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.

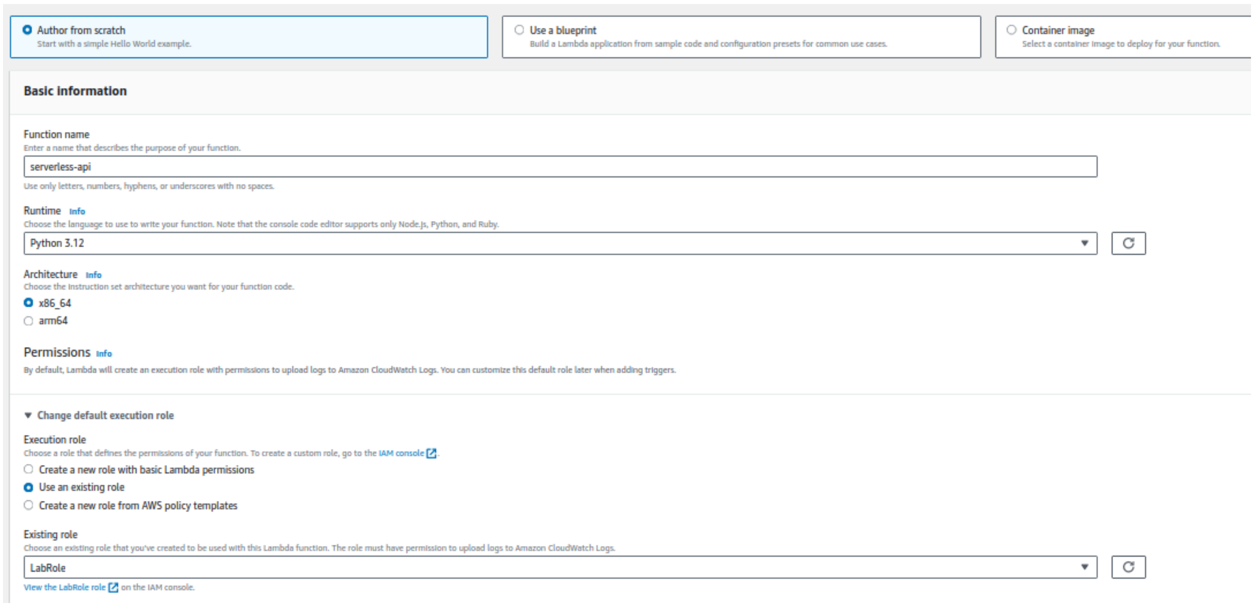
• API Gateway: Set up API Gateway to create RESTful endpoints that trigger the Lambda functions.

• 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.

- Create Lambda function with labRole as role



[✓ 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
Enter a name that describes the purpose of your function.
serverless-api
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 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64] / arm64

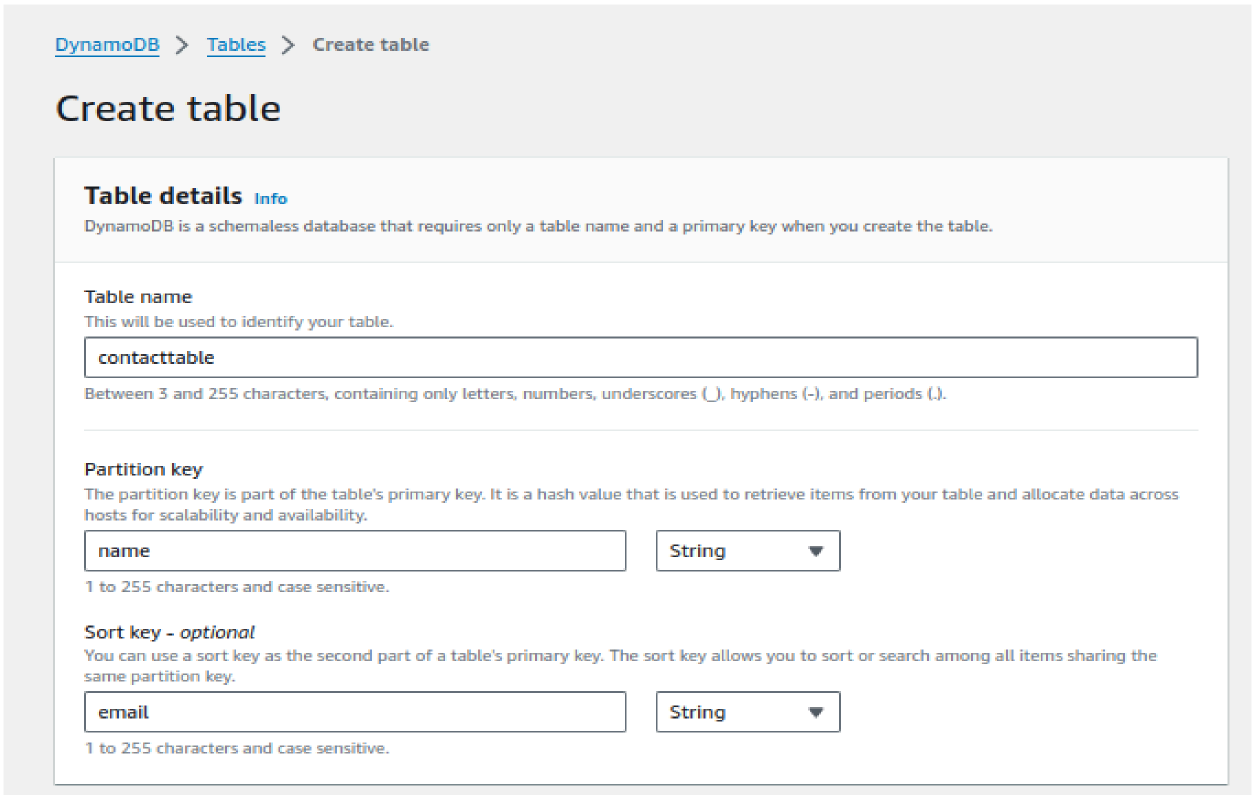
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

