

Name: Manav Jawrani

Roll No.: 19

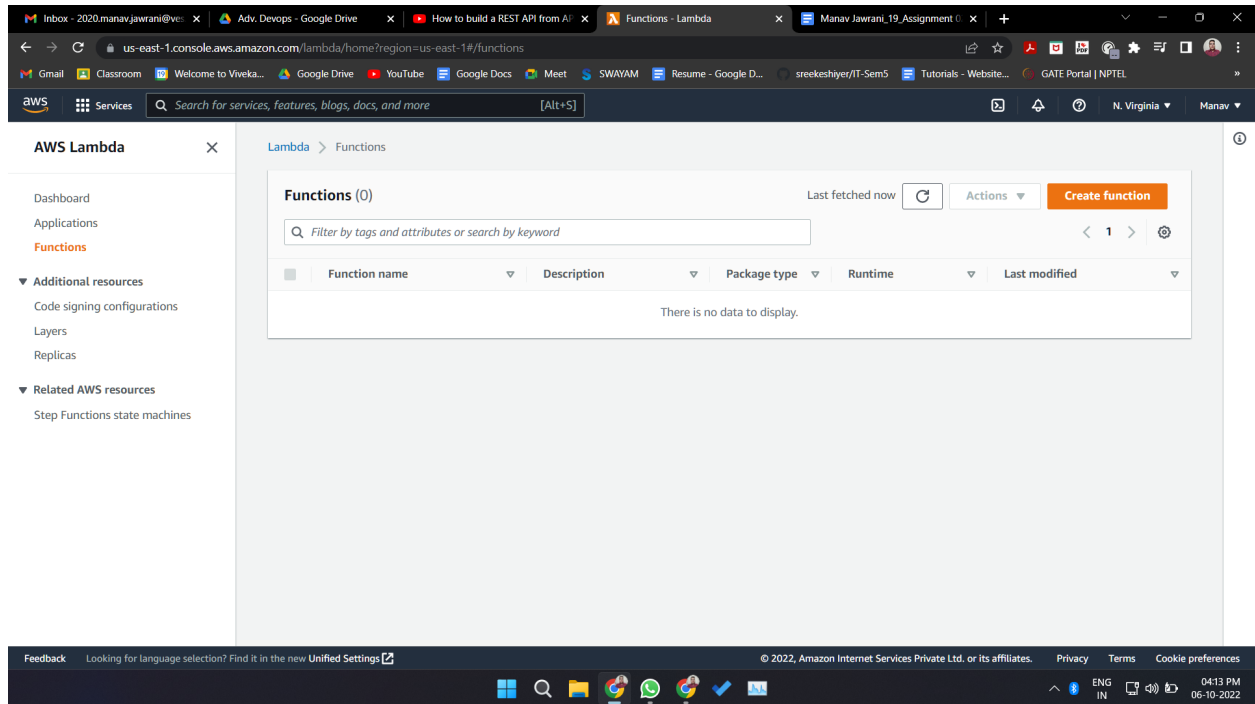
Subject: Advanced DevOps

Assignment No.: 2

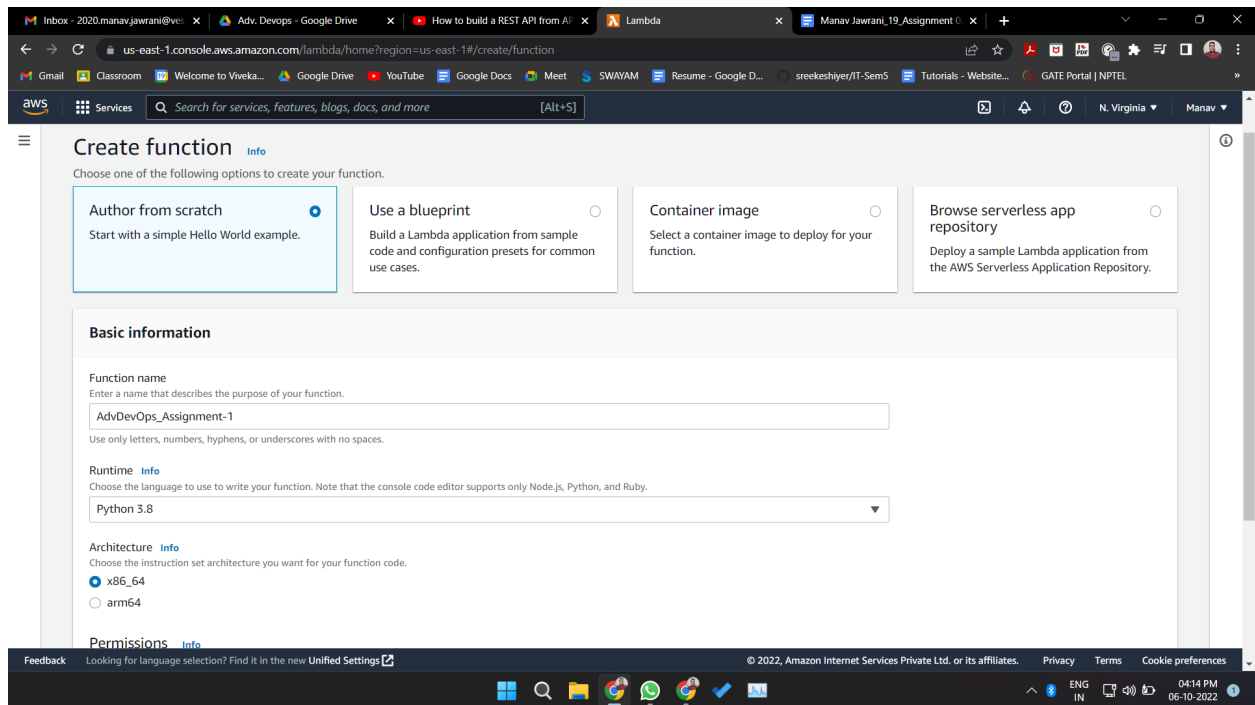
Implementation:

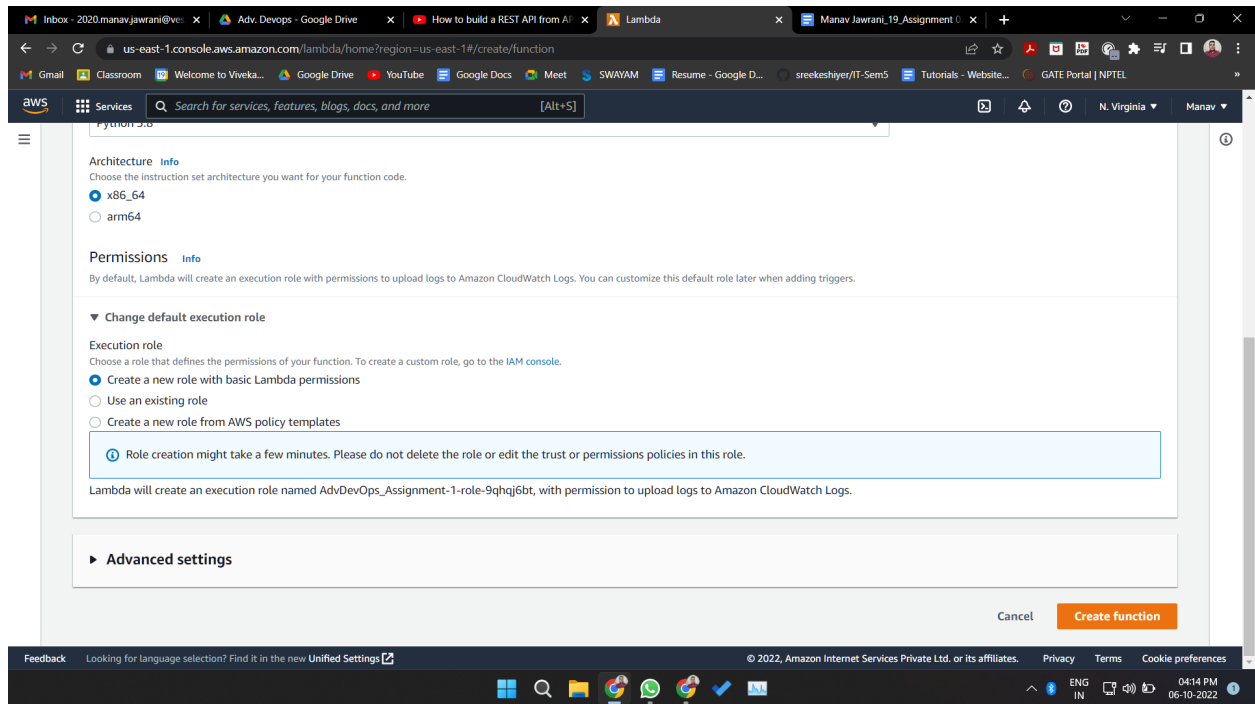
Prerequisites: AWS Free Tier account

Step 1: Login to your AWS Account and the Lambda Function Console.

