Mayuri Varade
D15B – 70

ASSIGNMENT – 2

- Create a REST API with serverless framework

THEORY :-

Representational State Transfer (REST) is an architectural style that defines a set of constraints to be used for creating web services. REST API is a way of accessing web services in a simple and flexible way without having any processing

REST technology is generally preferred to more robust Simple Object Access Protocol (SOAP) technology because REST uses less bandwith, simple and flexible making it more suitable for internet usage. Its used to fetch or give some information from a web service. All communication done via REST API uses only HTTP request.

A request is sent from client to server in form of web URL as HTTP GET or POST or PUT or DELETE request.

API Gateway acts as "front door" for applications to access data, business logic or functionality from your backend services, such as workloads running on Amazon EC2, code running on AWS Lambda or any application

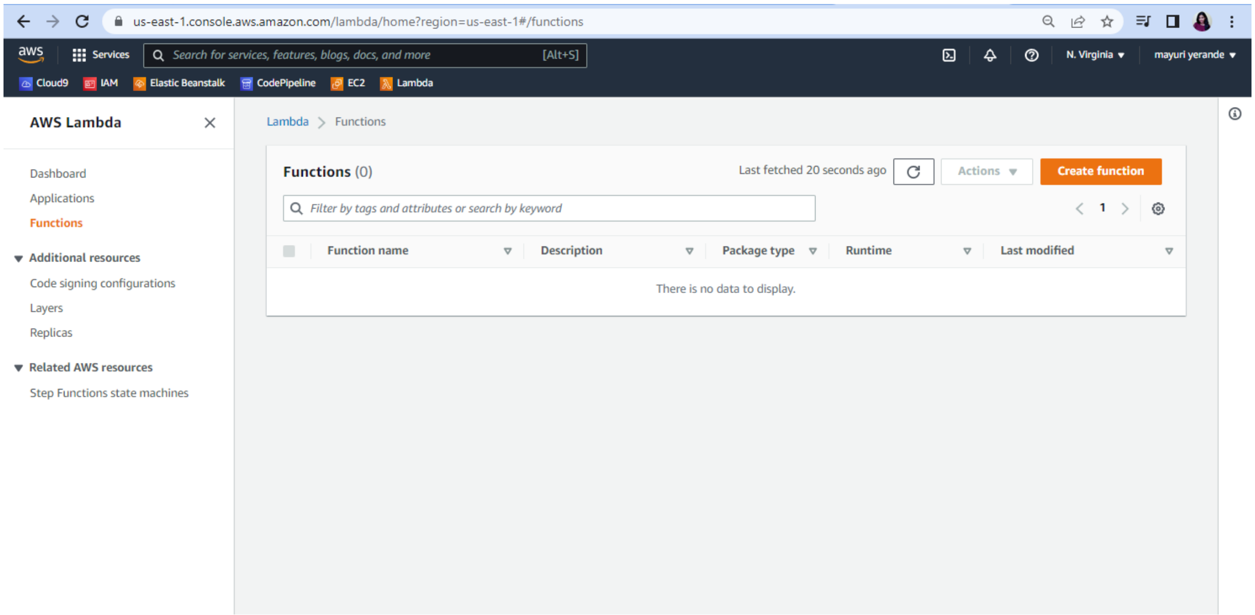
AWS Lambda is a serverless compute service that runs a code in response to events and automatically manages that underlying compute resources for you.
These events may include changes in state, or an update.

AWS Lambda automatically runs code in response to multiple events such as HTTP request via Amazon API Gateway, modifications to objects in amazon simple storage service buckets and state transitions in AWS step functions.

Lambda runs your code on high availability compute infrastructure and performs all administration of your compute resources. This includes server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment and code monitoring and logging. All you need to do is supply the code.

IMPLEMENTATION:

- Go to your lambda console



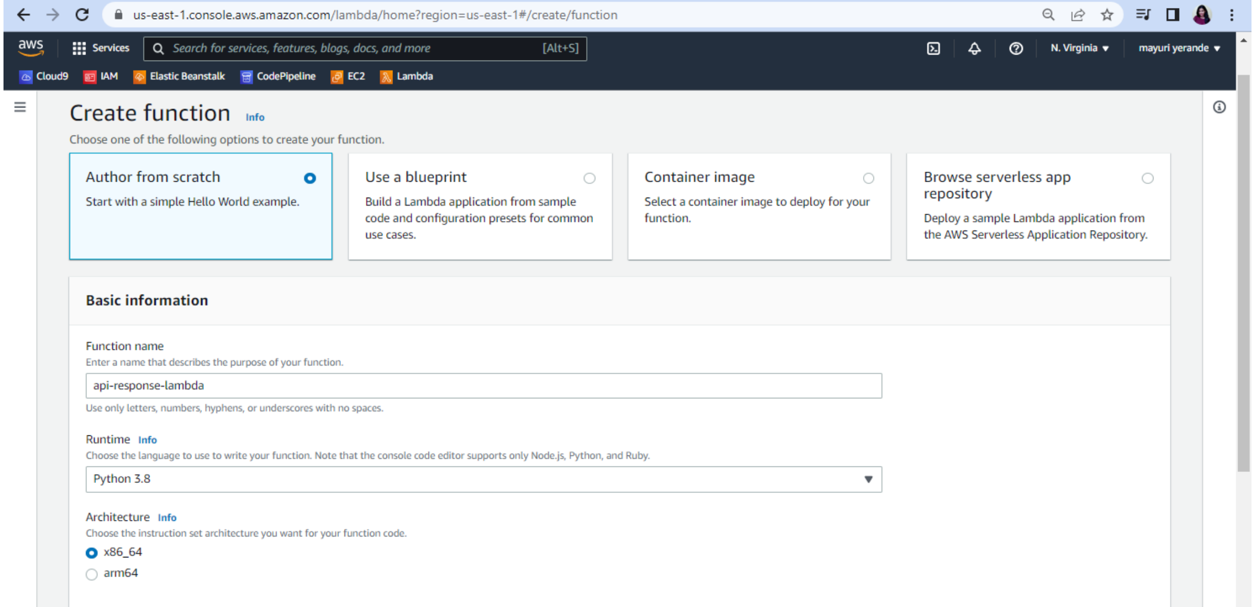
Functions (0)

Last fetched 20 seconds ago

Function name	Description	Package type	Runtime	Last modified

There is no data to display.

- Create new function
- Choose Author from scratch
- Name your function
- Choose "python 3.8"
- Click on "create function"


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia ▾
mayuri yerande ▾
Cloud9
IAM
Elastic Beanstalk
CodePipeline
EC2
Lambda

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 for your function.

○ Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

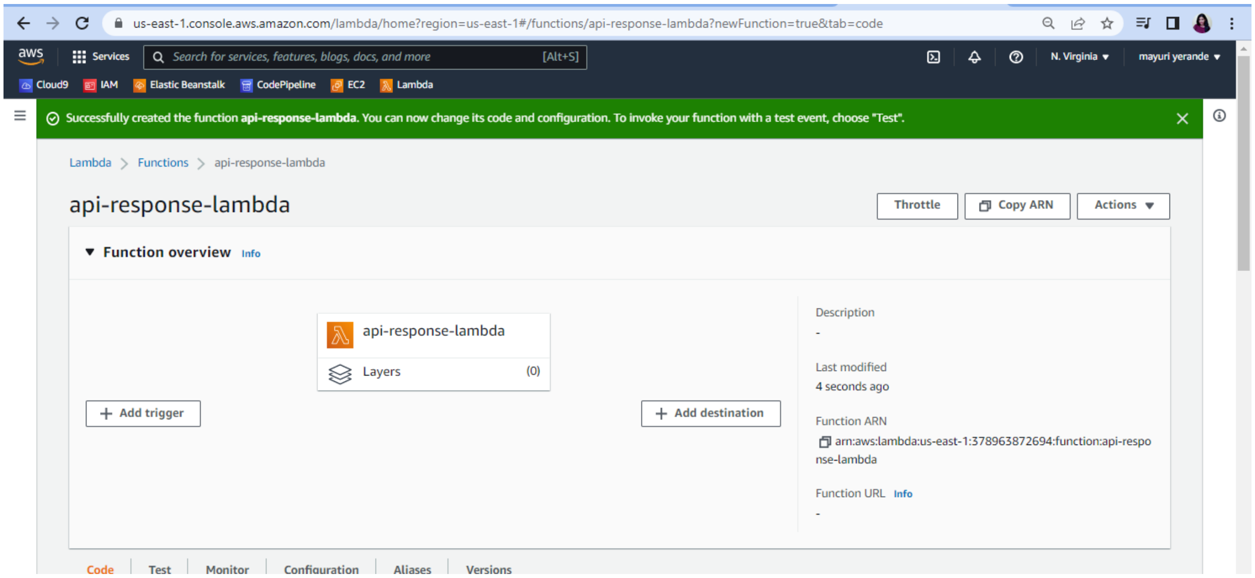
Basic information

Function name
Enter a name that describes the purpose of your function.
api-response-lambda
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 3.8

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

- Your function is now successfully created.



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

api-response-lambda

Description
-

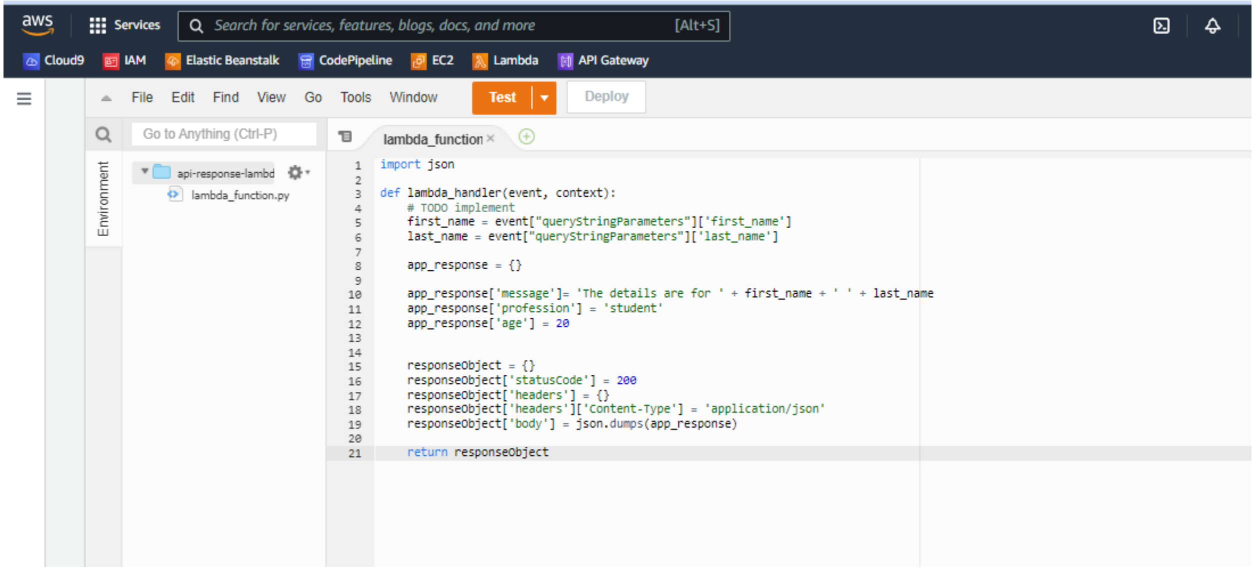
Last modified
4 seconds ago

Function ARN
arn:aws:lambda:us-east-1:378963872694:function:api-response-lambda

Function URL Info
-

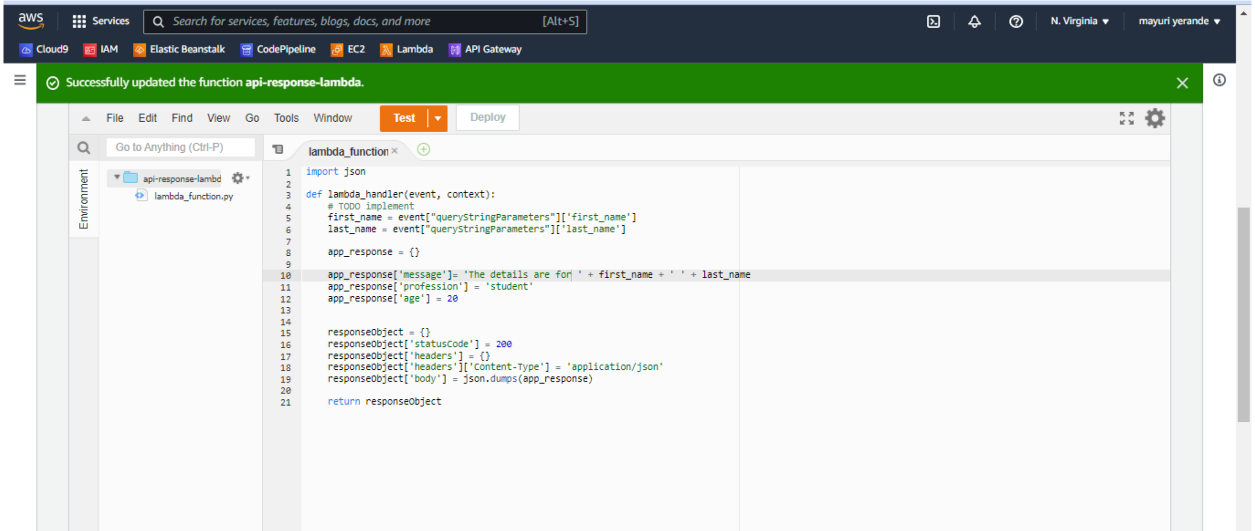
- Add this code in your function

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

    app_response = {}

    app_response['message']= 'The details are ' + first_name + ' ' + last_name
    app_response['profession'] = 'student'
    app_response['age'] = 20


