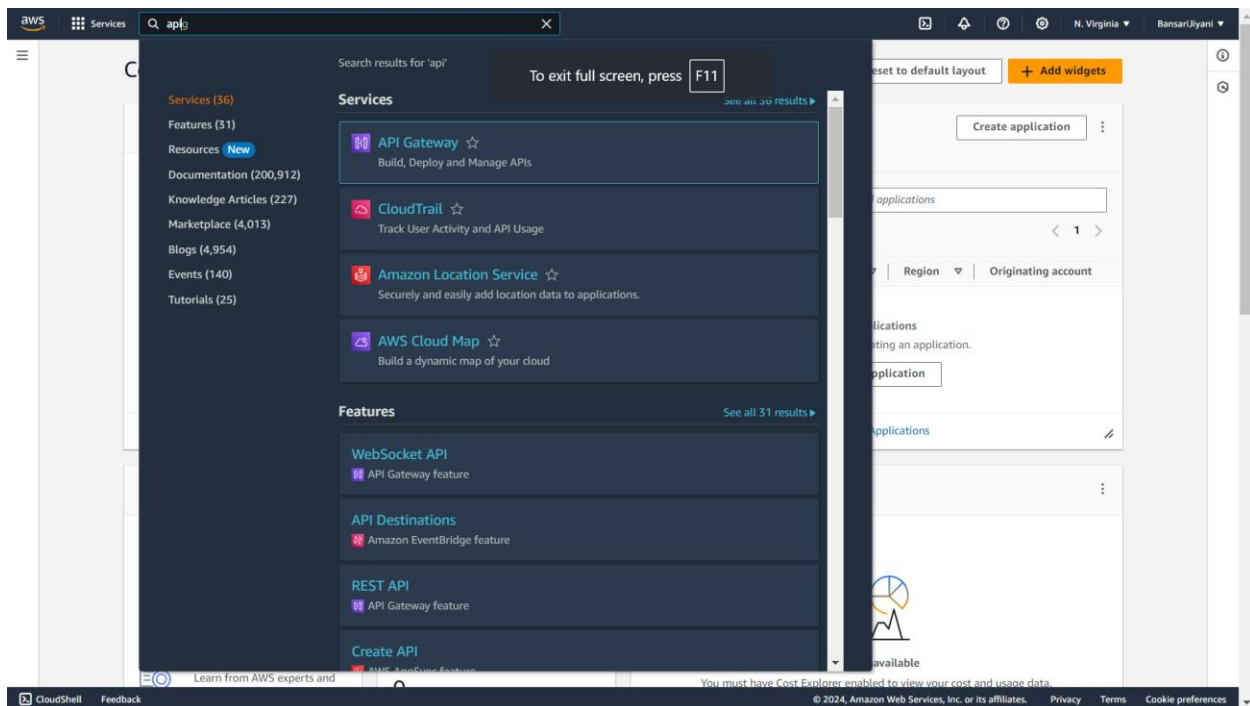


What is API gateway ?

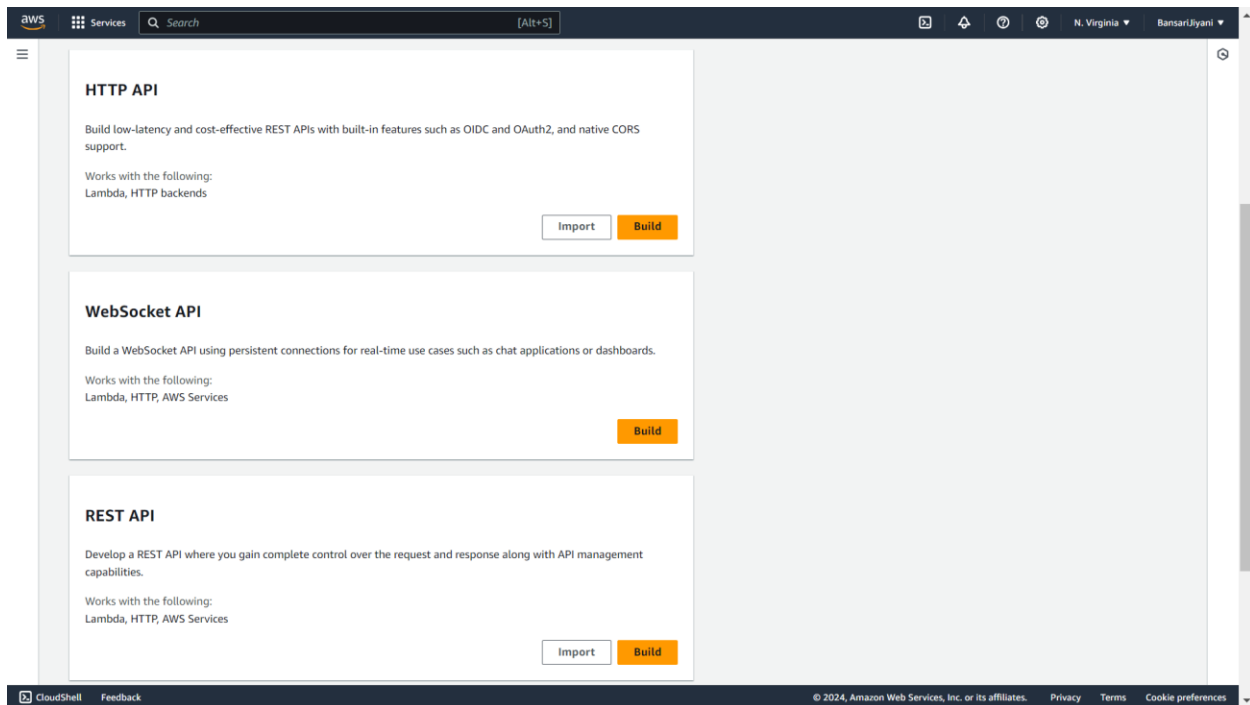
- Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and WebSocket APIs at any scale. API developers can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud. As an API Gateway API developer, you can create APIs for use in your own client applications. Or you can make your APIs available to third-party app developers.

PRACTICAL :

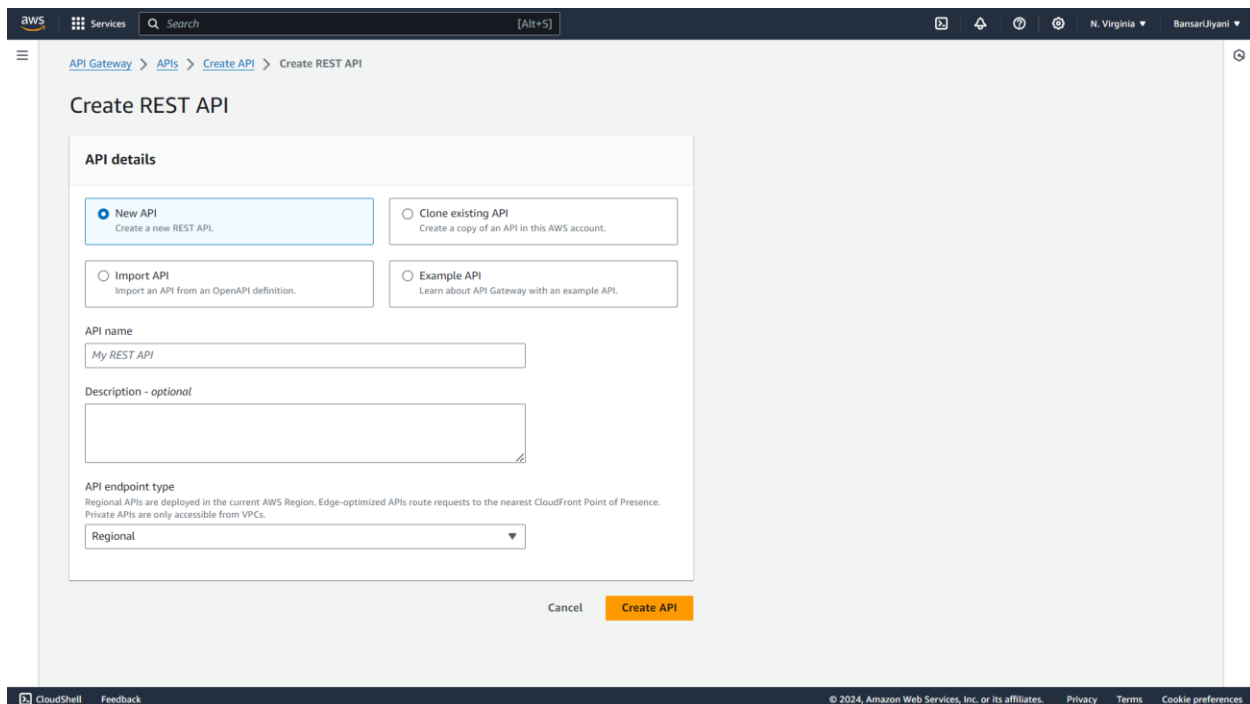
1. How create API. Go to the API gateway