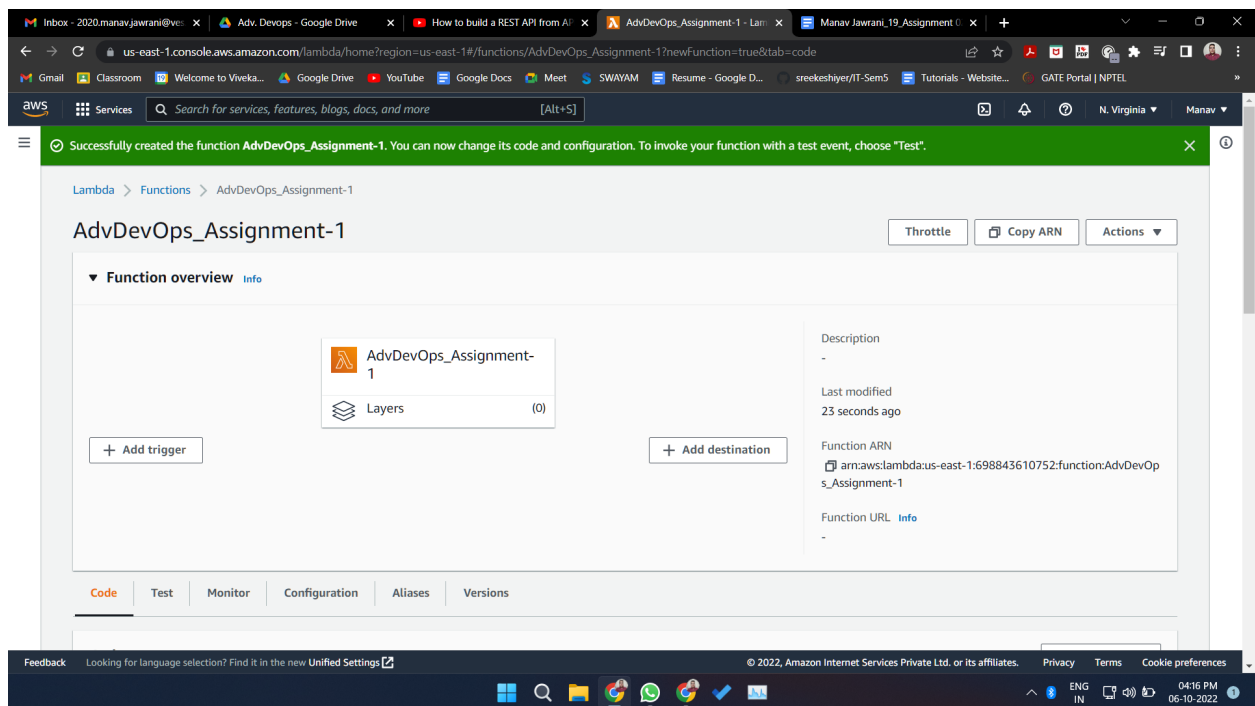


Step 2: Create an AWS Lambda function with runtime as Python 3.8.





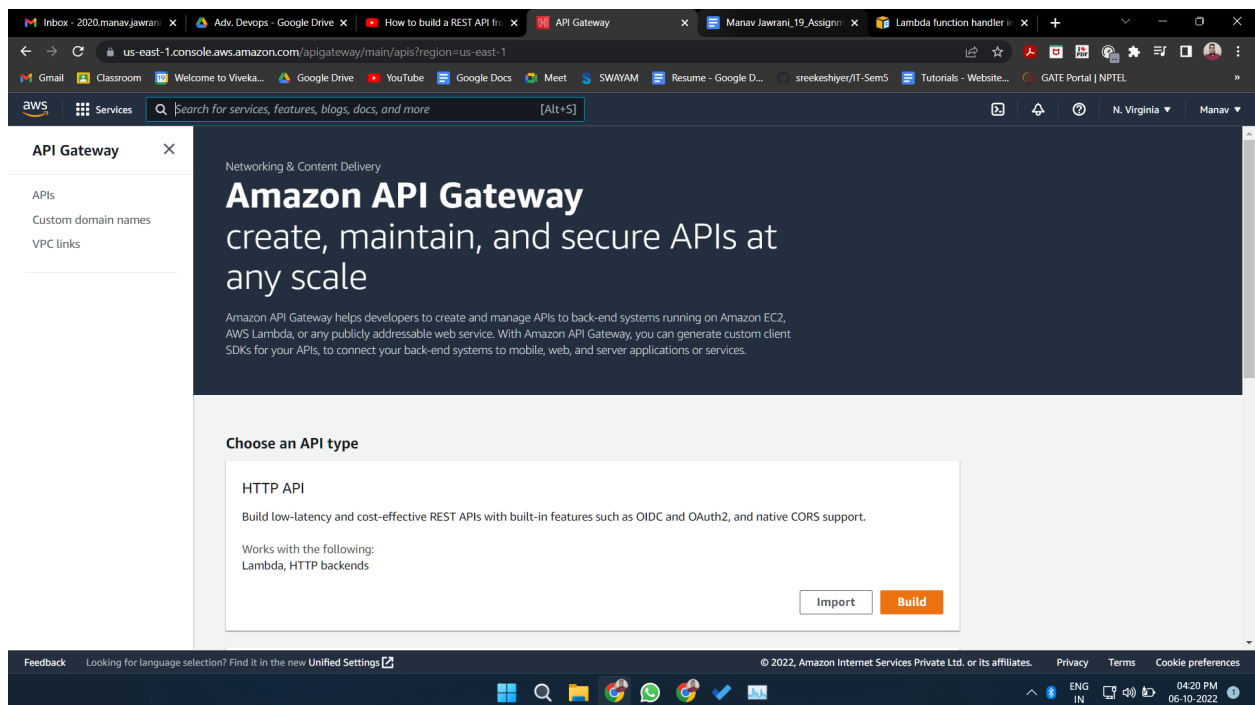
Then click on “Create Function”



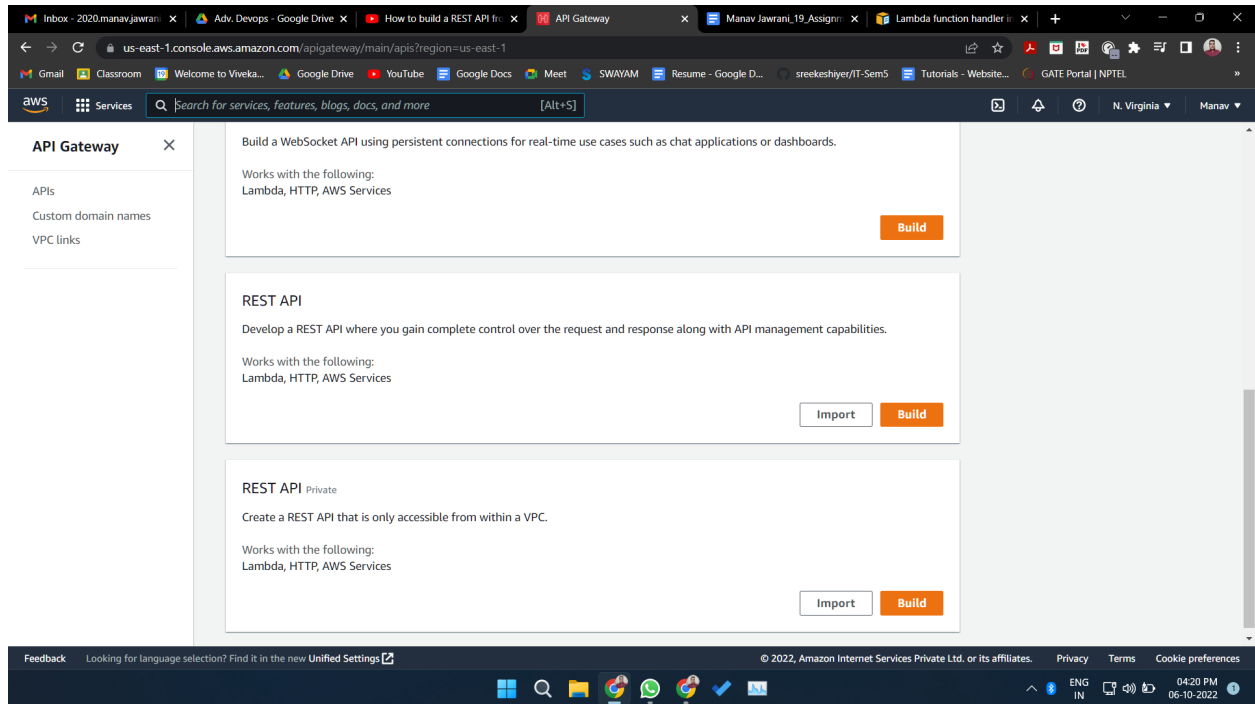
Step 3: Write the code for the handler which will be invoked after input from the user and save it.

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     first_name = event["queryStringParameters"]["first_name"]
6     last_name = event["queryStringParameters"]["last_name"]
7
8     app_response = {}
9
