Name: Hrishikesh Kumbhar

Div: D15A

Roll no: 32

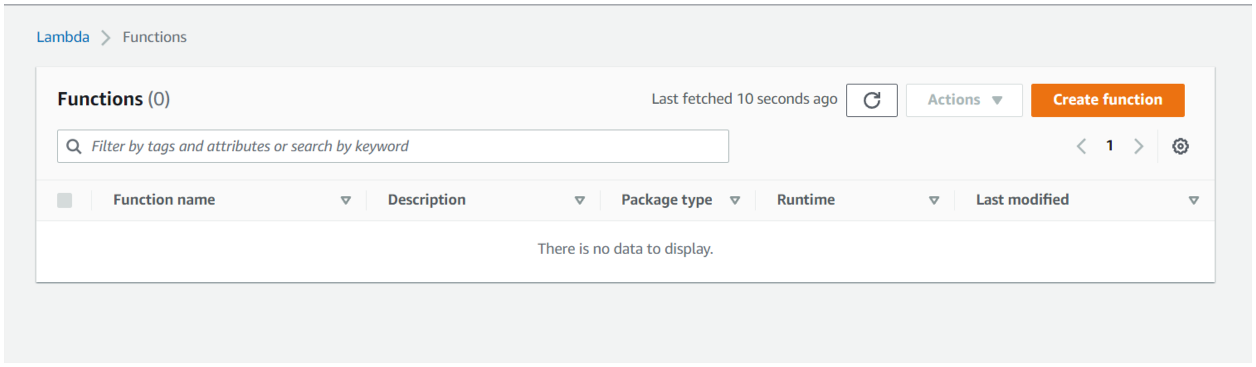
Sub: Advanced DevOps

Assignment No: 02

Implementation: Create a REST API with the Serverless Framework.

Prerequisites: AWS Free Tier account

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


Lambda > Functions

Functions (0)

Last fetched 10 seconds ago

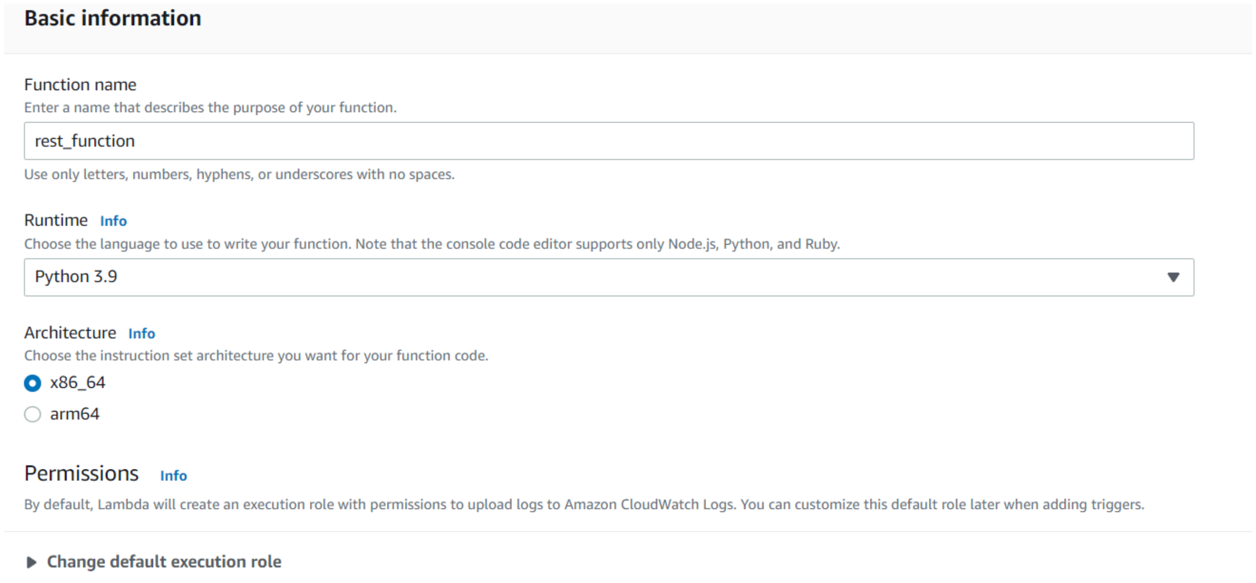
Actions ▼ | Create function

Filter by tags and attributes or search by keyword

Function name	Description	Package type	Runtime	Last modified

There is no data to display.

Step 2: Create an AWS Lambda function with runtime as Python 3.9.


Basic information

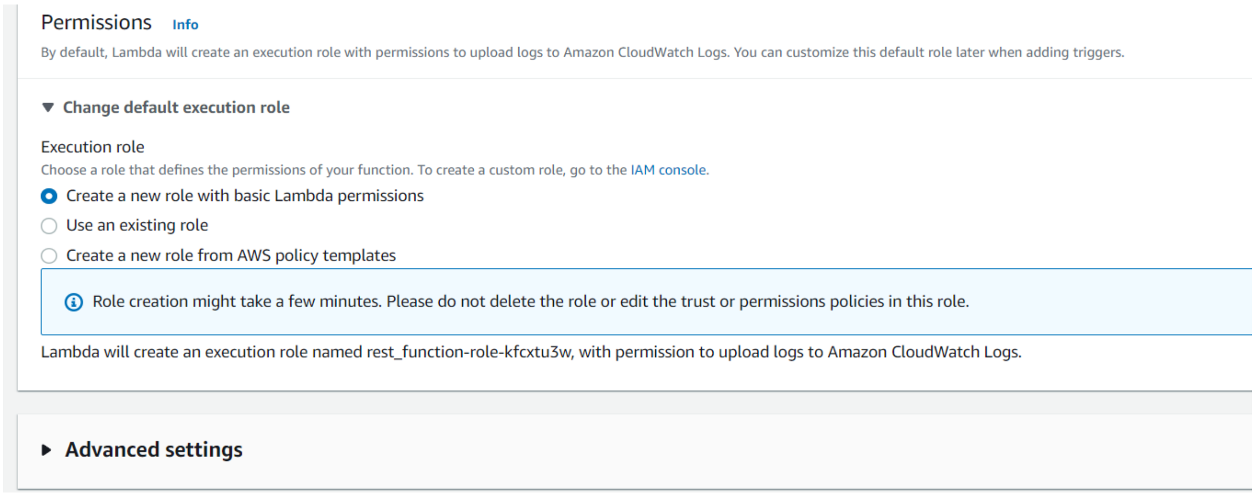
Function name
Enter a name that describes the purpose of your function.
rest_function
Use only letters, numbers, hyphens, or underscores with no spaces.

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 3.9

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ 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



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.