2. There is many types of API . you can see here.



3. We create REST API.



4. Give the name of API and select the API endpoint type Regional. And create API.

The screenshot shows the 'Create REST API' page in the AWS API Gateway console. The breadcrumb navigation is 'API Gateway > APIs > Create API > Create REST API'. The page title is 'Create REST API'. Under 'API details', there are four radio button options: 'New API' (selected), 'Clone existing API', 'Import API', and 'Example API'. Below these, there is a text input for 'API name' containing 'FirstApi' and a text area for 'Description - optional'. The 'API endpoint type' is set to 'Regional' in a dropdown menu. At the bottom, there are 'Cancel' and 'Create API' buttons.

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
FirstApi

Description - optional

API endpoint type
Regional

Cancel Create API

5. Our API is successfully created.

The screenshot shows the 'Resources' page in the AWS API Gateway console. A green banner at the top indicates 'Successfully created REST API 'FirstApi' (ngmq6u0s54)'. The breadcrumb navigation is 'API Gateway > APIs > Resources - FirstApi (ngmq6u0s54)'. The page title is 'Resources'. On the left, there is a sidebar with a list of API components: 'APIs', 'Custom domain names', 'VPC links', 'API: FirstApi' (expanded), 'Resources' (selected), 'Stages', 'Authorizers', 'Gateway responses', 'Models', 'Resource policy', 'Documentation', 'Dashboard', 'API settings', 'Usage plans', 'API keys', 'Client certificates', and 'Settings'. The main content area shows a 'Create resource' button and a table with one resource: '/'. To the right, there is a 'Resource details' section with 'Path' as '/' and 'Resource ID' as 'fe7mqzso15'. Below this, there is a 'Methods (0)' section with a 'Delete' button and a 'Create method' button. A table header is visible with columns: 'Method type', 'Integration type', 'Authorization', and 'API key'. The table is currently empty, showing 'No methods' and 'No methods defined.'.

API Gateway

Successfully created REST API 'FirstApi' (ngmq6u0s54).

API Gateway > APIs > Resources - FirstApi (ngmq6u0s54)

Resources

Create resource

Resource details

Path: /

Resource ID: fe7mqzso15

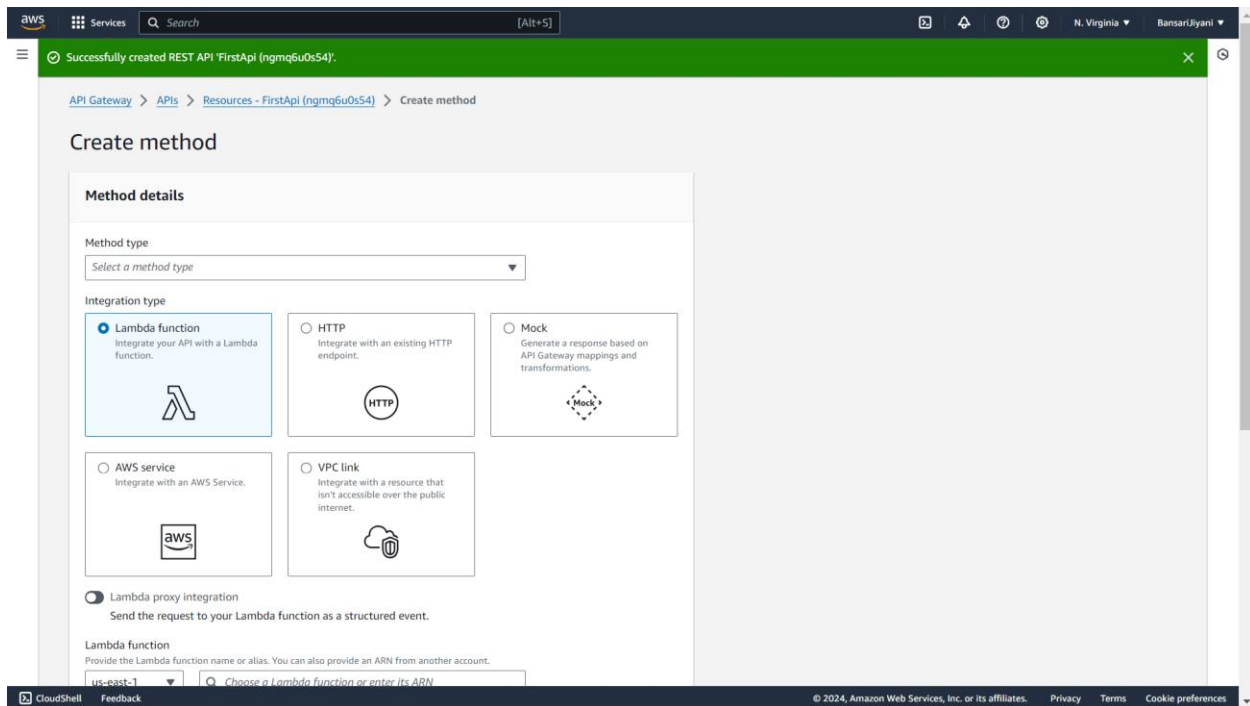
Methods (0)

Delete Create method

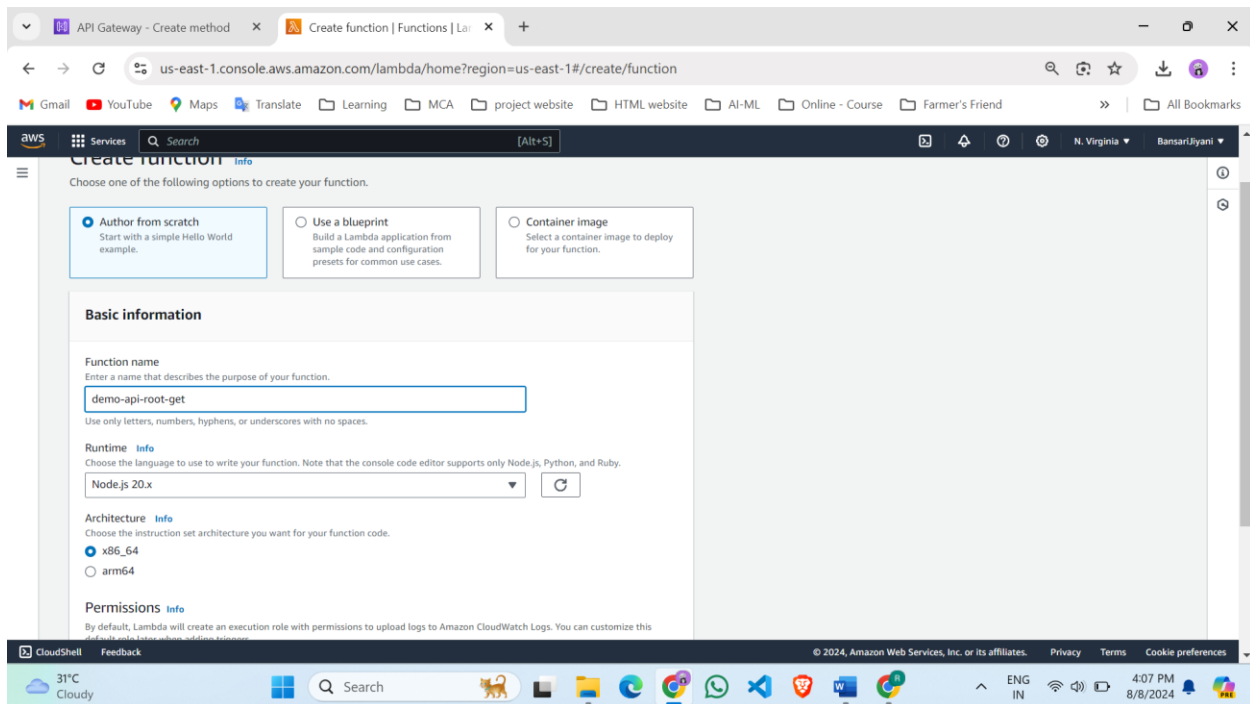
No methods

No methods defined.