10    app_response['message'] = f'The details are {first_name} and {last_name}'
11    app_response['profession'] = 'Student'
12    app_response['age'] = 19
13
14
15    responseObject = {}
16    responseObject['statusCode'] = 200
17    responseObject['headers'] = {}
18    responseObject['headers']['Content-Type'] = 'application/json'
19    responseObject['body'] = json.dumps(app_response)
20
21    return responseObject
```

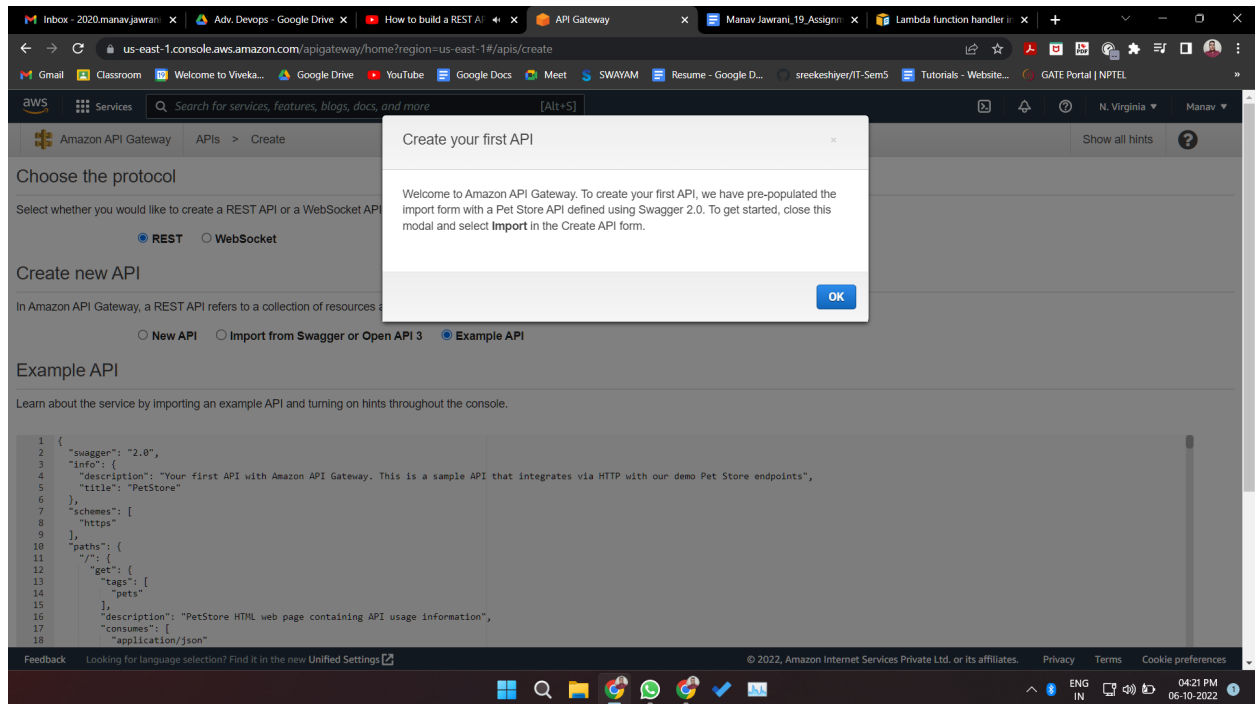
Step 4: Now leave the function as it is and the go to the Amazon API Gateway.



