

# 1. Building a Serverless Web Application

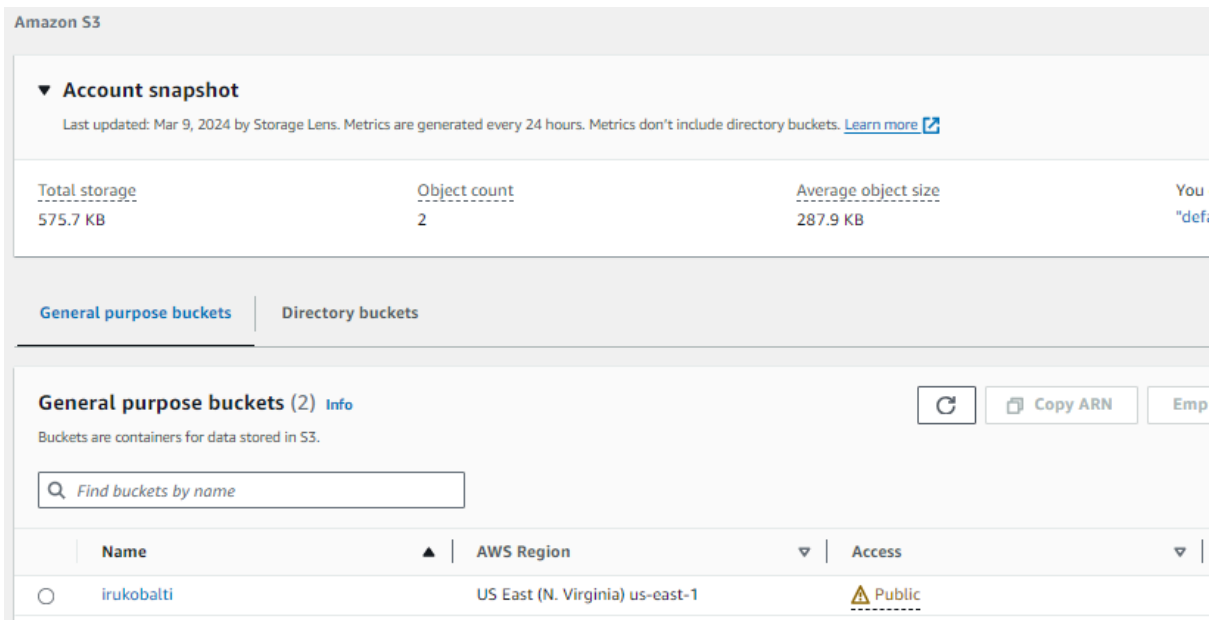
**Objective:** Create a serverless web application using AWS Lambda, API Gateway, S3, and DynamoDB.

**Approach:**

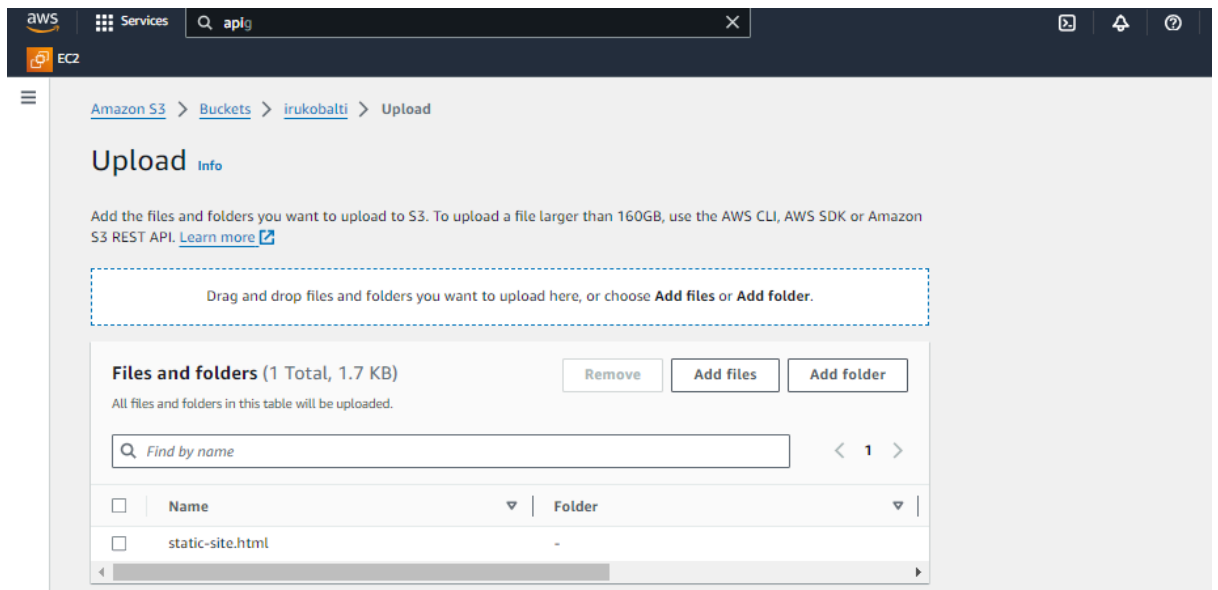
- **Set Up Backend:** Create Lambda functions to handle backend logic. These functions will interact with a DynamoDB table for data storage.
- **API Gateway:** Set up API Gateway to create RESTful endpoints that trigger the Lambda functions.
- **Frontend Hosting:** Host a static website on S3 that interacts 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.