6. Create method for API.



7. For create API we need a lambda function so first we create lambda function than create API and select python Runtime

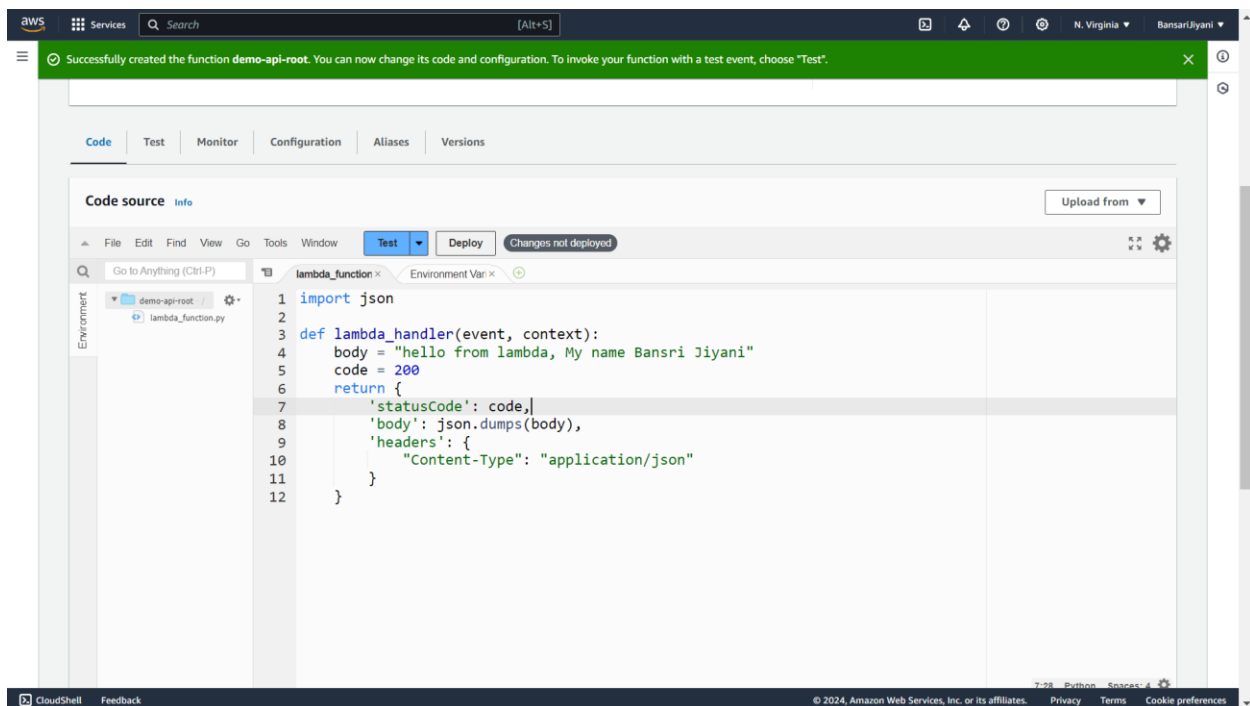


It is my lambda code

import json

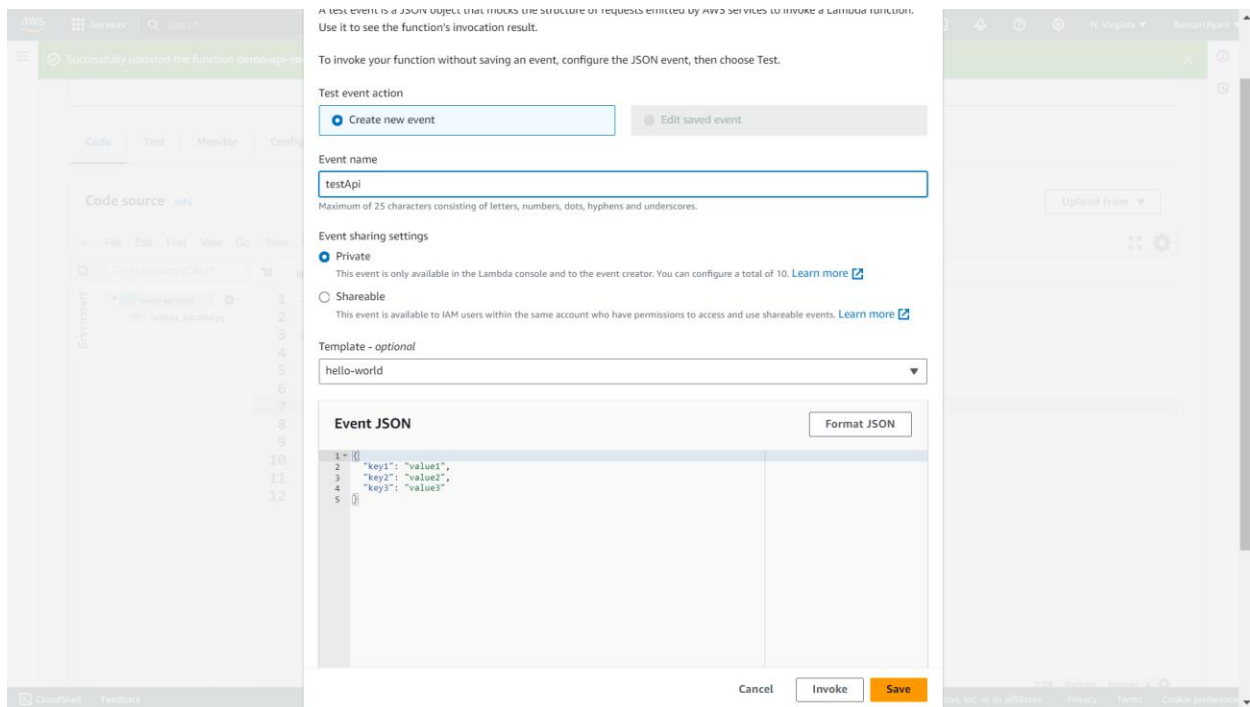
```
def lambda_handler(event, context):
    body = "hello from lambda, My name Bansri Jiyan"
    code = 200
    return {
        'statusCode': code,
        'body': json.dumps(body),
        'headers': {
            "Content-Type": "application/json"
        }
    }
```