Now look for the REST API and build it



Click on “Build”.



Step 5: You see this type of page choose “**REST**” and “**New API**”. Then click on “**Create API**”.

The screenshot shows the 'Create API' page in the Amazon API Gateway console. The browser tabs include 'Inbox - 2020.manav.jawrani@v...', 'Adv. Devops - Google Drive', 'How to build a REST API', 'API Gateway', 'Manav Jawrani_19_Assignm...', and 'Lambda function handler...'. The URL is 'us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/create'. The page has a header with the AWS logo, 'Services', a search bar, and user information 'N. Virginia' and 'Manav'. The main content area is titled 'Choose the protocol' and 'Create new API'. Under 'Choose the protocol', 'REST' is selected. Under 'Create new API', 'New API' is selected. The 'Settings' section has a form with 'API name*' set to 'Assignment-2', 'Description' empty, and 'Endpoint Type' set to 'Regional'. A 'Create API' button is at the bottom right. The footer includes 'Feedback', 'Looking for language selection?', '© 2022, Amazon Internet Services Private Ltd. or its affiliates.', 'Privacy', 'Terms', 'Cookie preferences', and system icons.

Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

☒ REST ☐ WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API ☐ Import from Swagger or Open API 3 ☐ Example API

Settings

Choose a friendly name and description for your API.

API name* Assignment-2

Description

Endpoint Type Regional

* Required

Create API

After successful creation you will see this.

The screenshot shows the 'Resources' page in the Amazon API Gateway console. The browser tabs are the same as the previous screenshot. The URL is 'us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/jk0psan7he/resources/89jvodxf6'. The page has a header with the AWS logo, 'Services', a search bar, and user information 'N. Virginia' and 'Manav'. The main content area is titled 'Resources' and shows a list of resources. The 'API: Assignment-2' is selected, and the 'Resources' tab is active. The 'Resources' list shows a single resource with a path '/' and a status of 'No methods defined for the resource.' The left sidebar shows the navigation menu with 'APIs', 'Custom Domain Names', 'VPC Links', 'API: Assignment-2', 'Resources', 'Stages', 'Authorizers', 'Gateway Responses', 'Models', 'Resource Policy', 'Documentation', 'Dashboard', 'Settings', and 'Usage Plans'. The footer is the same as the previous screenshot.

APIs

Custom Domain Names

VPC Links

API: Assignment-2

Resources

Stages

Authorizers

Gateway Responses

Models

Resource Policy

Documentation

Dashboard

Settings

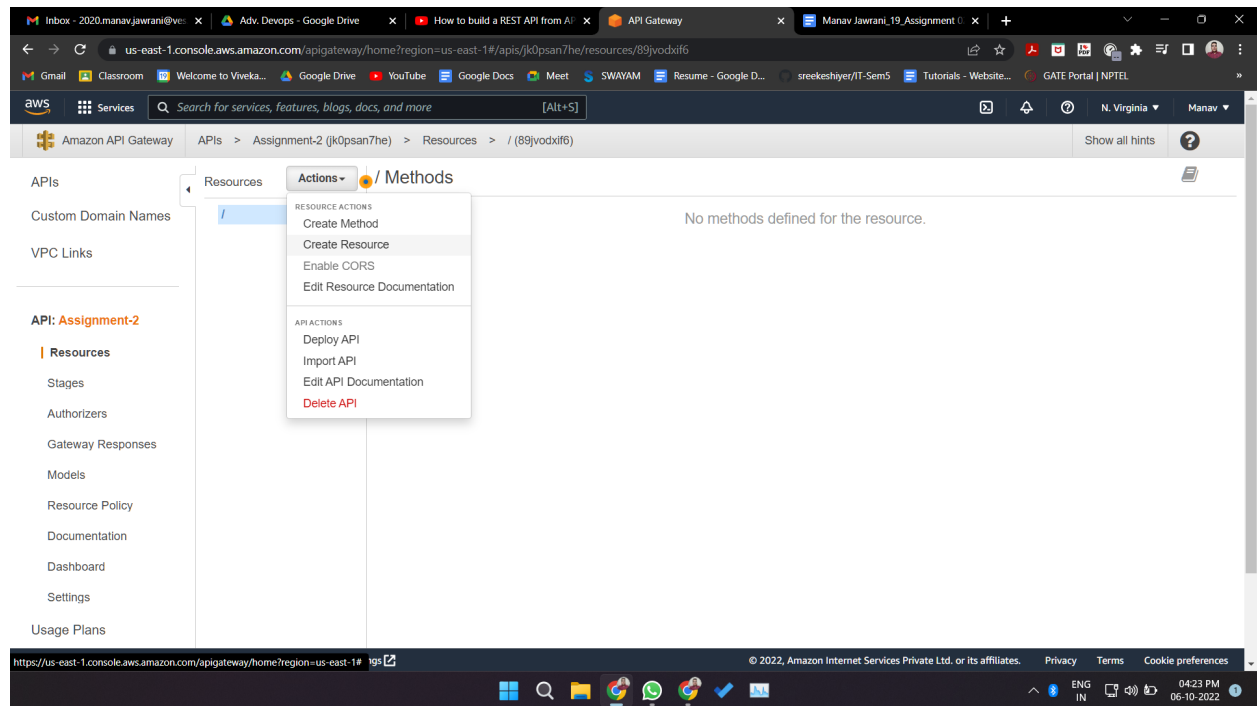
Usage Plans

Resources

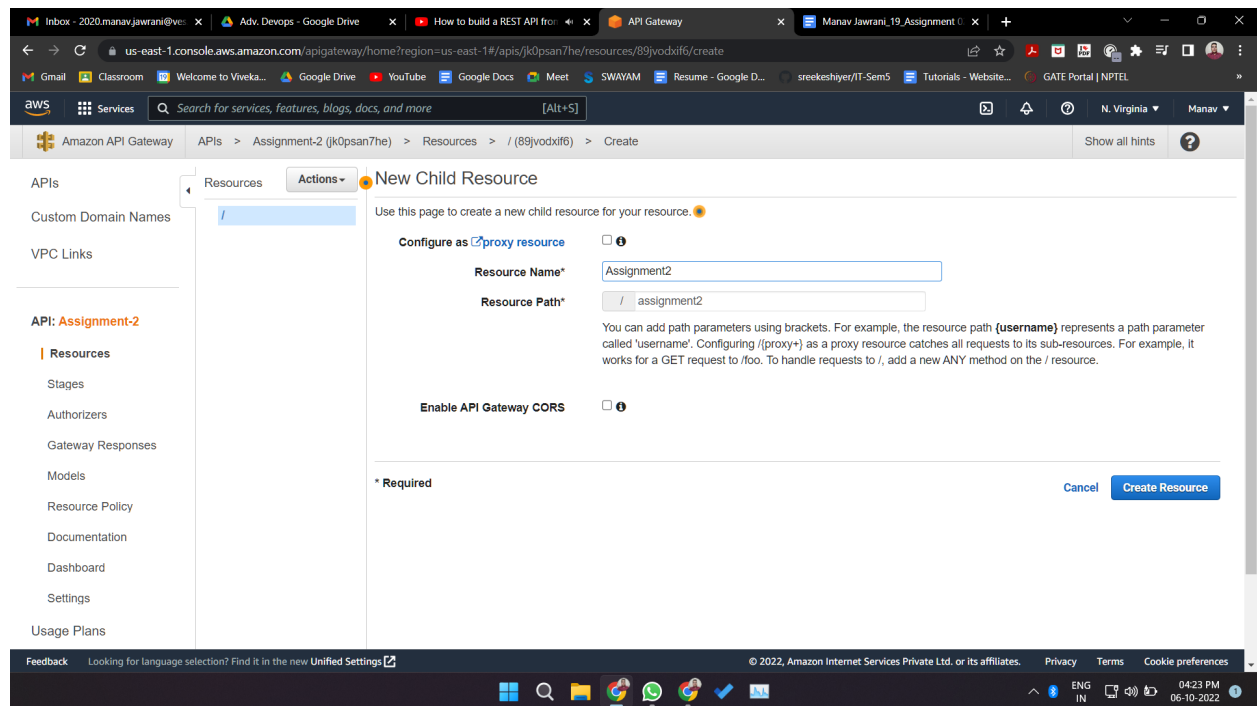
/

No methods defined for the resource.