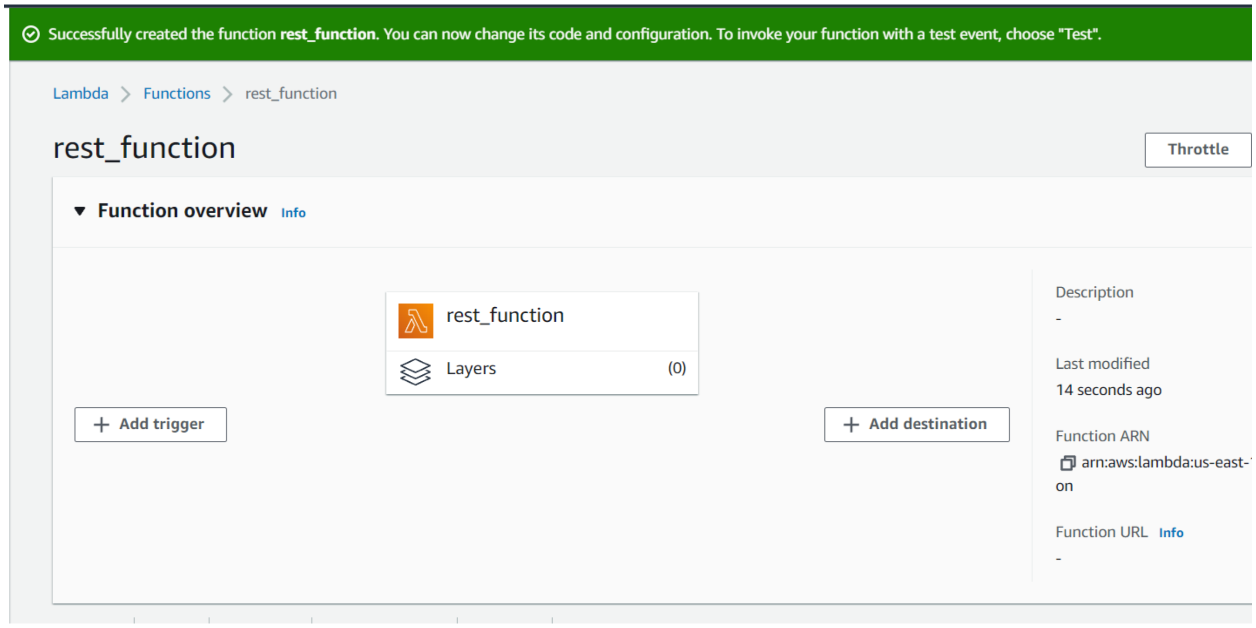
- [✓ Create a new role with basic Lambda permissions]
- Use an existing role
- Create a new role from AWS policy templates

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named rest_function-role-kfcxtu3w, with permission to upload logs to Amazon CloudWatch Logs.

▶ Advanced settings

Then click on "Create Function"



Successfully created the function **rest_function**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > rest_function

rest_function

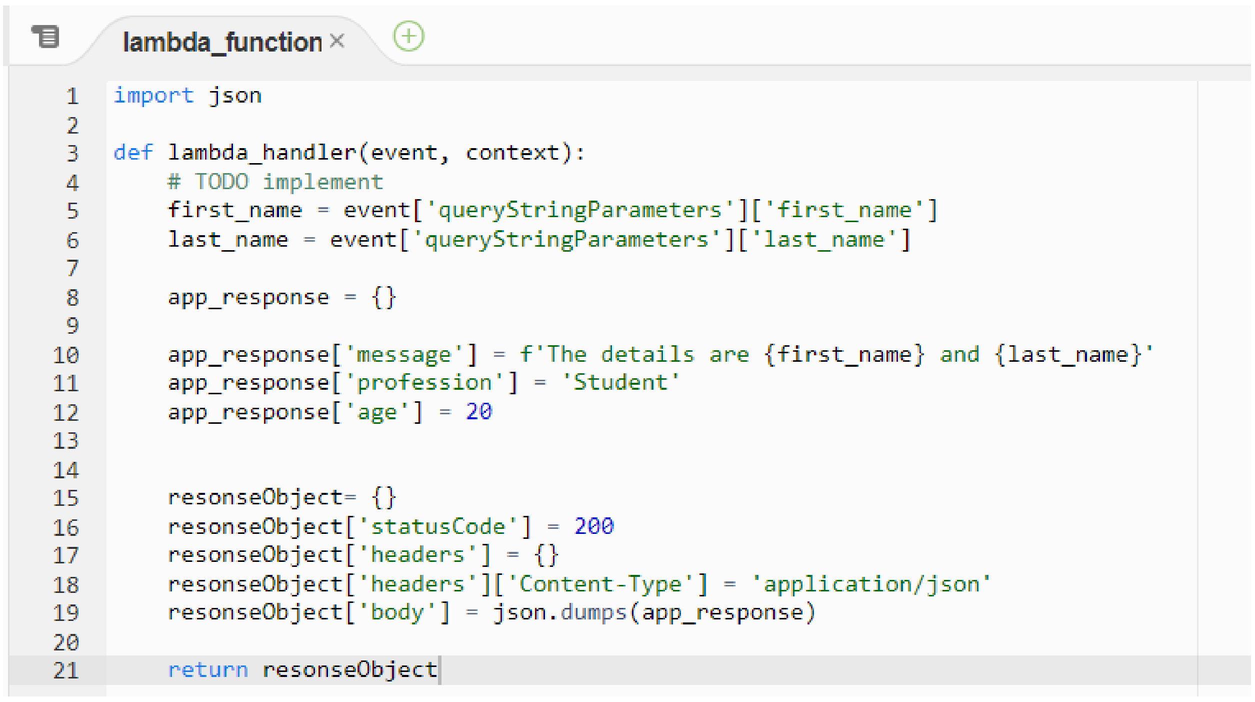
Description
-

Last modified
14 seconds ago

Function ARN
arn:aws:lambda:us-east-
on

Function URL Info
-

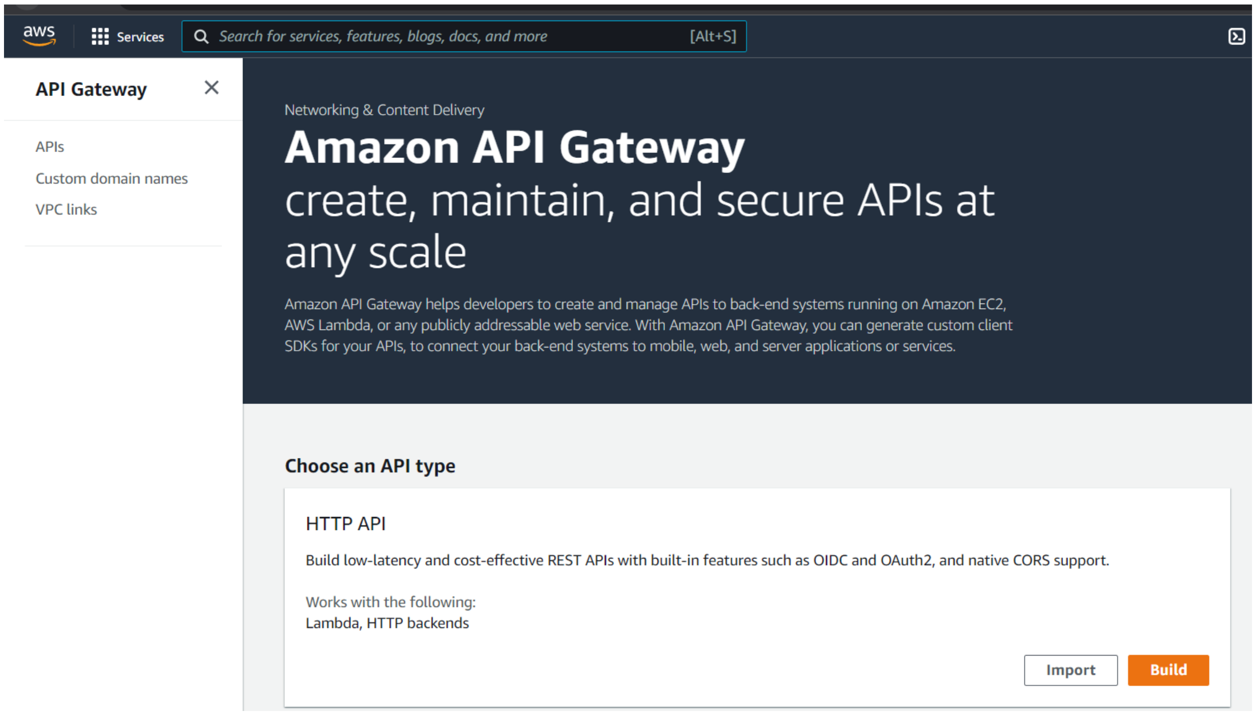
Step 3: Write the code for the handler which will be invoked after input from the user and save it



```python
import json

def lambda_handler(event, context):
    # TODO implement
    first_name = event['queryStringParameters']['first_name']
    last_name = event['queryStringParameters']['last_name']

    app_response = {}

    app_response['message'] = f'The details are {first_name} and {last_name}'
    app_response['profession'] = 'Student'
    app_response['age'] = 20


    resonseObject= {}
    resonseObject['statusCode'] = 200
    resonseObject['headers'] = {}
    resonseObject['headers']['Content-Type'] = 'application/json'
    resonseObject['body'] = json.dumps(app_response)

    return resonseObject
```

```python
import json

def lambda_handler(event, context):
    # TODO implement
    first_name = event['queryStringParameters']['first_name']
    last_name = event['queryStringParameters']['last_name']

    app_response = {}

    app_response['message'] = f'The details are {first_name} and {last_name}'
    app_response['profession'] = 'Student'
    app_response['age'] = 20


    resonseObject= {}
    resonseObject['statusCode'] = 200
    resonseObject['headers'] = {}
    resonseObject['headers']['Content-Type'] = 'application/json'
    resonseObject['body'] = json.dumps(app_response)

    return resonseObject
```

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



Networking & Content Delivery

Amazon API Gateway
create, maintain, and secure APIs at any scale

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.

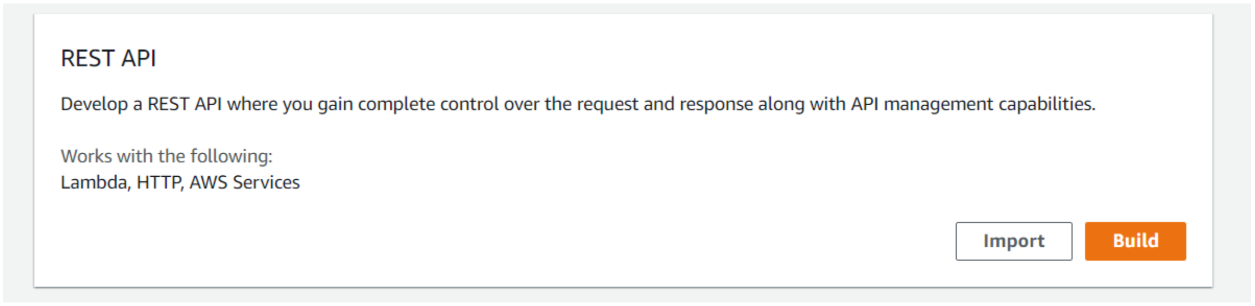
Choose an API type

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

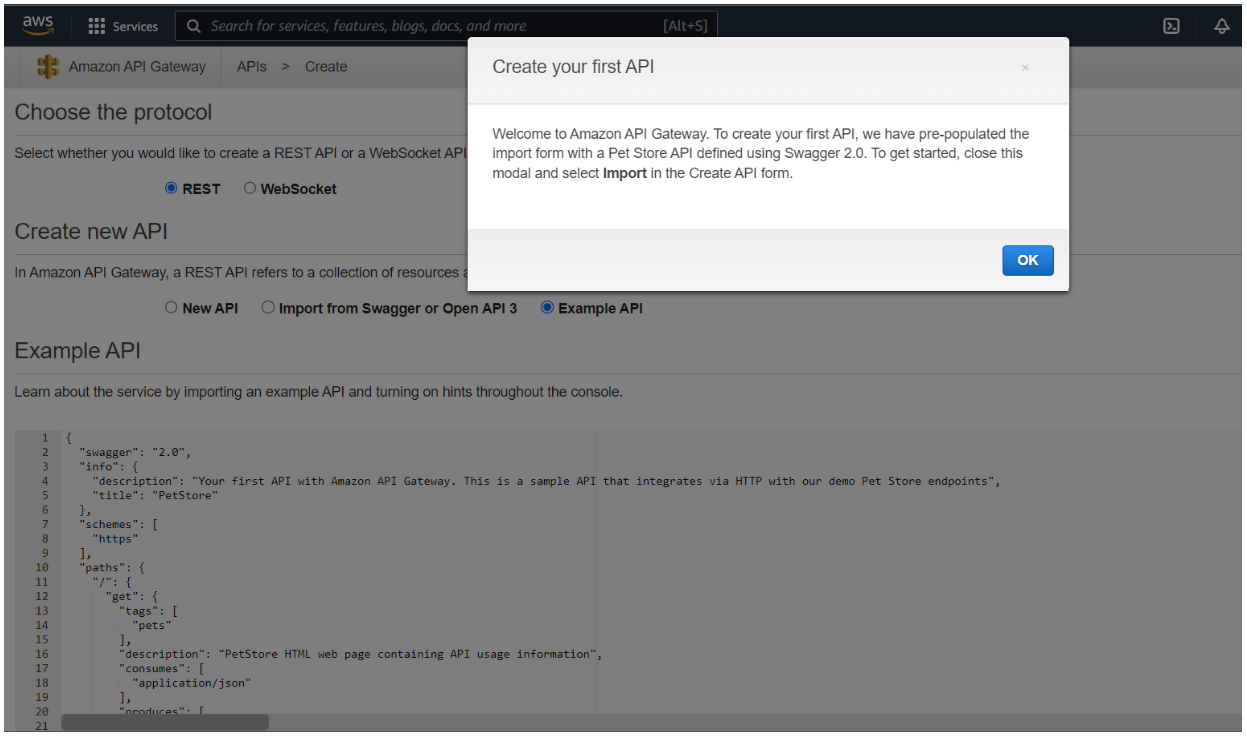
Works with the following:
Lambda, HTTP backends