Copy this code and paste it into lambda function and deploy it

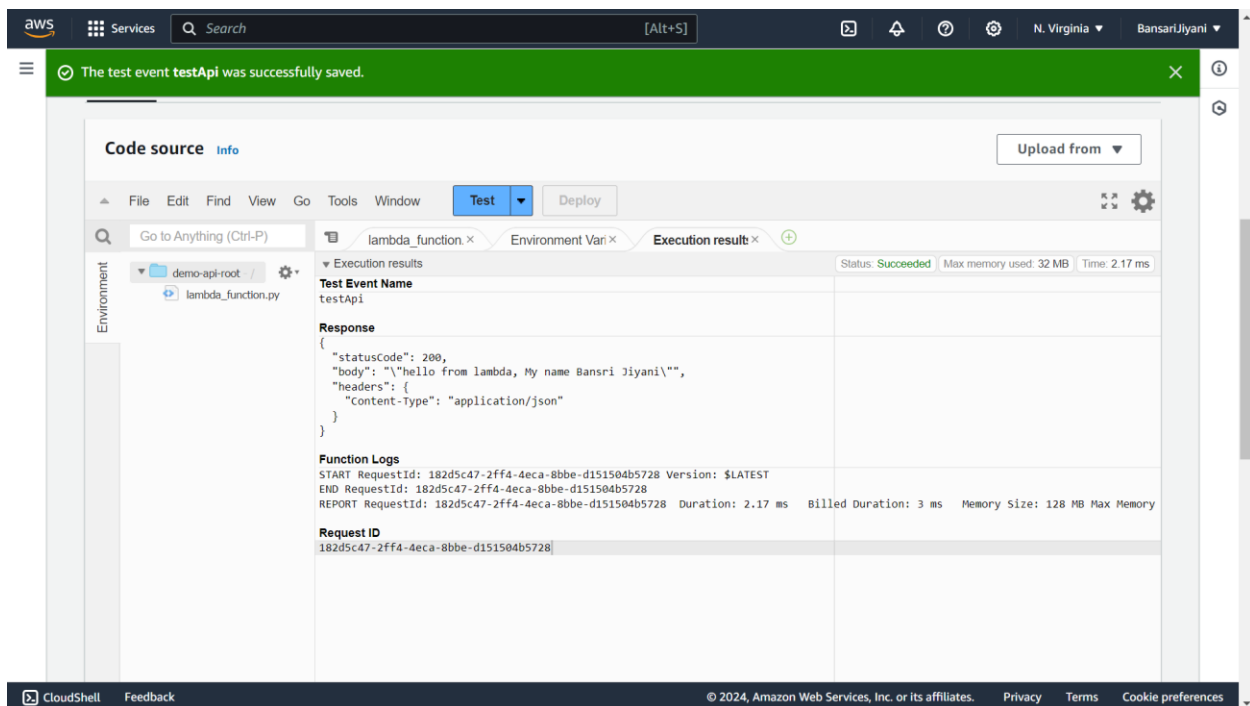


We can also check it is working or not .

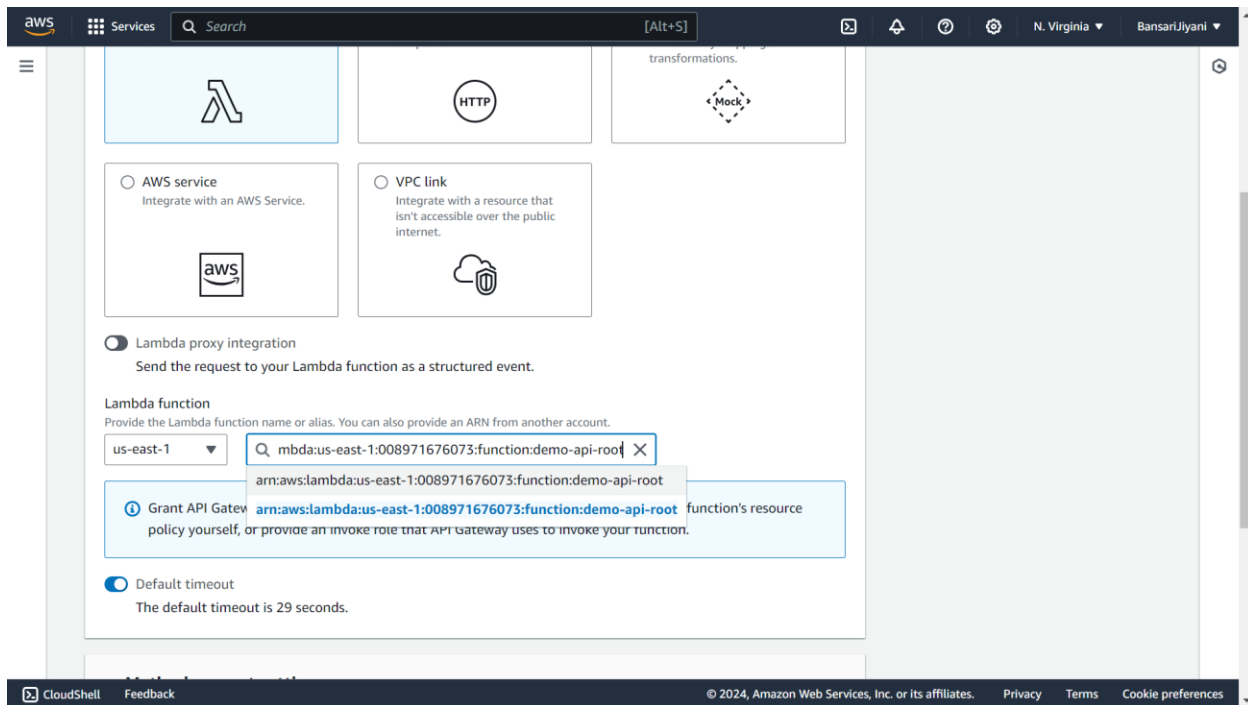
Go to the Test and create event and save it and again test it.



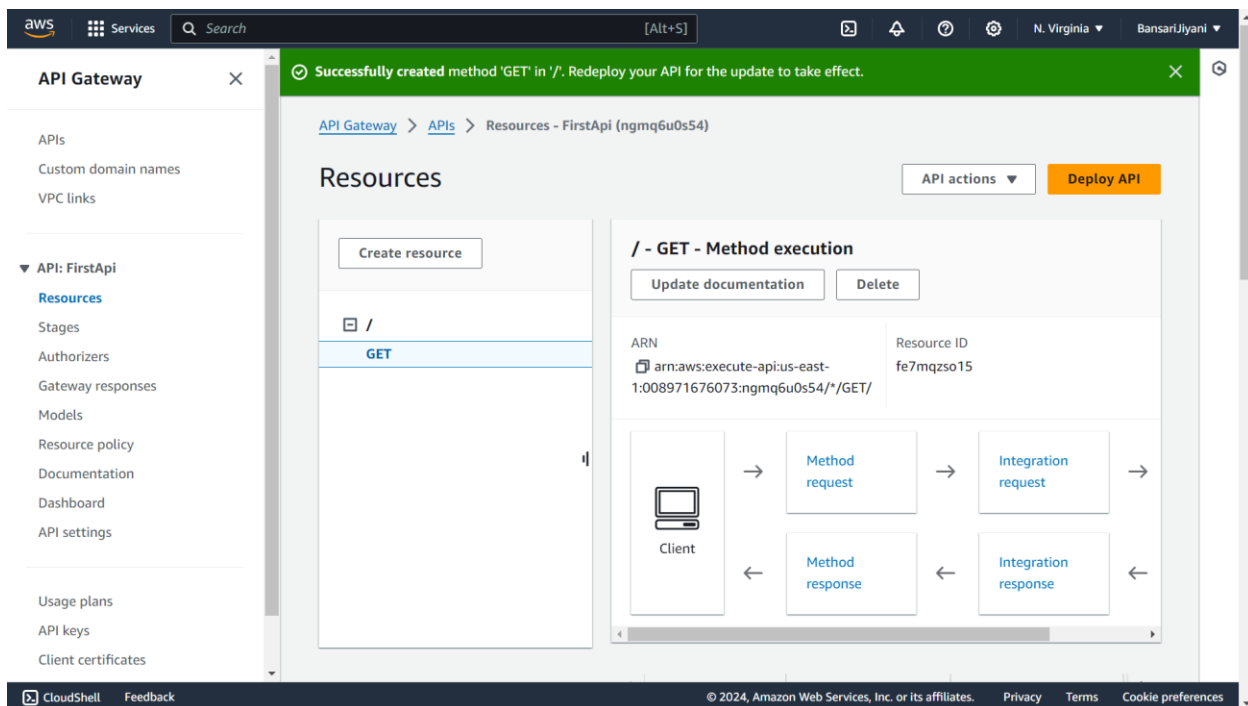
When we again test it than we can get the Output