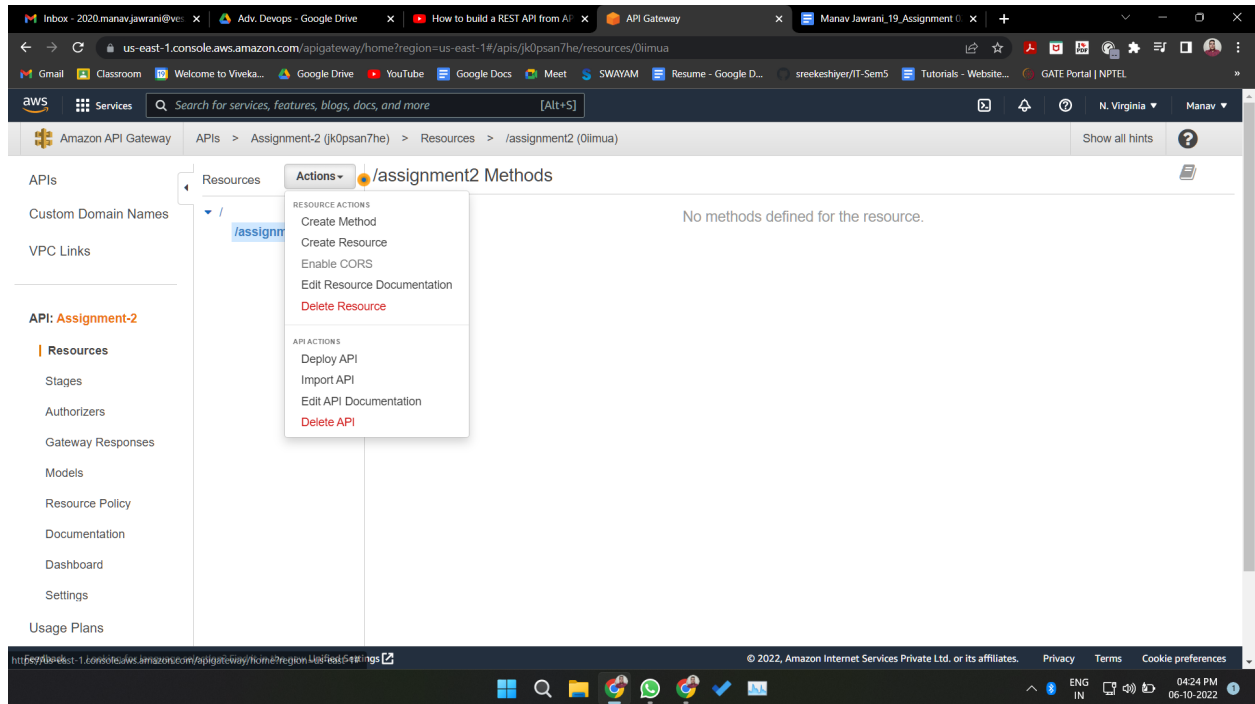
Step 6: Now , under “Actions” choose “Create Resource”.



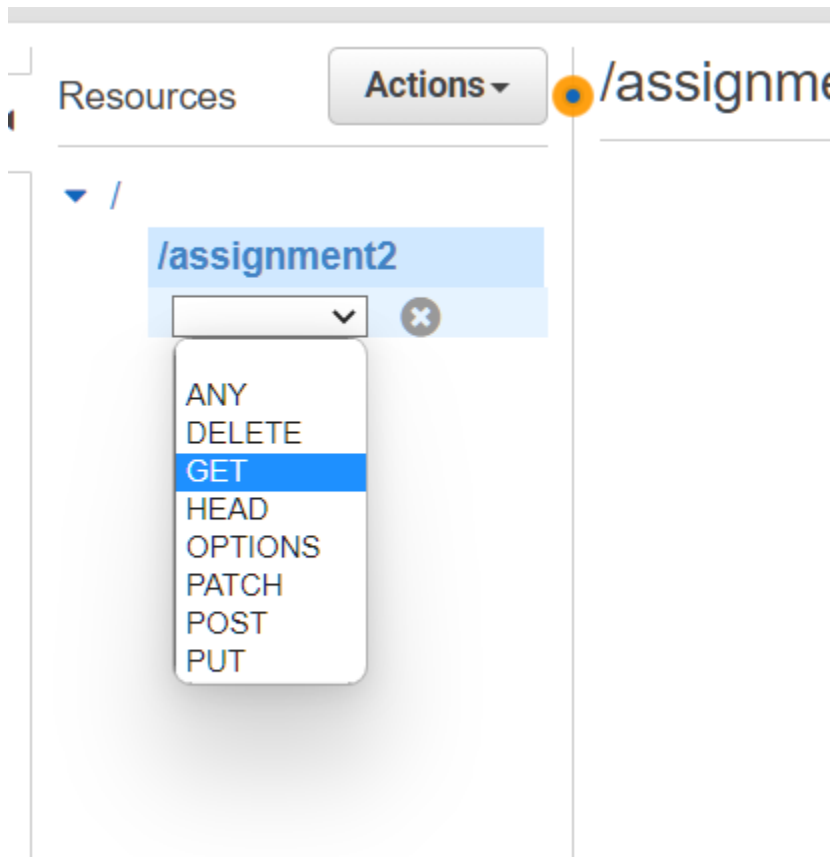
Configure the Resource and the create it.



