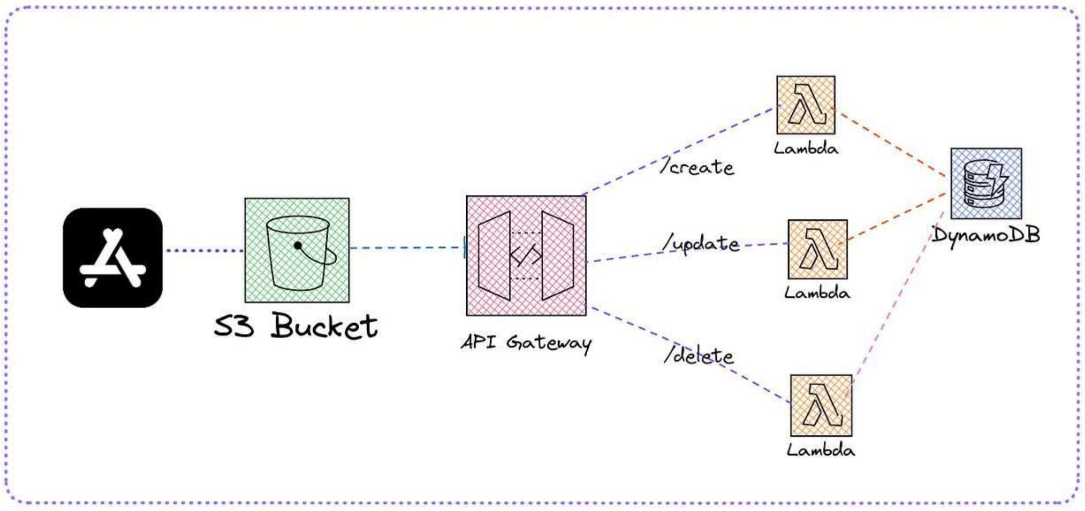
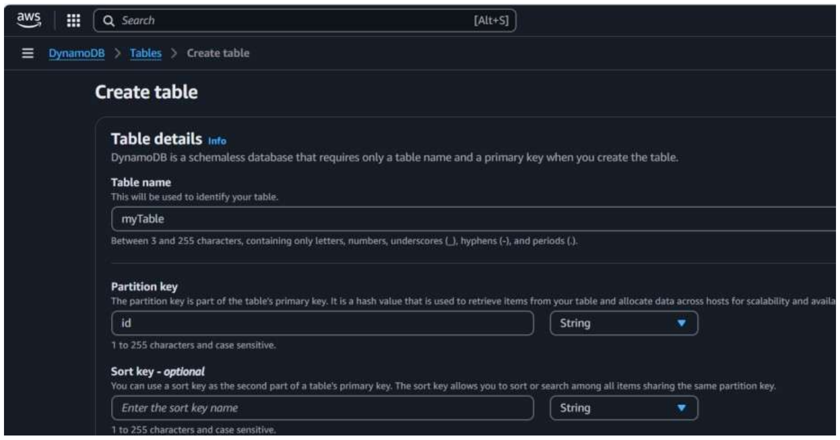
AWS Serverless Hands on Project:



In today's app world, going serverless isn't just a trend; it's a game-changer. Free up from traditional hassles and focus on creating awesome user experiences. Let's dive in, break free from the norm, and craft a serverless app that speaks the language of future tech!

We explore serverless with AWS — API Gateway, Lambda, DynamoDB, and S3. Where AWS API Gateway conducts requests seamlessly, Lambda powers the backend, DynamoDB scales effortlessly, and S3 hosts your static site.

1. **First, we have to create DynamoDB table for our application:**
   a. Navigate to the AWS console -> DynamoDB -> Tables
   b. Click on create table



DynamoDB > Tables > Create table

Create table

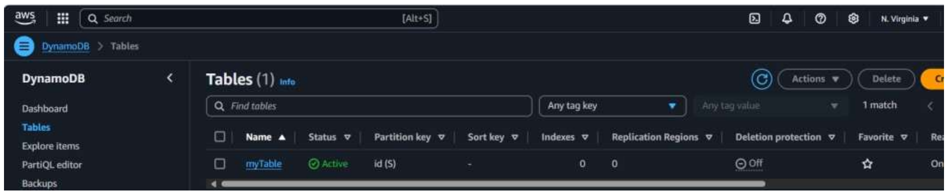
Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
myTable
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availa
id — String
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name — String
1 to 255 characters and case sensitive.



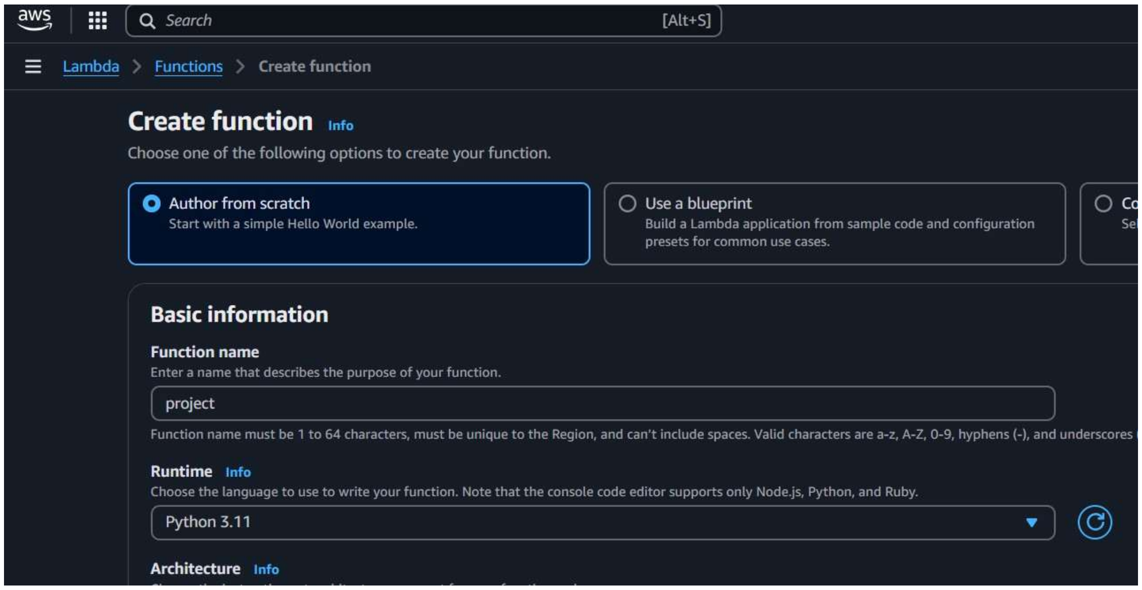
DynamoDB > Tables

Tables (1) Info

1 match

	Name ▲	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Re
☐	myTable	Active	id (S)	-	0	0	Off	☆	On

2. **Creating Lambda Functions**
   a. Navigate to AWS Console -> Lambda -> Create Function



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.
- ( ) Co… Se…

Basic information

Function name
Enter a name that describes the purpose of your function.
project
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores

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 3.11

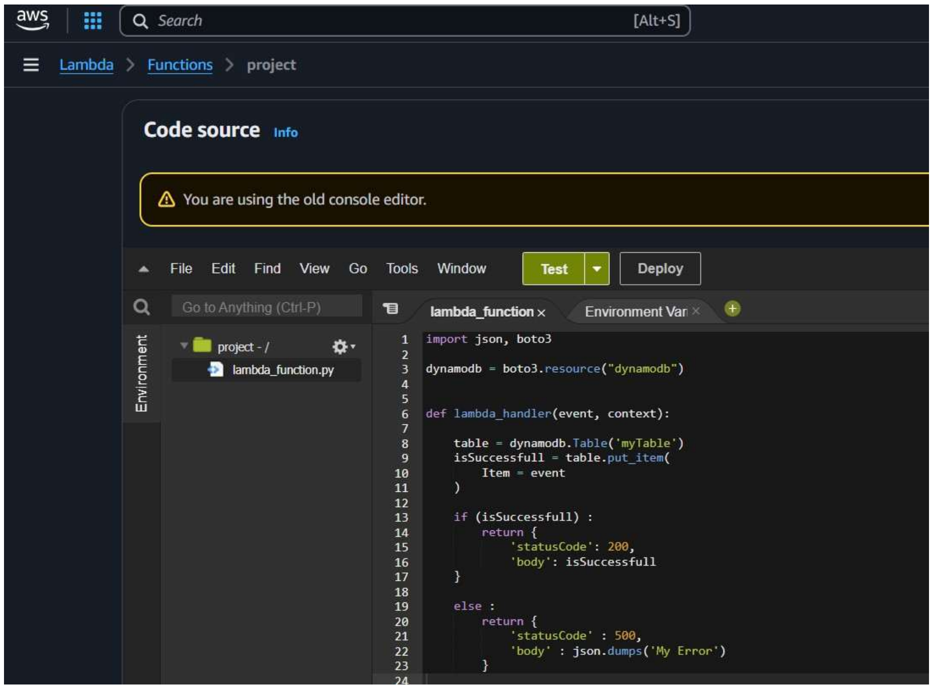
Architecture Info

b. Follow this configuration and click on create function

c. Copy the given code in the code editor:

```
import json, boto3

dynamodb = boto3.resource("dynamodb")


def lambda_handler(event, context):

  table = dynamodb.Table('myTable')
  isSuccessfull = table.put_item(
    Item = event
  )

  if (isSuccessfull) :
    return {
      'statusCode': 200,
      'body': isSuccessfull
  }

  else :
    return {
      'statusCode' : 500,
      'body' : json.dumps('My Error')
    }
```

d. Click on deploy to save it.

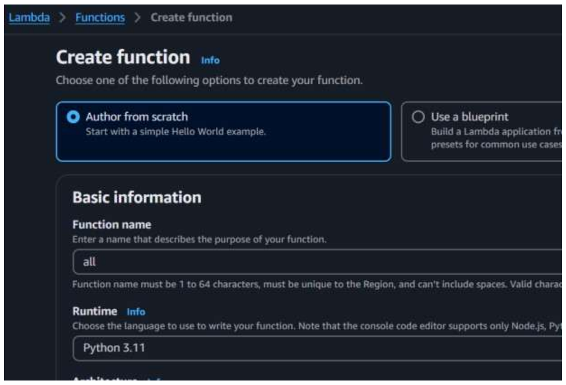


You are using the old console editor.

```
import json, boto3

dynamodb = boto3.resource("dynamodb")


def lambda_handler(event, context):

    table = dynamodb.Table('myTable')
    isSuccessfull = table.put_item(
        Item = event
    )

    if (isSuccessfull) :
        return {
            'statusCode': 200,
            'body': isSuccessfull
    }

    else :
        return {
            'statusCode' : 500,
            'body' : json.dumps('My Error')
        }
```

e. Create 2 more functions using the same steps, just name and code of the functions will change

Function name: all



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 fr
presets for common use cases

Basic information

Function name
Enter a name that describes the purpose of your function.
all
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid charac

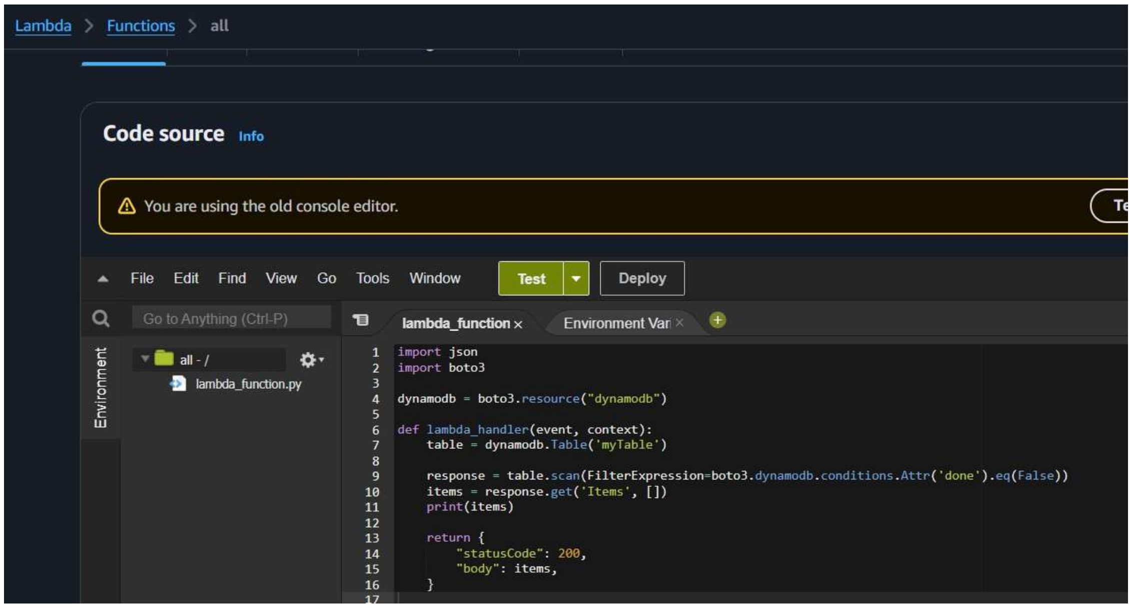
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Py
Python 3.11

Add this code into that:

```
import json
import boto3

dynamodb = boto3.resource("dynamodb")

def lambda_handler(event, context):
    table = dynamodb.Table('myTable')

    response = table.scan(FilterExpression=boto3.dynamodb.conditions.Attr('done').eq(False))
    items = response.get('Items', [])
```

```
print(items)

return {
  "statusCode": 200,
  "body": items,
}
```

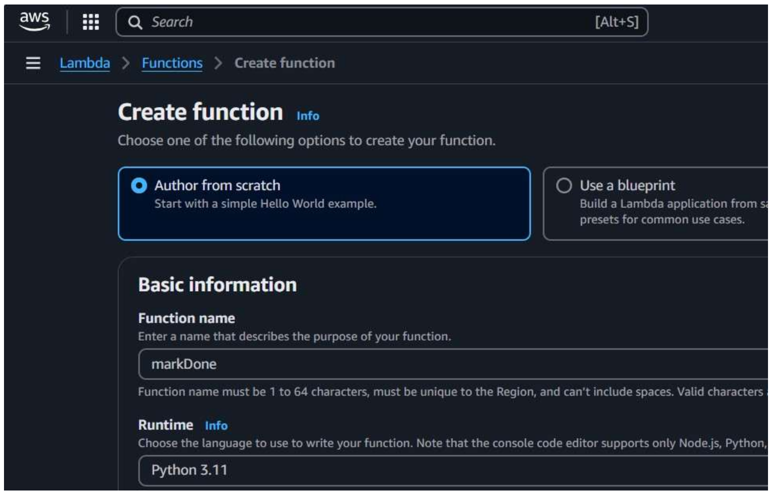
save it by deploy:



```
import json
import boto3

dynamodb = boto3.resource("dynamodb")

def lambda_handler(event, context):
    table = dynamodb.Table('myTable')

    response = table.scan(FilterExpression=boto3.dynamodb.conditions.Attr('done').eq(False))
    items = response.get('Items', [])
    print(items)

    return {
        "statusCode": 200,
        "body": items,
    }
```

Next, one more:

Function name: markDone



Create function

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 s
presets for common use cases.