Copy the URL of function and paste it in API creation and create the method



Now API is automatic runway



We can check API configuration in lambda function

demo-api-root

Function overview

Diagram Template

demo-api-root

Layers (0)

API Gateway

+ Add trigger

+ Add destination

Description

Last modified 27 minutes ago

Function ARN: arn:aws:lambda:us-east-1:008971676073:function:demo-api-root

Function URL

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

Triggers (1)

Find triggers

Trigger

API Gateway: FirstApi

arn:aws:execute-api:us-east-1:008971676073:ngmq6u0s54/*/*GET/

Details

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

And also check the permission

Execution role

Role name: demo-api-root-role-lyzh2ow0

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

Amazon CloudWatch Logs

3 actions, 2 resources

By action By resource

Resource	Actions
arn:aws:logs:us-east-1:008971676073:*	Allow: logs:CreateLogGroup
arn:aws:logs:us-east-1:008971676073:log-group:/aws/lambda/demo-api-root:*	Allow: logs:CreateLogStream Allow: logs:PutLogEvents

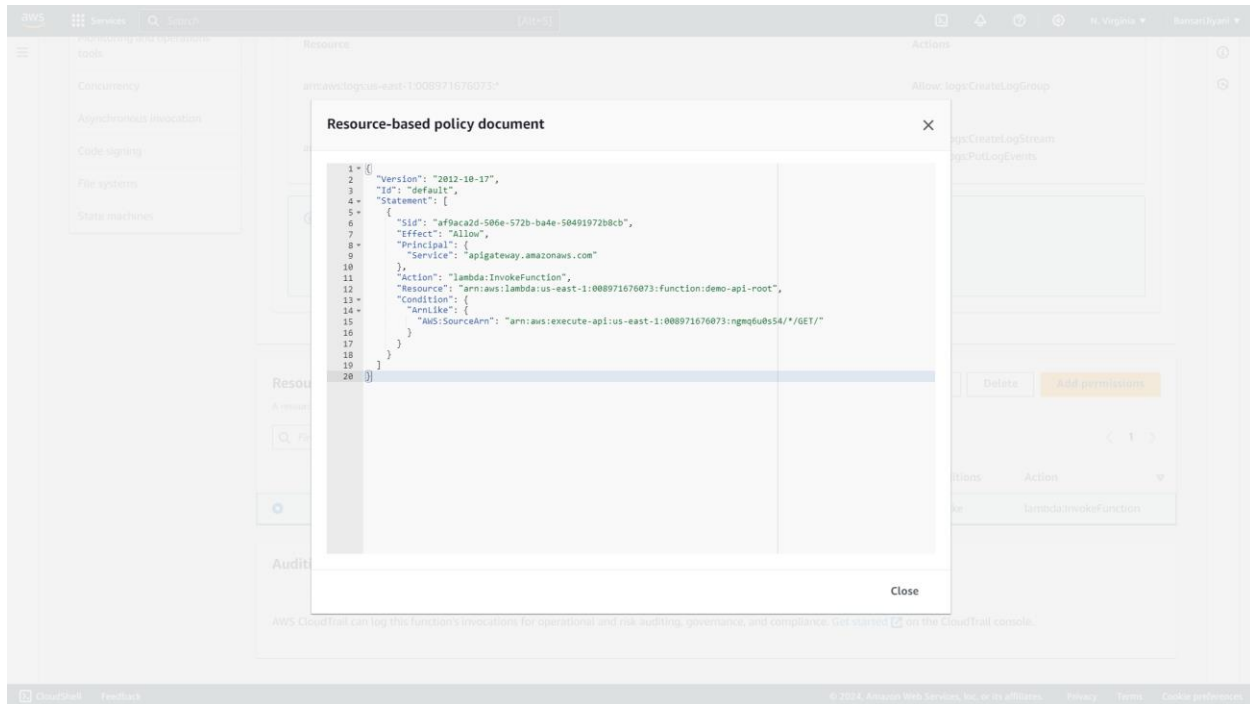
Lambda obtained this information from the following policy statements:

- Managed policy AWSLambdaBasicExecutionRole-b1f98de4-f2a4-4fec-abe6-4393e295a1ce, statement 0
- Managed policy AWSLambdaBasicExecutionRole-b1f98de4-f2a4-4fec-abe6-4393e295a1ce, statement 1