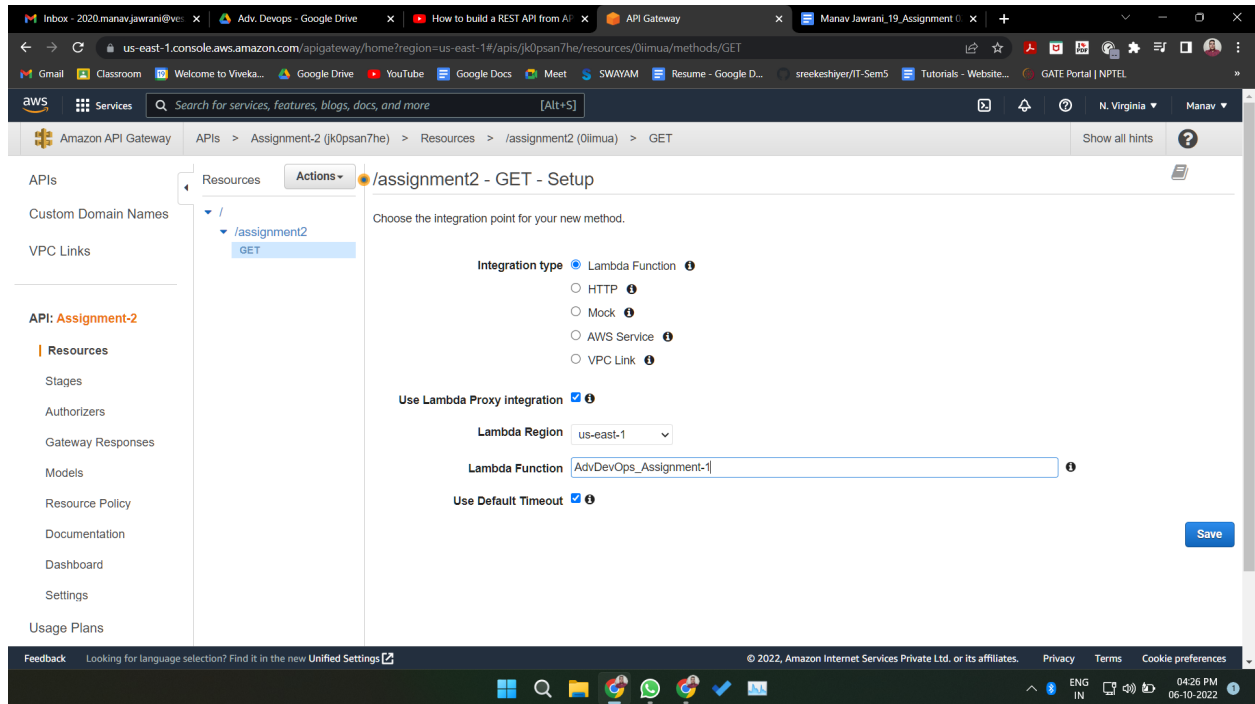
Step 7: Now , under Actions “Create Method”.



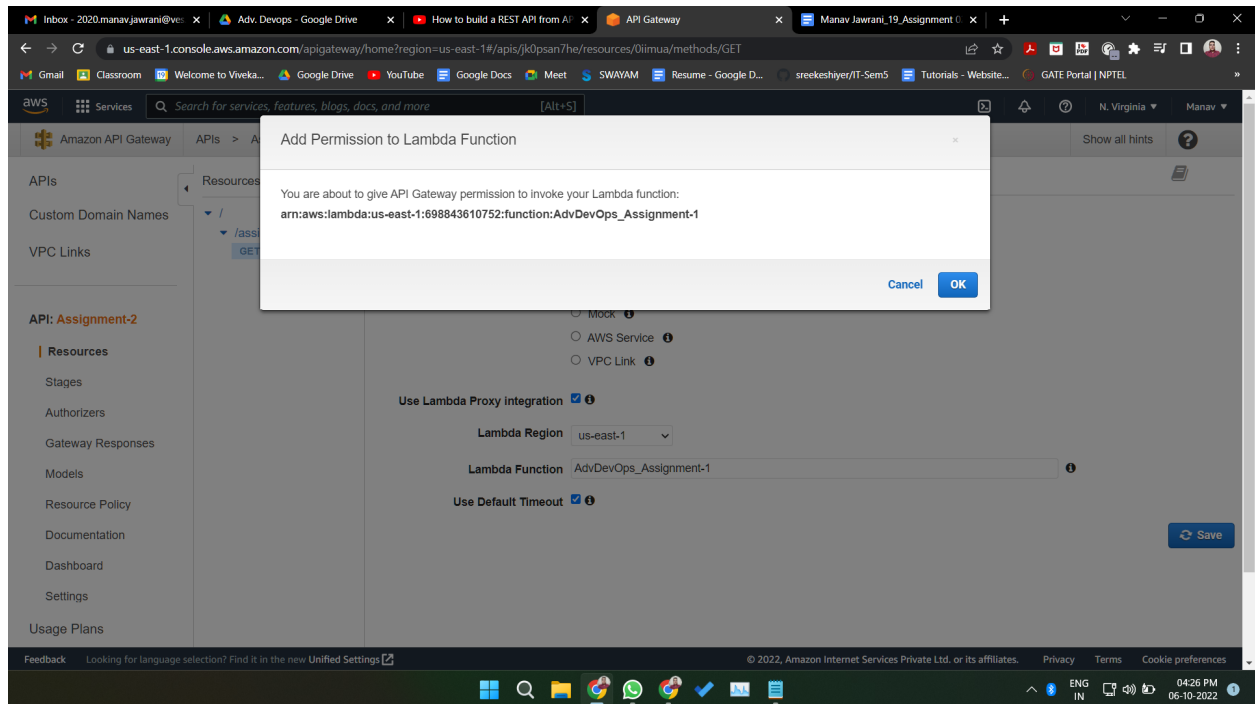
Select “GET”.



