1. Building a Serverless Web Application

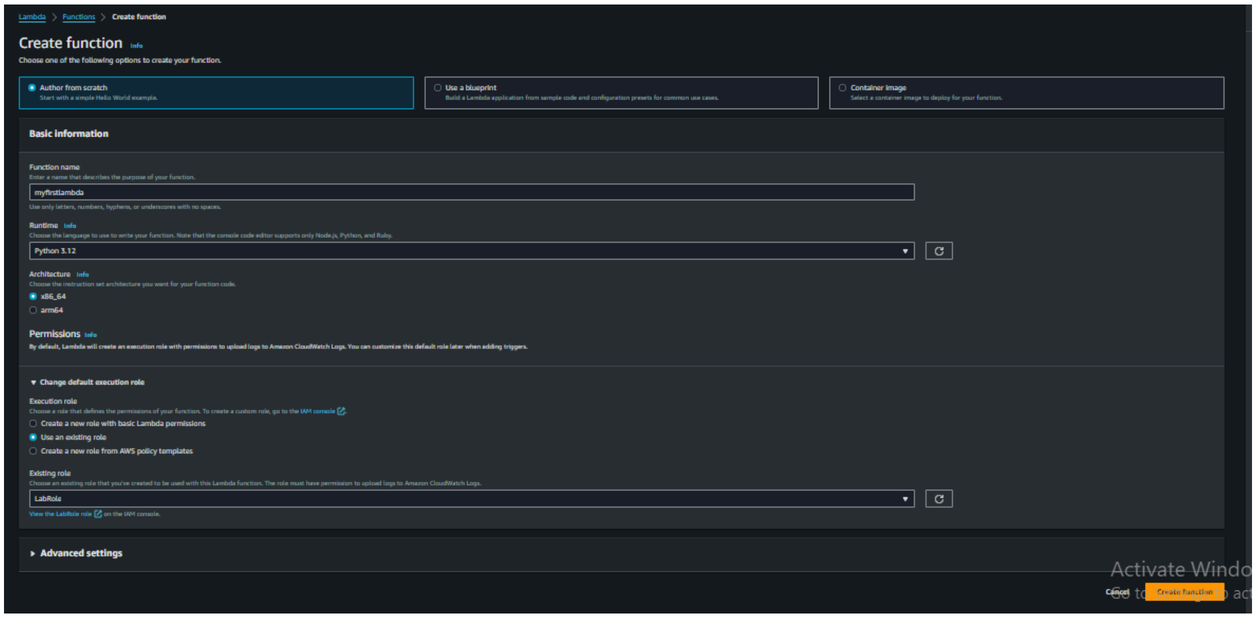
Objective: Create a serverless web application using AWS Lambda, API Gateway, S3, and DynamoDB.

Approach:

- **Set Up Backend**: Create Lambda functions to handle backend logic. These functions will interact with a DynamoDB table for data storage.
- **API Gateway**: Set up API Gateway to create RESTful endpoints that trigger the Lambda functions.
- **Frontend Hosting**: Host a static website on S3 that interacts with the backend via API Gateway.
- **Integration**: Ensure that the frontend can successfully send requests to the backend and display responses.

Goal: Understand the basics of building and connecting serverless backend services with a static frontend, enabling a fully serverless web application.

1. First create a lambda function, role was assigned to lab role



Lambda > Functions > 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.

Container image: Select a container image to deploy for your function.

Basic information

Function name: myfirstlambda

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

Runtime Info: Python 3.12

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

Architecture Info: [✓ x86_64] arm64

Choose the instruction set architecture you want for your function code.

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

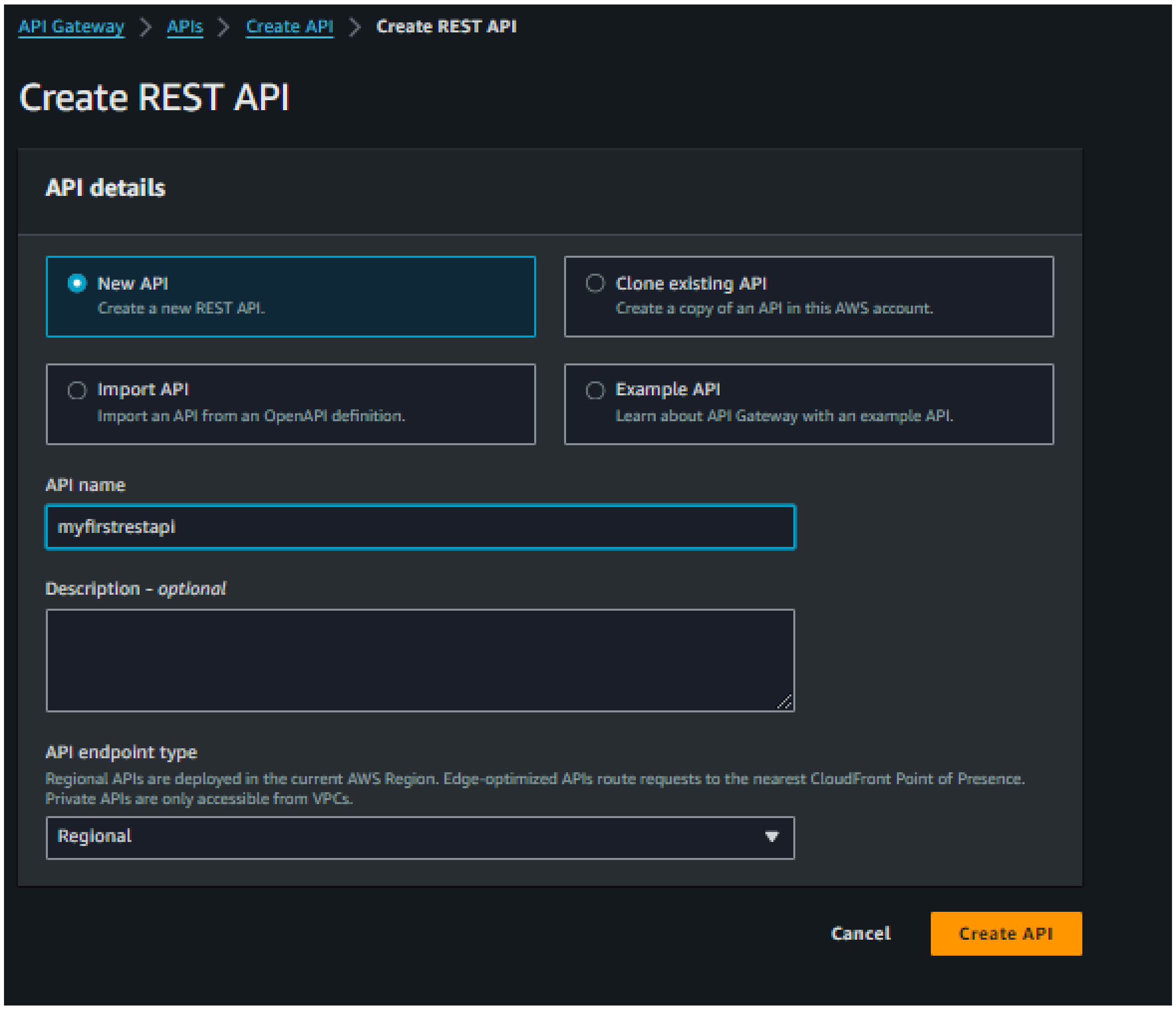
Existing role: LabRole

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

View the LabRole role on the IAM console.