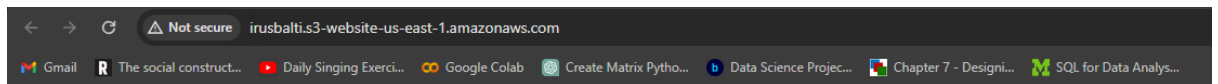
Start by creating an S3 bucket



Upload static file created in the bucket



Our static site looks like this



## Contact Form

Name:

Email:

Message:

Now a DynamoDB table is created

The screenshot shows the AWS Management Console interface for creating a new DynamoDB table. The breadcrumb navigation at the top indicates the path: **DynamoDB** > **Tables** > **Create table**. The main heading is **Create table**. Below this, there is a section titled **Table details** with an **Info** link. A descriptive text states: "DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table." The form contains three main sections: 1. **Table name**: A text input field containing "test-table". A note below the field states: "This will be used to identify your table." and "Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)". 2. **Partition key**: A text input field containing "name" and a dropdown menu set to "String". A note states: "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." and "1 to 255 characters and case sensitive." 3. **Sort key - optional**: A text input field with the placeholder "Enter the sort key name" and a dropdown menu set to "String". A note states: "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." The footer of the console shows "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates.

Then a lambda function is created and python code is written and deployed in it a trigger is also added.

The screenshot shows the AWS Management Console interface for creating a new Lambda function. The breadcrumb navigation at the top indicates the path: **EC2** > **Create function**. The main heading is **Create function** with an **Info** link. Below the heading, a prompt says: "Choose one of the following options to create your function." There are three radio button options: 1. **Author from scratch** (selected): "Start with a simple Hello World example." 2. **Use a blueprint**: "Build a Lambda application from sample code and configuration presets for common use cases." 3. **Container image**: "Select a container image to deploy for your function." Below these options is the **Basic information** section. It contains: 1. **Function name**: A text input field containing "test-function". A note states: "Enter a name that describes the purpose of your function." and "Use only letters, numbers, hyphens, or underscores with no spaces." 2. **Runtime**: A dropdown menu set to "Python 3.12". A note states: "Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby." 3. **Architecture**: A dropdown menu set to "x86\_64". A note states: "Choose the instruction set architecture you want for your function code." On the right side of the console, there is a **Tutorials** panel. It has tabs for **Info** and **Tutorials**. Under the **Tutorials** tab, it says: "Learn how to implement common use cases in AWS Lambda." Below this, there is a section titled **Create a simple web app** with a sub-heading "In this tutorial you will learn how to:". It lists two bullet points: "• Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage" and "• Invoke your function through its function URL". There is a **Learn more** link and a **Start tutorial** button.