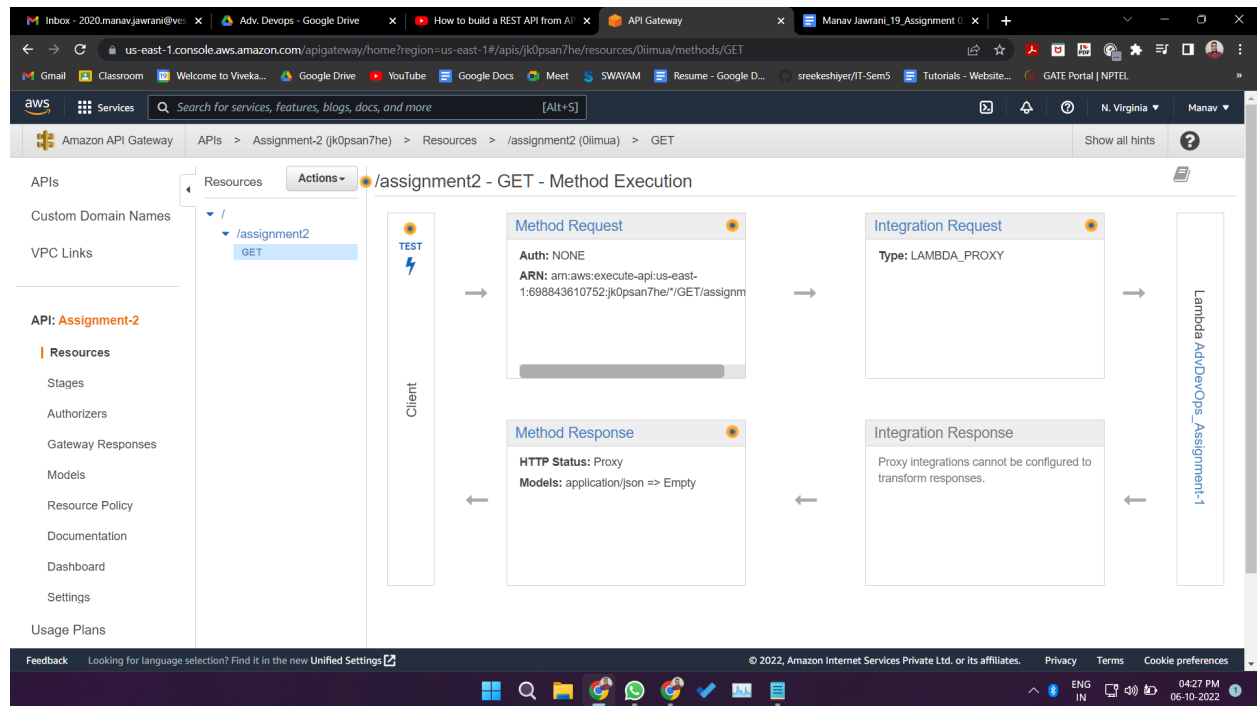
Do the necessary configuration and save it.



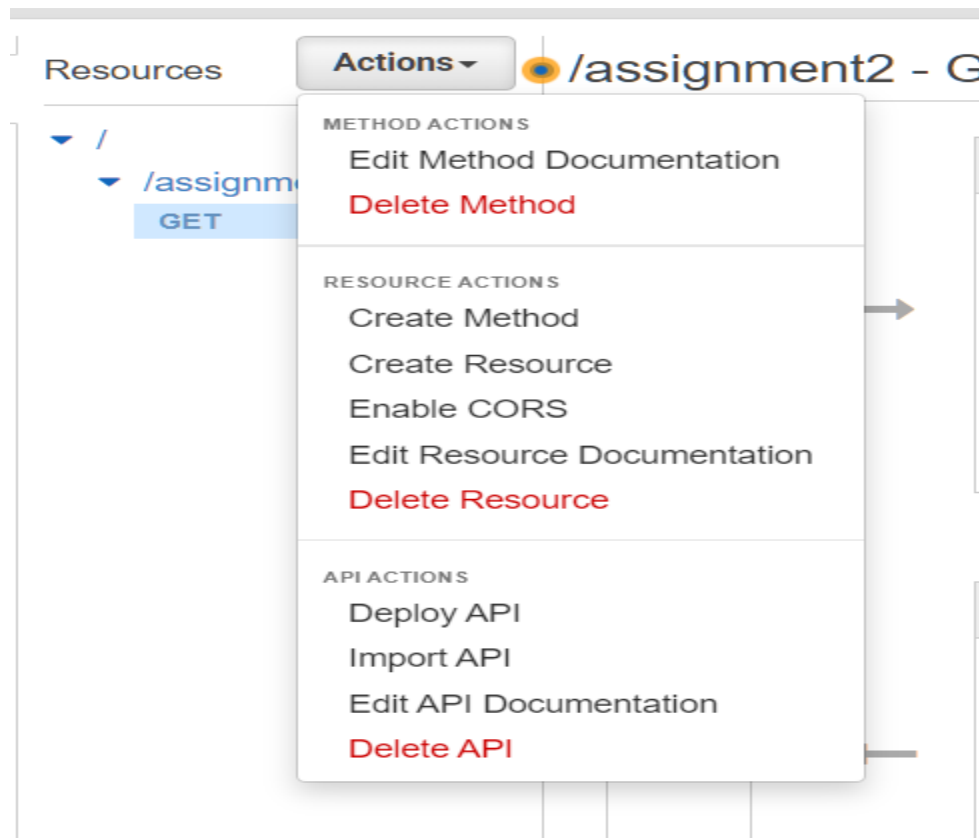
Step 8: Add the Permission of the method which we created previously.



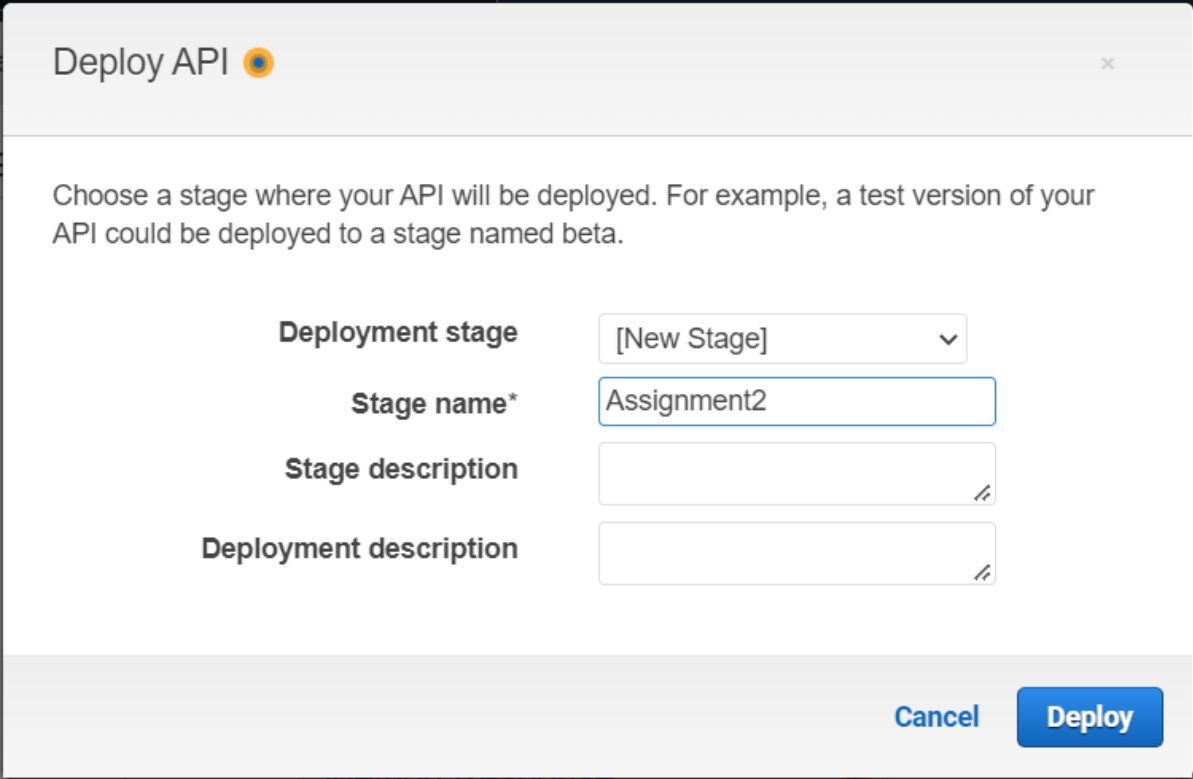
Step 9: After all the steps you are able to see this interface.