▸ Advanced settings

2. Created new REST Api:


API Gateway > APIs > Create API > Create REST API

Create REST API

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: myfirstrestapi

Description - *optional*

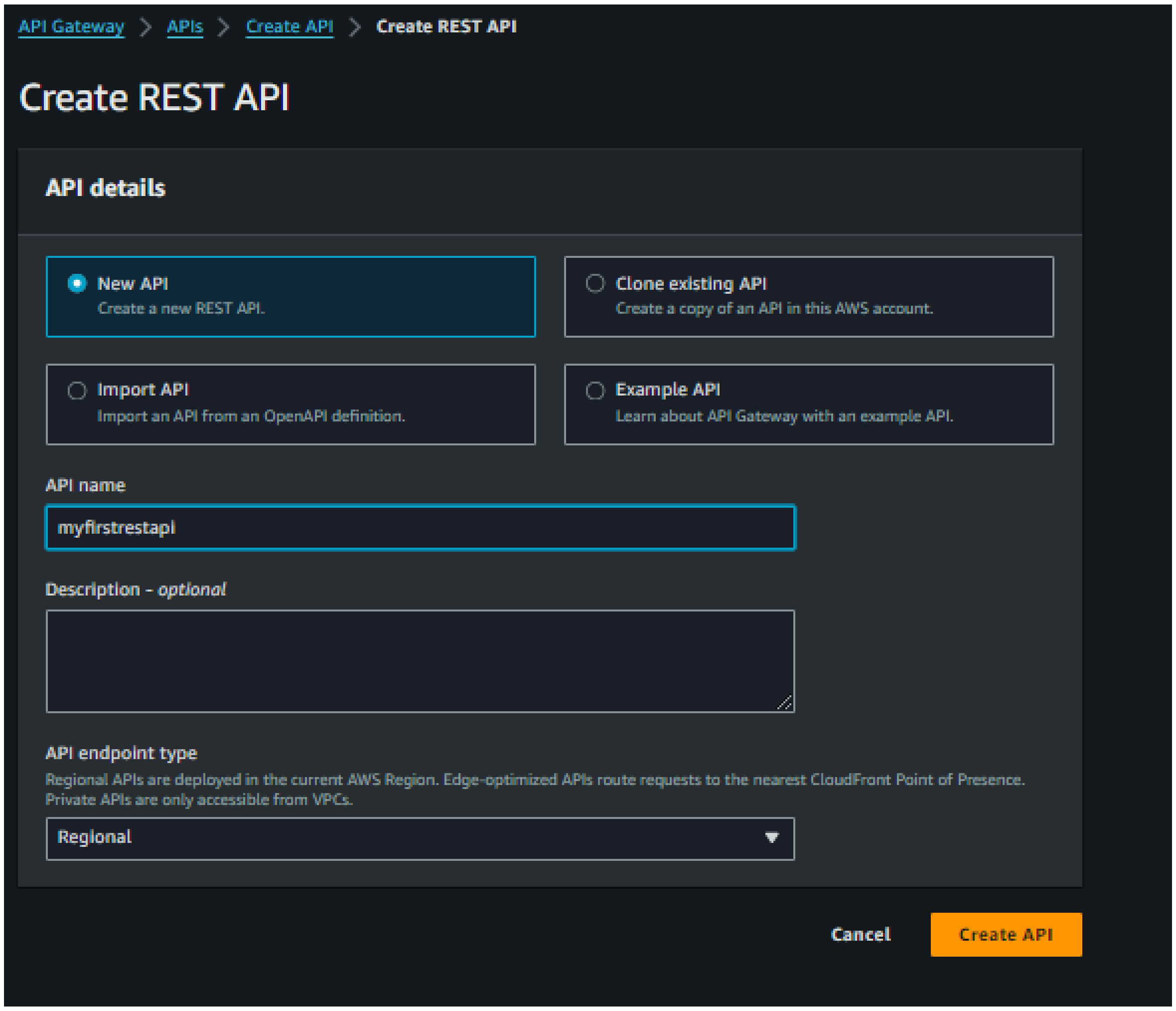
API endpoint type

Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

Regional

3.

3.POST Method was created within the previously created rest api



Create REST API

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.

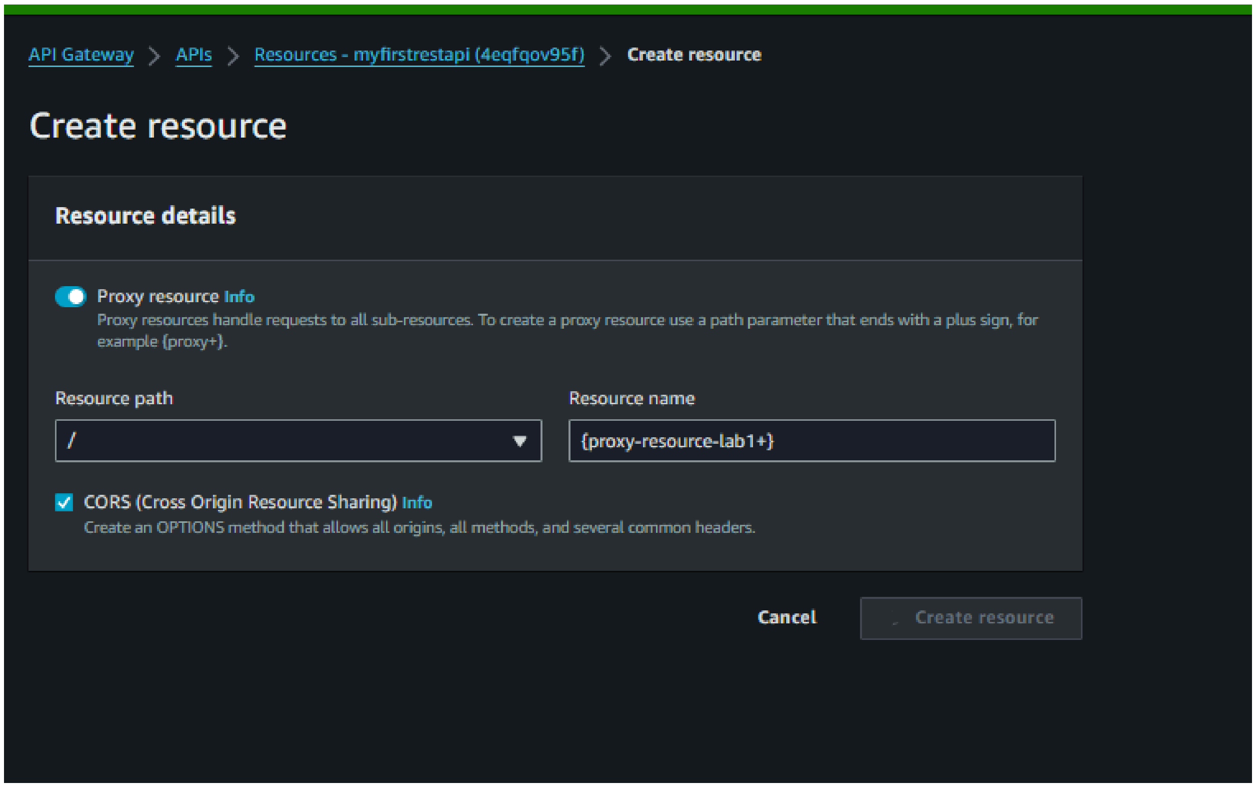
API name: myfirstrestapi

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 APIs are only accessible from VPCs.

Regional

3. Resource was created and path name was given as follow


API Gateway > APIs > Resources - myfirstrestapi (4eqfqov95f) > Create resource

Create resource

Resource details

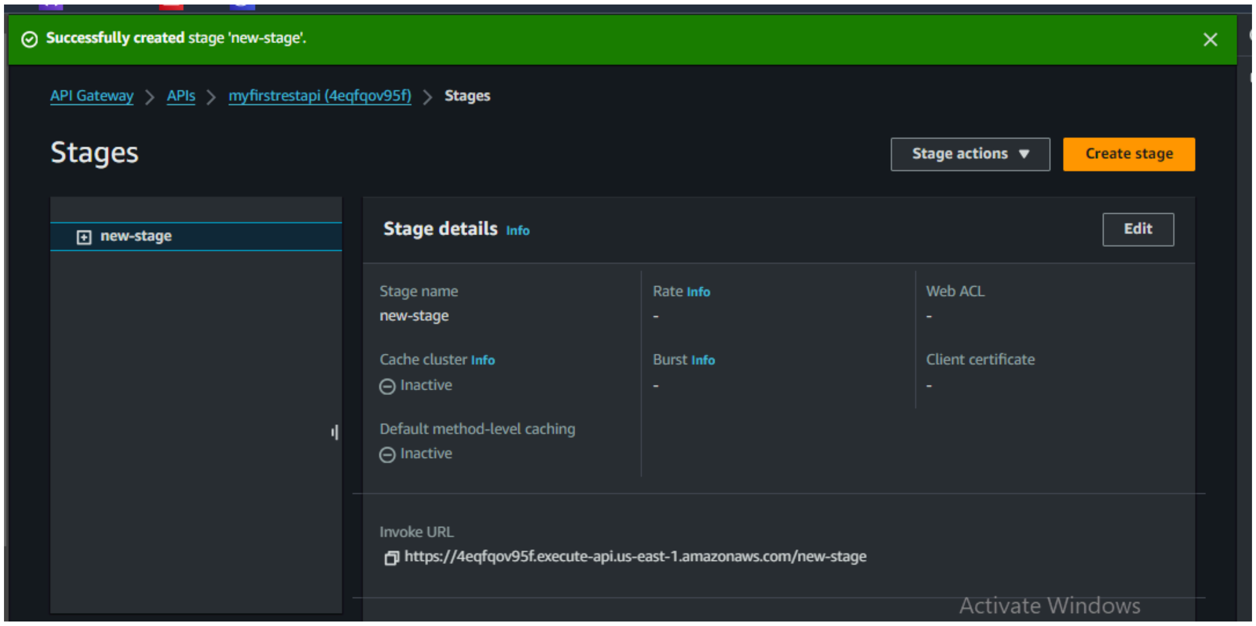
Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path: /

Resource name: {proxy-resource-lab1+}

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

Cancel | Create resource


Successfully created stage 'new-stage'.