Now look for the REST API and build it


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


Create your first API

Welcome to Amazon API Gateway. To create your first API, we have pre-populated the import form with a Pet Store API defined using Swagger 2.0. To get started, close this modal and select **Import** in the Create API form.

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 a

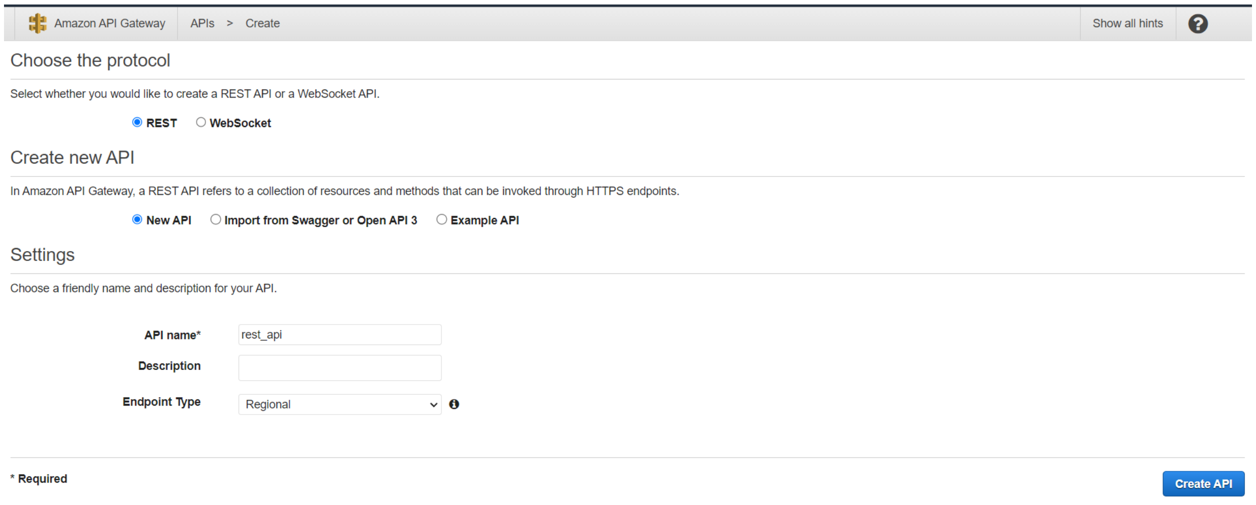
New API Import from Swagger or Open API 3 [✓ Example API]

Example API

Learn about the service by importing an example API and turning on hints throughout the console.

```
{
  "swagger": "2.0",
  "info": {
    "description": "Your first API with Amazon API Gateway. This is a sample API that integrates via HTTP with our demo Pet Store endpoints",
    "title": "PetStore"
  },
  "schemes": [
    "https"
  ],
  "paths": {
    "/": {
      "get": {
        "tags": [
          "pets"
        ],
        "description": "PetStore HTML web page containing API usage information",
        "consumes": [
          "application/json"
        ],
        "produces": [
```

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


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

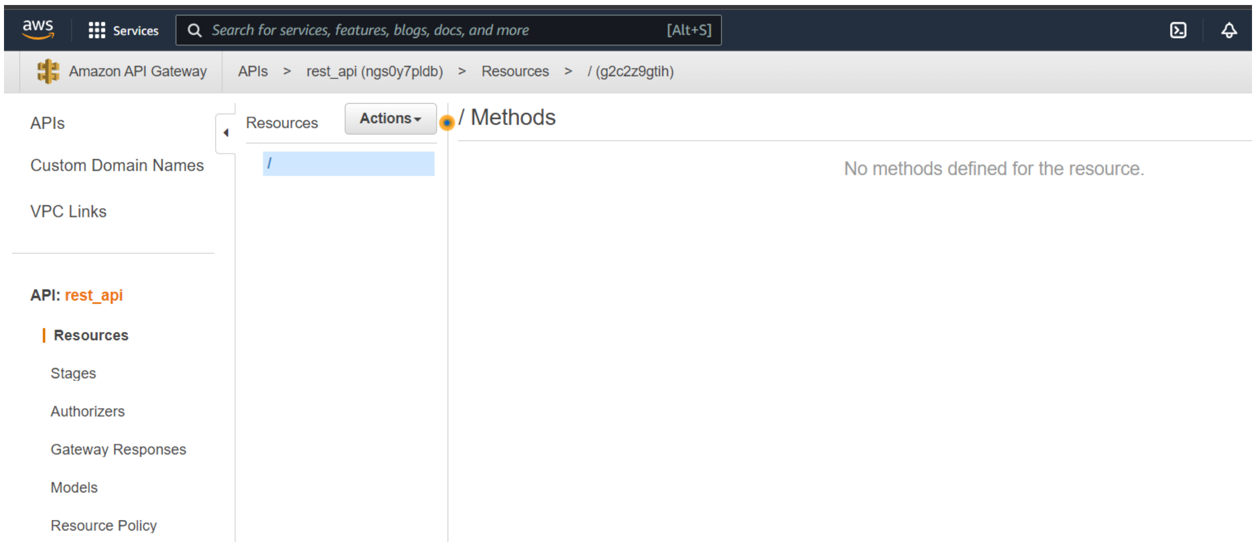
Settings

Choose a friendly name and description for your API.

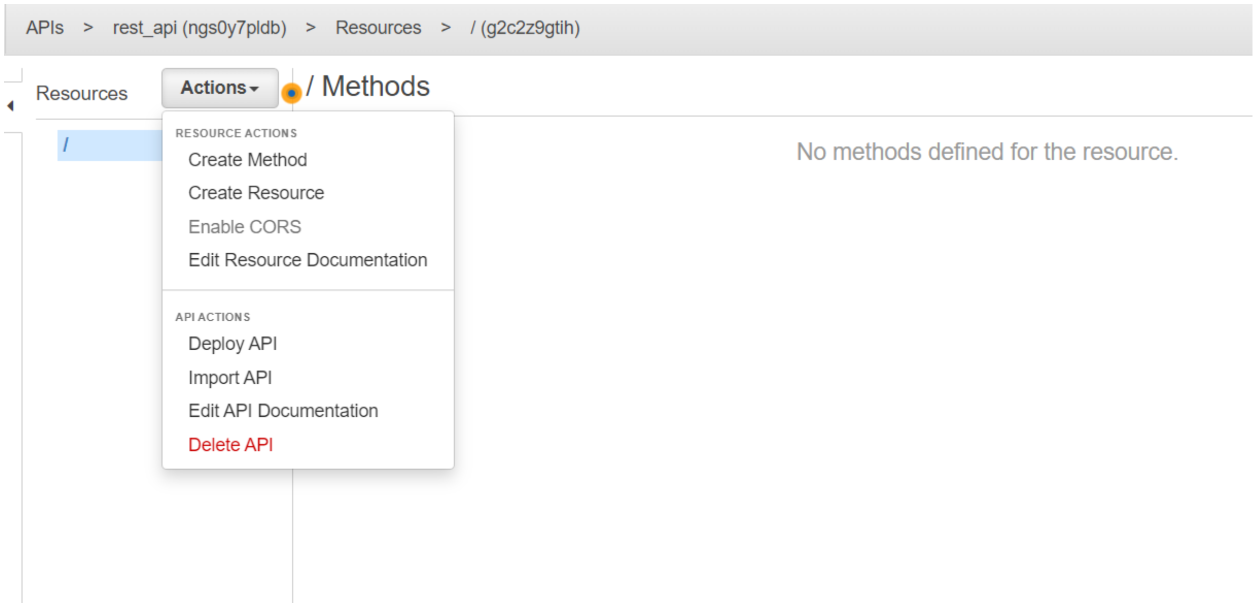
API name*	rest_api
Description	
Endpoint Type	Regional

* Required

After successful creation you will see this.

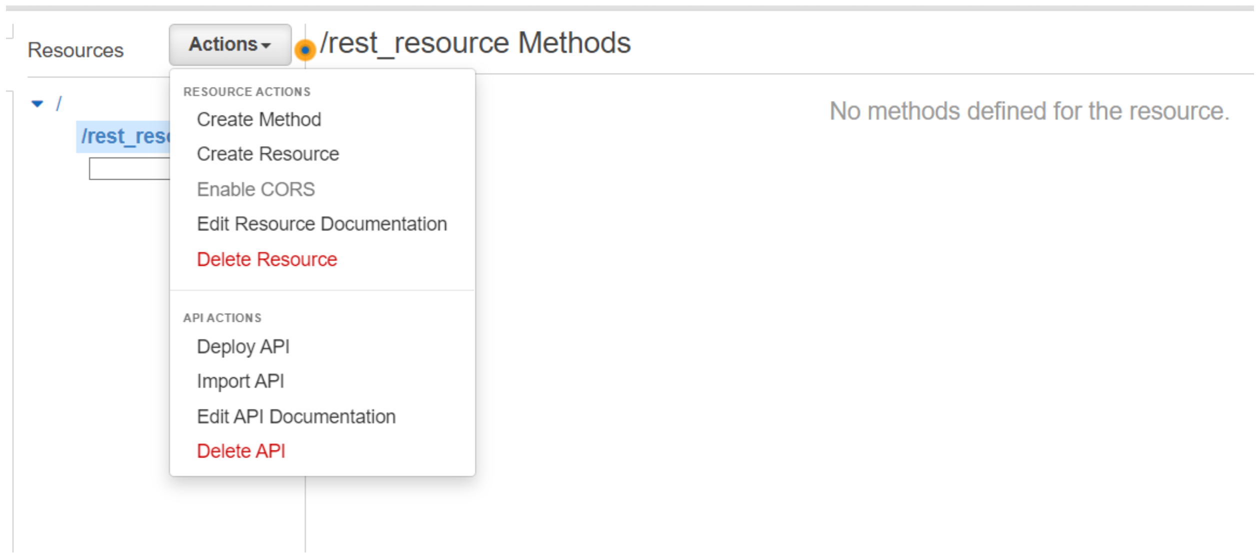


Step 6: Now , under "Actions" choose "Create Resource"



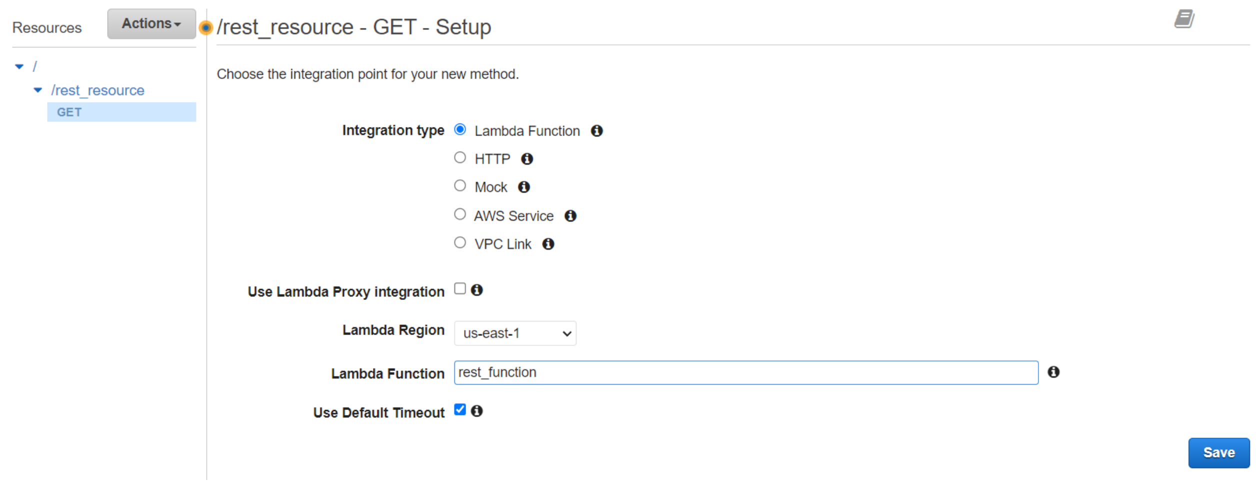
Configure the Resource and the create it

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



Select “GET”.

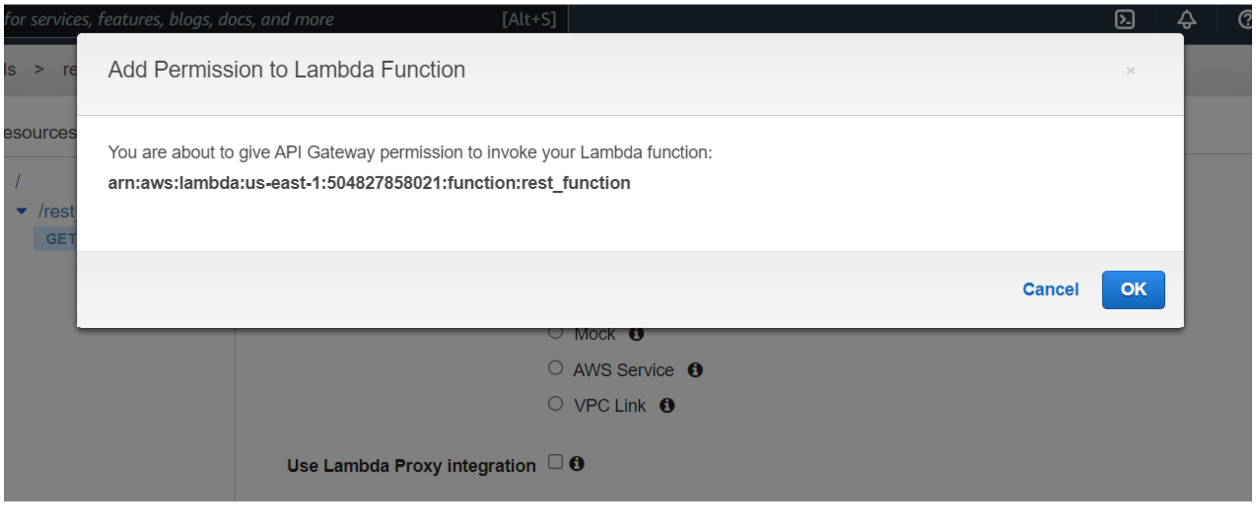
Do the necessary configuration and save it.



Choose the integration point for your new method.

Integration type	[✓ Lambda Function] HTTP Mock AWS Service VPC Link
Use Lambda Proxy integration	☐
Lambda Region	us-east-1
Lambda Function	rest_function
Use Default Timeout	✓

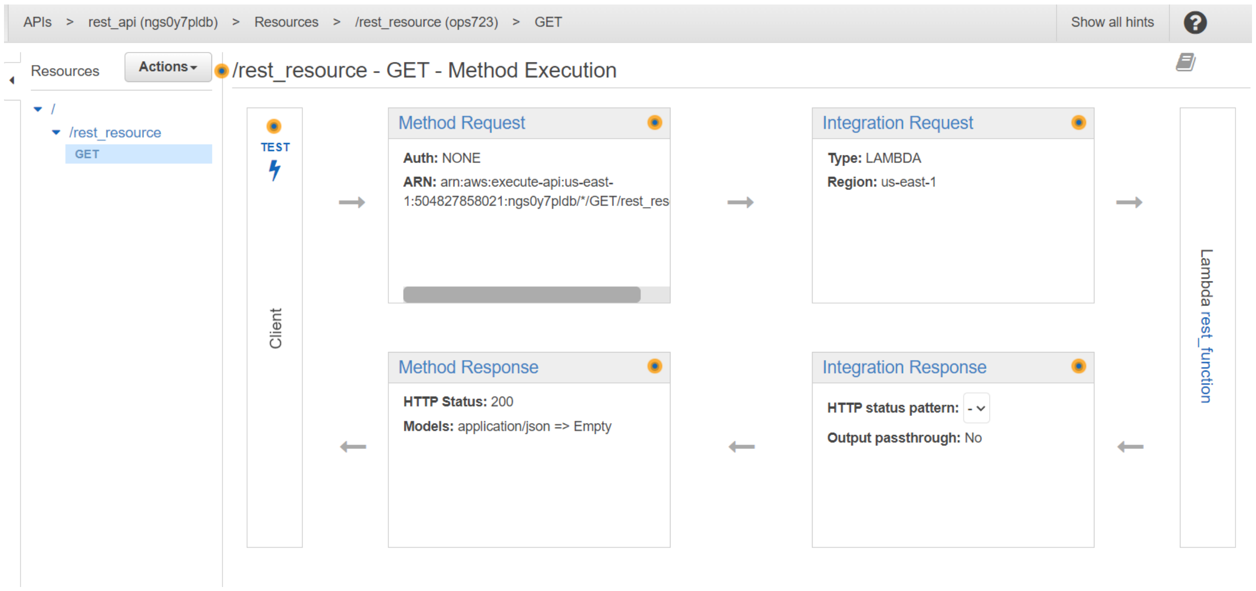
Step 8: Add the Permission of the method which we created previously.


Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:

arn:aws:lambda:us-east-1:504827858021:function:rest_function

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


APIs > rest_api (ngs0y7pldb) > Resources > /rest_resource (ops723) > GET

/rest_resource - GET - Method Execution

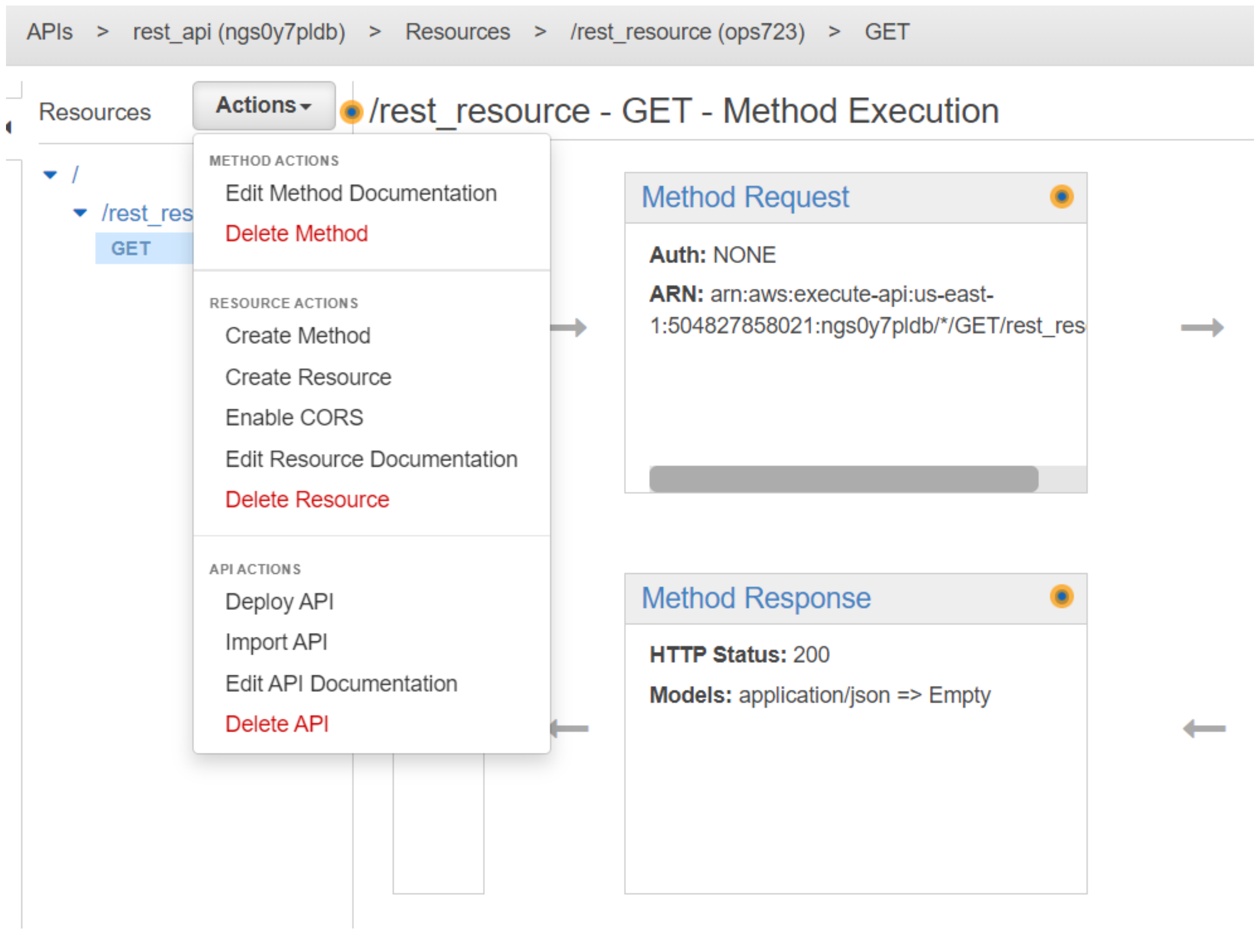
Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:504827858021:ngs0y7pldb/*/GET/rest_res

Integration Request
Type: LAMBDA
Region: us-east-1

Method Response
HTTP Status: 200
Models: application/json => Empty

Integration Response
HTTP status pattern: -
Output passthrough: No

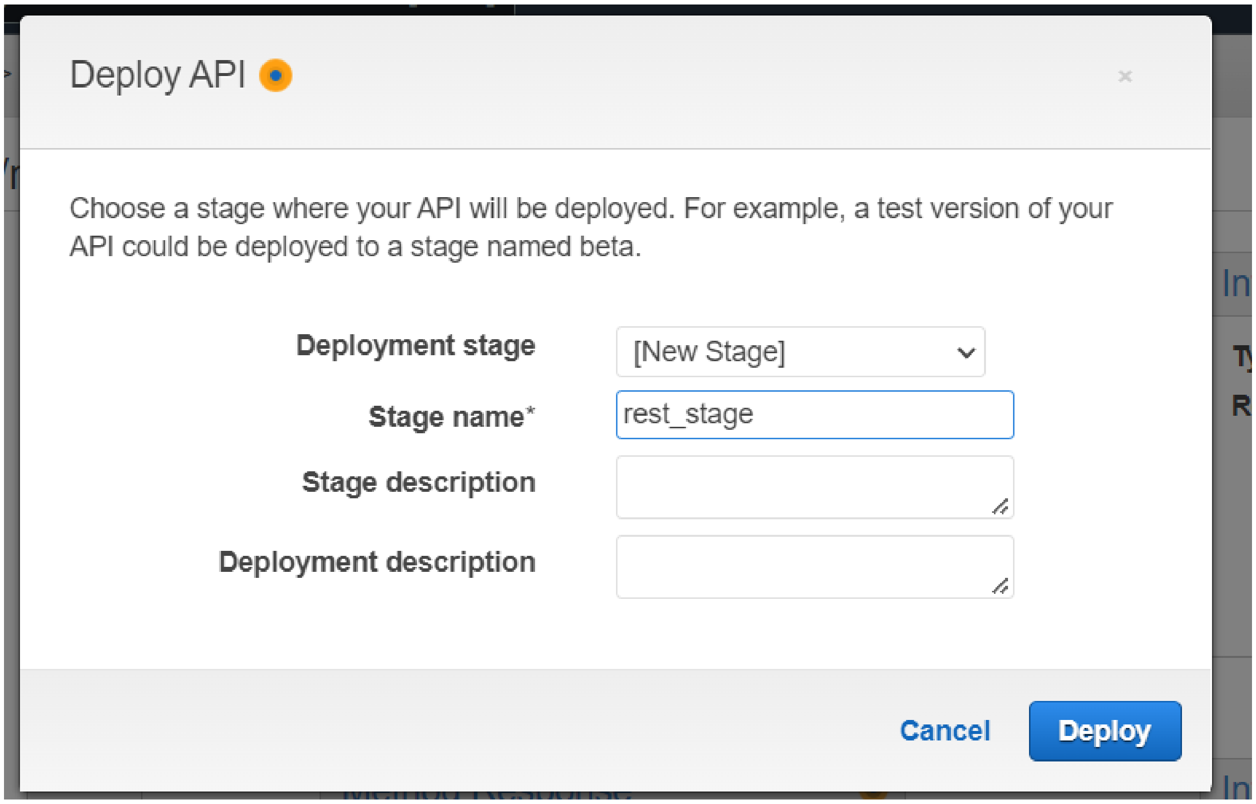
Step 10: Under Actions choose “Deploy API”.



Auth: NONE
ARN: arn:aws:execute-api:us-east-1:504827858021:ngs0y7pldb/*/GET/rest_res

HTTP Status: 200
Models: application/json => Empty

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

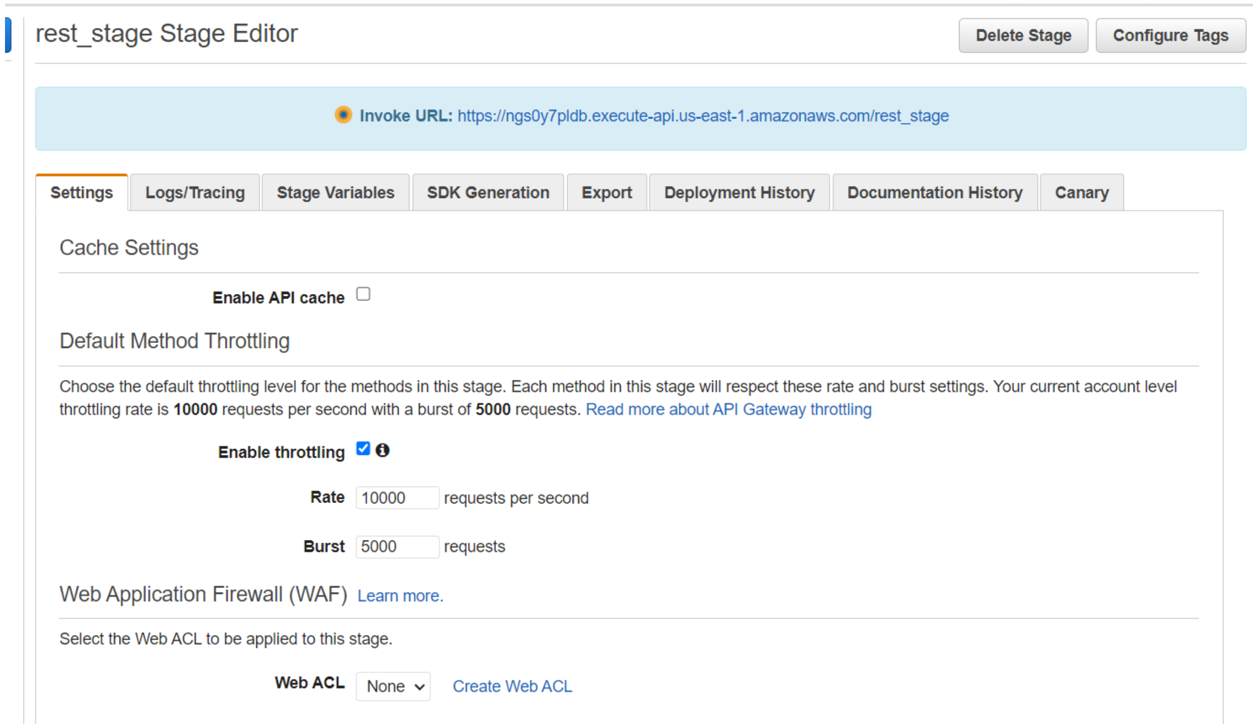


Deploy API

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.

Deployment stage	[New Stage]
Stage name*	rest_stage
Stage description	
Deployment description	

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



rest_stage Stage Editor

Invoke URL: https://ngs0y7pldb.execute-api.us-east-1.amazonaws.com/rest_stage

Cache Settings

Enable API cache ☐

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account level throttling rate is **10000** requests per second with a burst of **5000** requests. Read more about API Gateway throttling

Enable throttling ☑

Rate 10000 requests per second

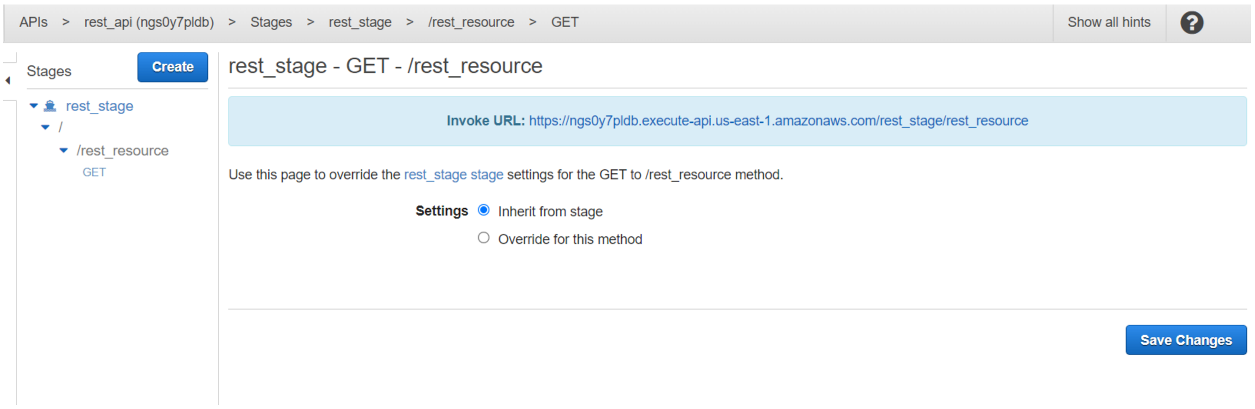
Burst 5000 requests

Web Application Firewall (WAF) Learn more.

Select the Web ACL to be applied to this stage.

Web ACL None Create Web ACL

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

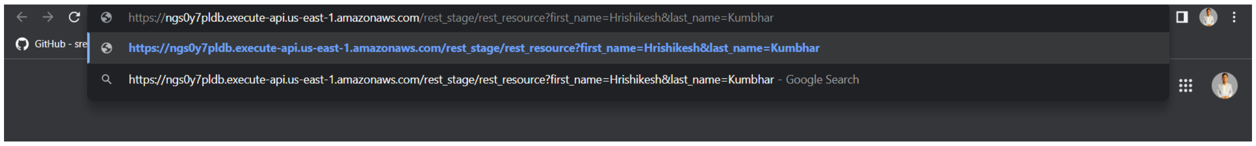

APIs > rest_api (ngs0y7pldb) > Stages > rest_stage > /rest_resource > GET
Show all hints
Stages
Create
rest_stage
/
/rest_resource
GET
Save Changes

rest_stage - GET - /rest_resource

Invoke URL: https://ngs0y7pldb.execute-api.us-east-1.amazonaws.com/rest_stage/rest_resource

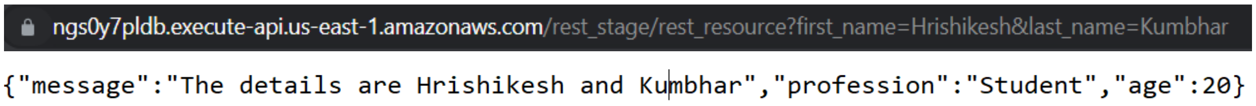
Use this page to override the rest_stage stage settings for the GET to /rest_resource method.

Settings [✓ Inherit from stage] Override for this method



```
https://ngs0y7pldb.execute-api.us-east-1.amazonaws.com/rest_stage/rest_resource?first_name=Hrishikesh&last_name=Kumbhar
```

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



```
ngs0y7pldb.execute-api.us-east-1.amazonaws.com/rest_stage/rest_resource?first_name=Hrishikesh&last_name=Kumbhar
{"message":"The details are Hrishikesh and Kumbhar","profession":"Student","age":20}
```