Step 10: Under Actions choose “Deploy API”.



Step 11: Create a new stage and then deploy it.



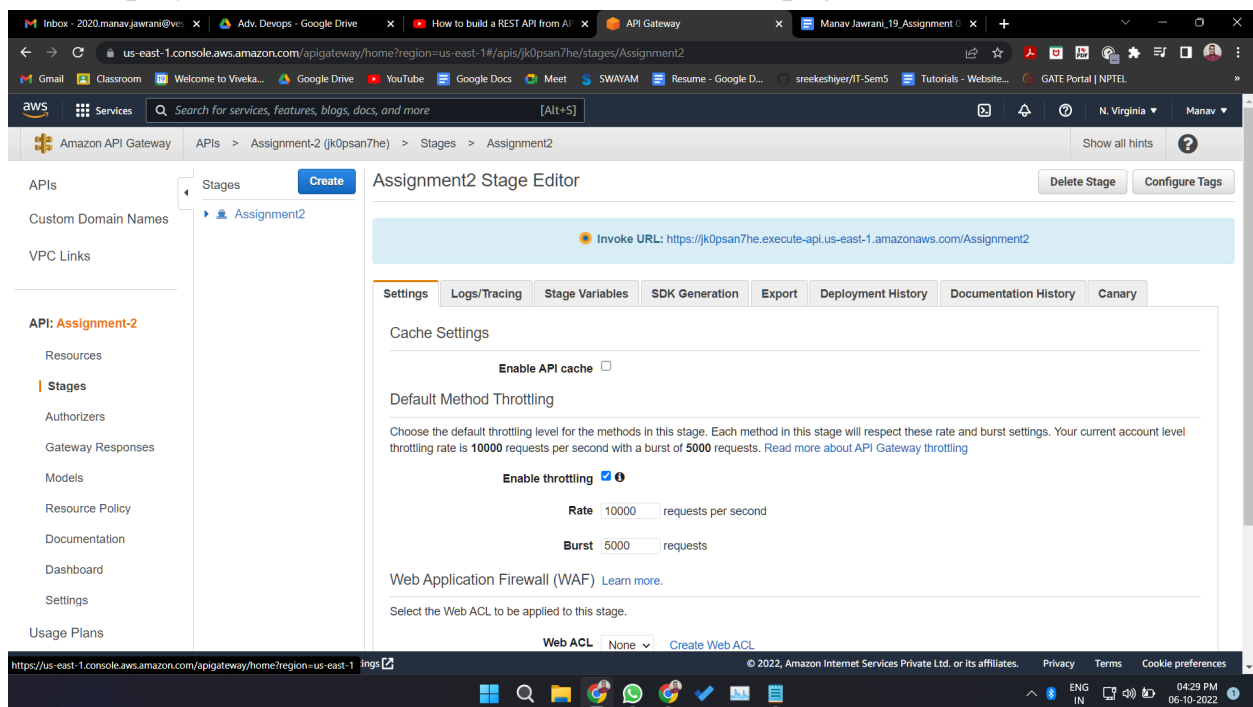
The image shows a 'Deploy API' dialog box with a close button in the top right corner. The text inside says: 'Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.'

The form contains the following fields:

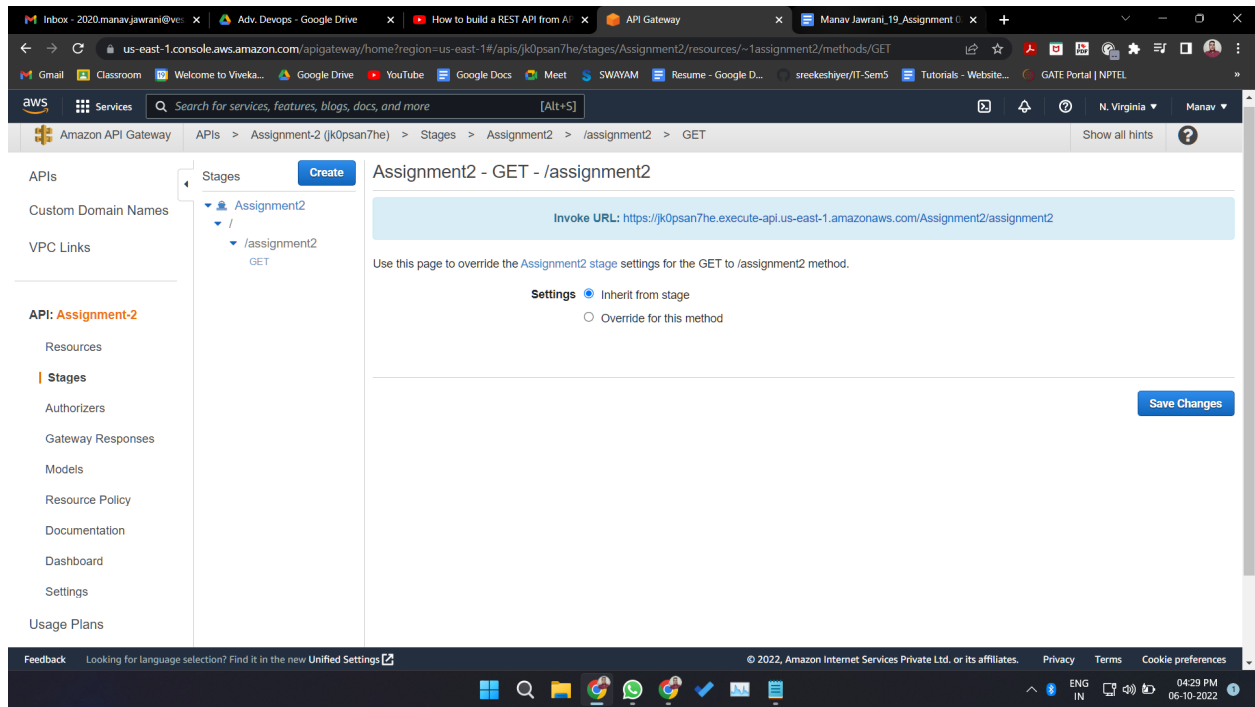
- Deployment stage:** A dropdown menu currently showing '[New Stage]'.
- Stage name*:** A text input field containing 'Assignment2'.
- Stage description:** An empty text area with a small icon in the bottom right corner.
- Deployment description:** An empty text area with a small icon in the bottom right corner.

At the bottom right of the dialog are two buttons: 'Cancel' and 'Deploy'.

After deployment a link will be shown on the home page of the API.



Step 12: Now the Get Method which created previously you see the link copy that and open in your browser , pass the arguments.



If all goes you will be able to see the output.