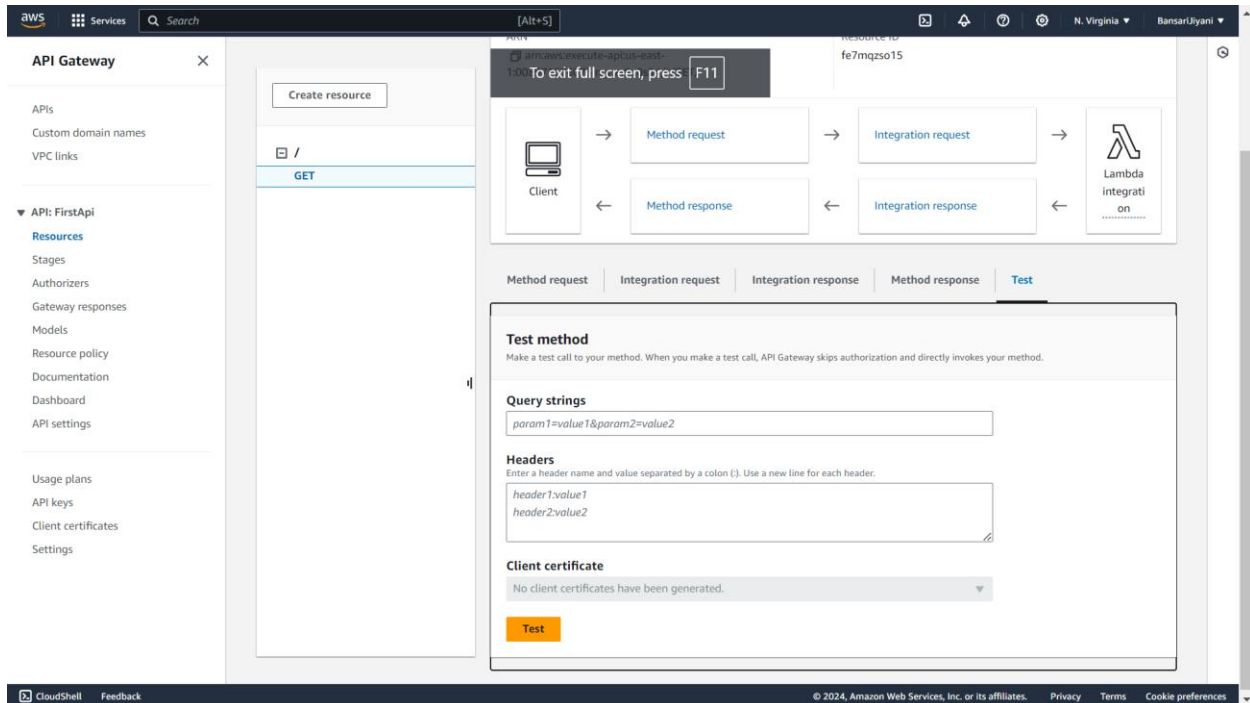
CloudShell Feedback

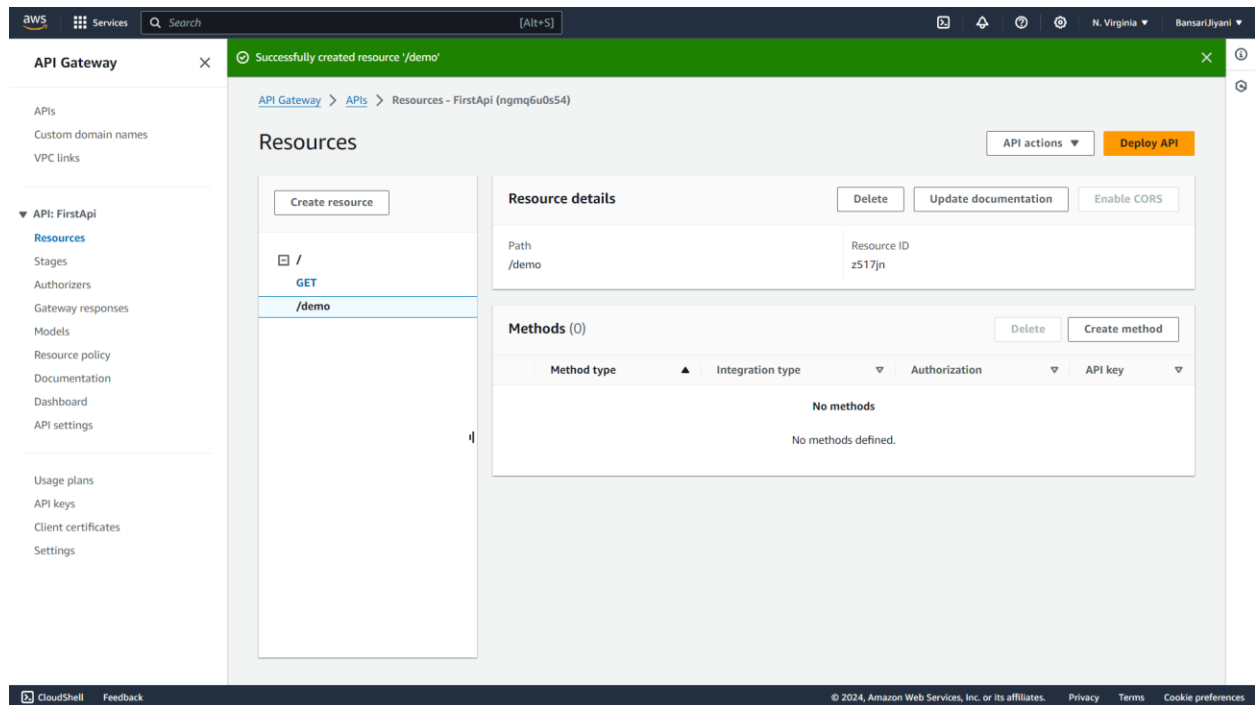
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Check the view policy of API



Now test the api

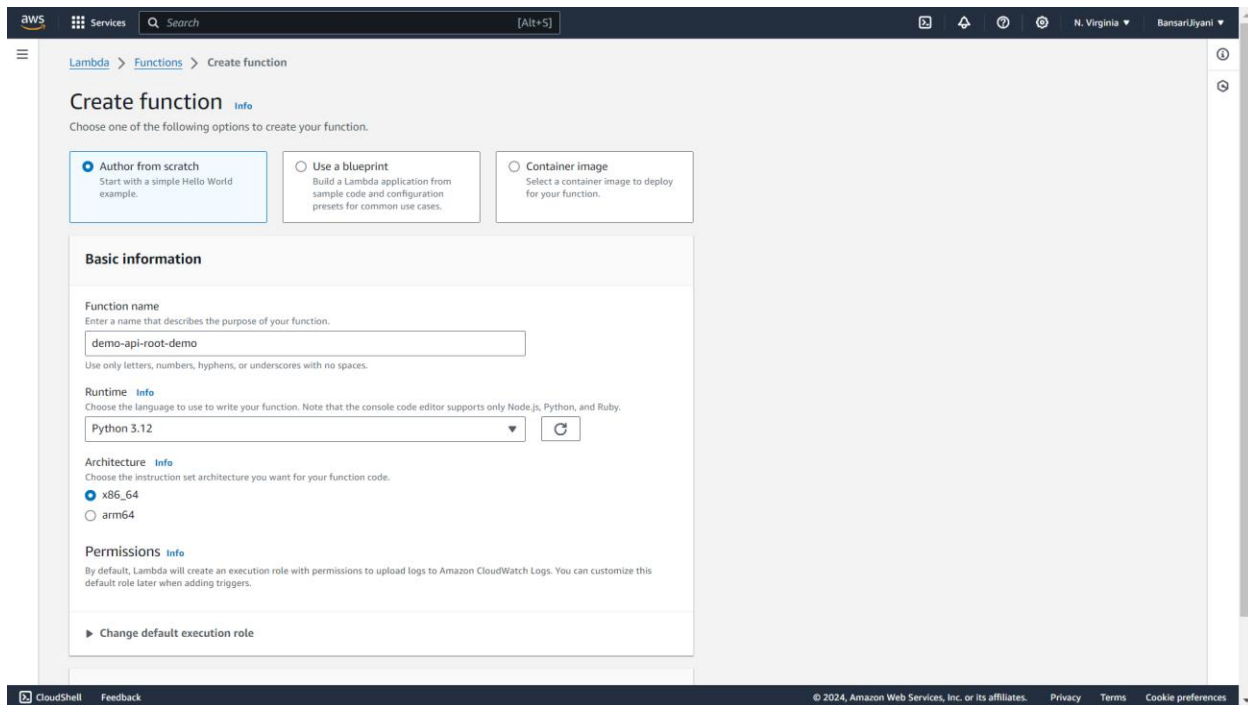




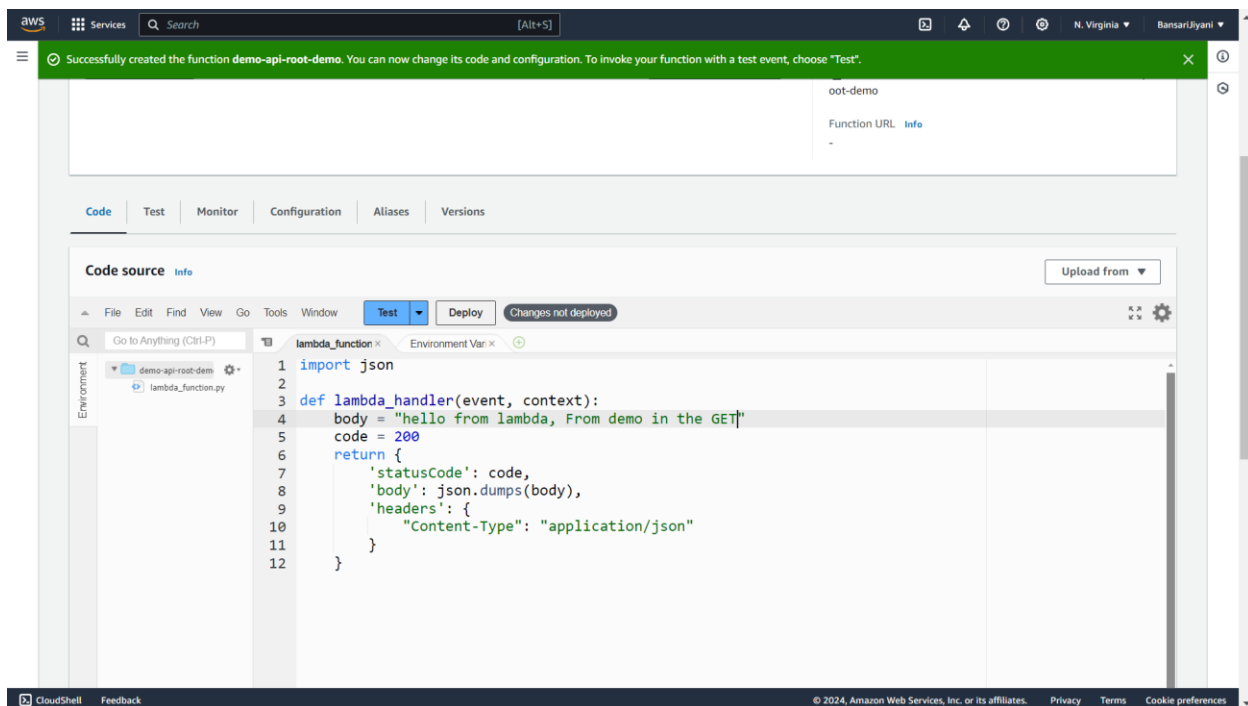
In this demo we create another new method

And for create new method we need the lambda function so firstly create it and then we create new method.

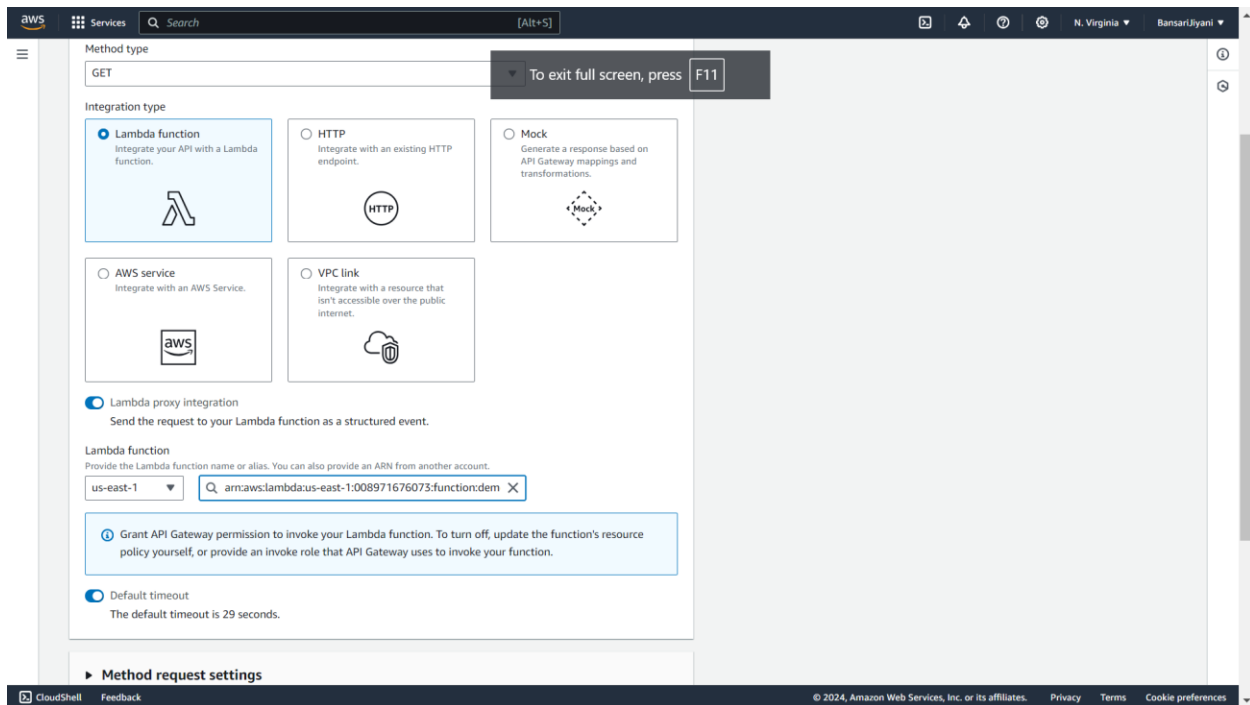
Create lambda fun :-



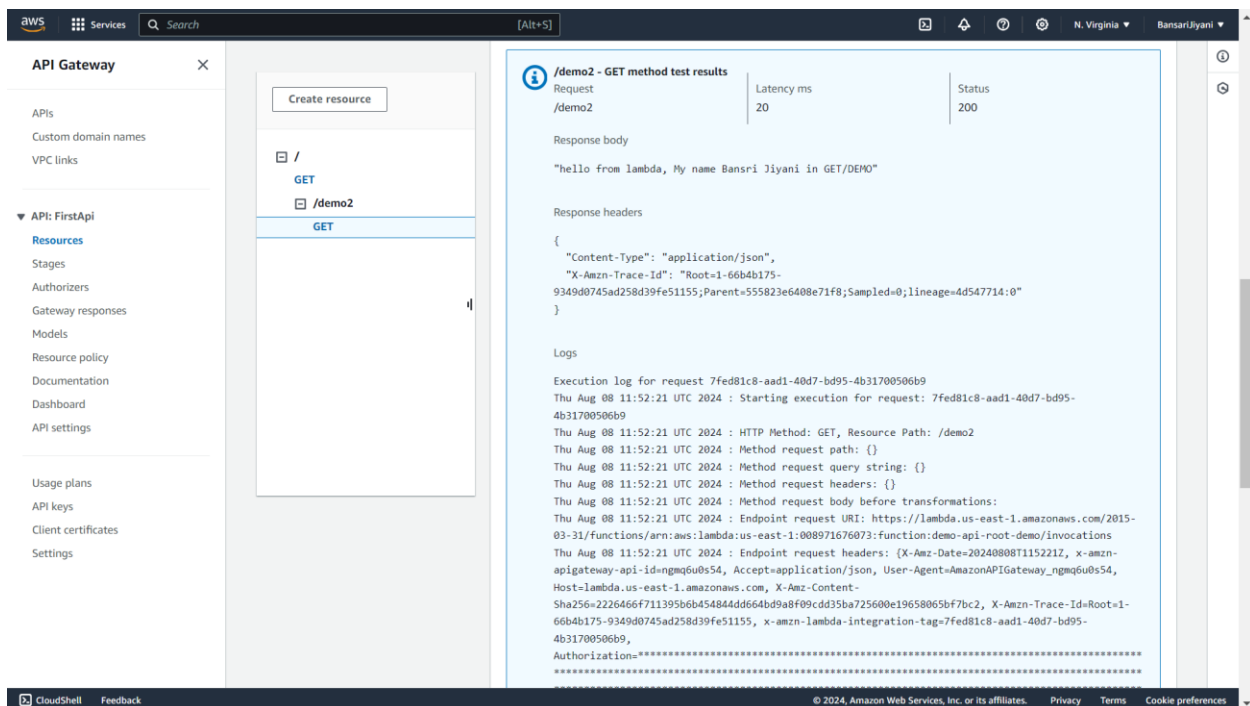
Add above code into this function also



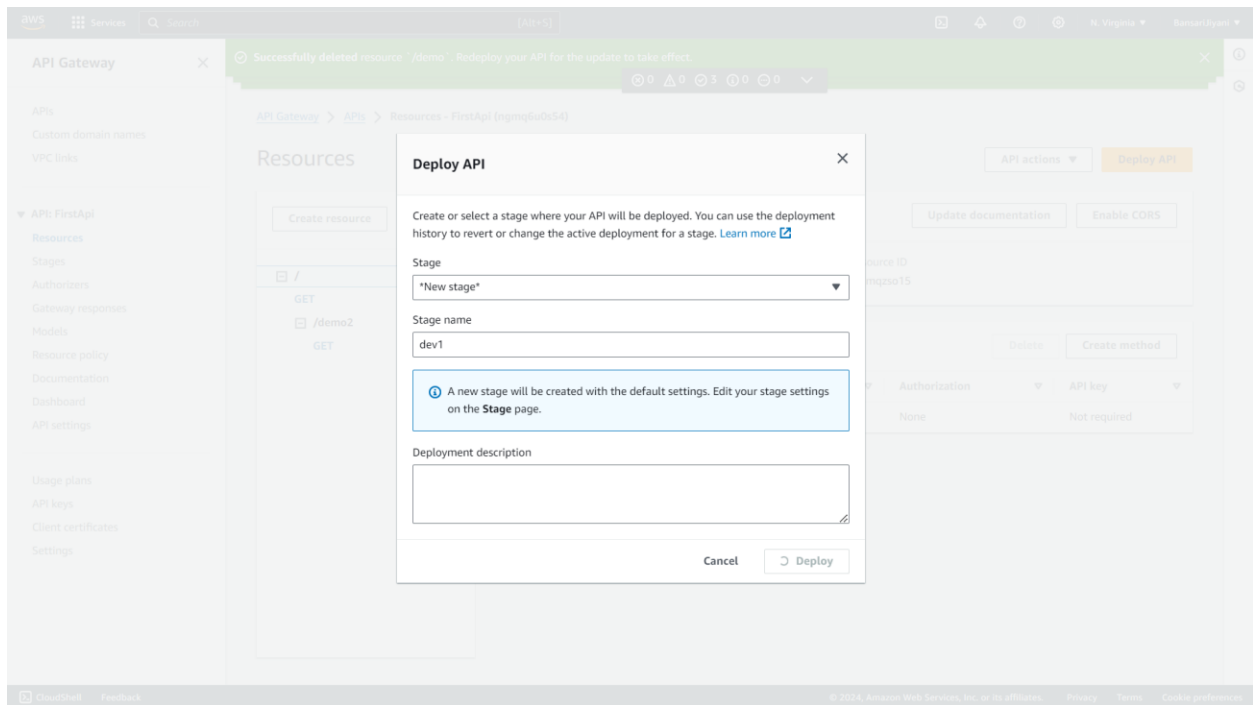
Copy this ARN link and paste it into new demo api



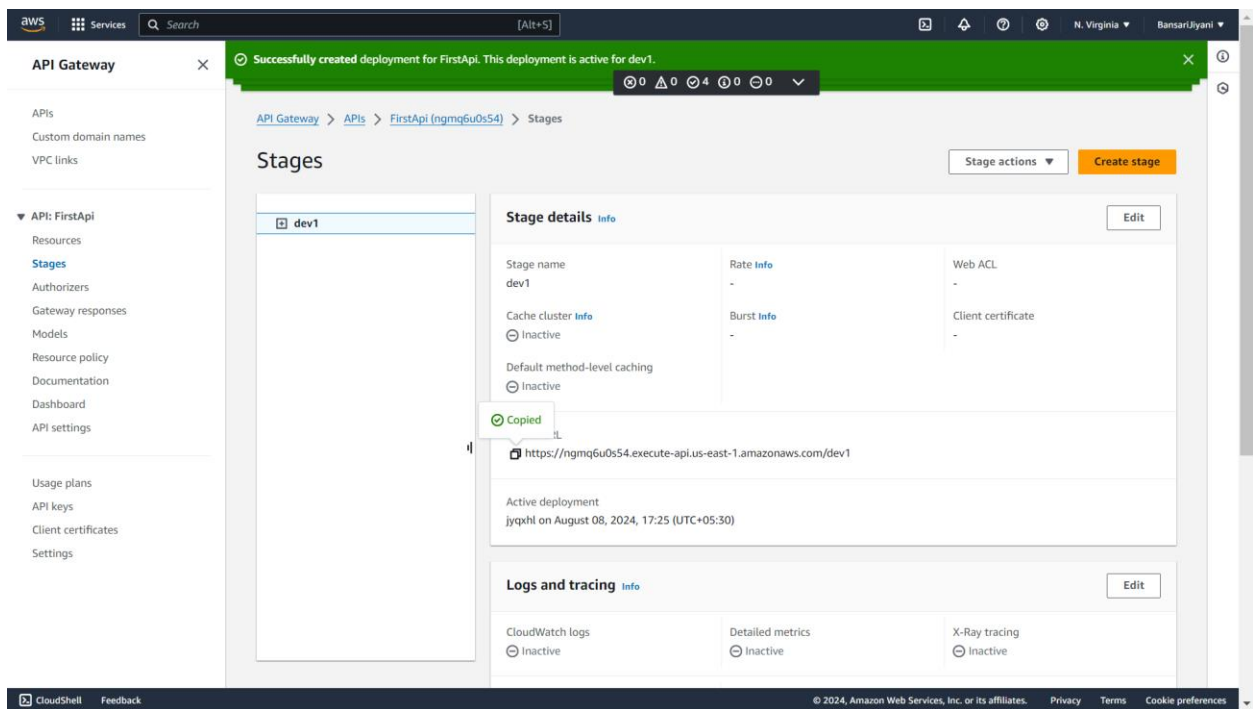
Create and test it.



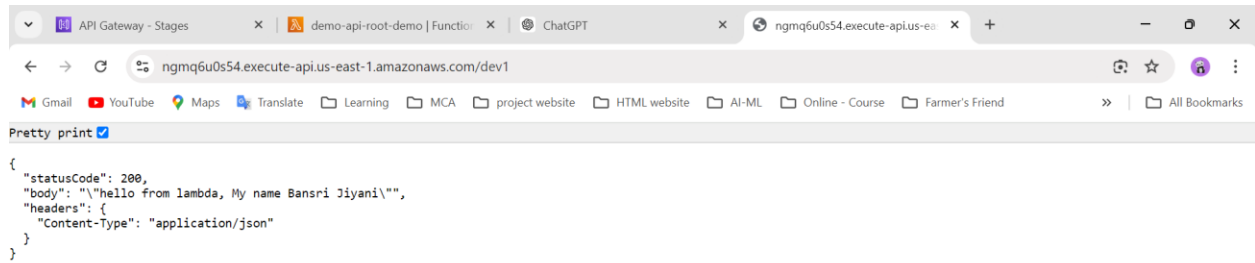
Now we deploy or test API using browser



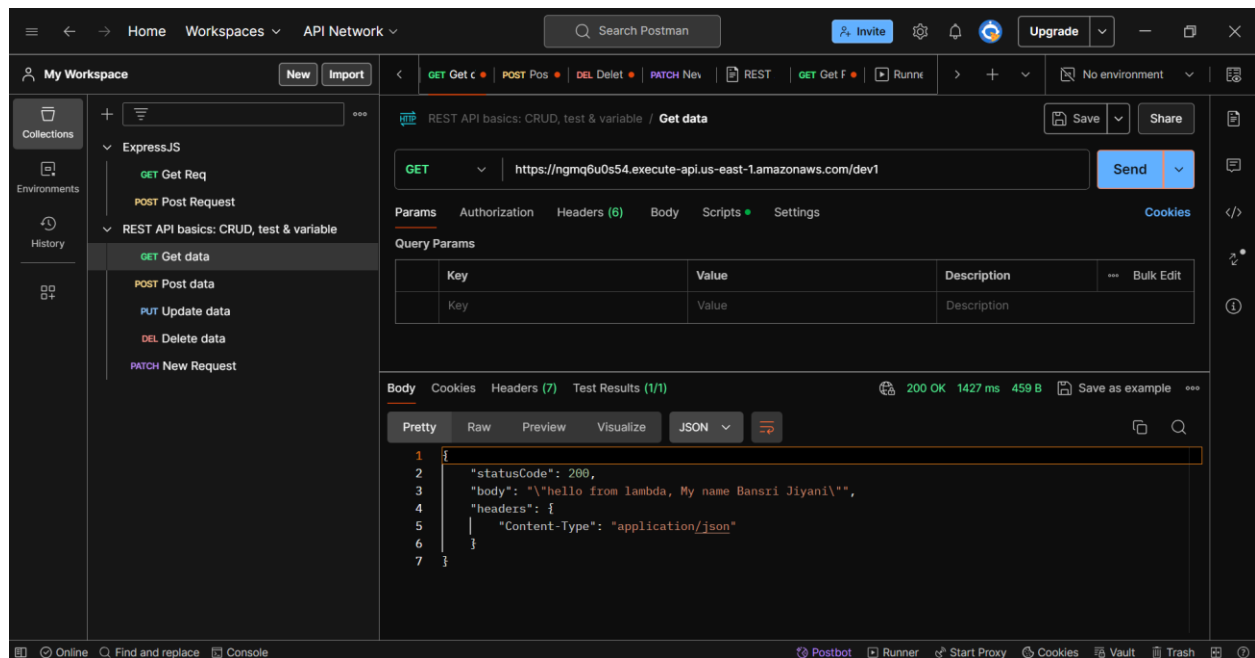
Copy this invoke URL



And paste it in browser



Check this get req in postman



For root api give this path :

