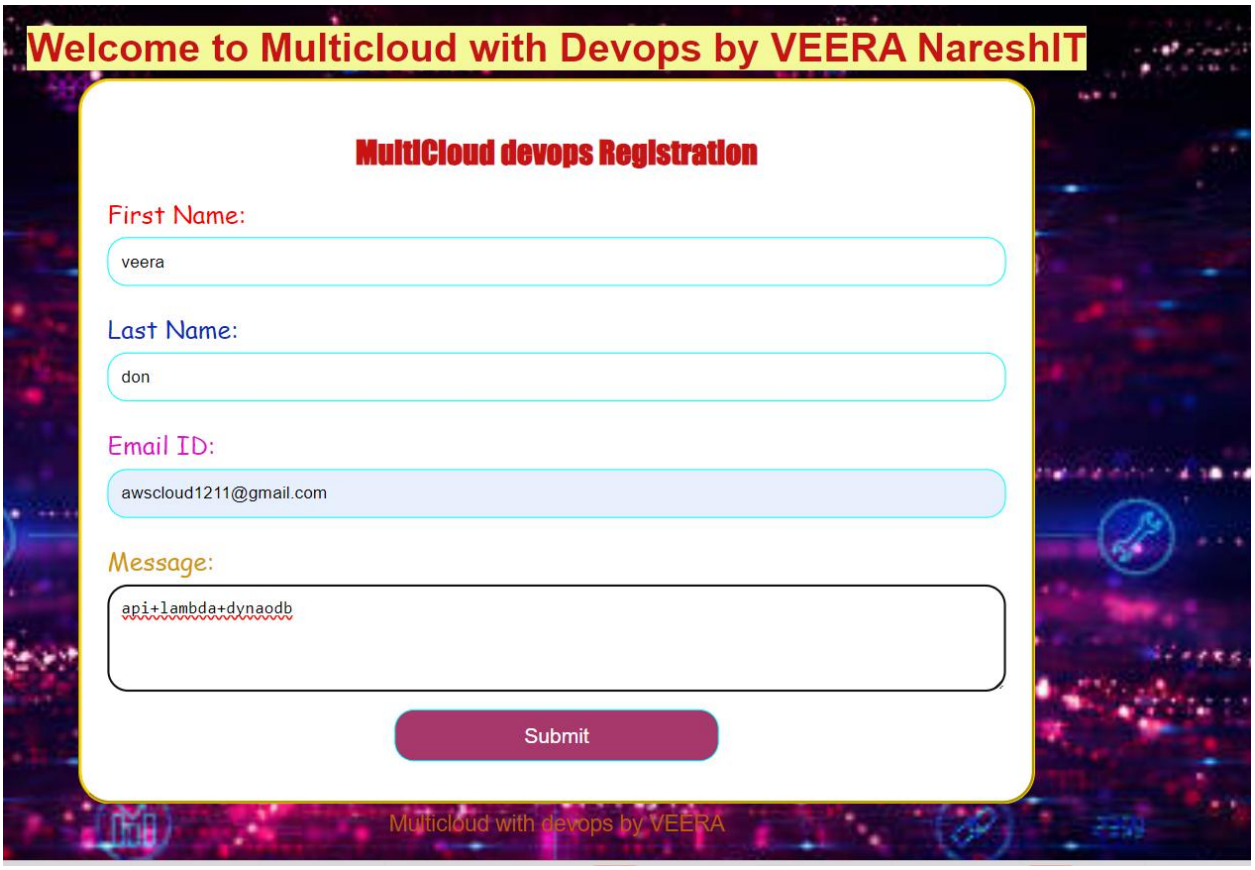
- Copy the stage invoke url



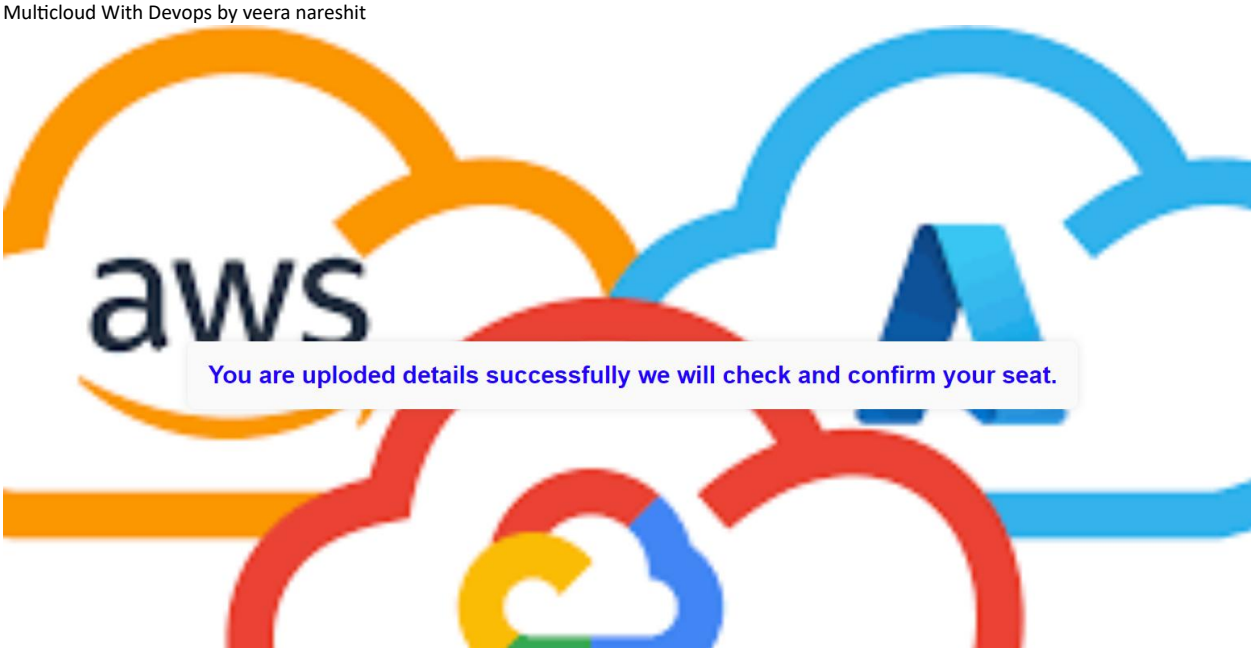
- Search it on web
- You will get the response



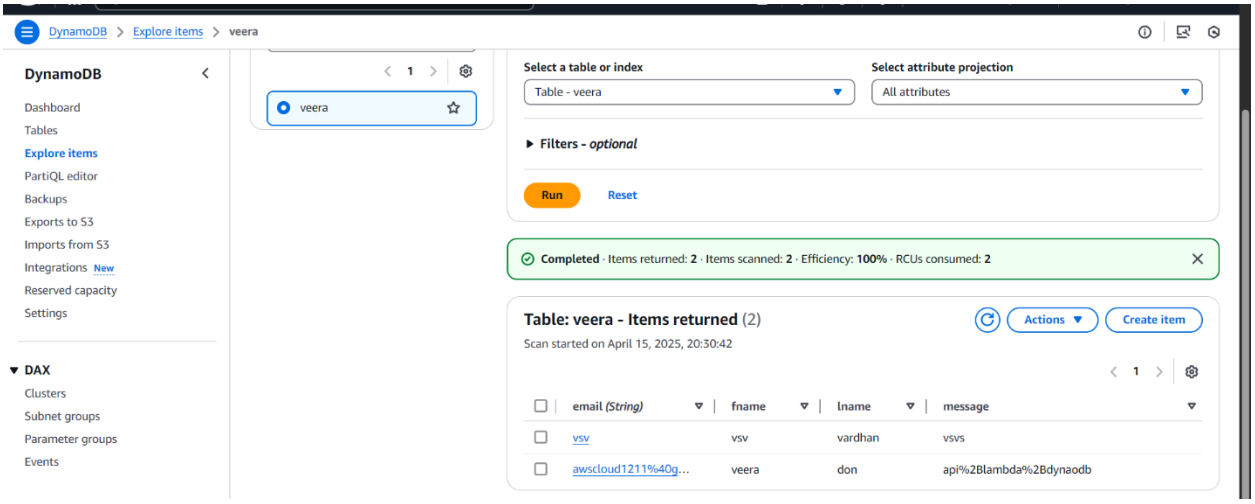
- Add the details
- Click on submit



- This is successful submit output



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



THANK YOU

✅ Benefits of This Integration

- **Serverless:** No server maintenance or provisioning.
- **Scalable:** Automatically handles traffic spikes.
- **Cost-effective:** Pay only for what you use.
- **Decoupled:** Each service handles its specific role.