API Gateway > APIs > myfirstrestapi (4eqfqov95f) > Stages

Stages

Stage actions ▼ | Create stage

new-stage

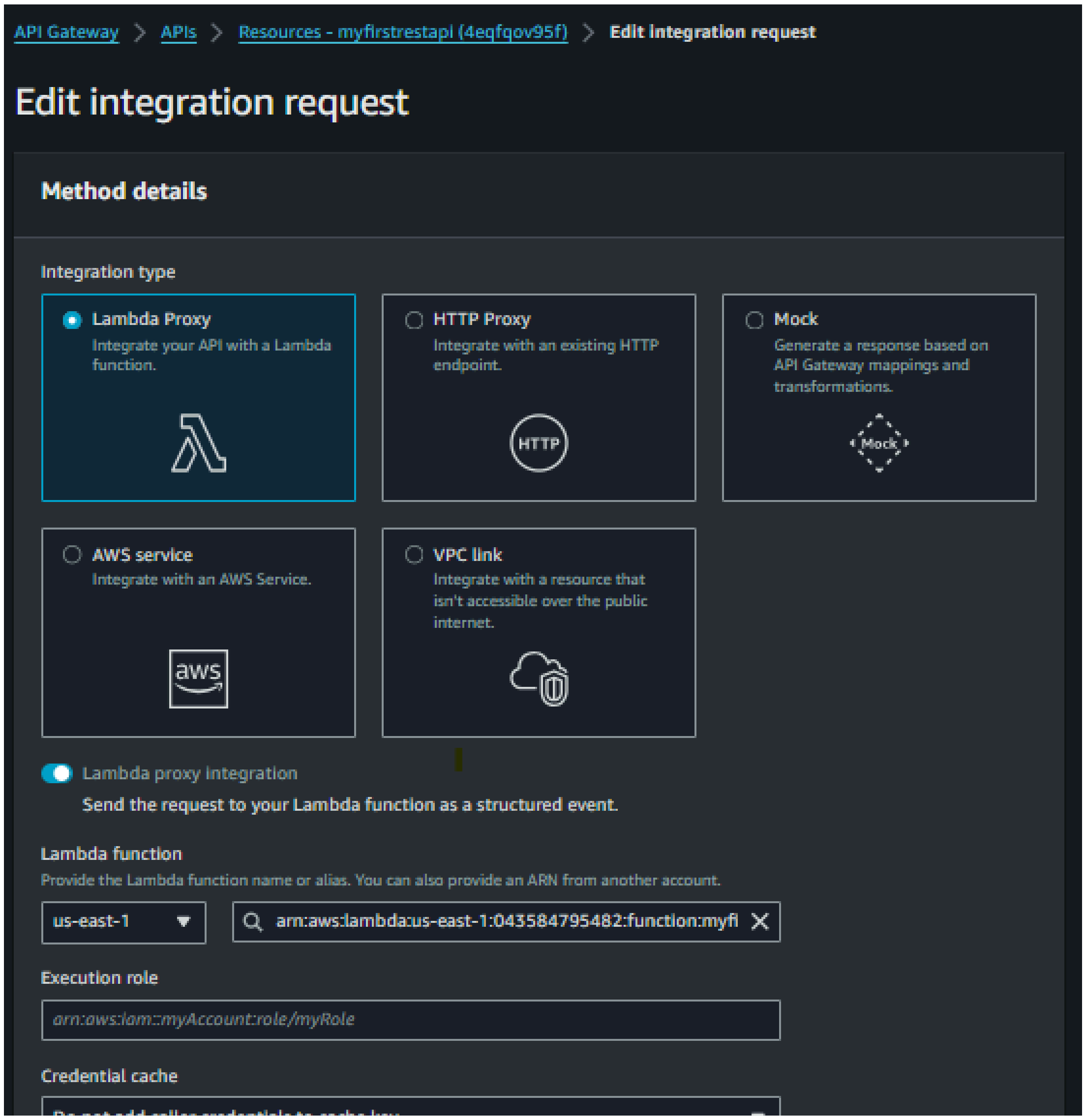
Stage details Info | Edit

Stage name: new-stage
Rate Info: -
Web ACL: -
Cache cluster Info: Inactive
Burst Info: -
Client certificate: -
Default method-level caching: Inactive

Invoke URL
https://4eqfqov95f.execute-api.us-east-1.amazonaws.com/new-stage

Activate Windows

4. Within the resource tab, click integration request, enable the proxy



API Gateway > APIs > Resources - myfirstrestapi (4eqfqov95f) > Edit integration request

Edit integration request

Method details

Integration type

[✓ Lambda Proxy] Integrate your API with a Lambda function.

HTTP Proxy: Integrate with an existing HTTP endpoint.

Mock: Generate a response based on API Gateway mappings and transformations.

AWS service: Integrate with an AWS Service.

VPC link: Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration: on

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

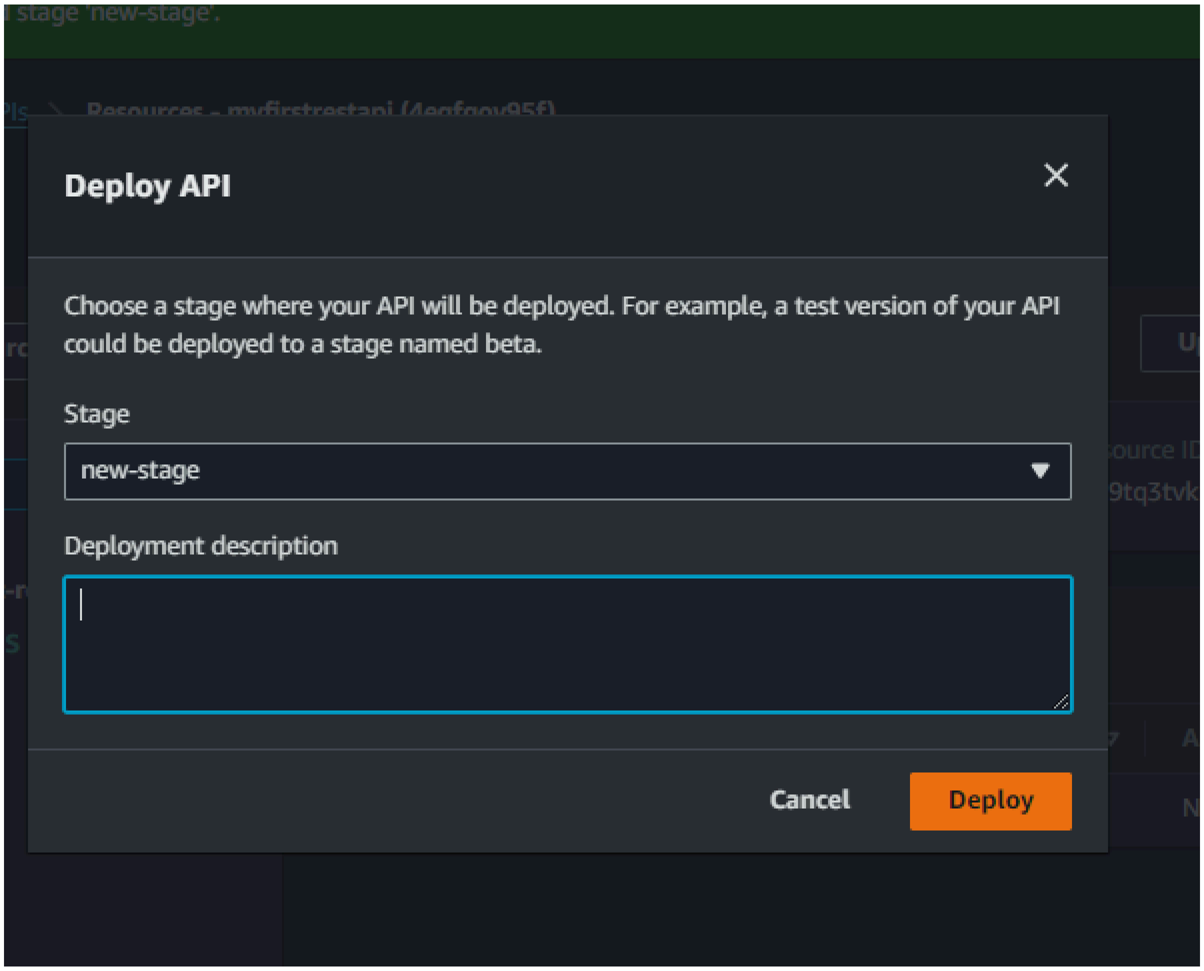
arn:aws:lambda:us-east-1:043584795482:function:myfi