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

    return responseObject
```



```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']= 'The details are for ' + first_name + ' ' + last_name
    app_response['profession'] = 'student'
    app_response['age'] = 20


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

    return responseObject
```

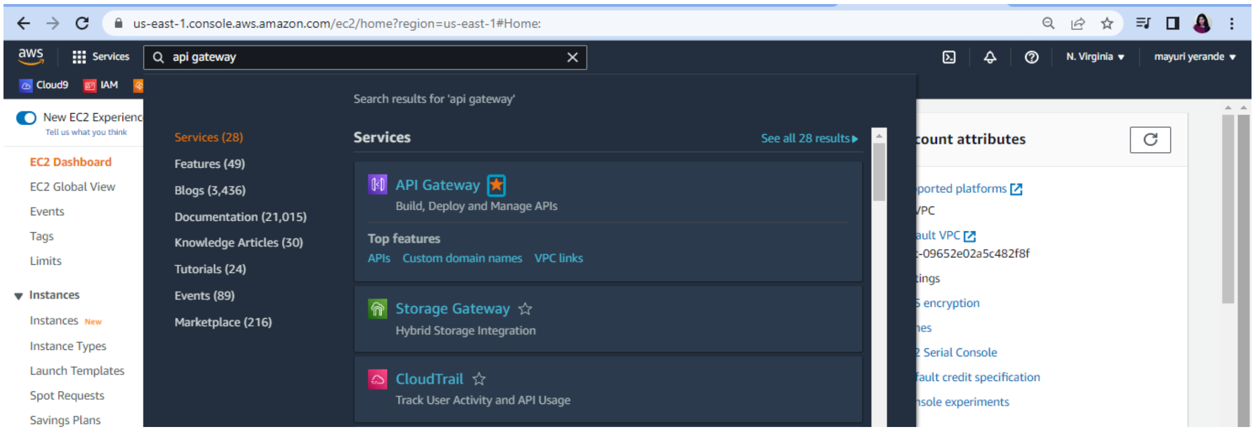
- Deploy the function



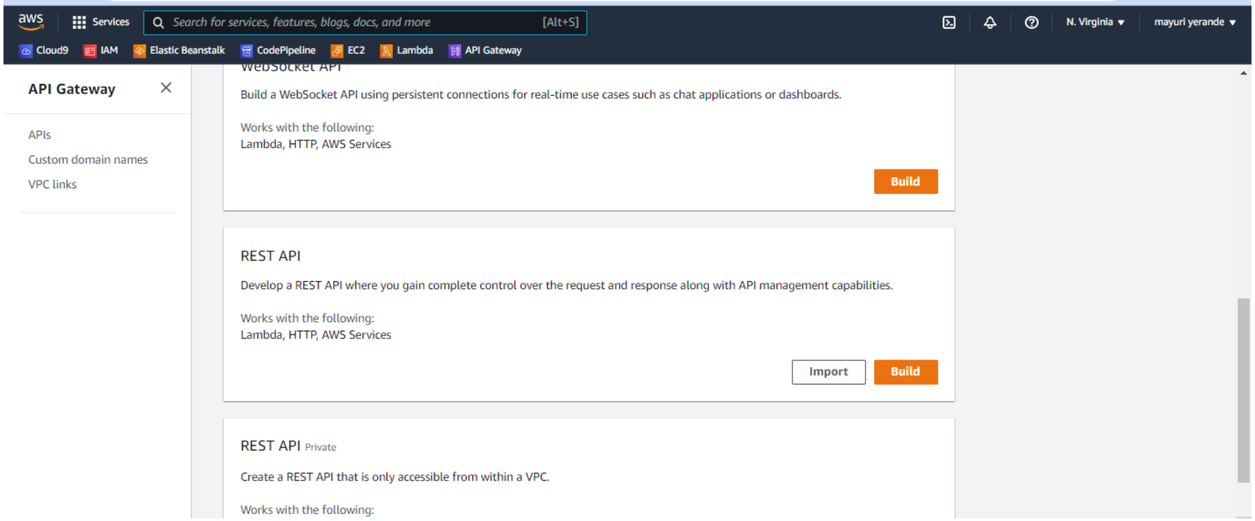
Successfully updated the function **api-response-lambda**.

```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']= 'The details are for ' + first_name + ' ' + last_name
    app_response['profession'] = 'student'
    app_response['age'] = 20