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
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.
LabRole
View the LabRole role on the IAM console.

- Create Dynamo DB table



DynamoDB > Tables > Create table

Create table

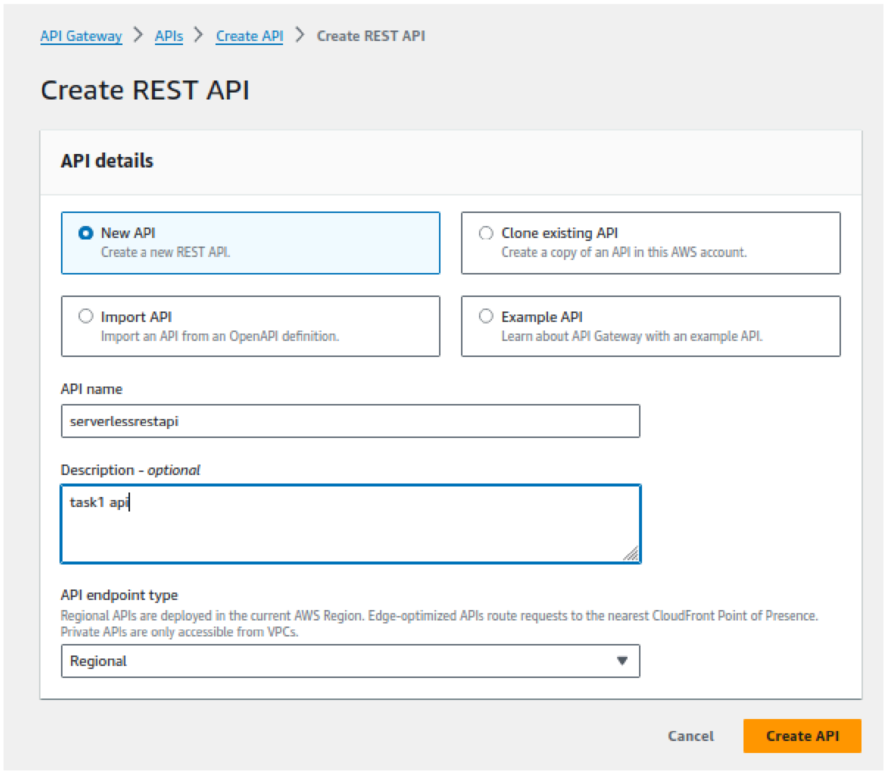
Table details Info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
contacttable
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.
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.
email — String
1 to 255 characters and case sensitive.

- Create 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.

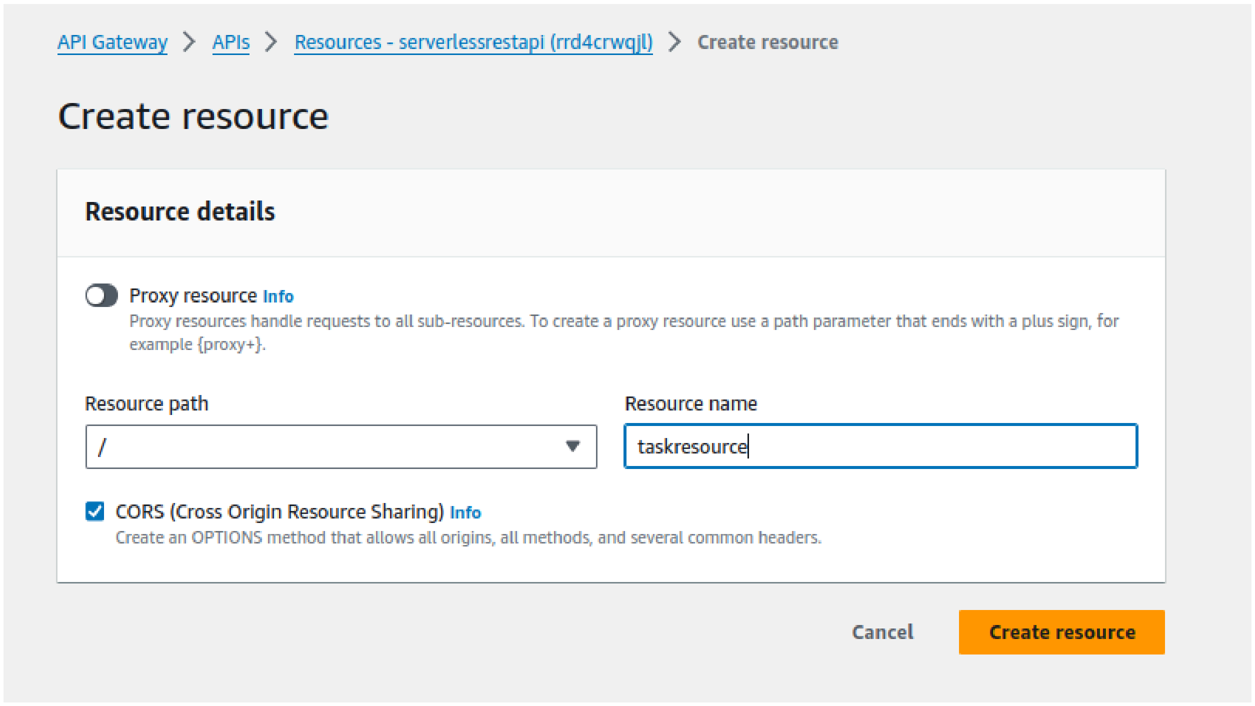
API name: serverlessrestapi

Description - *optional*: task1 api

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

- Create resource



API Gateway > APIs > Resources - serverlessrestapi (rrd4crwqjl) > Create resource

Create resource

Resource details

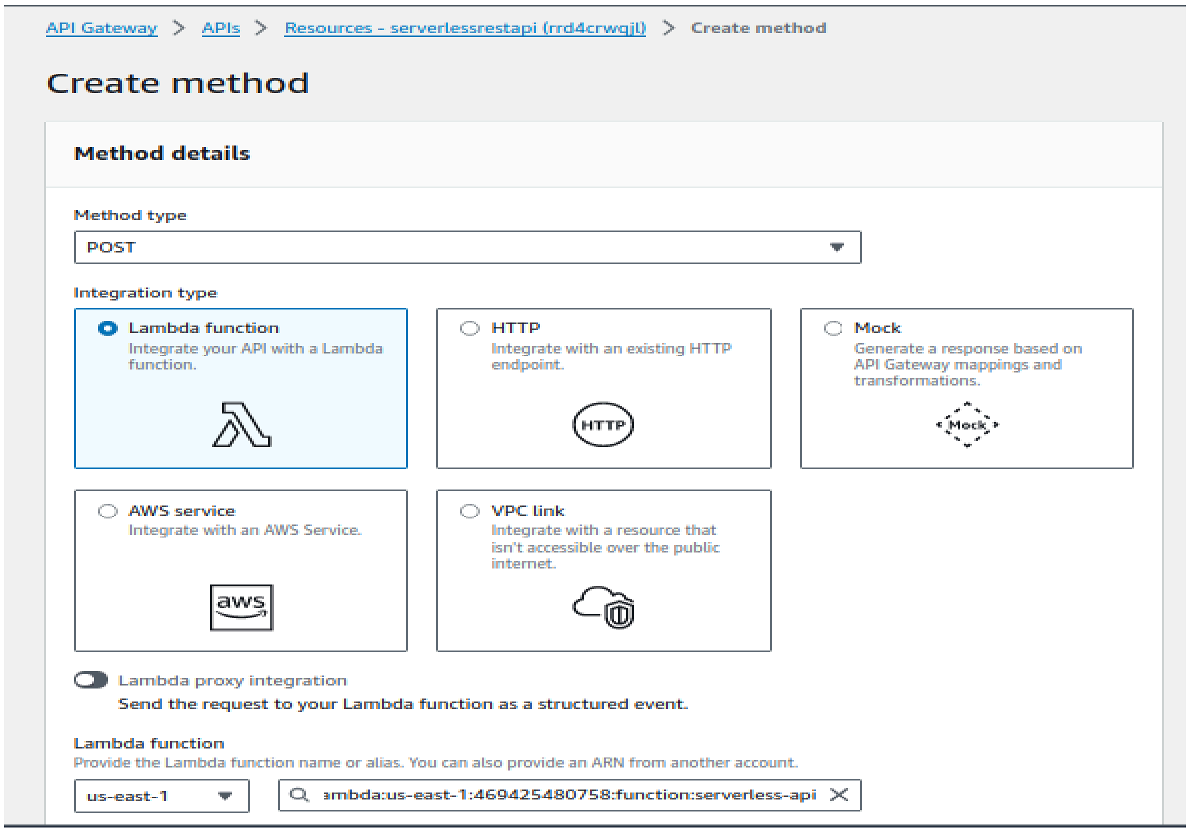
Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path: /