Create Rest API, then create a Resource, then a method and stage within it and finally deploy it

The screenshot shows the 'Create REST API' page in the AWS Management Console. The page has a dark header with the AWS logo, 'Services' menu, a search bar, and user information. The main content area is titled 'Create REST API' and contains a form for creating a new API. The 'API details' section has four radio button options: 'New API' (selected), 'Clone existing API', 'Import API', and 'Example API'. Below these options are text input fields for 'API name' (containing 'static-api') and 'Description - optional'. At the bottom, there is a section for 'API endpoint type' with explanatory text. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc.

aws Services Search [Alt+S] N. Virginia voclabs/user3011687~iruniroula@gmail.com @ 1716-3552-5037

### 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  
static-api

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.  
Deliver to API is only accessible from VPCs.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Create resource' page in the AWS Management Console. A green banner at the top indicates 'Successfully created REST API 'static-api (spn3hxum4m)''. The breadcrumb navigation shows 'API Gateway > APIs > Resources - static-api (spn3hxum4m) > Create resource'. The main content area is titled 'Create resource' and contains a form for creating a new resource. The 'Resource details' section has two radio button options: 'Proxy resource' (selected) and 'CORS (Cross Origin Resource Sharing)'. Below these options are text input fields for 'Resource path' (containing '/') and 'Resource name' (containing '{my-resource+}'). At the bottom, there is a 'Cancel' button and a 'Create resource' button. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc.

aws Services Search [Alt+S] N. Virginia voclabs/user3011687~iruniroula@gmail.com @ 1716-3552-5037

Successfully created REST API 'static-api (spn3hxum4m)'

API Gateway > APIs > Resources - static-api (spn3hxum4m) > 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  
{my-resource+}

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

Cancel Create resource

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Resource details

Delete

Update documentation

Enable CORS

Path

`/my-resource+`

Resource ID

b07gn1

Methods (2)

Delete

Create method

	Method type ▲	Integration type ▼	Authorization ▼	API key ▼
<input type="radio"/>	<a href="#">ANY</a>	Not setup	None	Not required
<input type="radio"/>	<a href="#">OPTIONS</a>	Mock	None	Not required

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3011687=iuniroula@gmail.com @ 1716-3552-5037

EC2

POST

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 function

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

us-east-1

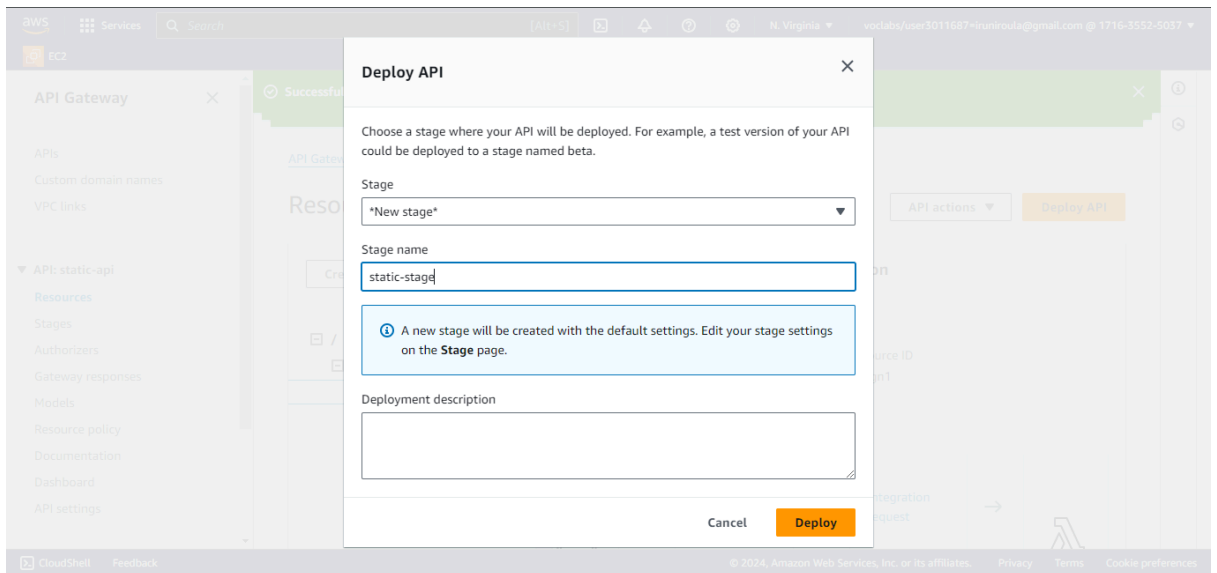
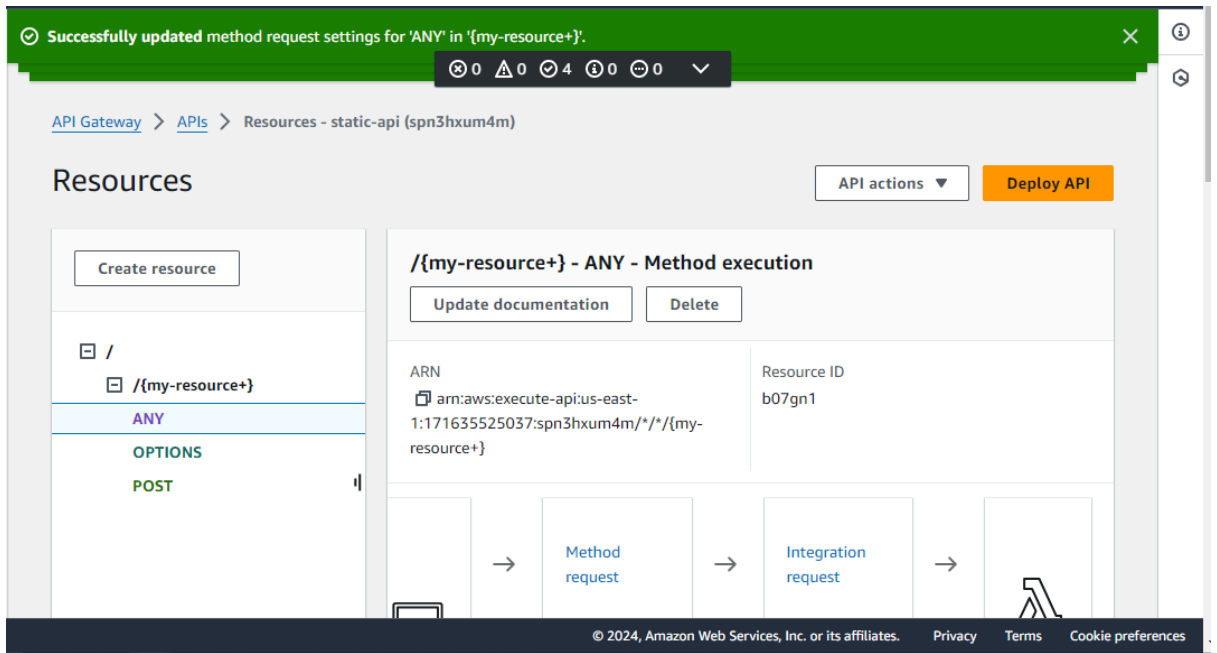
arn:aws:lambda:us-east-1:171635525037:function:test-