Execution role

arn:aws:iam::myAccount:role/myRole

Credential cache

6. Finally deploy the api



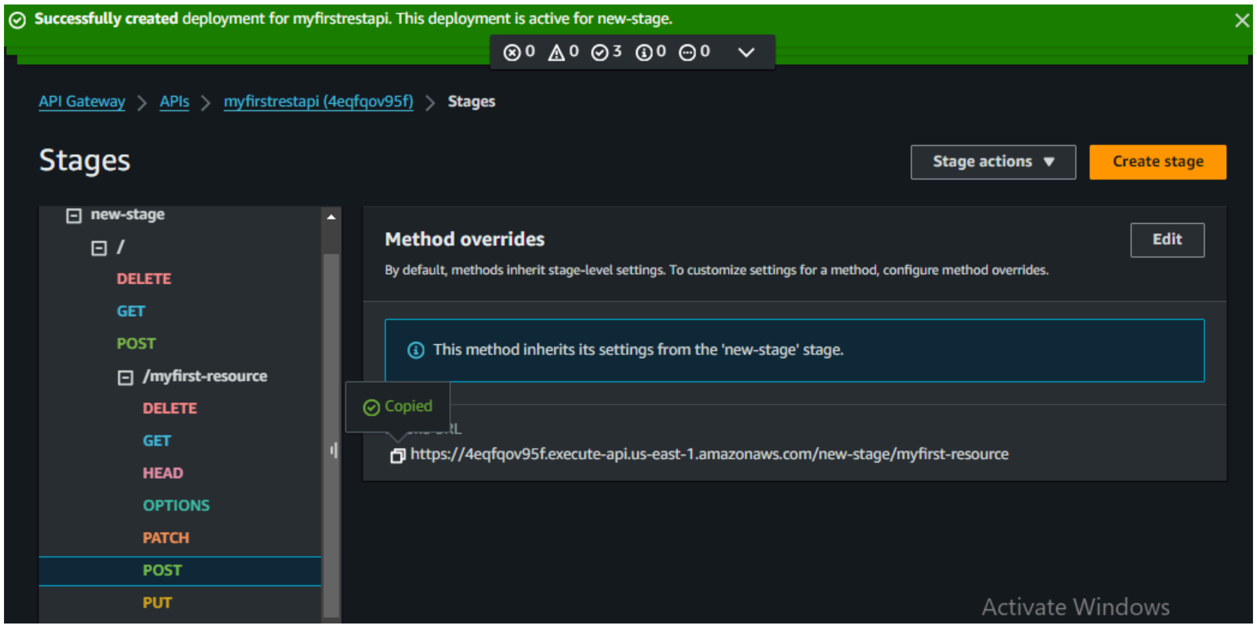
Deploy API

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.

Stage: new-stage

Deployment description:

7. Within the stages, click post and copy the url as below



Successfully created deployment for myfirstrestapi. This deployment is active for new-stage.

API Gateway > APIs > myfirstrestapi (4eqfqov95f) > Stages

Stages

Method overrides

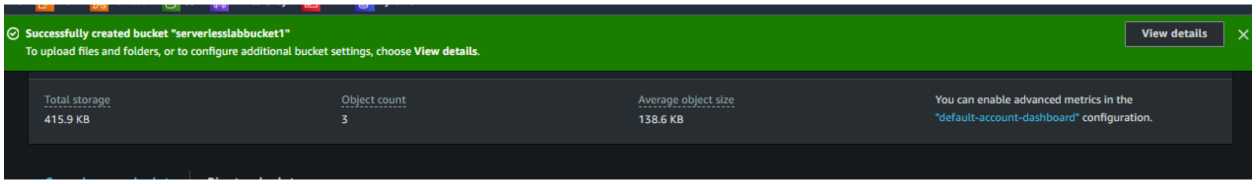
By default, methods inherit stage-level settings. To customize settings for a method, configure method overrides.

This method inherits its settings from the 'new-stage' stage.

Copied

https://4eqfqov95f.execute-api.us-east-1.amazonaws.com/new-stage/myfirst-resource

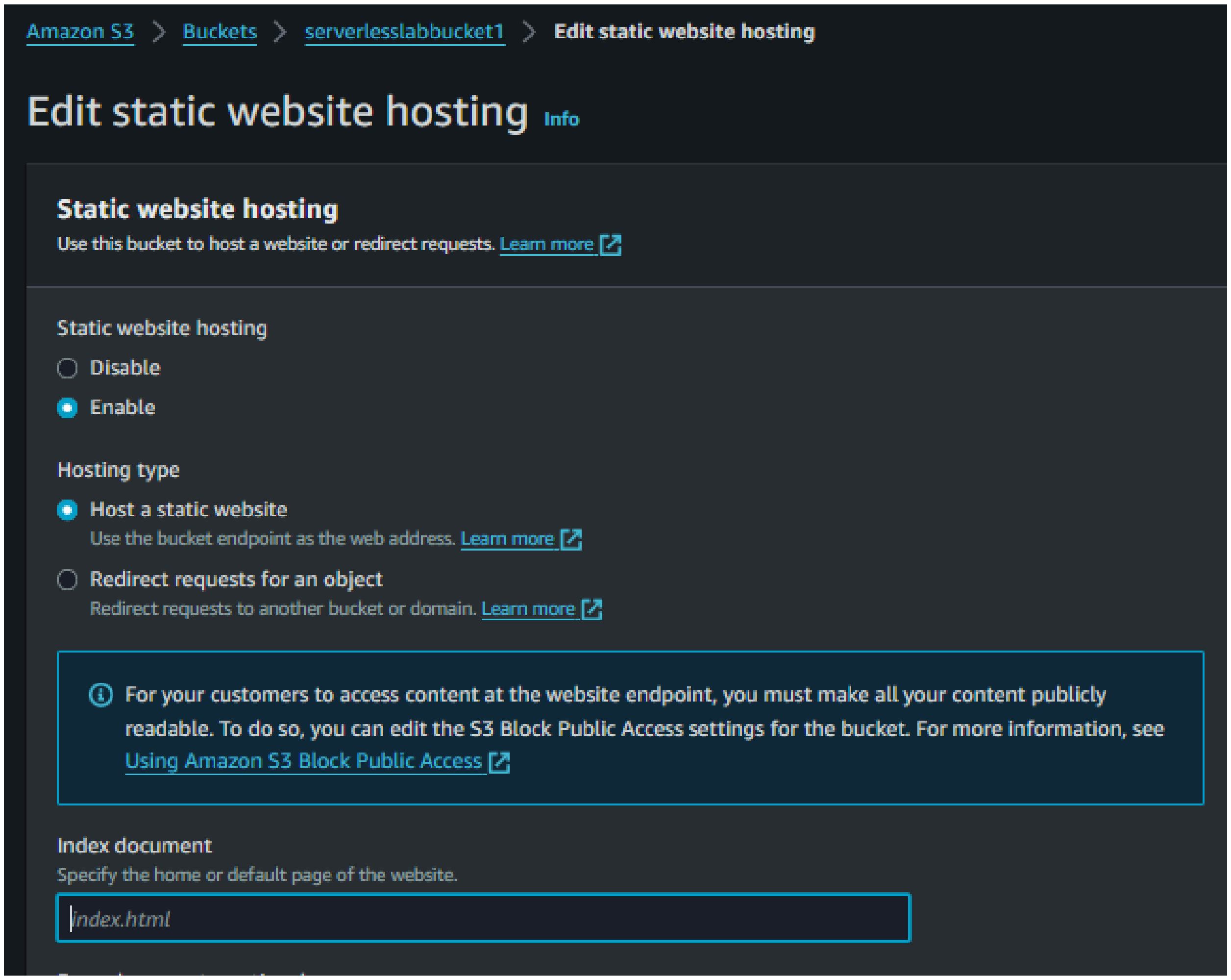
8. Create S3 bucket



Successfully created bucket "serverlesslabbucket1"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Total storage	Object count	Average object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
415.9 KB	3	138.6 KB	

9. Edit static website hosting



Amazon S3 > Buckets > serverlesslabbucket1 > **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

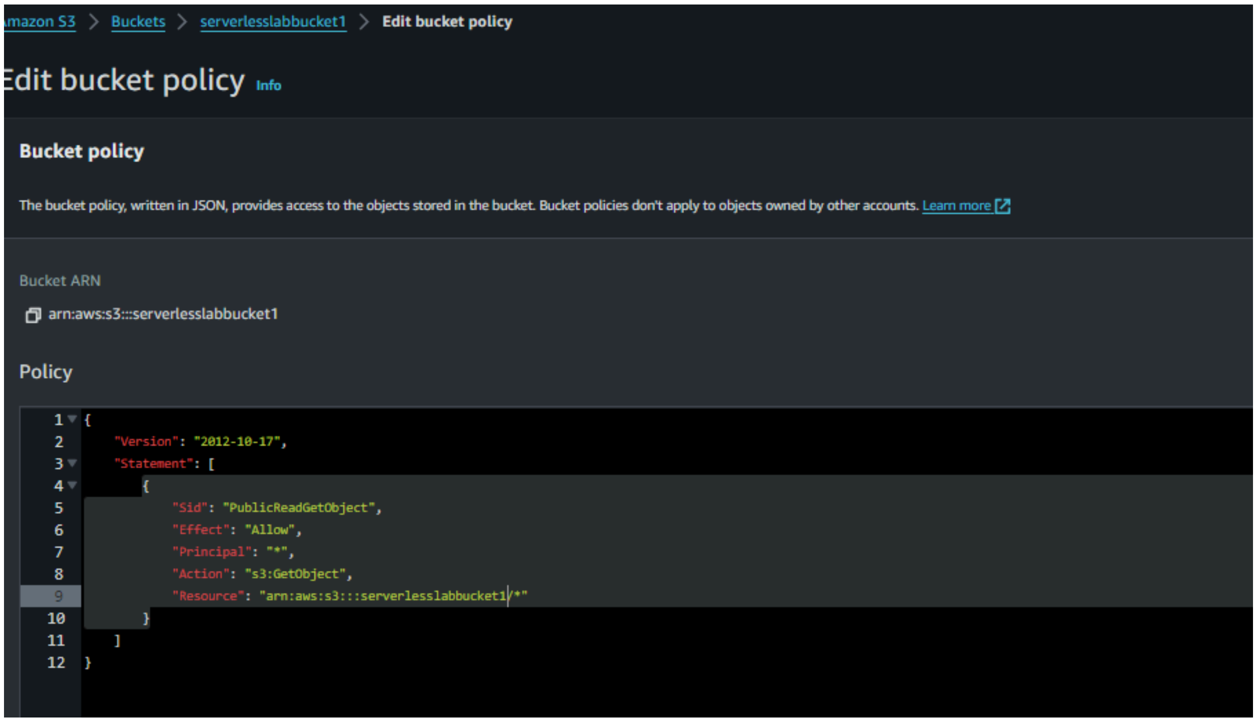
- [✓ Host a static website]
  Use the bucket endpoint as the web address. Learn more
- Redirect requests for an object
  Redirect requests to another bucket or domain. Learn more

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

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

index.html

10. Edit bucket policy



mazon S3 > Buckets > serverlesslabbucket1 > Edit bucket policy

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

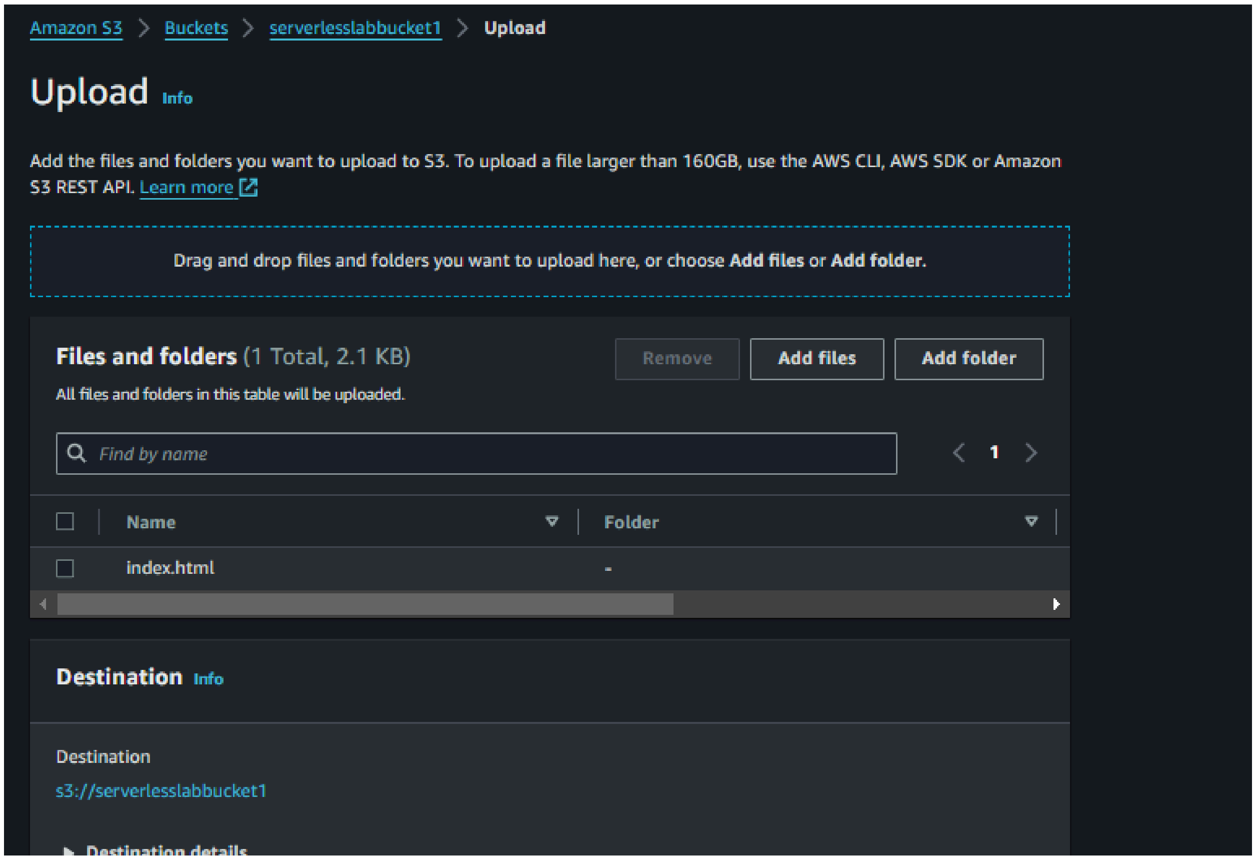
Bucket ARN

arn:aws:s3:::serverlesslabbucket1

Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::serverlesslabbucket1/*"
        }
    ]
}
```

11. Upload the required website file



Amazon S3 > Buckets > serverlesslabbucket1 > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 2.1 KB)

All files and folders in this table will be uploaded.

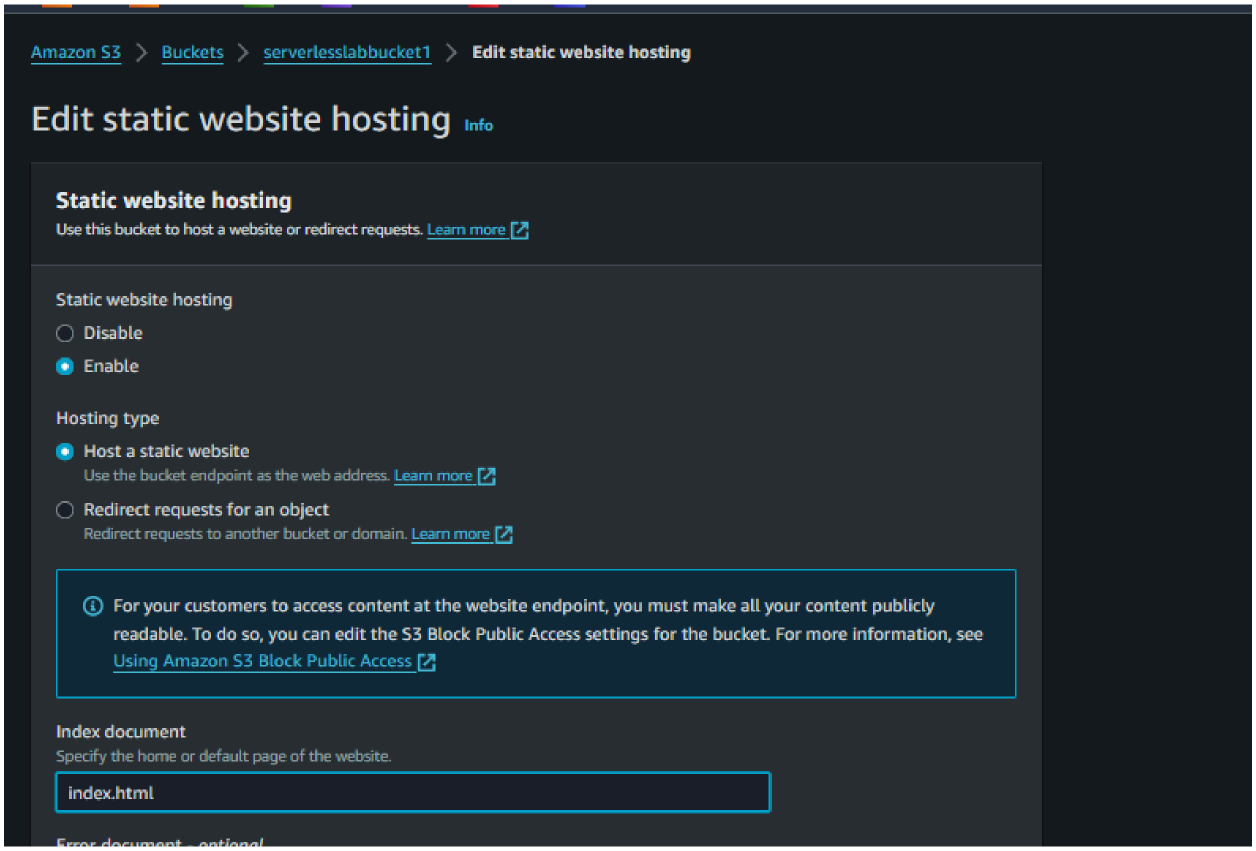
Name	Folder
index.html	-

Destination Info

Destination

s3://serverlesslabbucket1

Destination details


Amazon S3 > Buckets > serverlesslabbucket1 > 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 readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

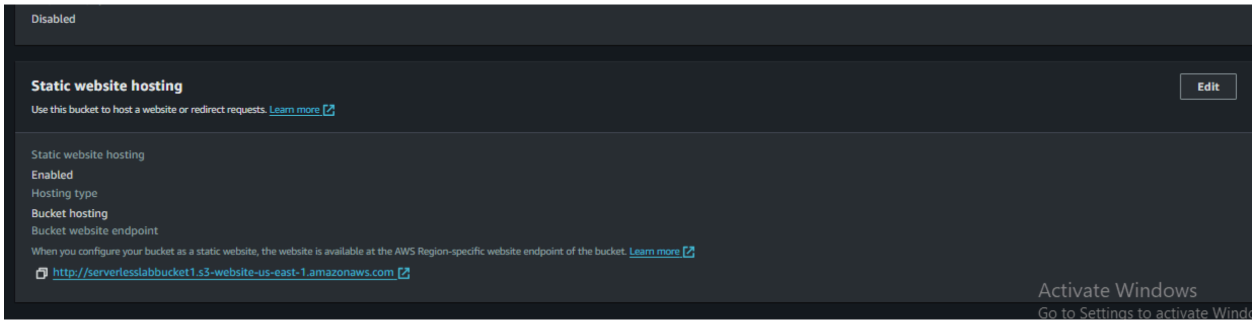
Index document

Specify the home or default page of the website.

index.html

Error document - *optional*

12. Find the link and click to check if working

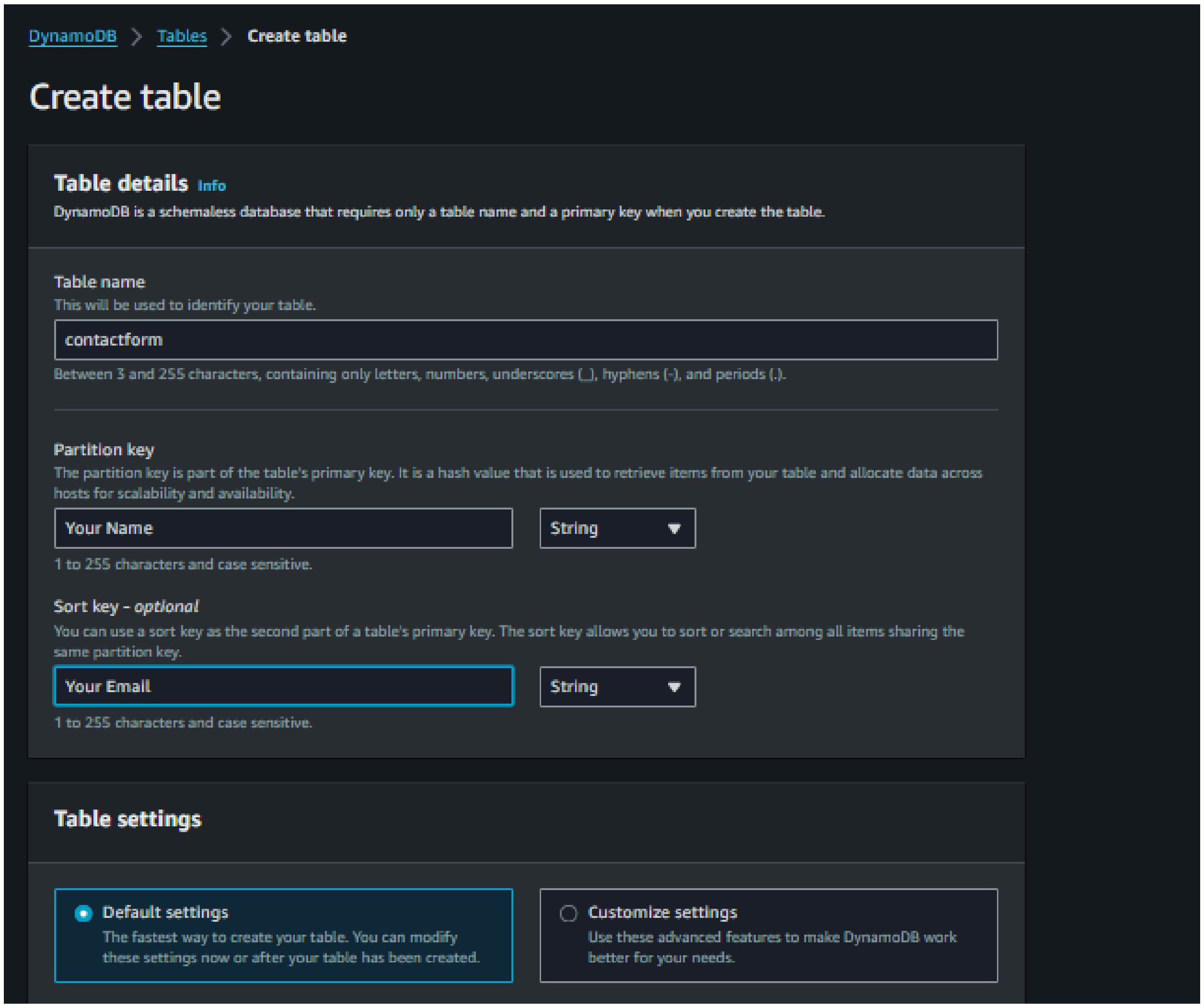

Disabled

Static website hosting

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

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://serverlesslabbucket1.s3-website-us-east-1.amazonaws.com

13. Go to DynamoDb to create tables



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.

contactform

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 availability.

Your Name | 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.

Your Email | String

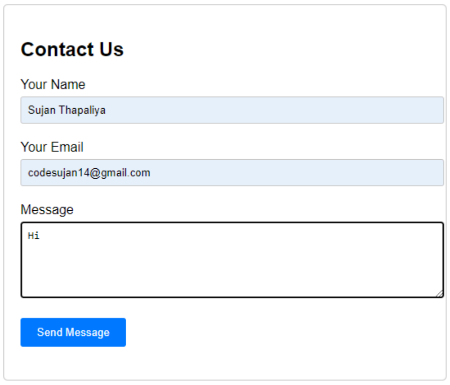
1 to 255 characters and case sensitive.

Table settings

[✓ Default settings] The fastest way to create your table. You can modify these settings now or after your table has been created.

Customize settings: Use these advanced features to make DynamoDB work better for your needs.

14. Write code to receive data in the table in lambda function,Check if data is updated in table



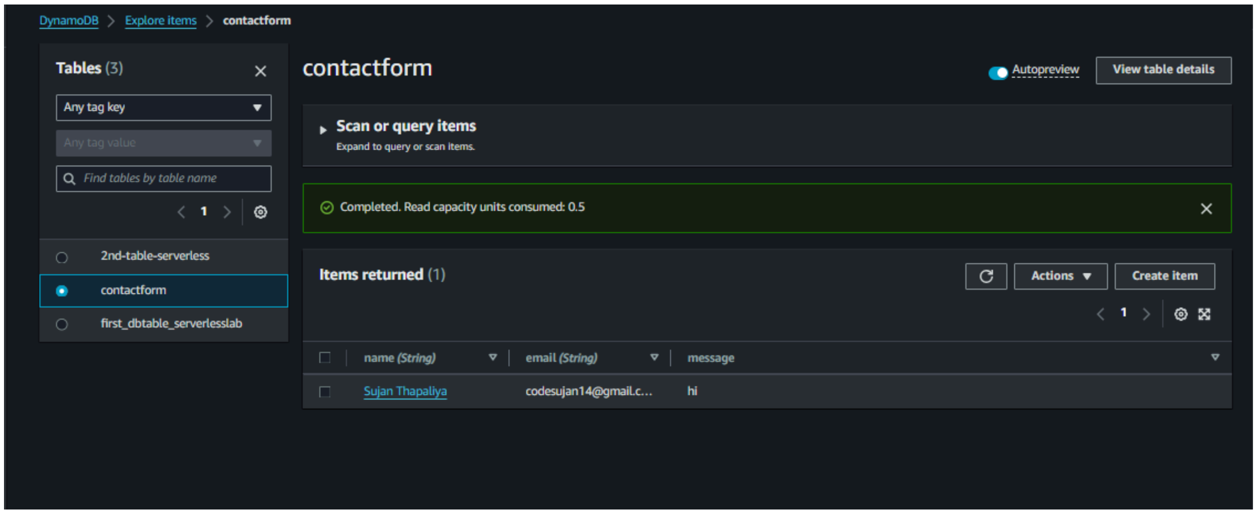
Contact Us

Your Name: Sujan Thapaliya

Your Email: codesujan14@gmail.com

Message: Hi

15. frontend interacted with the backend. Data is saved to the table



Completed. Read capacity units consumed: 0.5

name (String)	email (String)	message
Sujan Thapaliya	codesujan14@gmail.c...	hi