Resource name: taskresource

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

- Create POST method



API Gateway > APIs > Resources - serverlessrestapi (rrd4crwqjl) > Create method

Create method

Method details

Method type: POST

Integration type

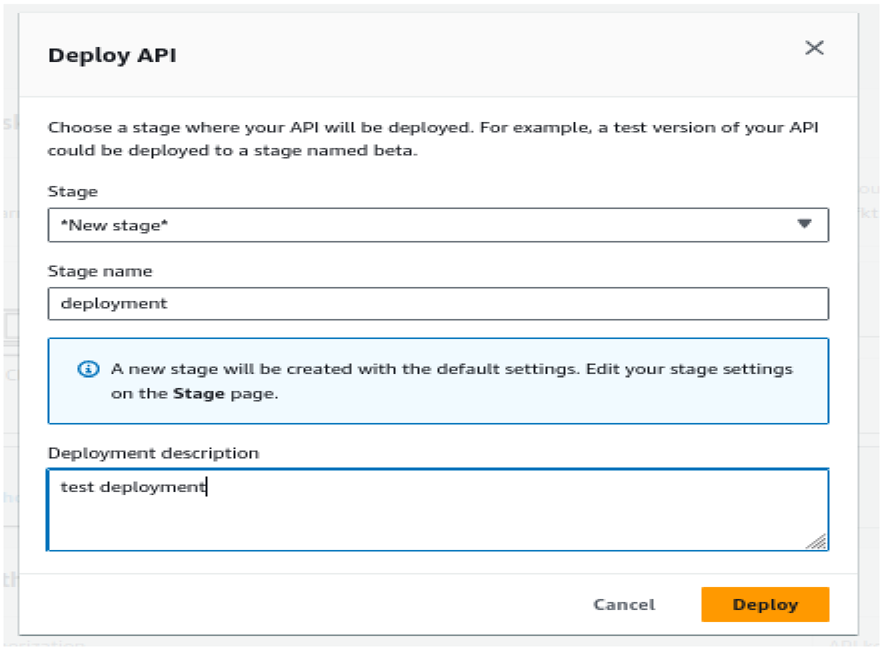
- [✓ Lambda function] Integrate your API with a Lambda function.
- HTTP: Integrate with an existing HTTP endpoint.
- Mock: Generate a response based on API Gateway mappings and transformations.
- AWS service: Integrate with an AWS Service.
- VPC link: Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration
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 | ambda:us-east-1:469425480758:function:serverless-api

- Deploy api with new stage as deployment



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.

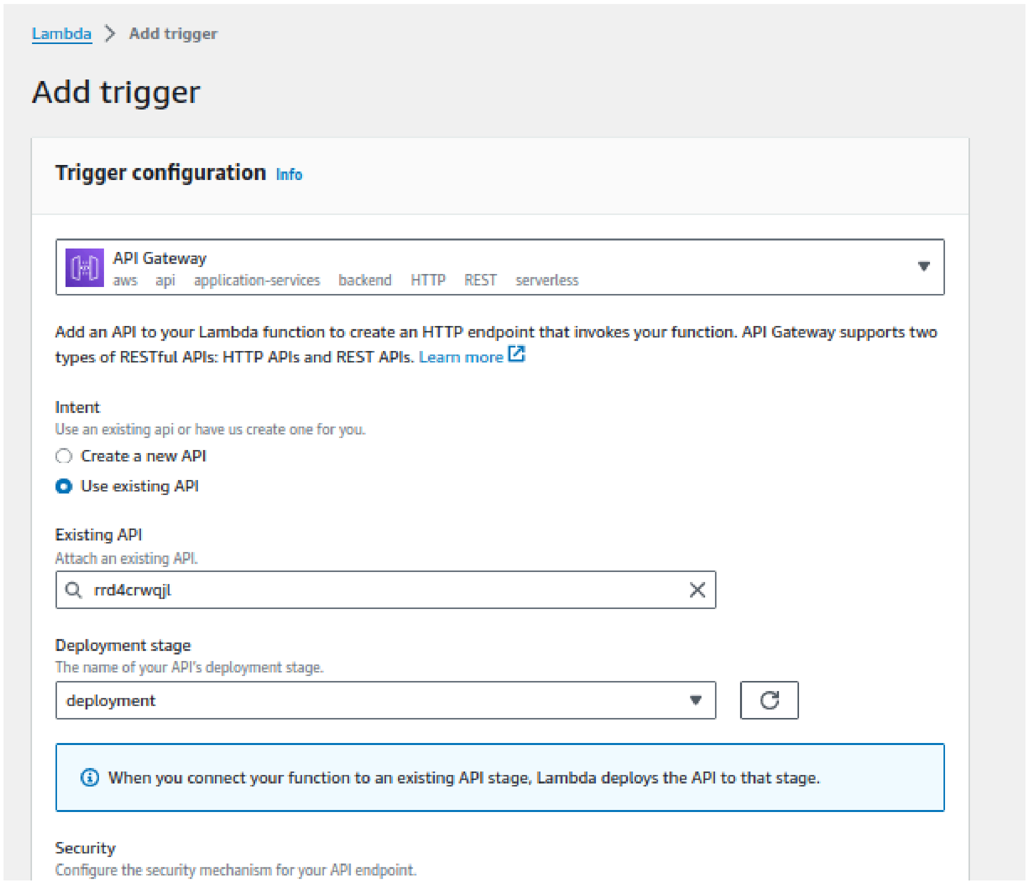
Stage: *New stage*

Stage name: deployment

A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description: test deployment

- Add trigger(API gateway)



Lambda > Add trigger

Add trigger

Trigger configuration Info

API Gateway
aws api application-services backend HTTP REST serverless

Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. Learn more

Intent
Use an existing api or have us create one for you.
○ Create a new API
● Use existing API

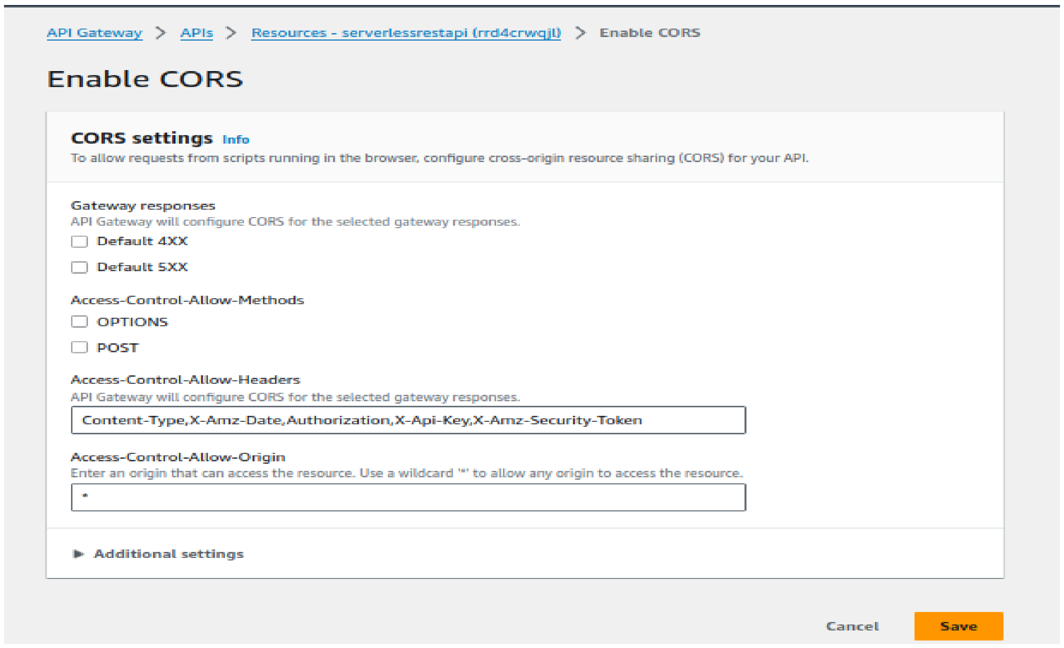
Existing API
Attach an existing API.
rrd4crwqjl

Deployment stage
The name of your API's deployment stage.
deployment

When you connect your function to an existing API stage, Lambda deploys the API to that stage.

Security
Configure the security mechanism for your API endpoint.

- Enable CORS



API Gateway > APIs > Resources - serverlessrestapi (rrd4crwqjl) > Enable CORS

Enable CORS

CORS settings Info
To allow requests from scripts running in the browser, configure cross-origin resource sharing (CORS) for your API.

Gateway responses
API Gateway will configure CORS for the selected gateway responses.
☐ Default 4XX
☐ Default 5XX

Access-Control-Allow-Methods
☐ OPTIONS
☐ POST

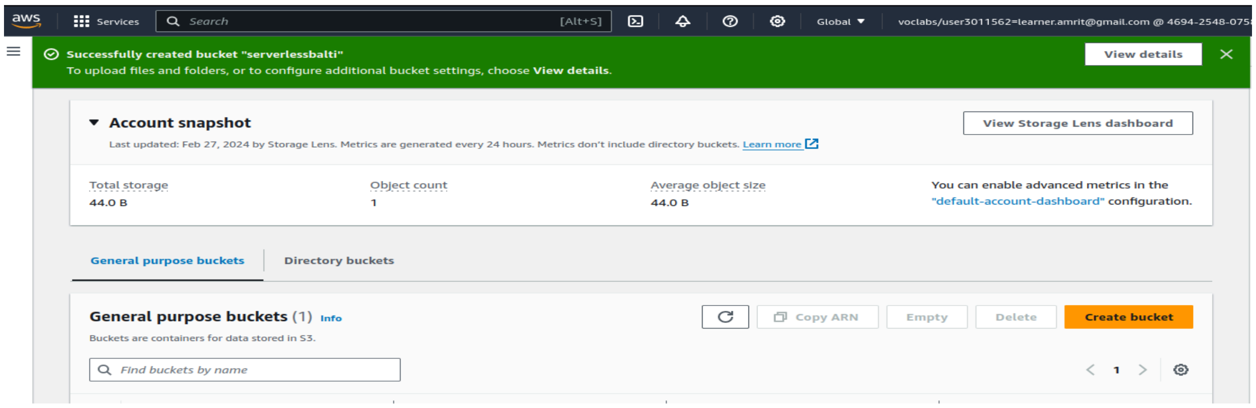
Access-Control-Allow-Headers
API Gateway will configure CORS for the selected gateway responses.
Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token

Access-Control-Allow-Origin
Enter an origin that can access the resource. Use a wildcard '*' to allow any origin to access the resource.
*

▸ Additional settings

Cancel Save

- Create S3 Bucket



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

Account snapshot

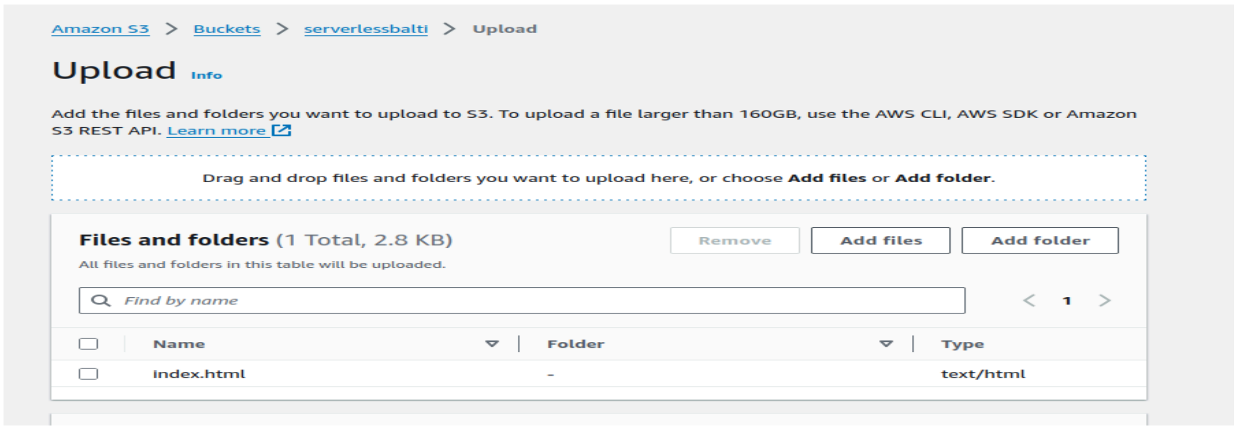
Last updated: Feb 27, 2024 by Storage Lens. Metrics are generated every 24 hours. Metrics don't include directory buckets. Learn more

Total storage	Object count	Average object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
44.0 B	1	44.0 B	

General purpose buckets (1) Info

Buckets are containers for data stored in S3.

- Upload HTML file



Amazon S3 > Buckets > serverlessbalti > 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