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

    return responseObject
```

- Go to “API gateway” console



- Go to “Rest API”



Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

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

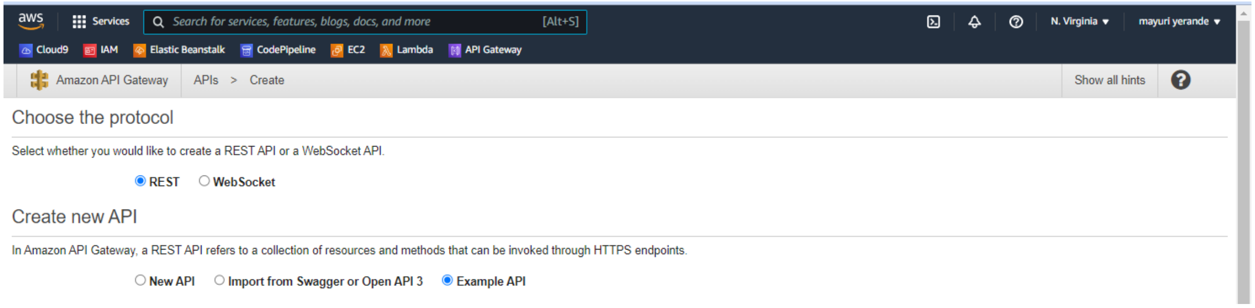
Works with the following:
Lambda, HTTP, AWS Services

Create a REST API that is only accessible from within a VPC.

Works with the following:

- Click on Build
- Then click “ok” for the pop-up that appears after that

- Click on “Rest” and “New API” option



Choose the protocol

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

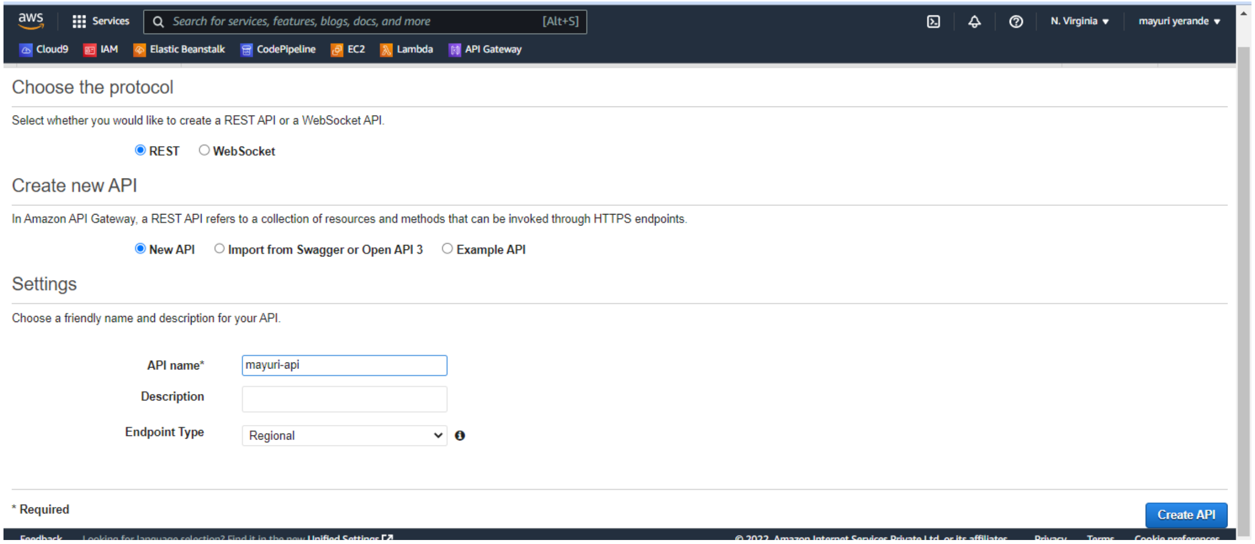
● REST ○ WebSocket

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

- Name your API. Keep other things as default.



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* mayuri-api

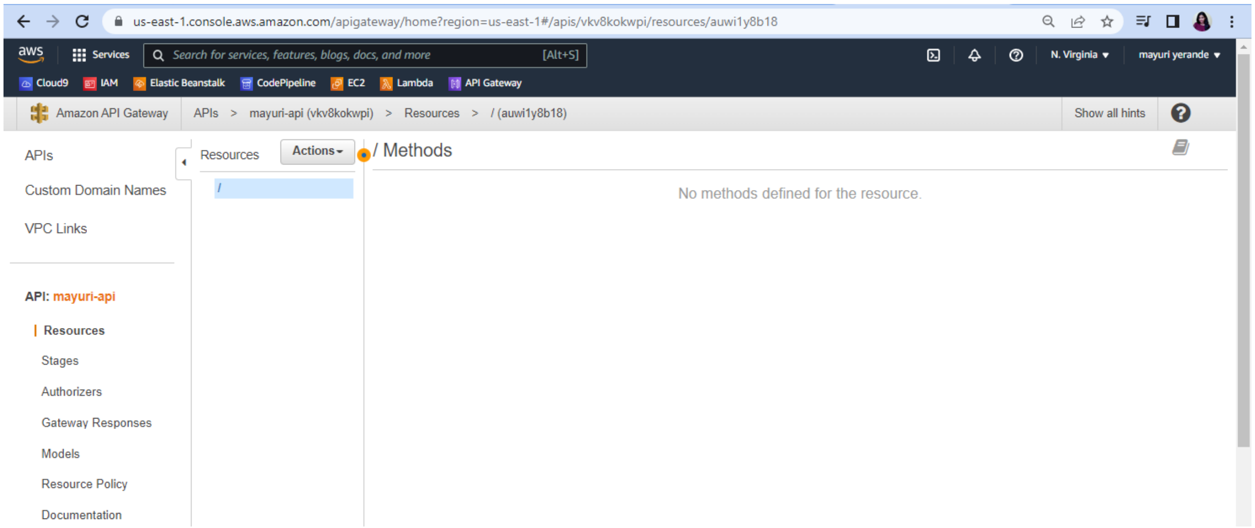
Description

Endpoint Type Regional

* Required

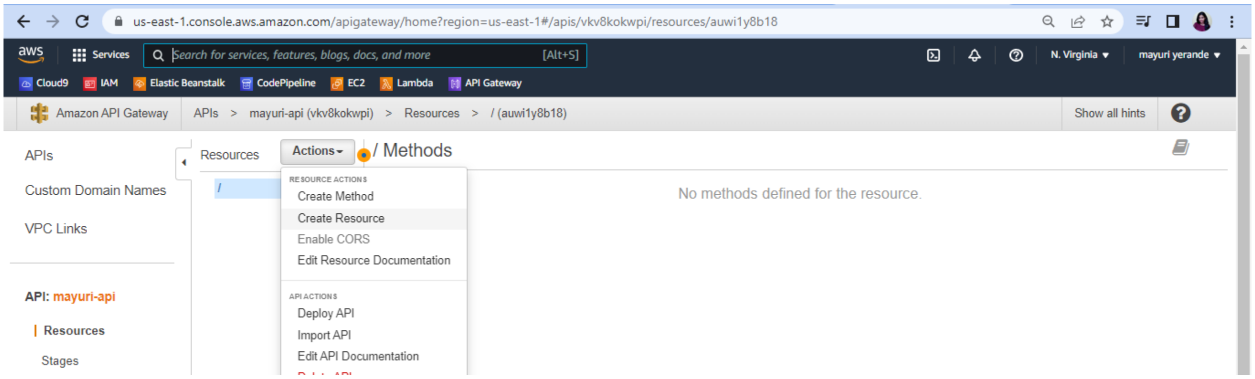
- Click on "Create API"

- Your API is successfully created.

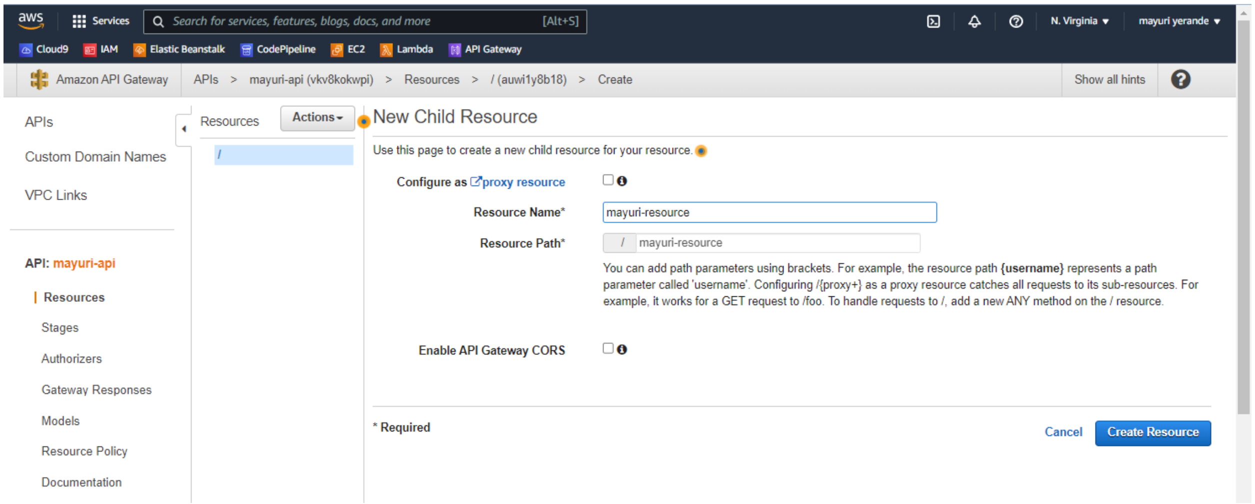


No methods defined for the resource.

- Click on “actions” and choose the “create resource” option.


us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/vkv8kokwpi/resources/auwi1y8b18
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
mayuri yerande
Cloud9 IAM Elastic Beanstalk CodePipeline EC2 Lambda API Gateway
Amazon API Gateway
APIs > mayuri-api (vkv8kokwpi) > Resources > / (auwi1y8b18)
Show all hints
APIs
Custom Domain Names
VPC Links
API: mayuri-api
Resources
Stages
Resources
Actions
/ Methods
RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Edit Resource Documentation
API ACTIONS
Deploy API
Import API
Edit API Documentation
No methods defined for the resource.

- Name your resource and click on “create resource” option



New Child Resource

Use this page to create a new child resource for your resource.

Configure as proxy resource ☐

Resource Name* mayuri-resource

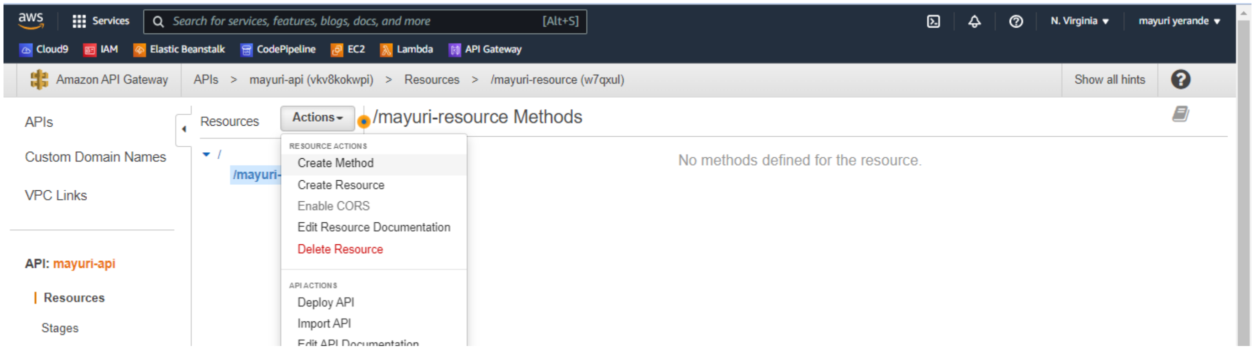
Resource Path* / mayuri-resource

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring /{proxy+} as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

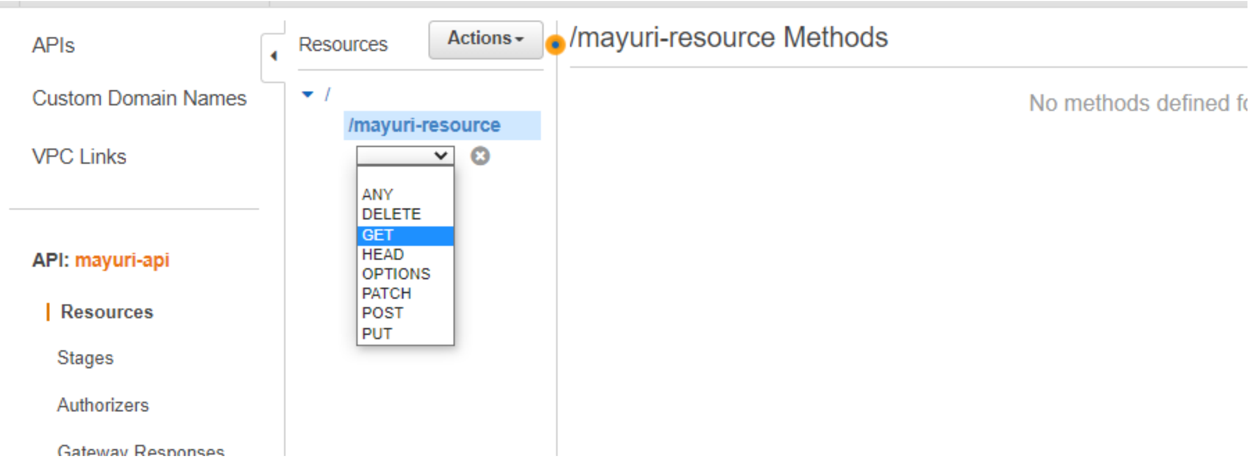
Enable API Gateway CORS ☐

* Required

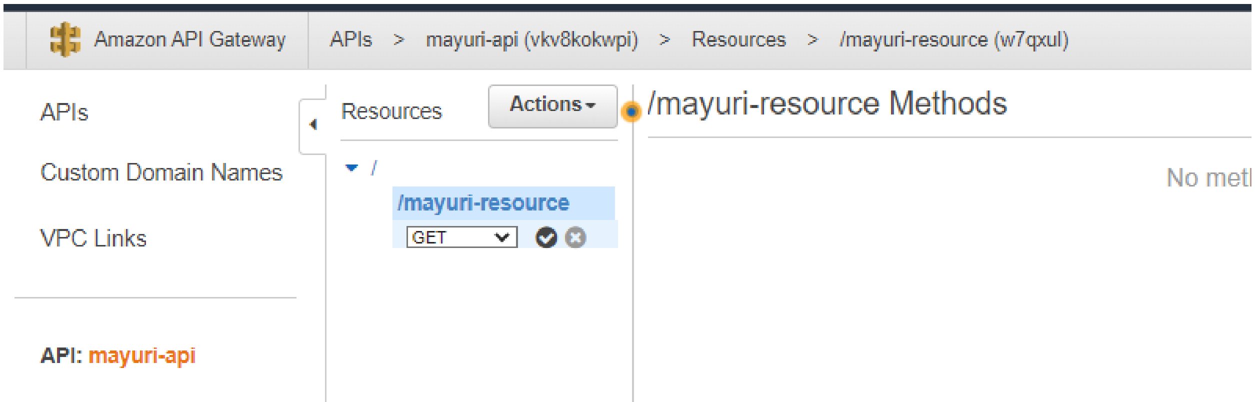
- Now click on “actions” and choose “create method”



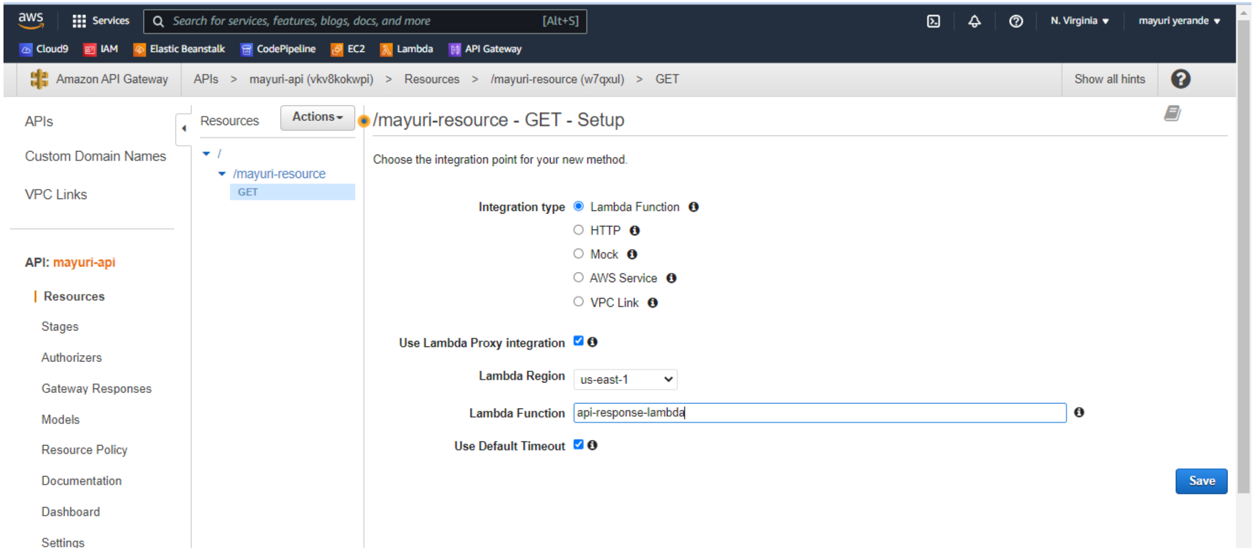
- On the drop down , choose “GET” method



- Click on the tick mark



- Here we are going to integrate our Api with our lambda function
- Enter your lambda function name here
- Check the “use lambda proxy integration” checkbox.
- Click on save



APIs > mayuri-api (vkv8kokwpi) > Resources > /mayuri-resource (w7qxul) > GET

API: mayuri-api

Resources

/mayuri-resource

/mayuri-resource - GET - Setup

Choose the integration point for your new method.

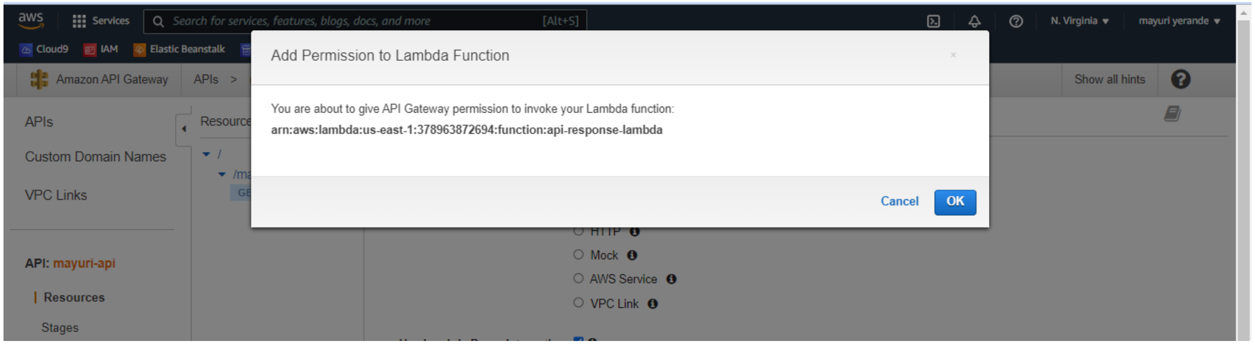
Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

Use Lambda Proxy integration ☑

Lambda Region us-east-1

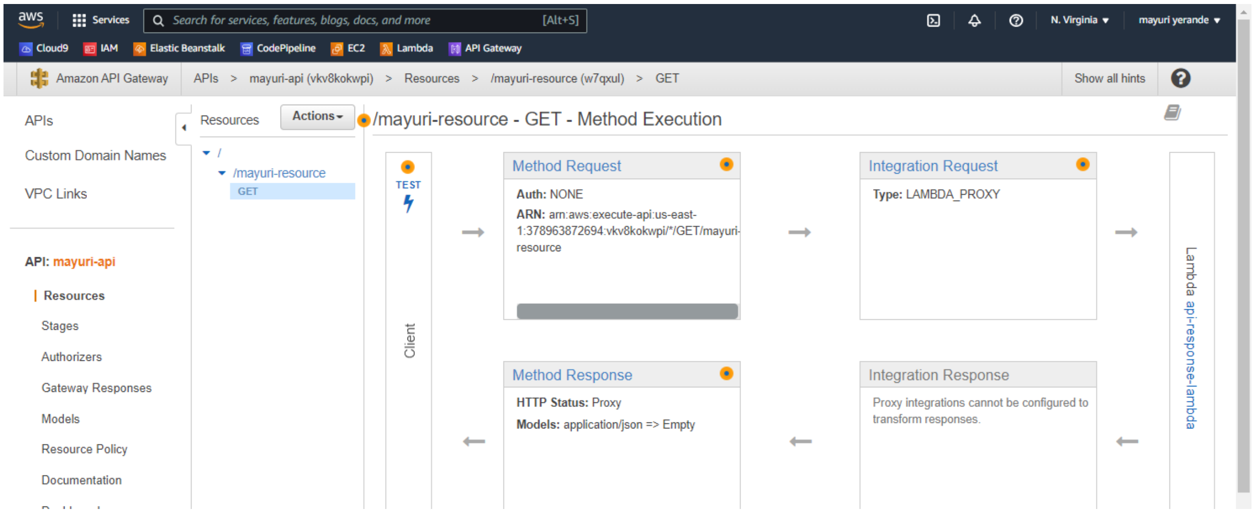
Lambda Function api-response-lambda

Use Default Timeout ☑



Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:us-east-1:378963872694:function:api-response-lambda



APIs > mayuri-api (vkv8kokwpi) > Resources > /mayuri-resource (w7qxul) > GET

API: mayuri-api

/mayuri-resource - GET - Method Execution

Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:378963872694:vkv8kokwpi/*/GET/mayuri-resource

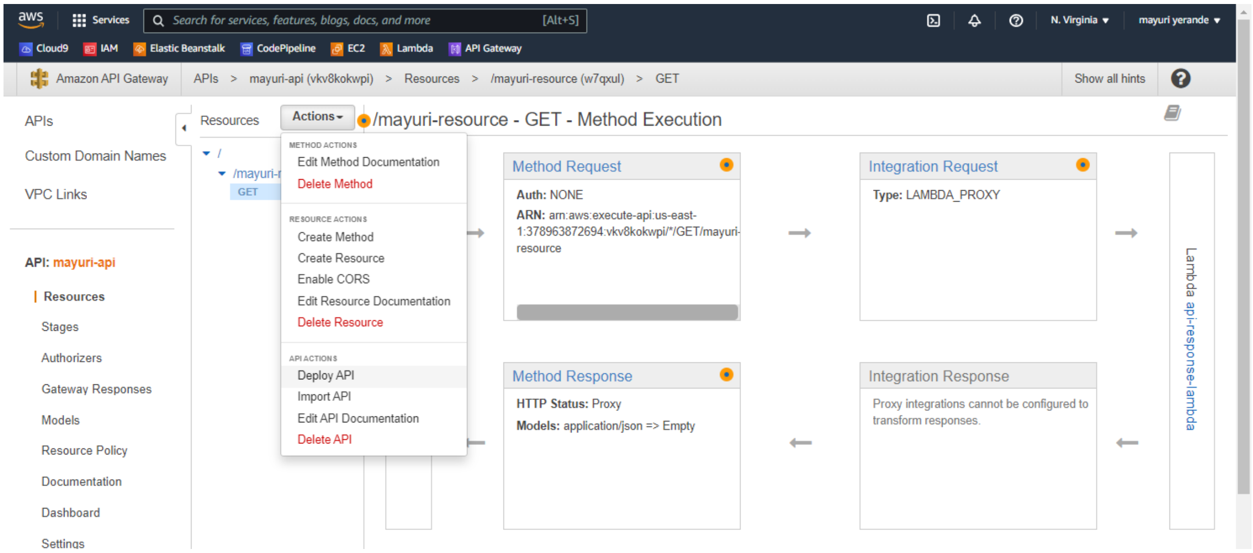
Integration Request
Type: LAMBDA_PROXY

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

Integration Response
Proxy integrations cannot be configured to transform responses.

- This shows the flow of how your method works

- Click on “actions” and choose “Deploy API”



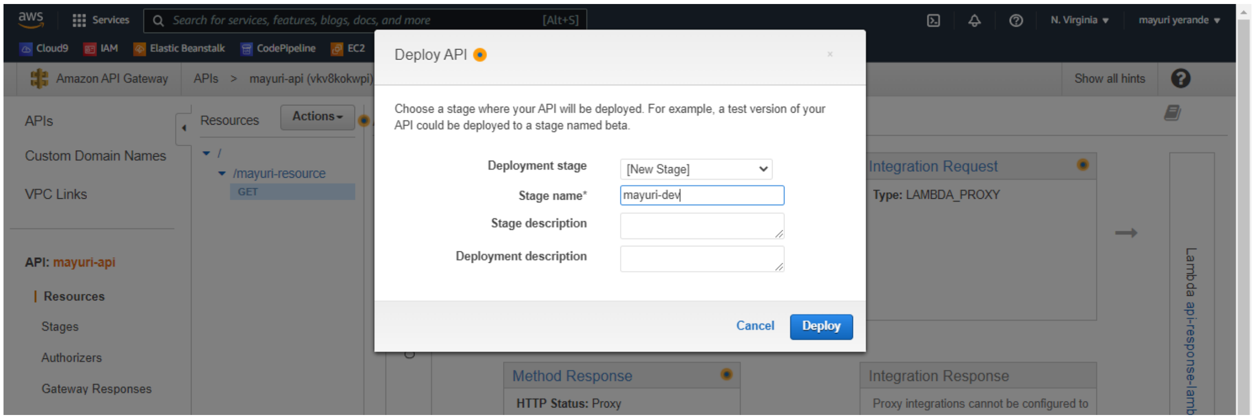
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:378963872694:vkv8kokwpi/*/GET/mayuri-resource

Type: LAMBDA_PROXY

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

Proxy integrations cannot be configured to transform responses.

- Click on “new stage” in deployment stage
- Name your stage
- Click on deploy



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.

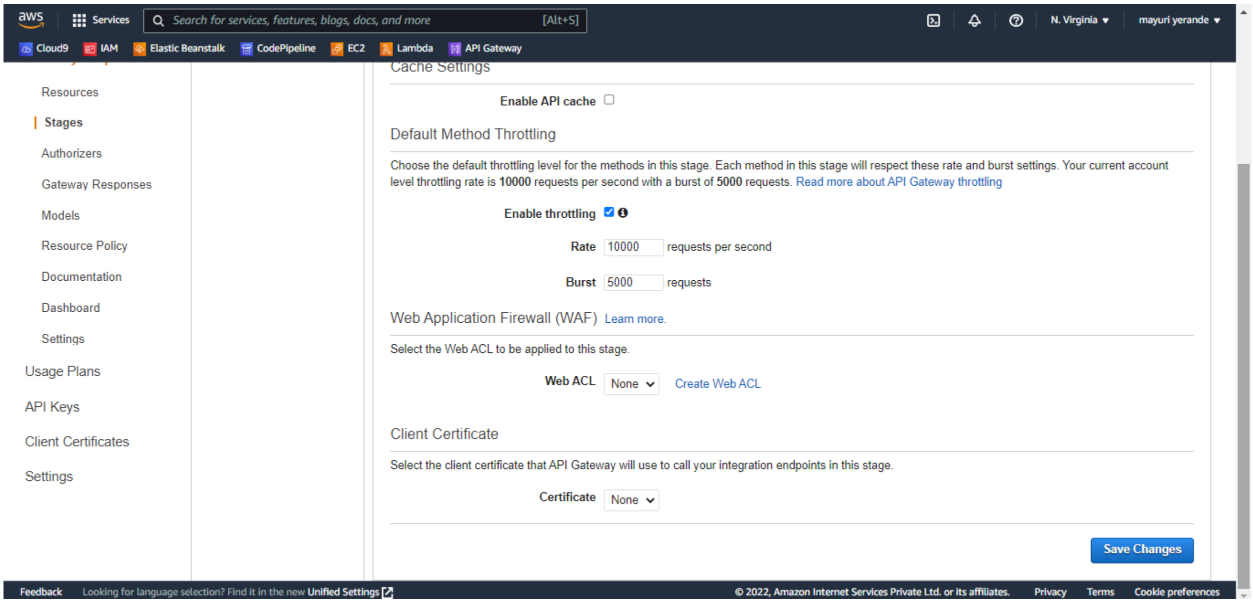
Deployment stage: [New Stage]
Stage name*: mayuri-dev
Stage description:
Deployment description:

Type: LAMBDA_PROXY

HTTP Status: Proxy

Proxy integrations cannot be configured to

- Click on save changes



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.

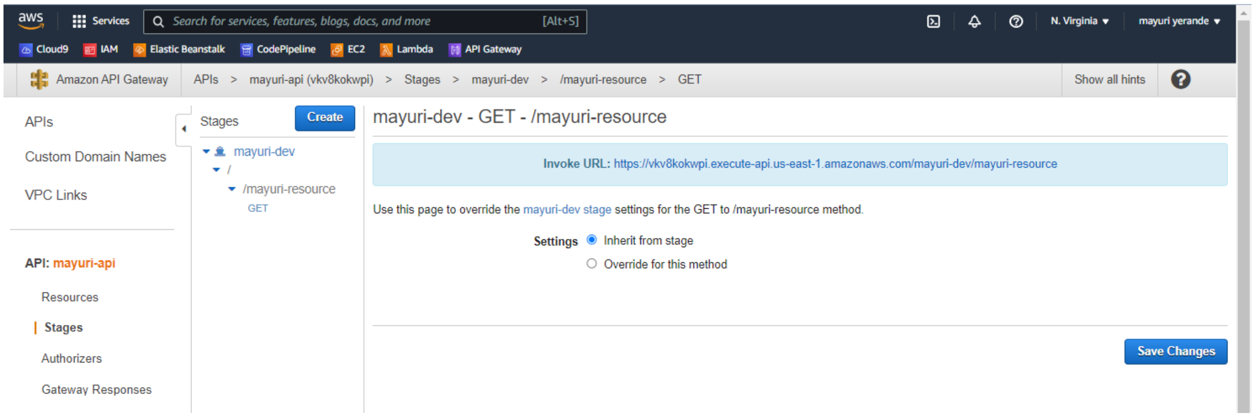
Web ACL None Create Web ACL

Client Certificate

Select the client certificate that API Gateway will use to call your integration endpoints in this stage.

Certificate None

- To get the Invoke URL
- Go to “your deployment stage name” (mayuri-dev), Go to the GET method.



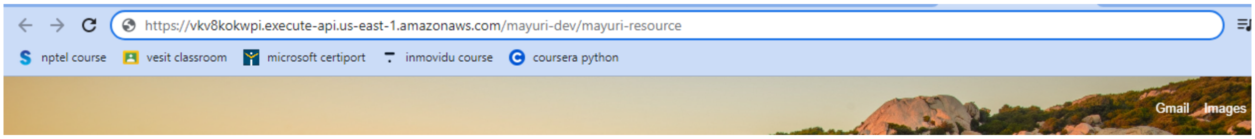
mayuri-dev - GET - /mayuri-resource

Invoke URL: https://vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev/mayuri-resource

Use this page to override the mayuri-dev stage settings for the GET to /mayuri-resource method.

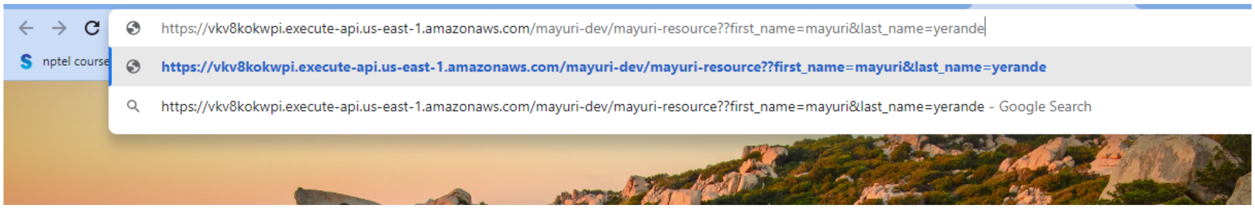
Settings ● Inherit from stage
○ Override for this method

- Copy the url
- Open new tab in your browser and paste it



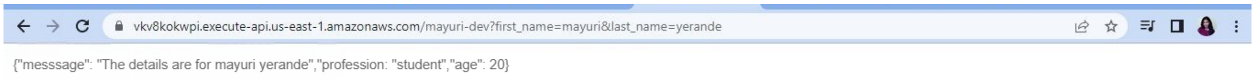
https://vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev/mayuri-resource

- Now add the parameters for first and last name ?first_name=yourfirstname&last_name=yourlastname



```
https://vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev/mayuri-resource??first_name=mayuri&last_name=yerande
https://vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev/mayuri-resource??first_name=mayuri&last_name=yerande
https://vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev/mayuri-resource??first_name=mayuri&last_name=yerande - Google Search
```

- **Now your Details are displayed successfully.**



```
vkv8kokwpi.execute-api.us-east-1.amazonaws.com/mayuri-dev?first_name=mayuri&last_name=yerande
{"messsage": "The details are for mayuri yerande","profession: "student","age": 20}
```

<u>CONCLUSION:</u> Thus we have successfully created an API using API gateway and Lambda Function.