Basic information

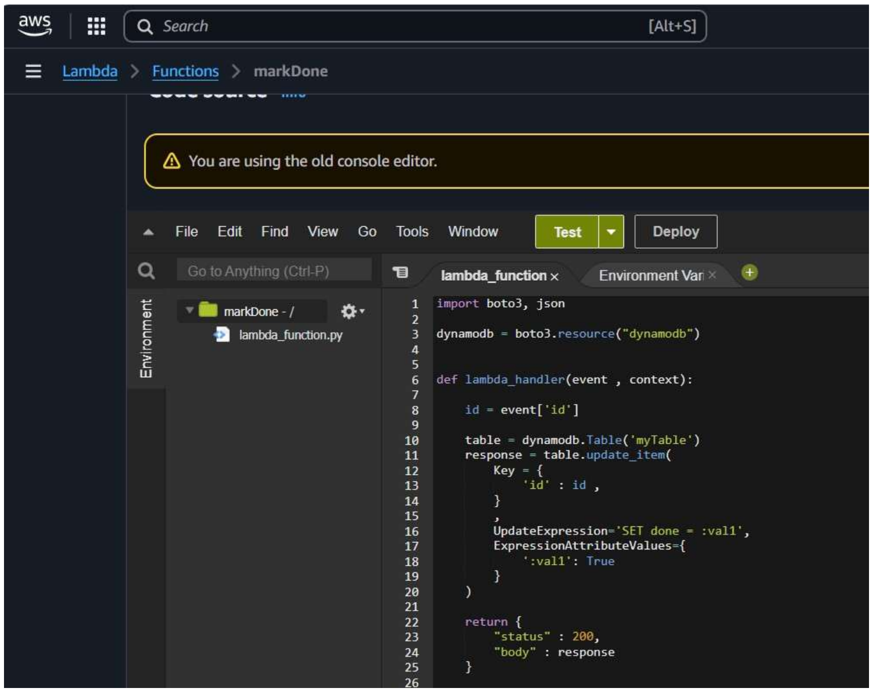
Function name
Enter a name that describes the purpose of your function.
markDone
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters

Runtime
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python,
Python 3.11

Add this code into that:

```
import boto3, json

dynamodb = boto3.resource("dynamodb")


def lambda_handler(event , context):

    id = event['id']

    table = dynamodb.Table('myTable')
    response = table.update_item(
        Key = {
            'id' : id ,
        }
        ,
        UpdateExpression='SET done = :val1',
        ExpressionAttributeValues={
            ':val1': True
        }
    )

    return {
        "status" : 200,
        "body" : response
    }
```

Use Deploy, to save the code.

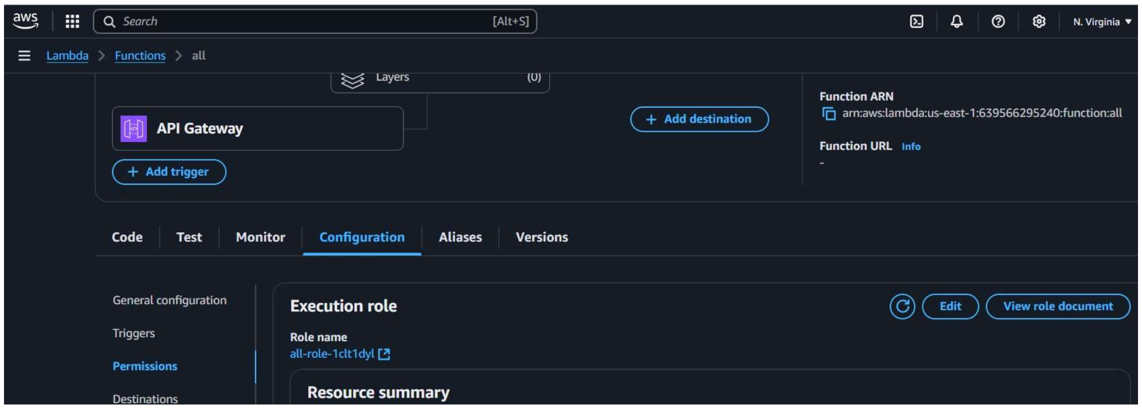


You are using the old console editor.

```
import boto3, json

dynamodb = boto3.resource("dynamodb")


def lambda_handler(event , context):

    id = event['id']

    table = dynamodb.Table('myTable')
    response = table.update_item(
        Key = {
            'id' : id ,
        }
        ,
        UpdateExpression='SET done = :val1',
        ExpressionAttributeValues={
            ':val1': True
        }
    )

    return {
        "status" : 200,
        "body" : response
    }
```

Note:

Make sure, in all lambda functions, must have these permissions, check under configuration permission tab, click on role and ensure these. If possible go to general configuration, set the timeout to 1min.


Lambda > Functions > all

API Gateway

Function ARN
arn:aws:lambda:us-east-1:639566295240:function:all

Function URL -

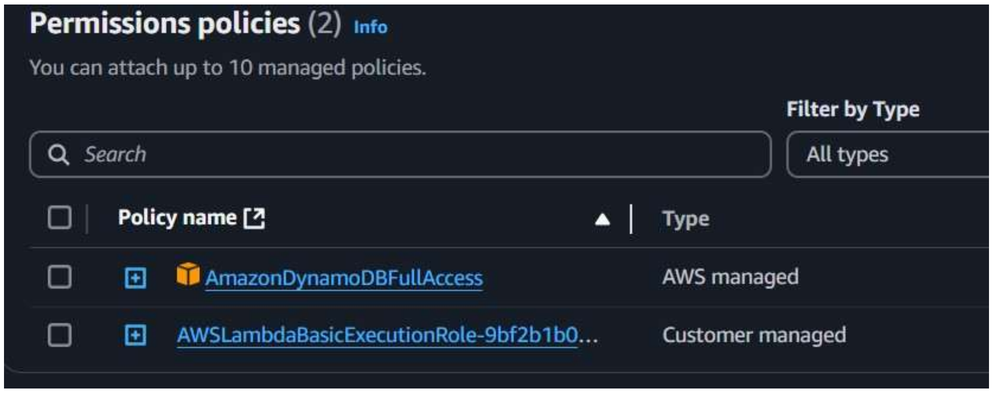
Code | Test | Monitor | Configuration | Aliases | Versions

General configuration
Triggers
Permissions
Destinations

Execution role

Role name
all-role-1clt1dyl

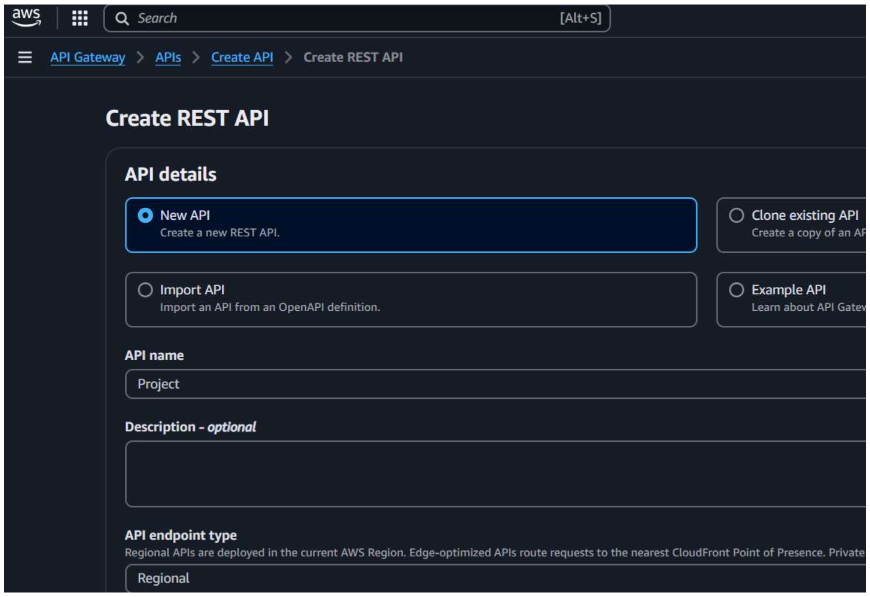
Resource summary


Permissions policies (2) Info

You can attach up to 10 managed policies.

Policy name	Type
AmazonDynamoDBFullAccess	AWS managed
AWSLambdaBasicExecutionRole-9bf2b1b0...	Customer managed

3. **Create Api Gateway**
   a. Navigate to AWS Console -> Api Gateway -> Creat API -> Create Rest API Gateway
   b. Select this configuration


API Gateway > APIs > Create API > Create REST API

Create REST API

API details

- [✓ New API] Create a new REST API.
- Clone existing API
- Import API — Import an API from an OpenAPI definition.
- Example API

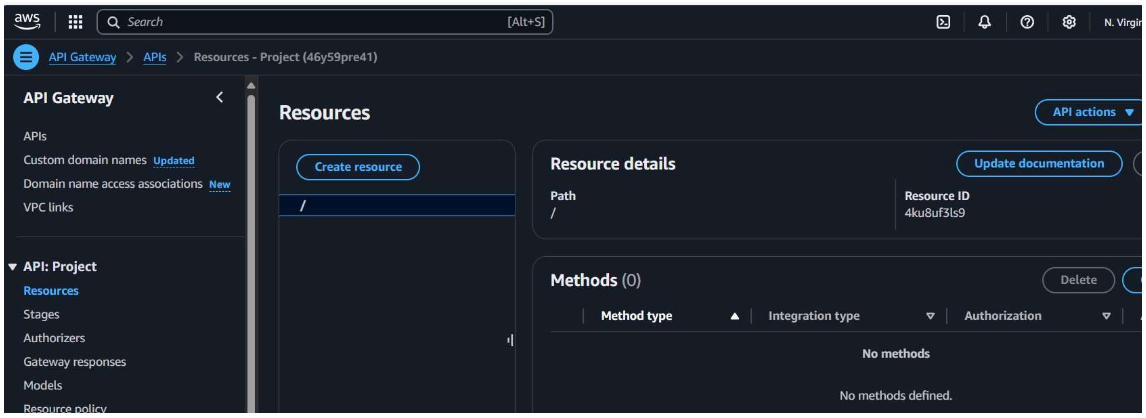
API name: Project

Description - *optional*:

API endpoint type: Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private

Regional

c. Click on create resource


API Gateway > APIs > Resources - Project (46y59pre41)

API Gateway: APIs, Custom domain names, Domain name access associations, VPC links

API: Project — Resources, Stages, Authorizers, Gateway responses, Models, Resource policy

Resources

Create resource

/

Resource details

Path: /

Resource ID: 4ku8uf3ls9

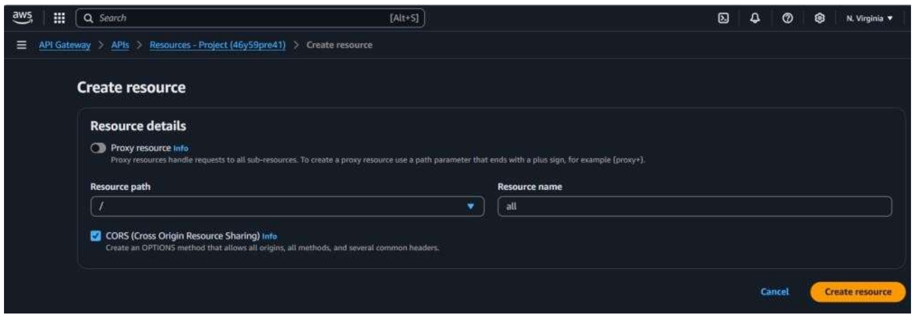
Methods (0)

Method type	Integration type	Authorization

No methods

No methods defined.

d. Follow this configuration


API Gateway > APIs > Resources - Project (46y59pre41) > Create resource

Create resource

Resource details

Proxy resource: off — Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

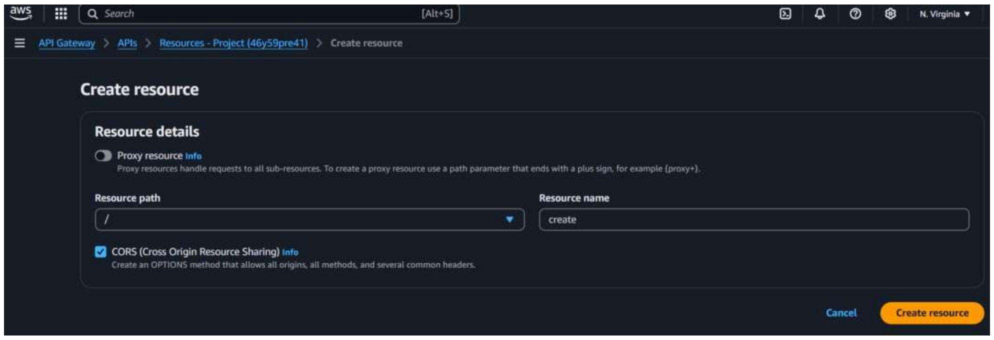
Resource path: /

Resource name: all

☑ CORS (Cross Origin Resource Sharing) — Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel | Create resource

e. Using the same steps, we have to 2 more resources, and name them as create and markDone.


API Gateway > APIs > Resources - Project (46y59pre41) > Create resource

Create resource

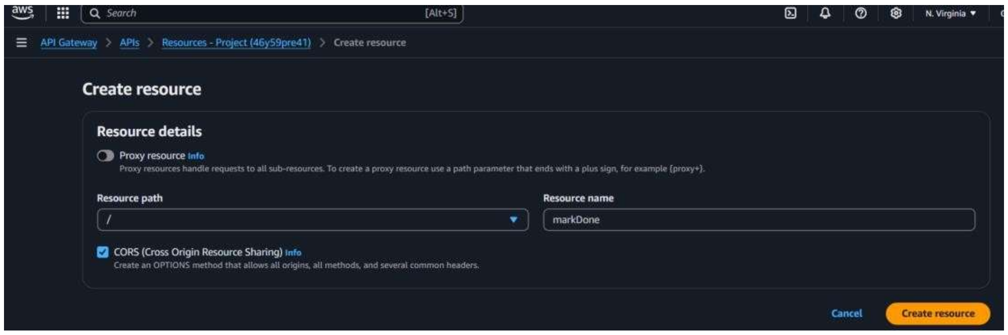
Resource details

Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path: /
Resource name: create

[✓] CORS (Cross Origin Resource Sharing) Info
Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel | Create resource


API Gateway > APIs > Resources - Project (46y59pre41) > Create resource

Create resource

Resource details

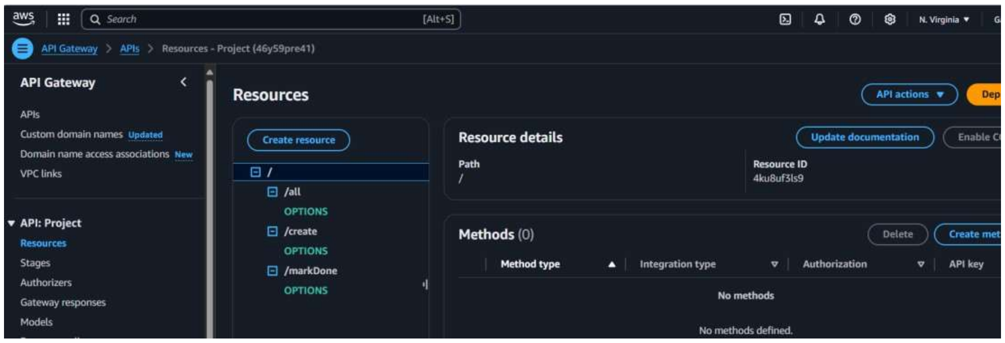
Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path: /
Resource name: markDone

[✓] CORS (Cross Origin Resource Sharing) Info
Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel | Create resource

f. After creating the resources console will look something like this.


API Gateway > APIs > Resources - Project (46y59pre41)

Resources

```
/
  /all
    OPTIONS
  /create
    OPTIONS
  /markDone
    OPTIONS
```

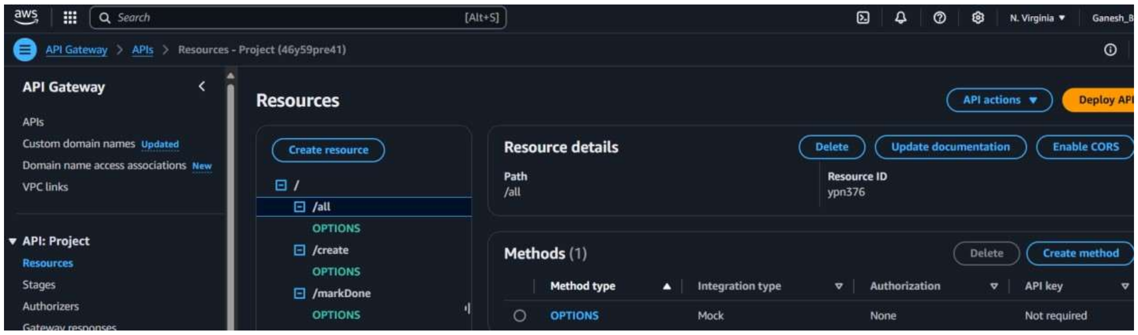
Resource details
Path: /
Resource ID: 4ku8uf3ls9

Methods (0)

Method type	Integration type	Authorization	API key

No methods
No methods defined.

g. Now click on the /all on the resources and the click on create method


API Gateway > APIs > Resources - Project (46y59pre41)

Resources

```
/
  /all
    OPTIONS
  /create
    OPTIONS
  /markDone
    OPTIONS
```

Resource details
Path: /all
Resource ID: ypn376

Methods (1)

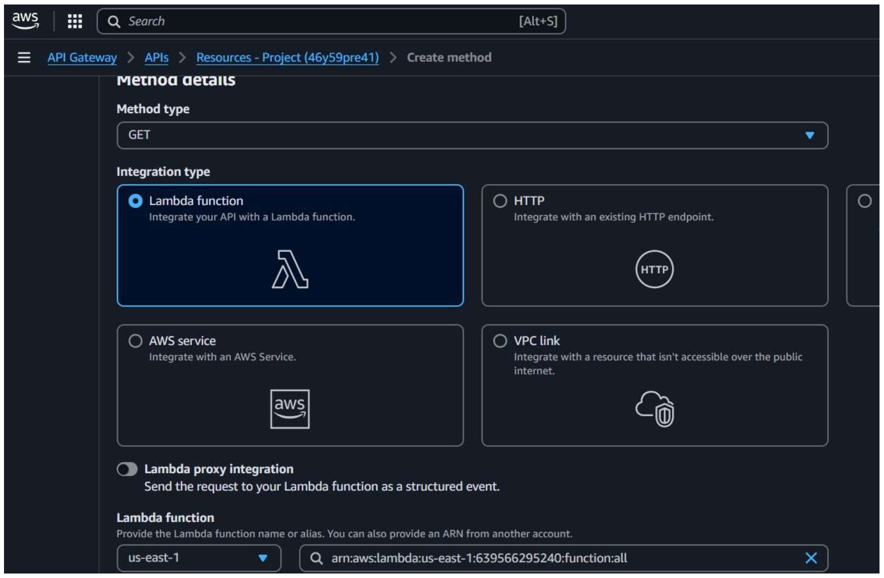
Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required

h. Click these settings,

Method: GET

Integration: Lambda

Lambda Function: all (we have created this function)



API Gateway > APIs > Resources - Project (46y59pre41) > Create method

Method details

Method type: GET

Integration type

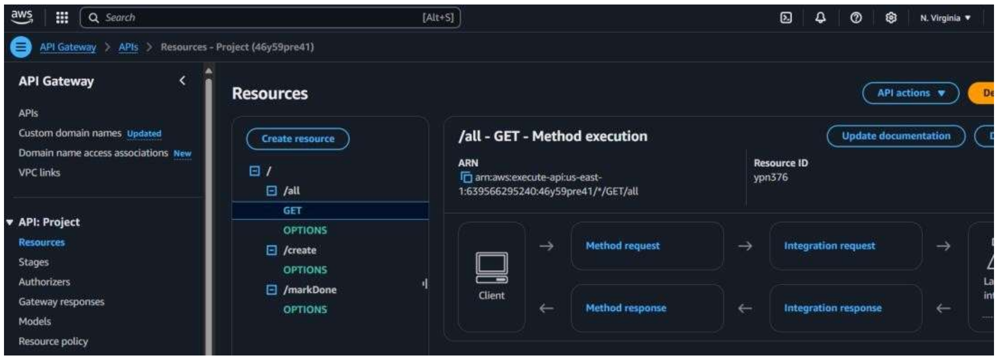
- (●) Lambda function — Integrate your API with a Lambda function.
- ( ) HTTP — Integrate with an existing HTTP endpoint.
- ( ) AWS service — Integrate with an AWS Service.
- ( ) VPC link — Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration — Send the request to your Lambda function as a structured event.

Lambda function — Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 | arn:aws:lambda:us-east-1:639566295240:function:all

Then click on create method



API Gateway > APIs > Resources - Project (46y59pre41)

Resources

```
/
  /all
    GET
    OPTIONS
  /create
    OPTIONS
  /markDone
    OPTIONS
```

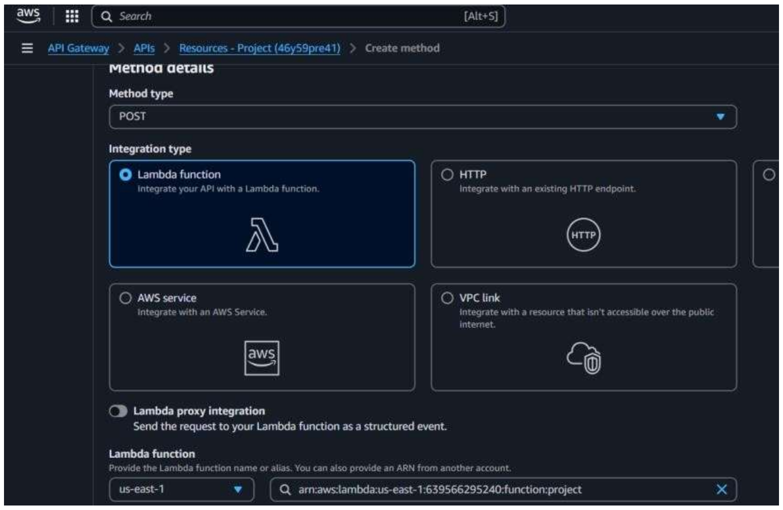
/all - GET - Method execution

ARN: arn:aws:execute-api:us-east-1:639566295240:46y59pre41/*/GET/all

Resource ID: ypn376

i. Similarly, we have to create the method for all the resource i.e /create and /markDone

j. For create, method type will be POST since we will be sending new todo objects using this and we keep lamdba function as project function.


API Gateway > APIs > Resources - Project (46y59pre41) > Create method

Method details

Method type: POST

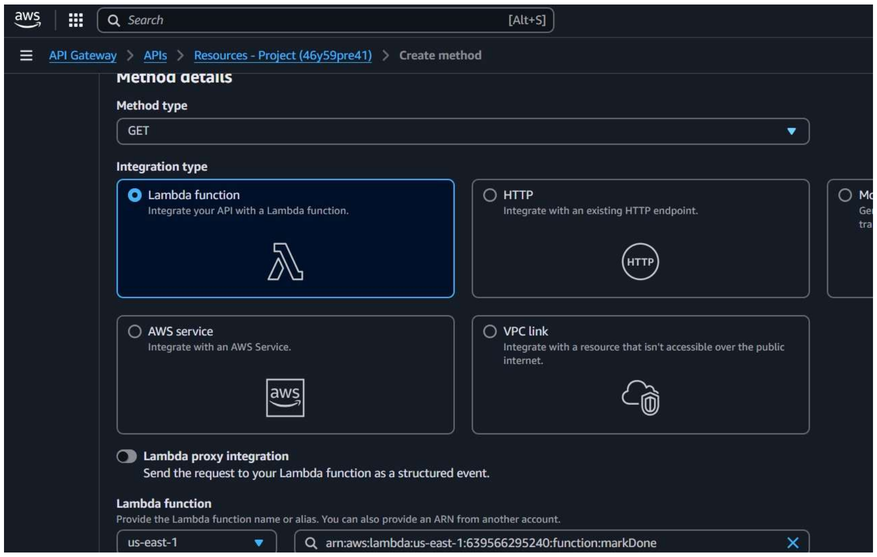
Integration type: Lambda function — Integrate your API with a Lambda function.

Lambda proxy integration — Send the request to your Lambda function as a structured event.

Lambda function — Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 | arn:aws:lambda:us-east-1:639566295240:function:project

k. Similary for markDone endpoint, using GET method and integrating with markDone lambda function.


API Gateway > APIs > Resources - Project (46y59pre41) > Create method

Method details

Method type: GET

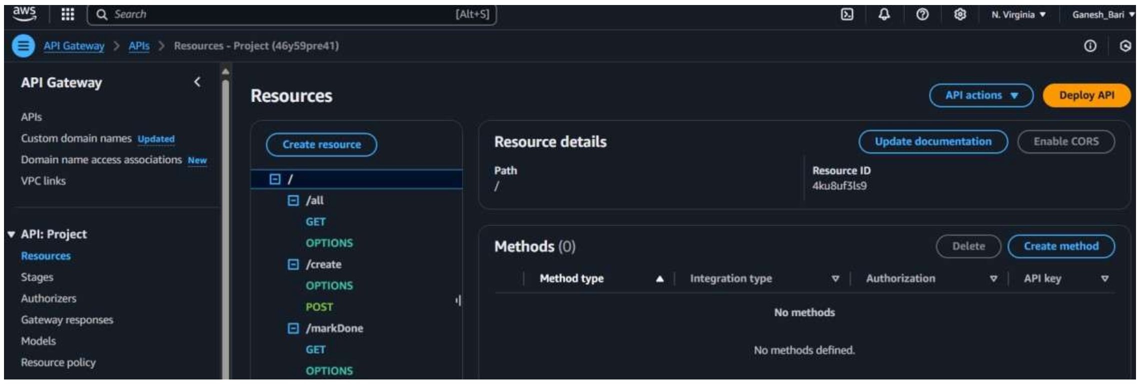
Integration type: Lambda function — Integrate your API with a Lambda function.

Lambda proxy integration — Send the request to your Lambda function as a structured event.

Lambda function — Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 | arn:aws:lambda:us-east-1:639566295240:function:markDone

L. After creating the endpoints now it is time to deploy our api server.


API Gateway > APIs > Resources - Project (46y59pre41)

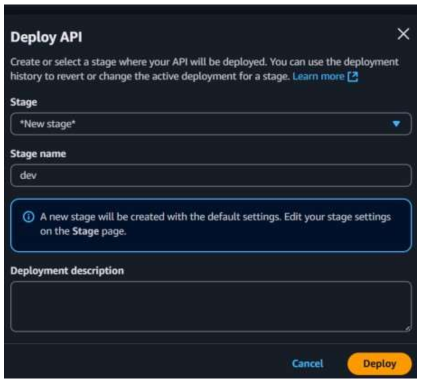
Resources

```
/
  /all
    GET
    OPTIONS
  /create
    OPTIONS
    POST
  /markDone
    GET
    OPTIONS
```

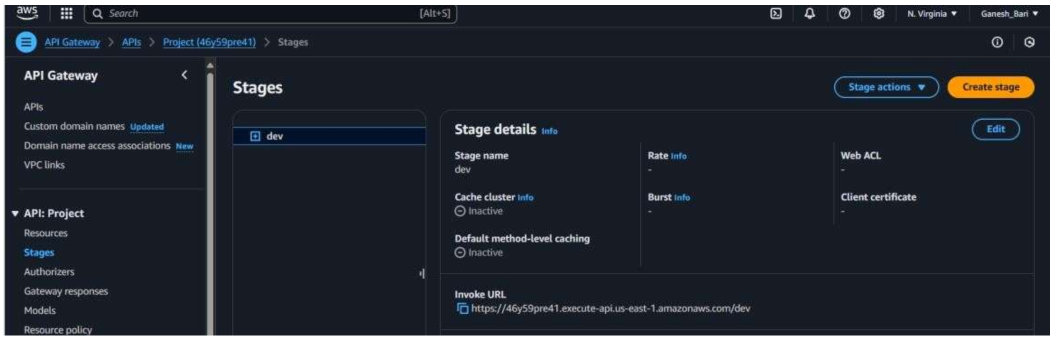
Resource details — Path: / — Resource ID: 4ku8uf3ls9

Methods (0) — No methods. No methods defined.

m. When prompted click on new stage, and name its dev. Then click on deploy.

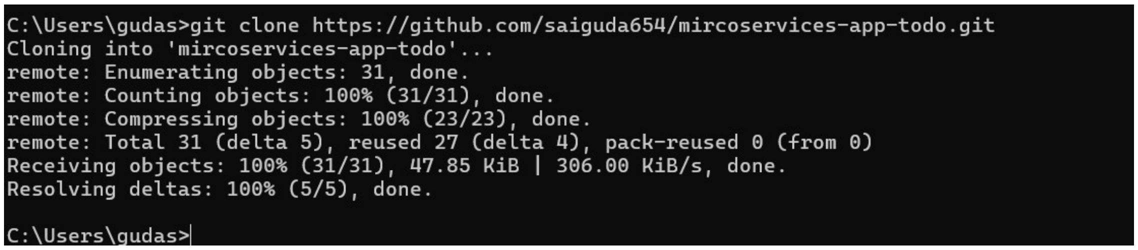

Deploy API
Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. Learn more
Stage
New stage
Stage name
dev
A new stage will be created with the default settings. Edit your stage settings on the Stage page.
Deployment description
Cancel
Deploy

n. This will take you to the stages section, copy the url of the api gateway.



4. Static Site Hosting

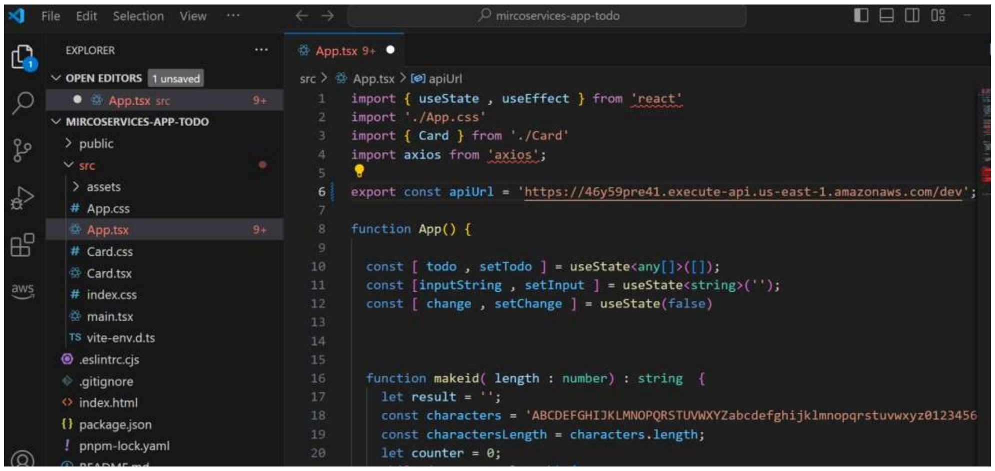
a. Clone this github repository https://github.com/saiguda654/mircoservices-app-todo.git



```
C:\Users\gudas>git clone https://github.com/saiguda654/mircoservices-app-todo.git
Cloning into 'mircoservices-app-todo'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 31 (delta 5), reused 27 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (31/31), 47.85 KiB | 306.00 KiB/s, done.
Resolving deltas: 100% (5/5), done.

C:\Users\gudas>
```

b. Open in your favourite code editor

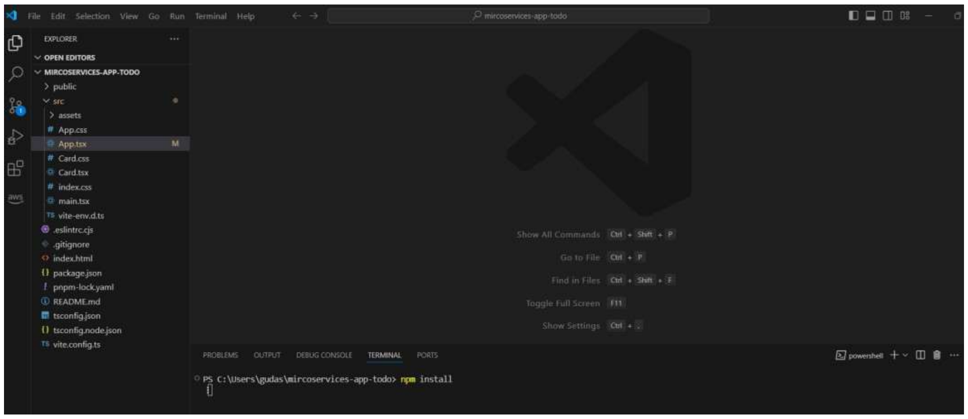
c. Open scr/App.tsx file and replace the api url with the api url of you api gateway.



```
import { useState , useEffect } from 'react'
import './App.css'
import { Card } from './Card'
import axios from 'axios';

export const apiUrl = 'https://46y59pre41.execute-api.us-east-1.amazonaws.com/dev';

function App() {

  const [ todo , setTodo ] = useState<any[]>([]);
  const [inputString , setInput ] = useState<string>('');
  const [ change , setChange ] = useState(false)


  function makeid( length : number) : string  {
    let result = '';
    const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
    const charactersLength = characters.length;
    let counter = 0;
```

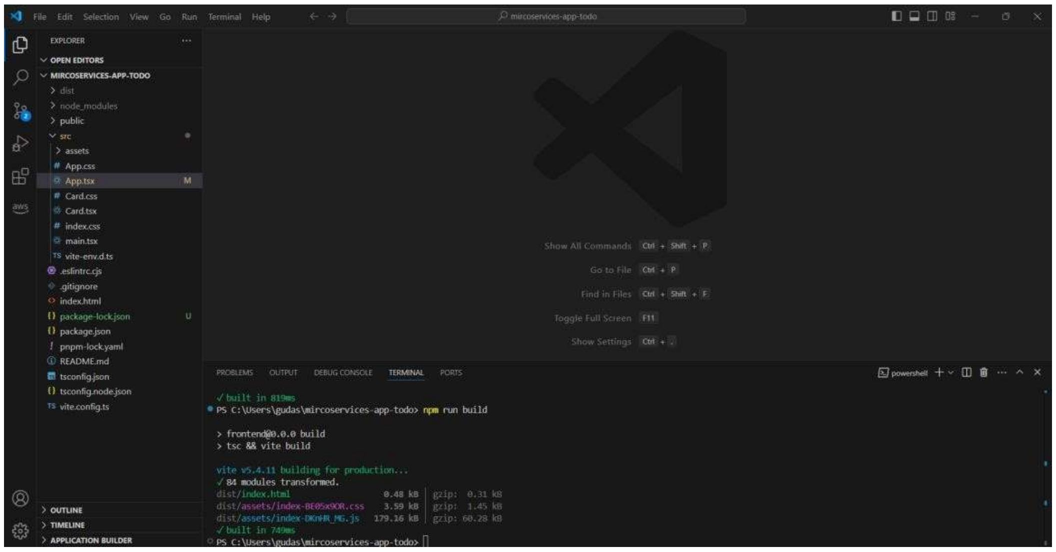
d. Run these command

npm install



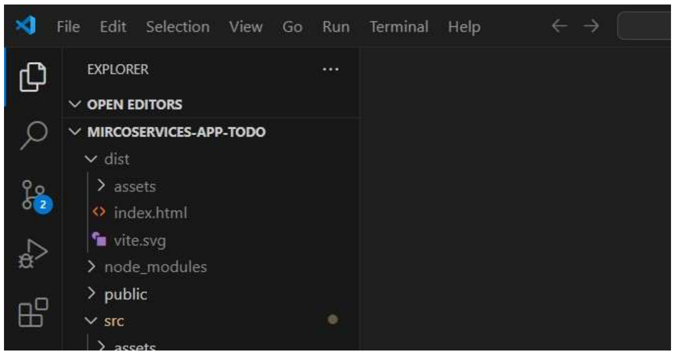
```
PS C:\Users\gudas\mircoservices-app-todo> npm install
```

npm run build



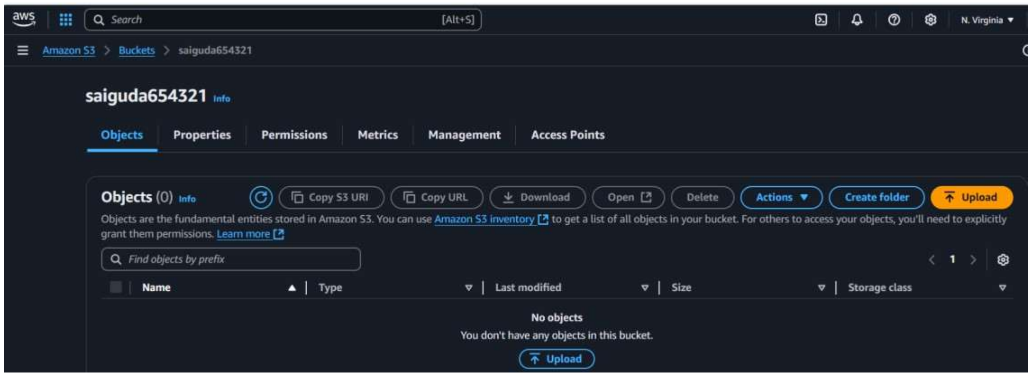
```
  ✓ built in 819ms
PS C:\Users\gudas\mircoservices-app-todo> npm run build

> frontend@0.0.0 build
> tsc && vite build

vite v5.4.11 building for production...
✓ 84 modules transformed.
dist/index.html                    0.48 kB │ gzip:  0.31 kB
dist/assets/index-BE05x9OR.css     3.59 kB │ gzip:  1.45 kB
dist/assets/index-DKnHR_MG.js    179.16 kB │ gzip: 60.28 kB
✓ built in 749ms
PS C:\Users\gudas\mircoservices-app-todo>
```

e. This will create dist folder in your repository with index.html file and assets.



f. We will put them in our s3 bucket, first let’s create one.

g. Navigate to aws console -> s3 -> create bucket, after creating the bucket, go to permission and unblock the public access.

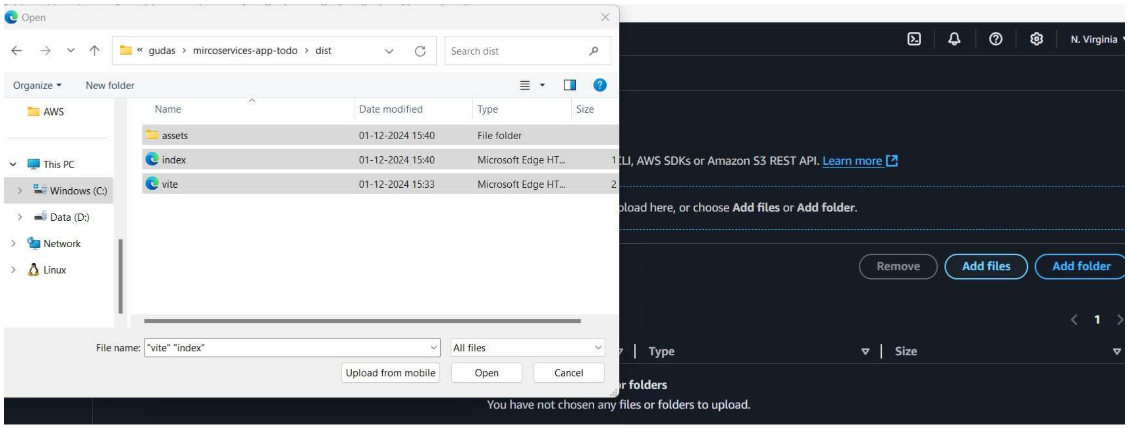


Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Now, upload:

h. Select the files in dist folder from your source code.

i. Upload all items one by one them.



Name	Date modified	Type	Size
assets	01-12-2024 15:40	File folder	
index	01-12-2024 15:40	Microsoft Edge HT...	1
vite	01-12-2024 15:33	Microsoft Edge HT...	2

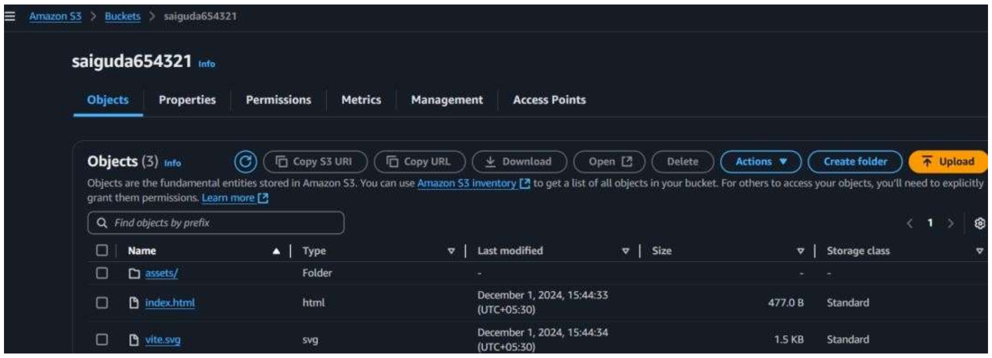
File name: "vite" "index"

LI, AWS SDKs or Amazon S3 REST API. Learn more

pload here, or choose Add files or Add folder.

r folders

You have not chosen any files or folders to upload.

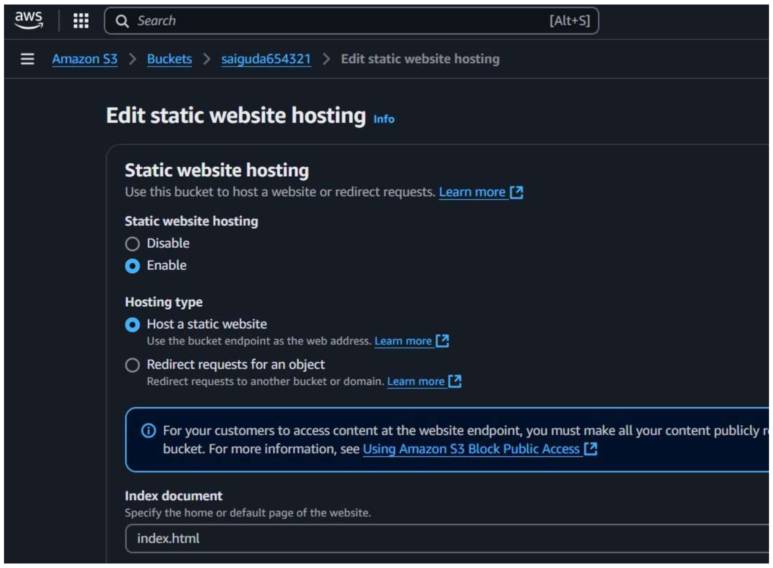


Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
assets/	Folder	-	-	-
index.html	html	December 1, 2024, 15:44:33 (UTC+05:30)	477.0 B	Standard
vite.svg	svg	December 1, 2024, 15:44:34 (UTC+05:30)	1.5 KB	Standard

j. To Configure S3 bucket, for static site host.

k. Go to properties in your bucket, scroll down and enable static site hosting


aws
Search
[Alt+S]
Amazon S3 > Buckets > saiguda654321 > Edit static website hosting
Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

Disable [✓ Enable]

Hosting type

[✓ Host a static website]
Use the bucket endpoint as the web address. Learn more

Redirect requests for an object
Redirect requests to another bucket or domain. Learn more

For your customers to access content at the website endpoint, you must make all your content publicly r
bucket. For more information, see Using Amazon S3 Block Public Access

Index document
Specify the home or default page of the website.
index.html

L. Click on create.

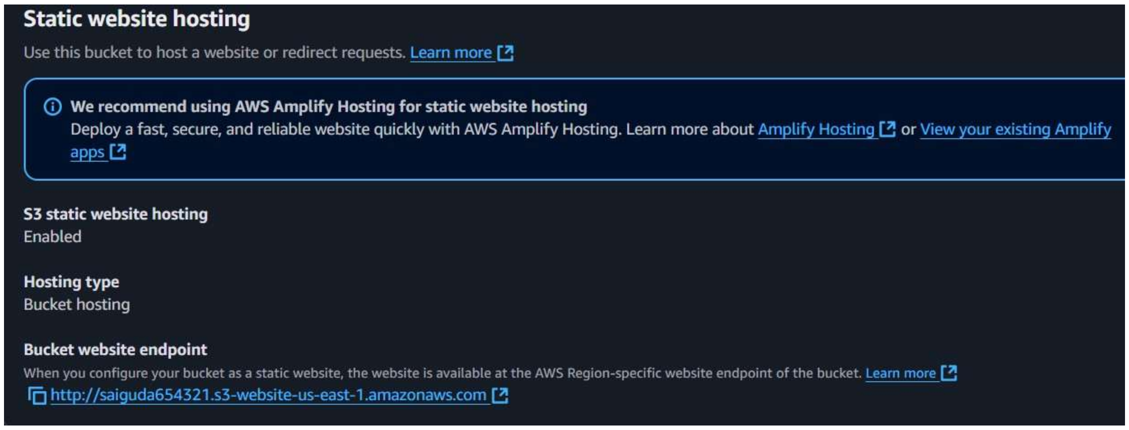
Now,

m. Setting up bucket policy in your bucket, go to permissions tab.

n. Click on permission, and then copy this bucket policy. Update the resource in this bucker policy with name of your bucket .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
            "Action": "s3:GetObject",
            "Resource":  "arn:aws:s3:::<---your-bucket-name---->/*"
        }
    ]
}
```

o. Now, move to the properties and search for URL under static website hosting and copy the URL of website.



Static website hosting

Use this bucket to host a website or redirect requests. Learn more

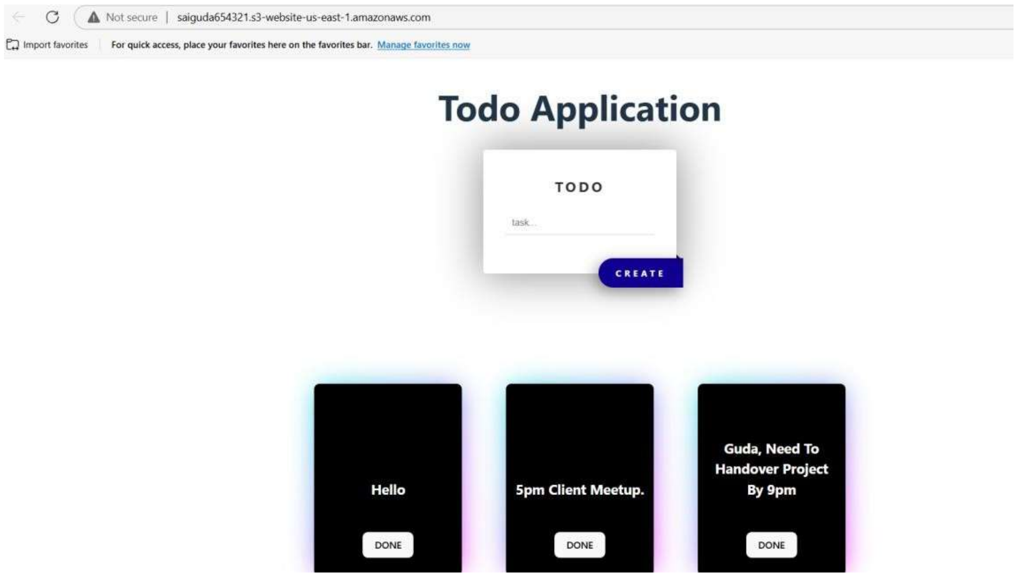
We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps

S3 static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://saiguda654321.s3-website-us-east-1.amazonaws.com

p. Open the URL in the browser.

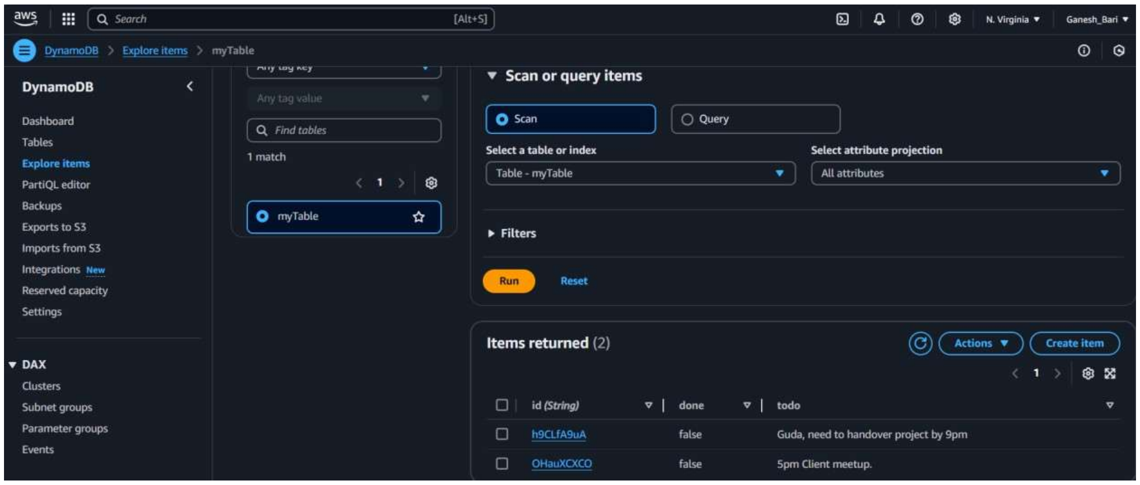


Not secure | saiguda654321.s3-website-us-east-1.amazonaws.com

Import favorites | For quick access, place your favorites here on the favorites bar. Manage favorites now

Todo Application

Now, check these data in the DynamoDB table, select the table, explore the table items:



Select a table or index: Table - myTable

Select attribute projection: All attributes

Items returned (2)

id (String)	done	todo
h9CLfA9uA	false	Guda, need to handover project by 9pm
OHauXCXCO	false	5pm Client meetup.

Note: Don't forgot to delete the resources once you done.

THANK YOU

Regards:

Shashank M P