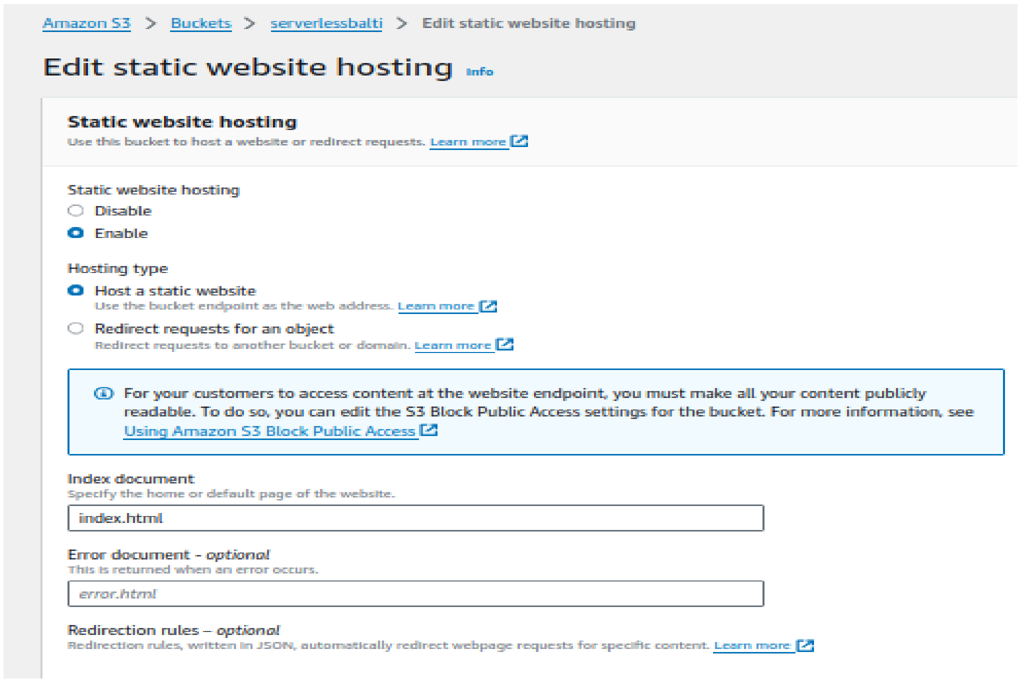
Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 2.8 KB)

All files and folders in this table will be uploaded.

Name	Folder	Type
index.html	-	text/html

- Enable Static Hosting



Amazon S3 > Buckets > serverlessbalti > 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

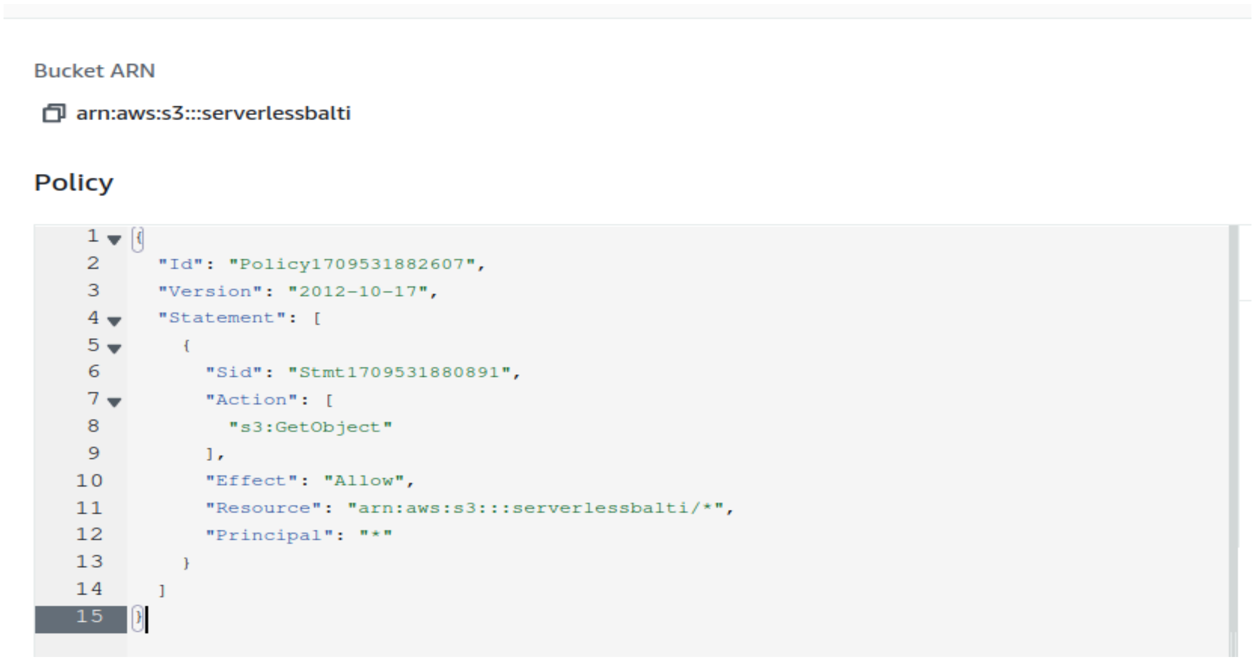
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

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

Error document - *optional*
This is returned when an error occurs.
error.html

Redirection rules – *optional*
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

- Update Bucket Policy



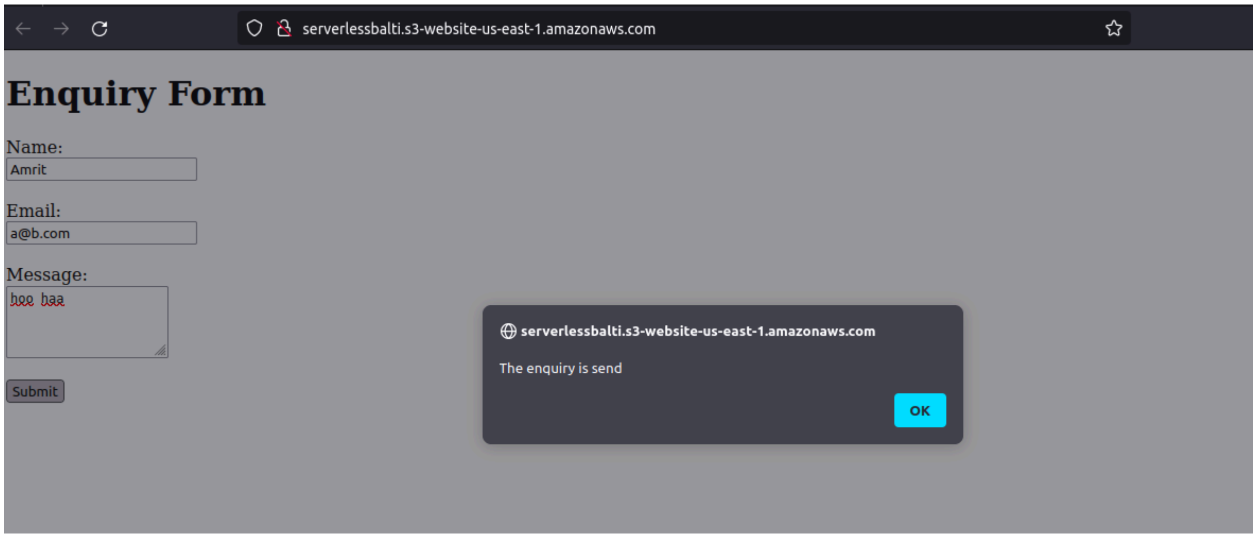
Bucket ARN

arn:aws:s3:::serverlessbalti

Policy

```
{
  "Id": "Policy1709531882607",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1709531880891",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::serverlessbalti/*",
      "Principal": "*"
    }
  ]
}
```

- Access using Static web hosting URL



serverlessbalti.s3-website-us-east-1.amazonaws.com

Enquiry Form

Name: Amrit

Email: a@b.com

Message: hoo haa

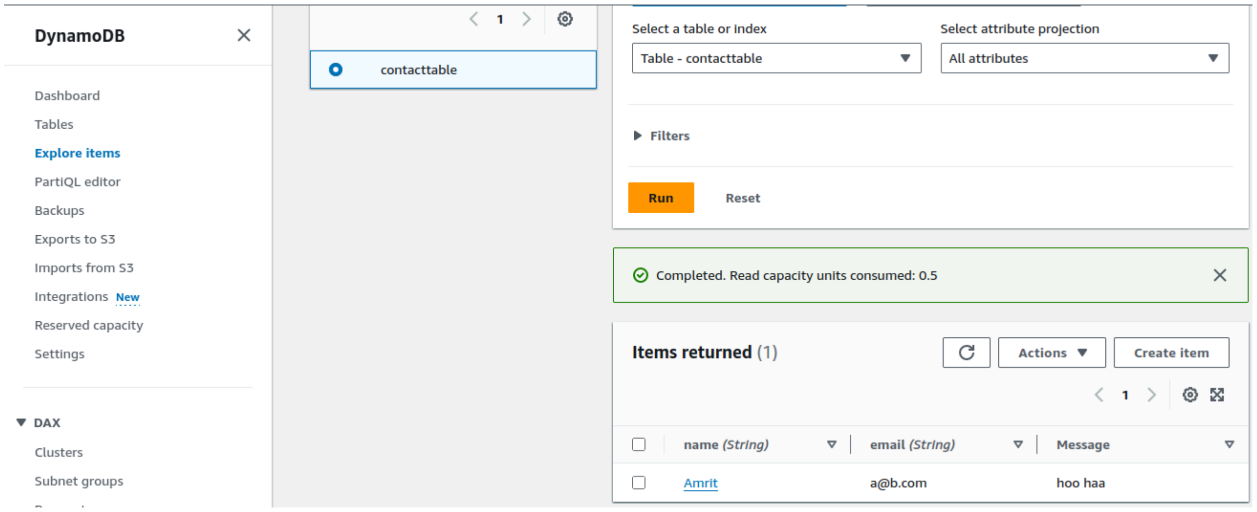
Submit

serverlessbalti.s3-website-us-east-1.amazonaws.com

The enquiry is send

OK

- Access item in DynamoDB table



Select a table or index: Table - contacttable

Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (1)

name (String)	email (String)	Message
Amrit	a@b.com	hoo haa