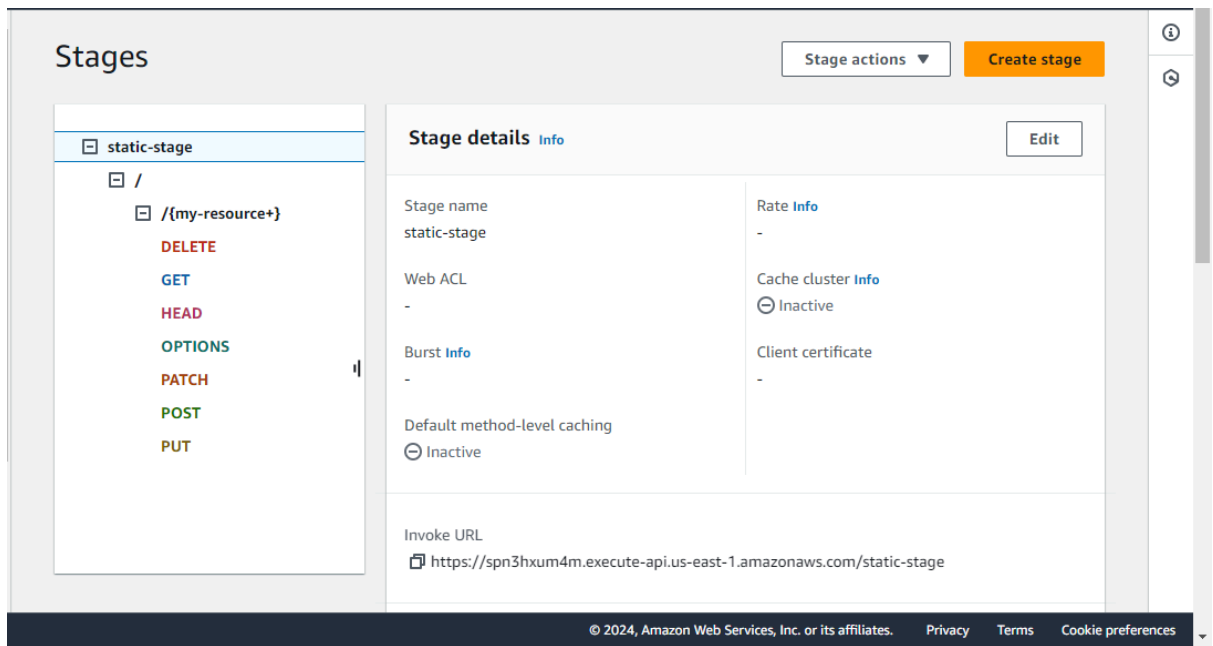
Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource

CloudShell

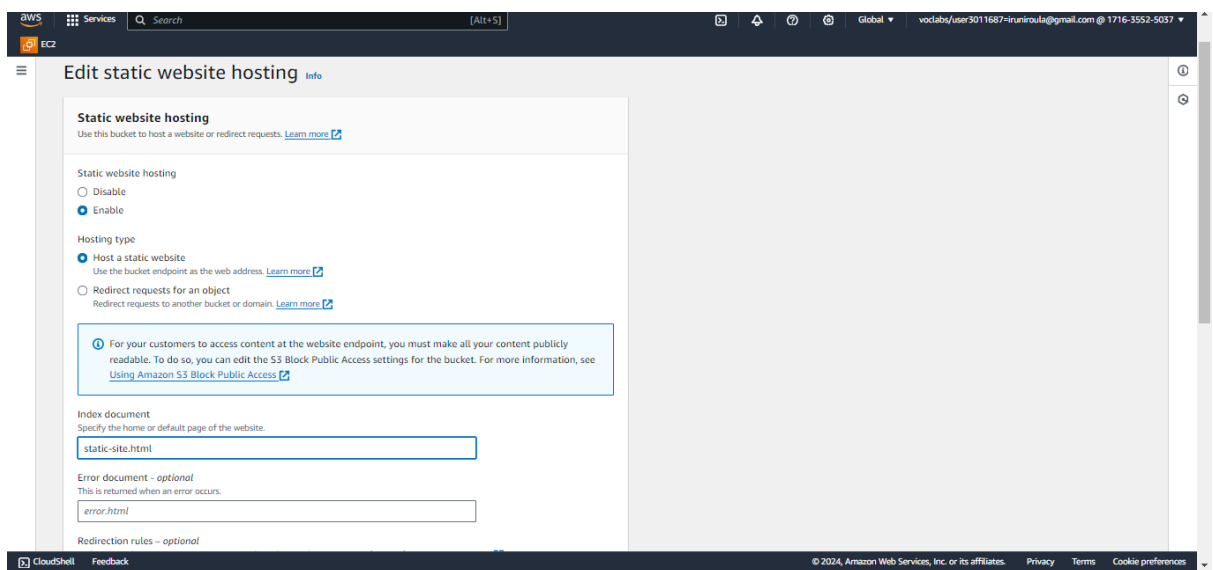
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences





Edit necessary policies within the bucket to allow static hosting



## Policy

```
1 {  
2   "Version": "2012-10-17",  
3   "Id": "Policy1709791241957",  
4   "Statement": [  
5     {  
6       "Sid": "Stmnt1709791239327",  
7       "Effect": "Allow",  
8       "Principal": "*",  
9       "Action": "s3:GetObject",  
10      "Resource": "arn:aws:s3:::irusbalti/*"  
11    }  
12  ]  
13 }
```

When information is filled in this contact form,

← → ↻ ⚠ Not secure irusbalti.s3-website-us-east-1.amazonaws.com

Gmail

The social construct...

Daily Singing Exerci...

Google Colab

Create Matrix Pytho...

Data Science Projec...

Chapter 7 - Designi...

SQL for Data Analys...

## Contact Form

Name:

Email:

Message:



It is now recorded in the DynamoDB as follows:

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3011687=iruniroula@gmail.com @ 1716-3552-5037

EC2

DynamoDB

test-table

Table - test-table

All attributes

Filters

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (2)

name (String)

email

message

lone tree

iru.lonetree...

feri

Iru Niroula

iru.lonetree...

la sakyo

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences