## 1. Building a Serverless Web Application

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

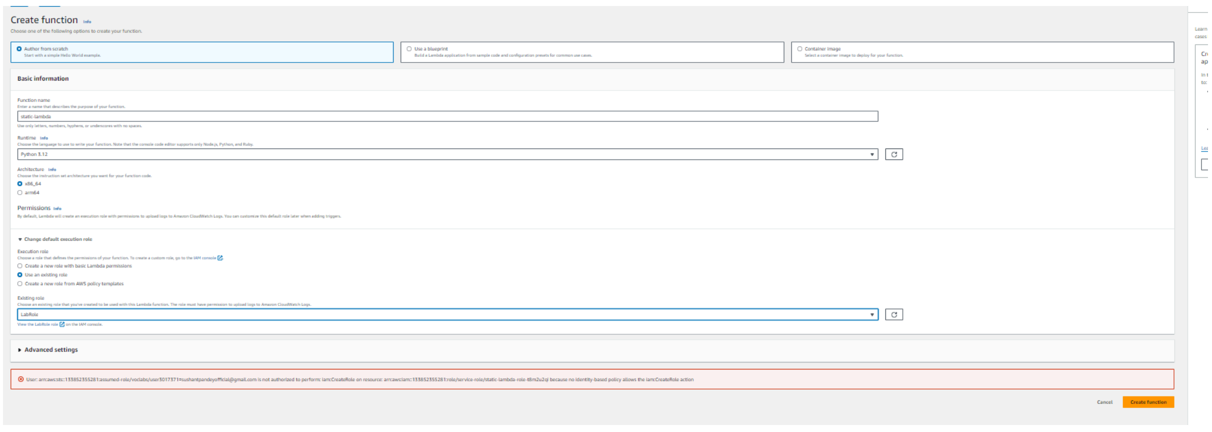
**Approach**:

- **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.

Steps:

1. First create a lambda function: static-lambda function was created, role was assigned to lab role



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
Enter a name that describes the purpose of your function.
static-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 3.12

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

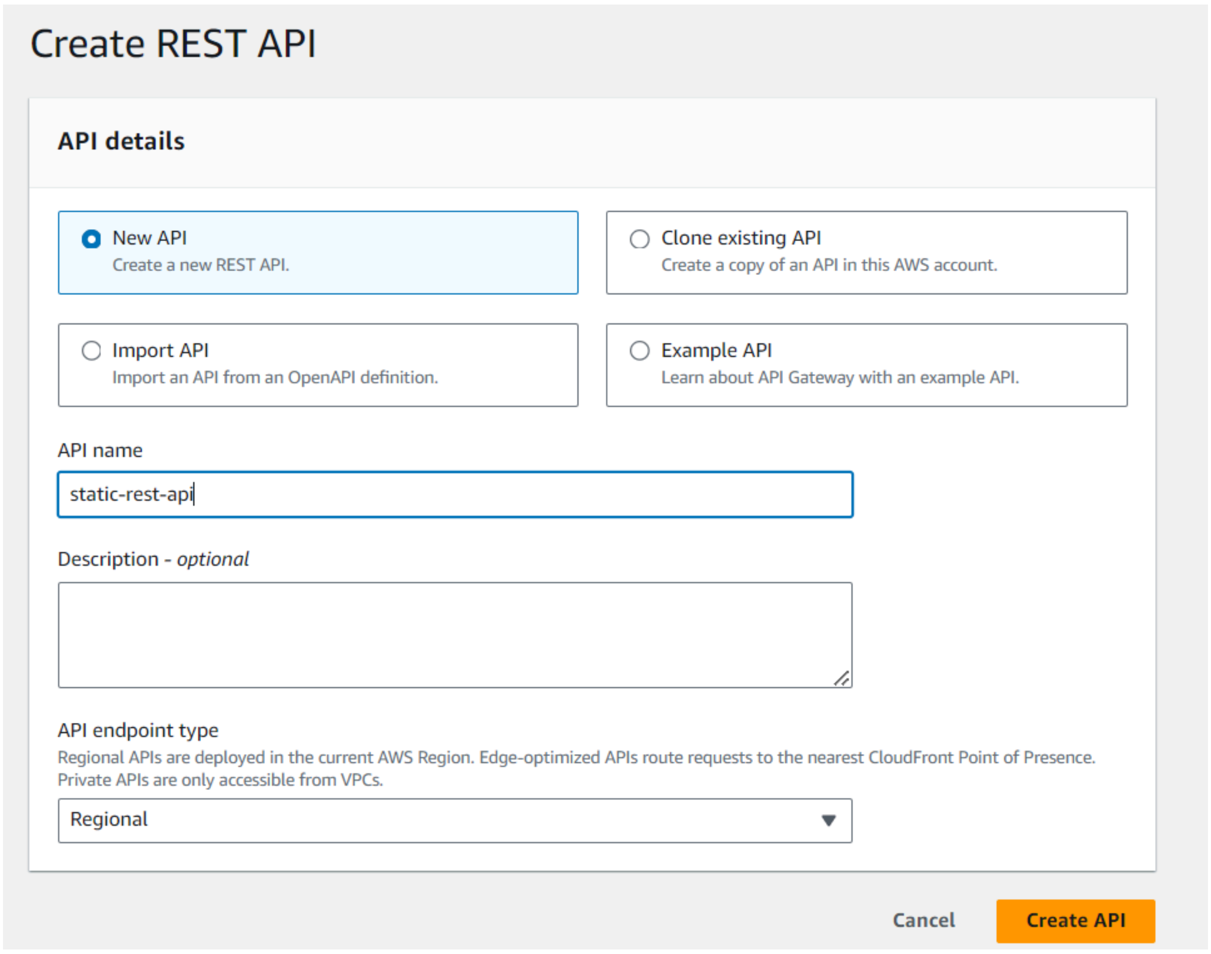
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
[✓ Use an existing role]
Create a new role from AWS policy templates

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.

▸ Advanced settings

```
User: arn:aws:sts::133852355281:assumed-role/voclabs/user3017371=sushantpandeyofficial@gmail.com is not authorized to perform: iam:CreateRole on resource: arn:aws:iam::133852355281:role/service-role/static-lambda-role-t8m2u2qi because no identity-based policy allows the iam:CreateRole action
```

2. Created new REST Api: static-rest-api was created


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.

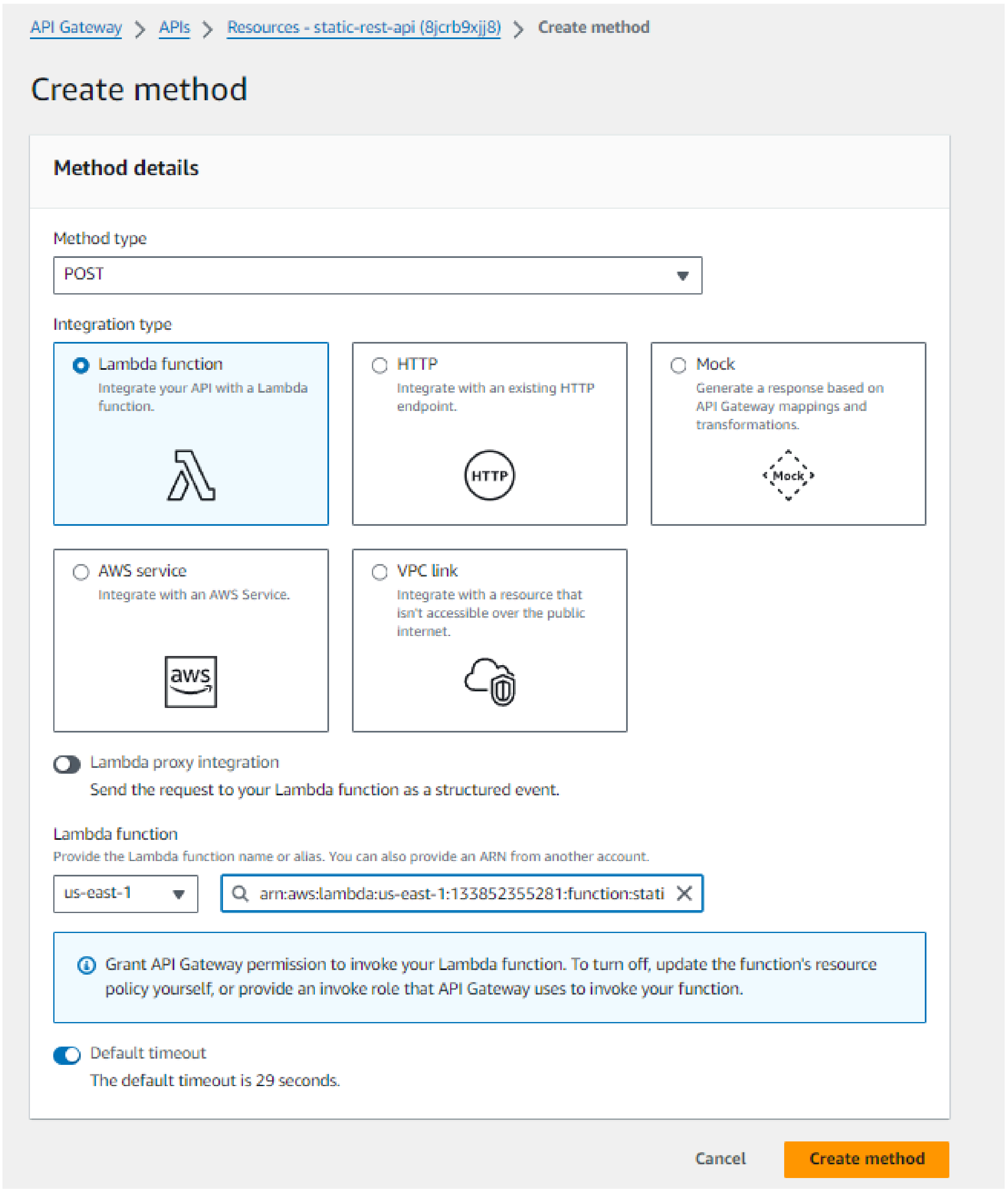
API name: static-rest-api

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. Private APIs are only accessible from VPCs.

Regional

3. POST Method was created within the previously created rest api, lambda function was assigned which was created in 1.


API Gateway > APIs > Resources - static-rest-api (8jcrb9xjj8) > 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 (off)
Send the request to your Lambda function as a structured event.

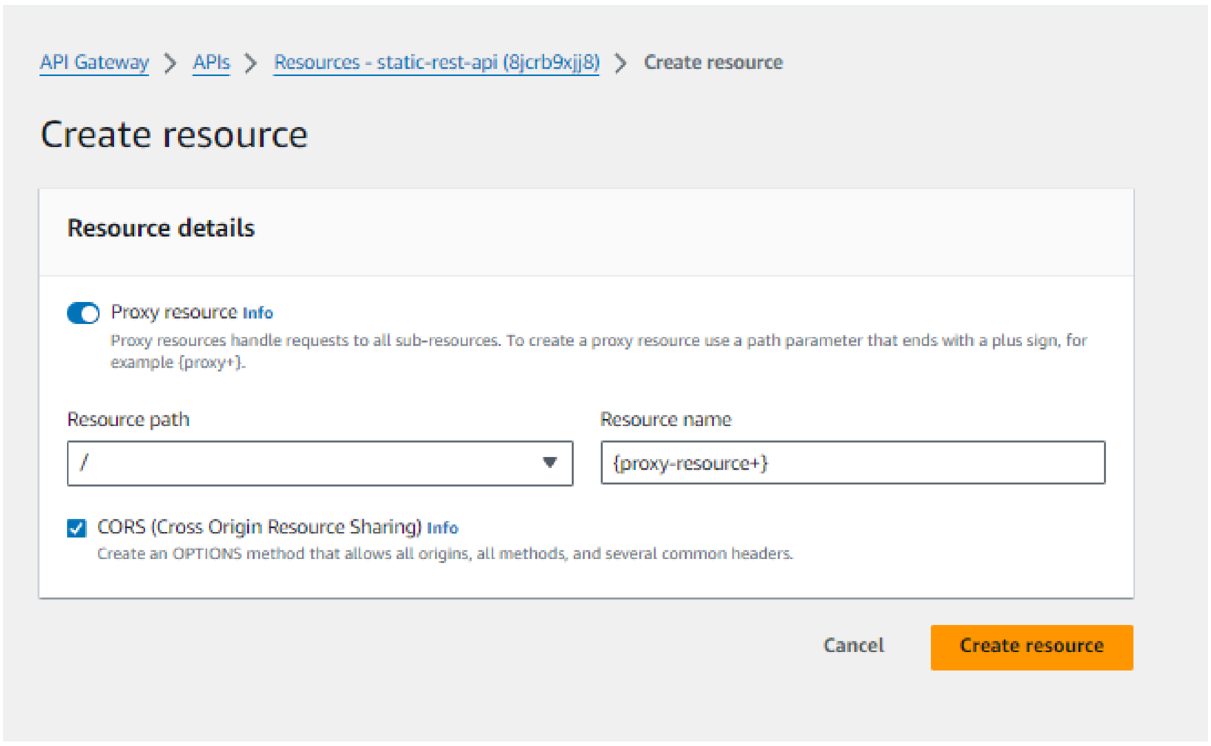
Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 | arn:aws:lambda:us-east-1:133852355281:function:stati

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

Default timeout (on)
The default timeout is 29 seconds.

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



API Gateway > APIs > Resources - static-rest-api (8jcrb9xjj8) > 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+}.

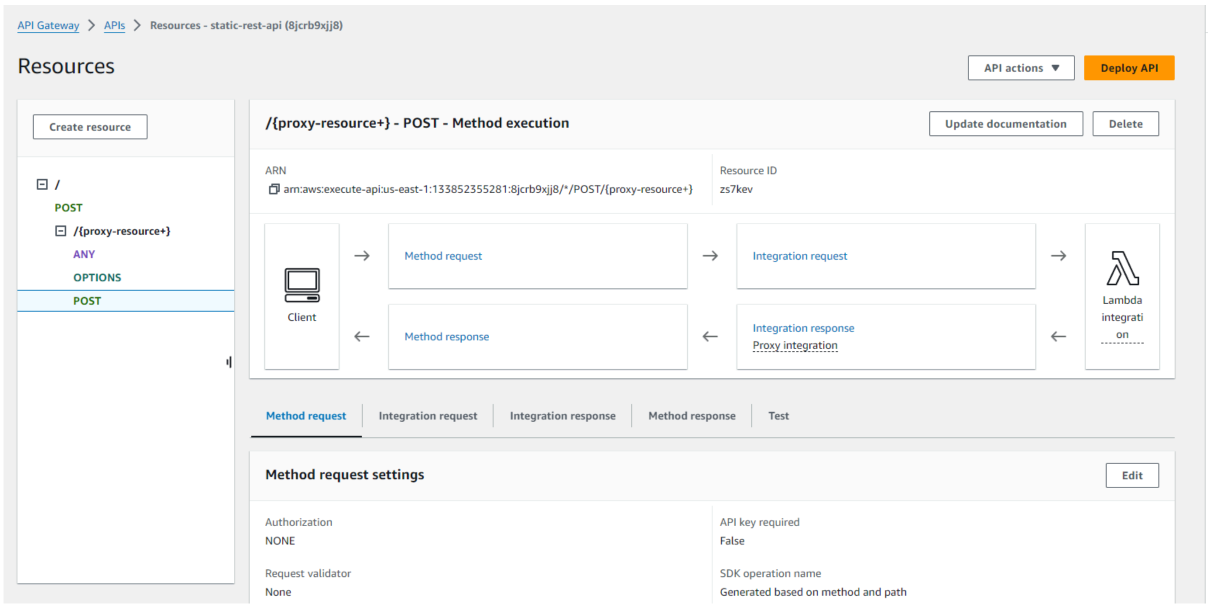
Resource path: /

Resource name: {proxy-resource+}

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

Cancel | Create resource

4. Within the resource tab, click integration request



API Gateway > APIs > Resources - static-rest-api (8jcrb9xjj8)

Resources

API actions | Deploy API

Create resource

/
POST
/{proxy-resource+}
ANY
OPTIONS
POST

/{proxy-resource+} - POST - Method execution

Update documentation | Delete

ARN
arn:aws:execute-api:us-east-1:133852355281:8jcrb9xjj8/*/POST/{proxy-resource+}

Resource ID
zs7kev

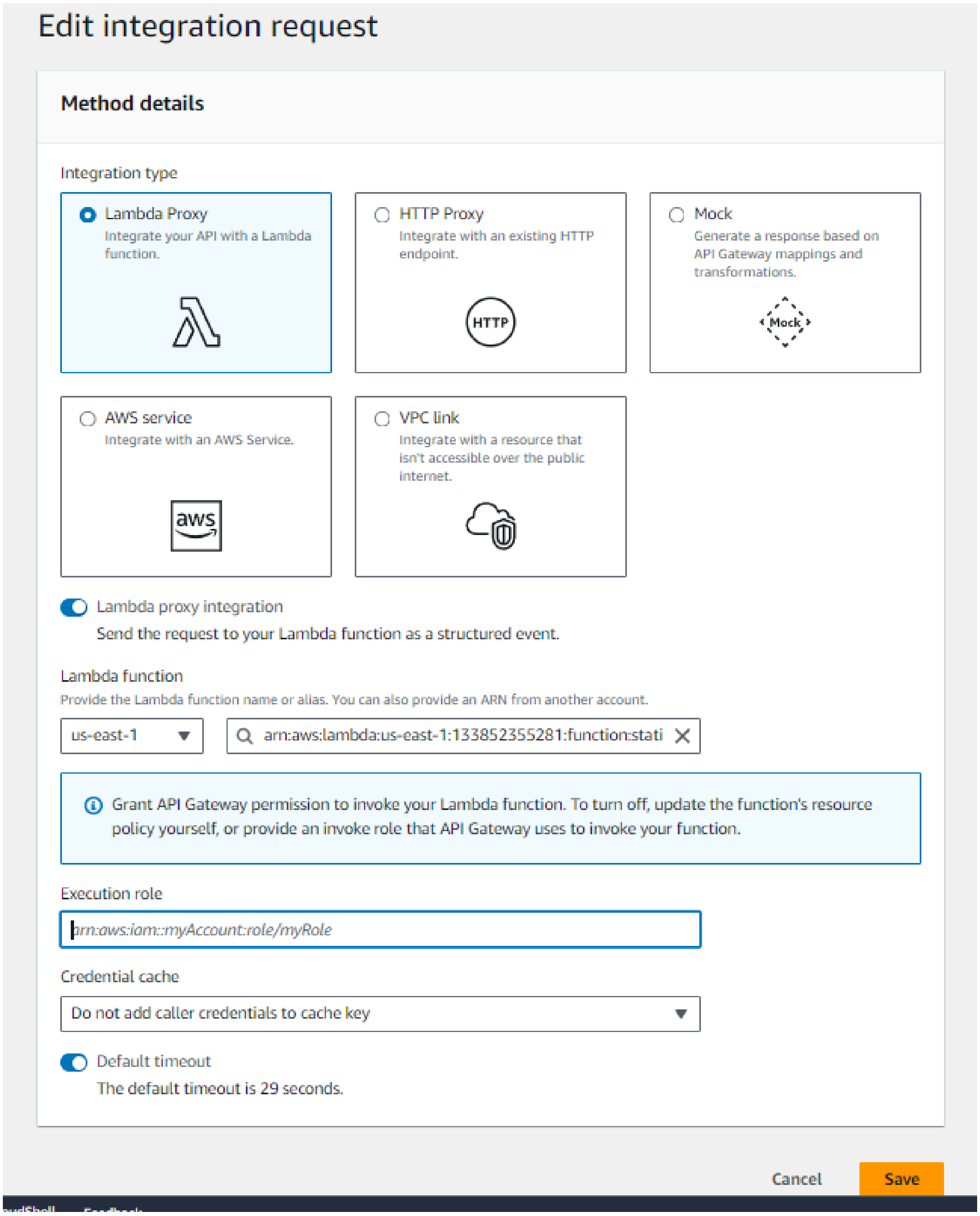
Method request | Integration request | Integration response | Method response | Test

Method request settings

Edit

| | |
|---|---|
| Authorization: NONE | API key required: False |
| Request validator: None | SDK operation name: Generated based on method and path |

5. Edit the integration request, enable the proxy



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-1:133852355281:function:stati

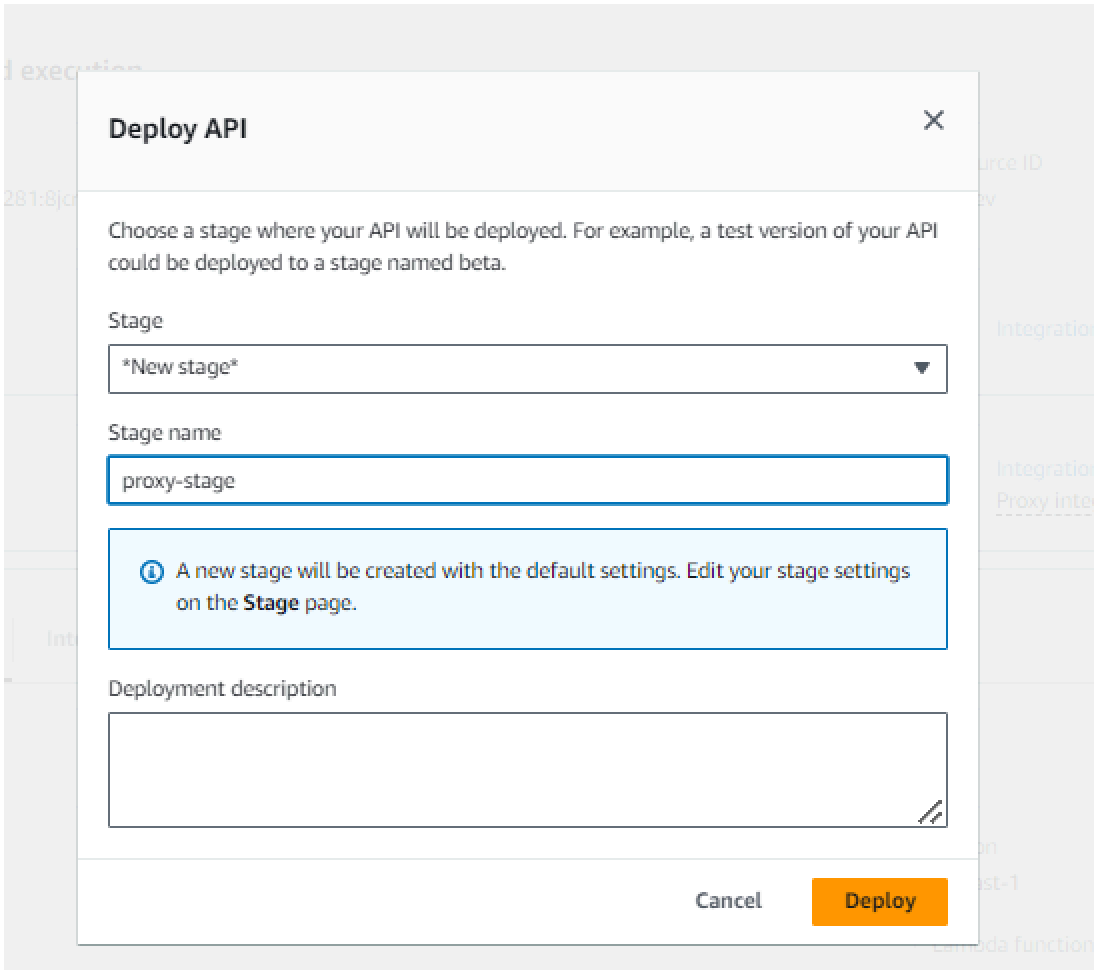
Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

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

Credential cache
Do not add caller credentials to cache key

Default timeout: on
The default timeout is 29 seconds.

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.

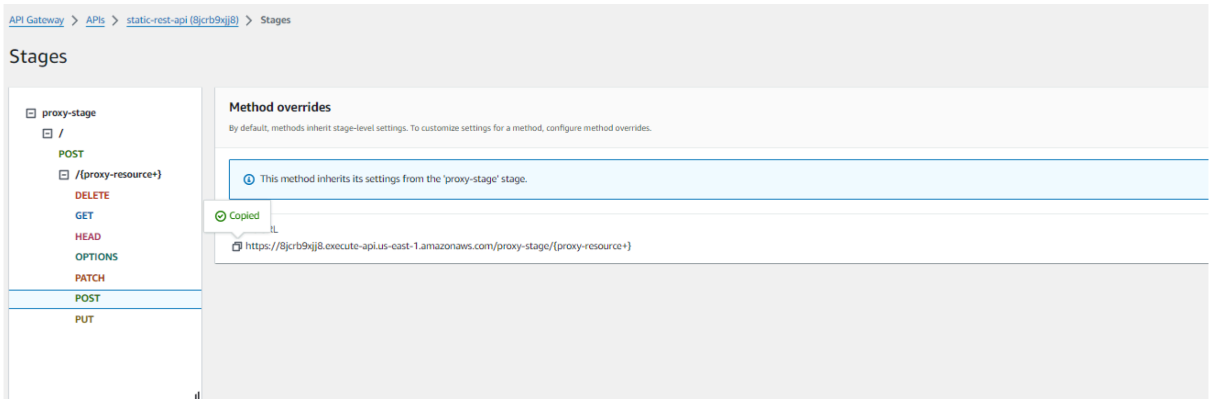
Stage: *New stage*

Stage name: proxy-stage

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

Deployment description:

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



API Gateway > APIs > static-rest-api (8jcrb9xjj8) > Stages

Stages

- proxy-stage
  - /
    - POST
    - /{proxy-resource+}
      - DELETE
      - GET
      - HEAD
      - OPTIONS
      - PATCH
      - POST
      - PUT

**Method overrides**

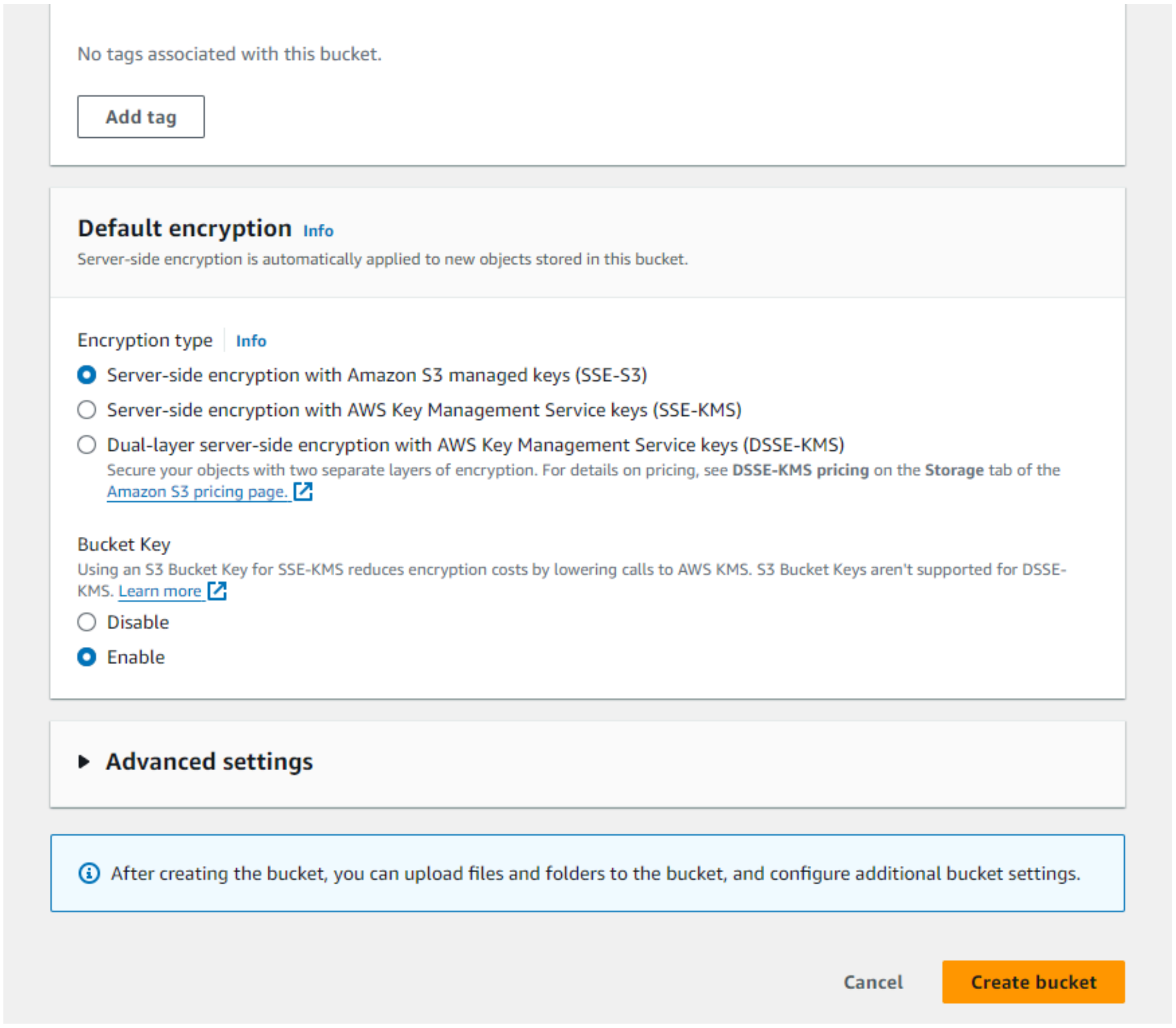
By default, methods inherit stage-level settings. To customize settings for a method, configure method overrides.

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

Copied

https://8jcrb9xjj8.execute-api.us-east-1.amazonaws.com/proxy-stage/{proxy-resource+}

8. Create S3 bucket



No tags associated with this bucket.

**Default encryption** Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type | Info

- [✓ Server-side encryption with Amazon S3 managed keys (SSE-S3)]
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
  Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

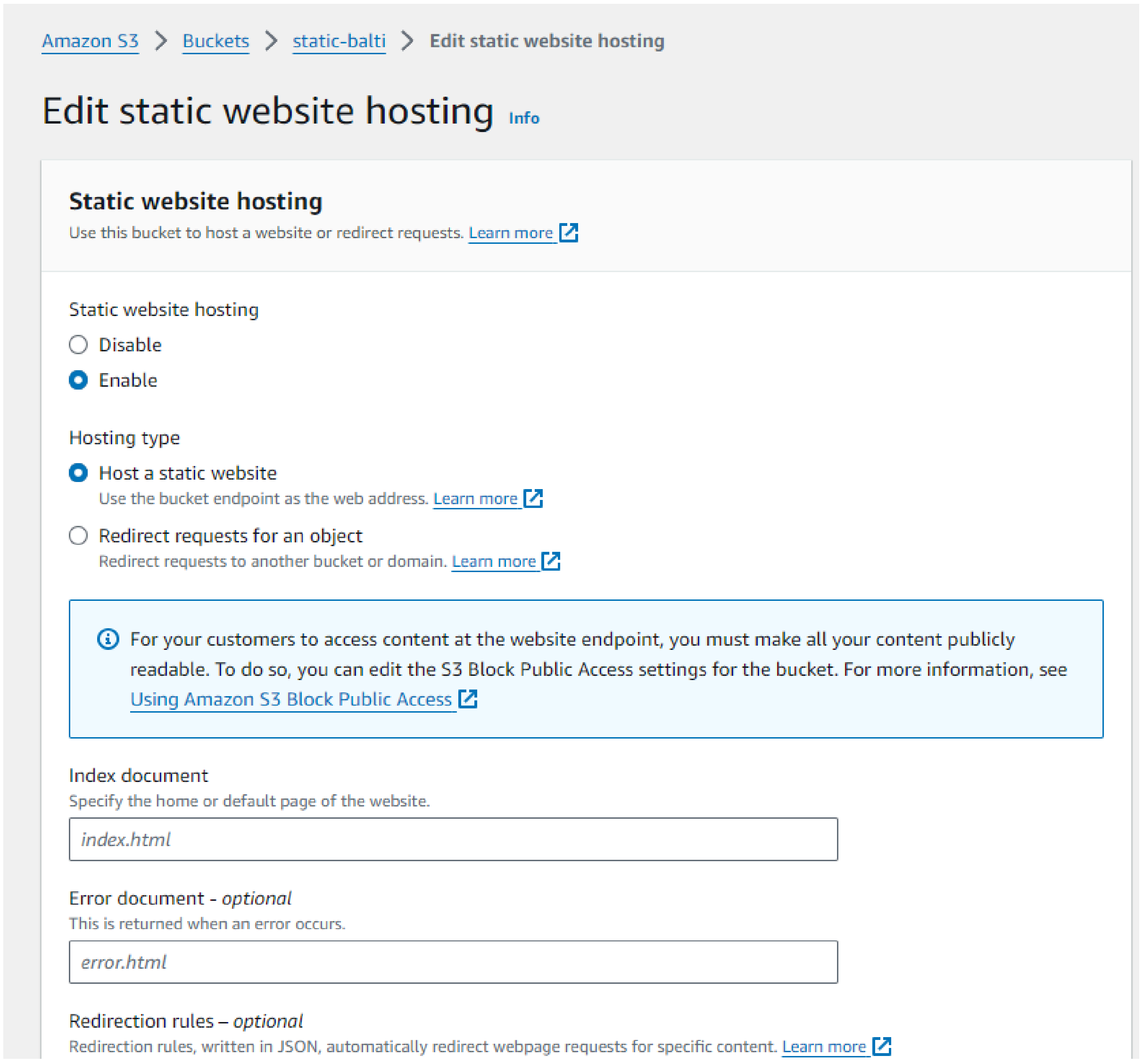
Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

- Disable
- [✓ Enable]

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

9. Edit static website hosting


Amazon S3 > Buckets > static-balti > 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*

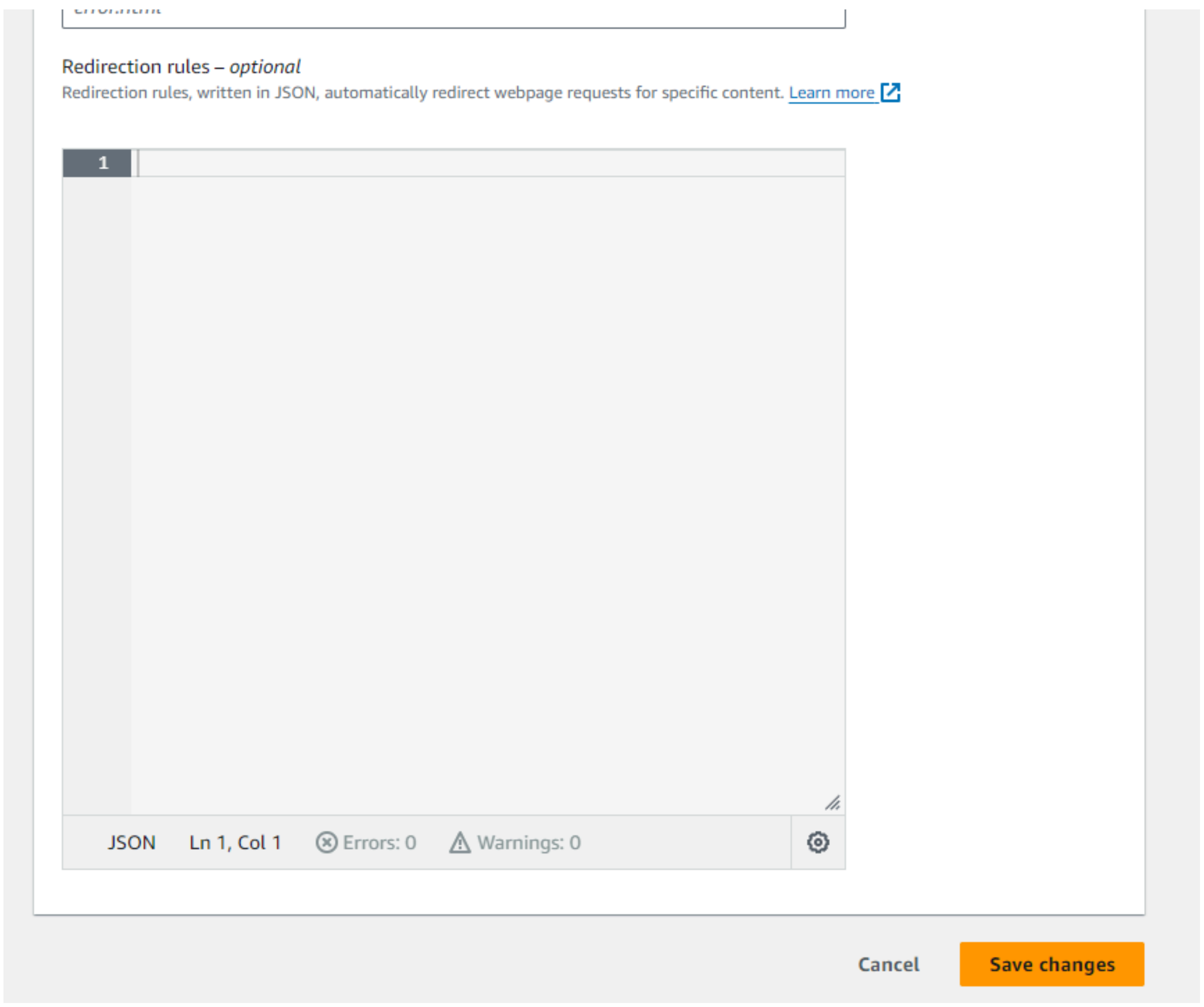
Error document - *optional*

This is returned when an error occurs.

*error.html*

Redirection rules – *optional*

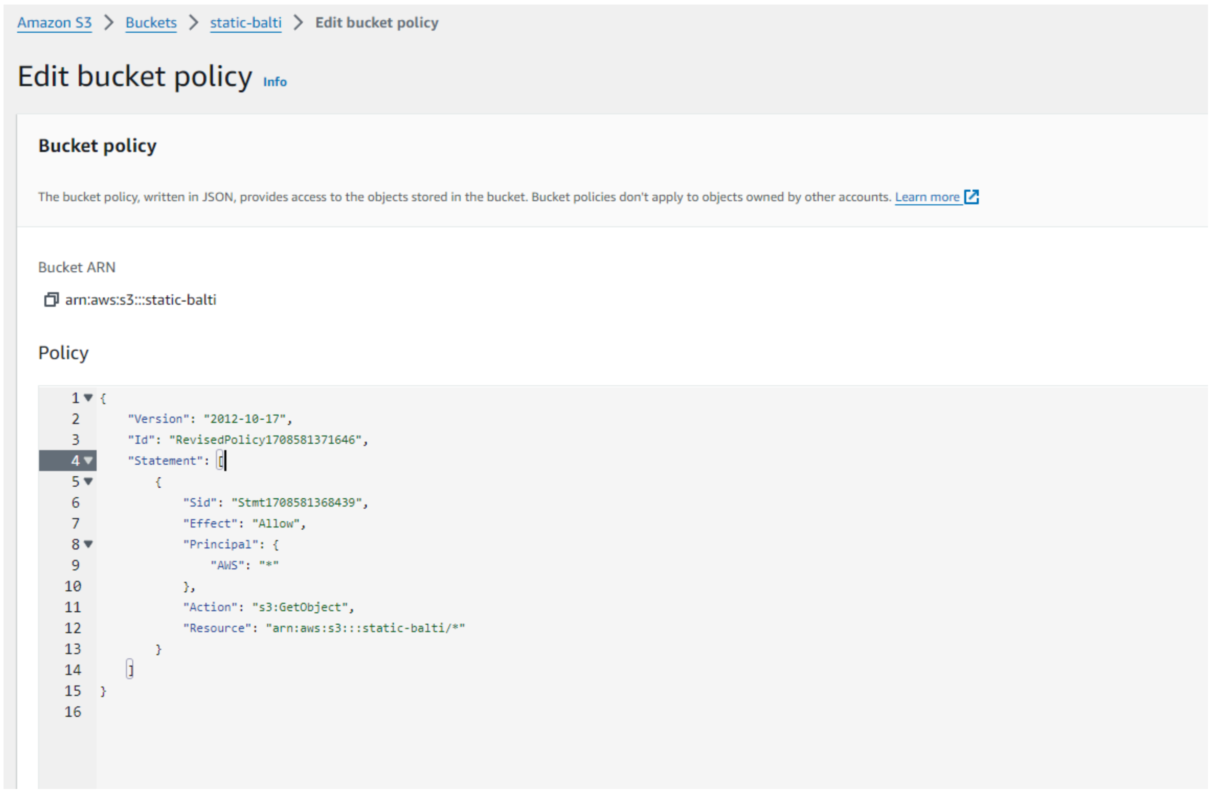
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more



Redirection rules – *optional*

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

10. Edit bucket policy



Amazon S3 > Buckets > static-balti > 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

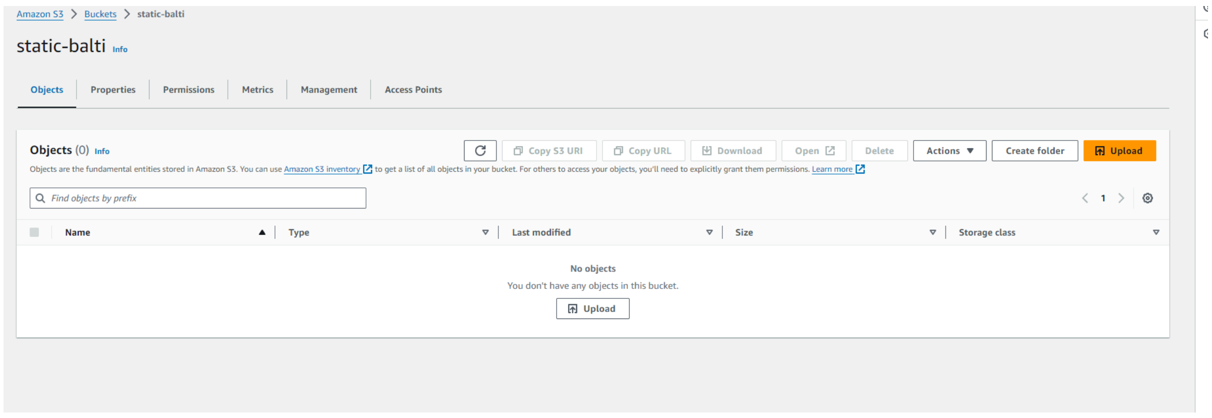
arn:aws:s3:::static-balti

Policy

```
{
    "Version": "2012-10-17",
    "Id": "RevisedPolicy1708581371646",
    "Statement": [
        {
            "Sid": "Stmt1708581368439",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::static-balti/*"
        }
    ]
}
```

```json
{
    "Version": "2012-10-17",
    "Id": "RevisedPolicy1708581371646",
    "Statement": [
        {
            "Sid": "Stmt1708581368439",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::static-balti/*"
        }
    ]
}
```

11. Upload the required website file


Amazon S3 > Buckets > static-balti

static-balti Info

Objects (0) Info

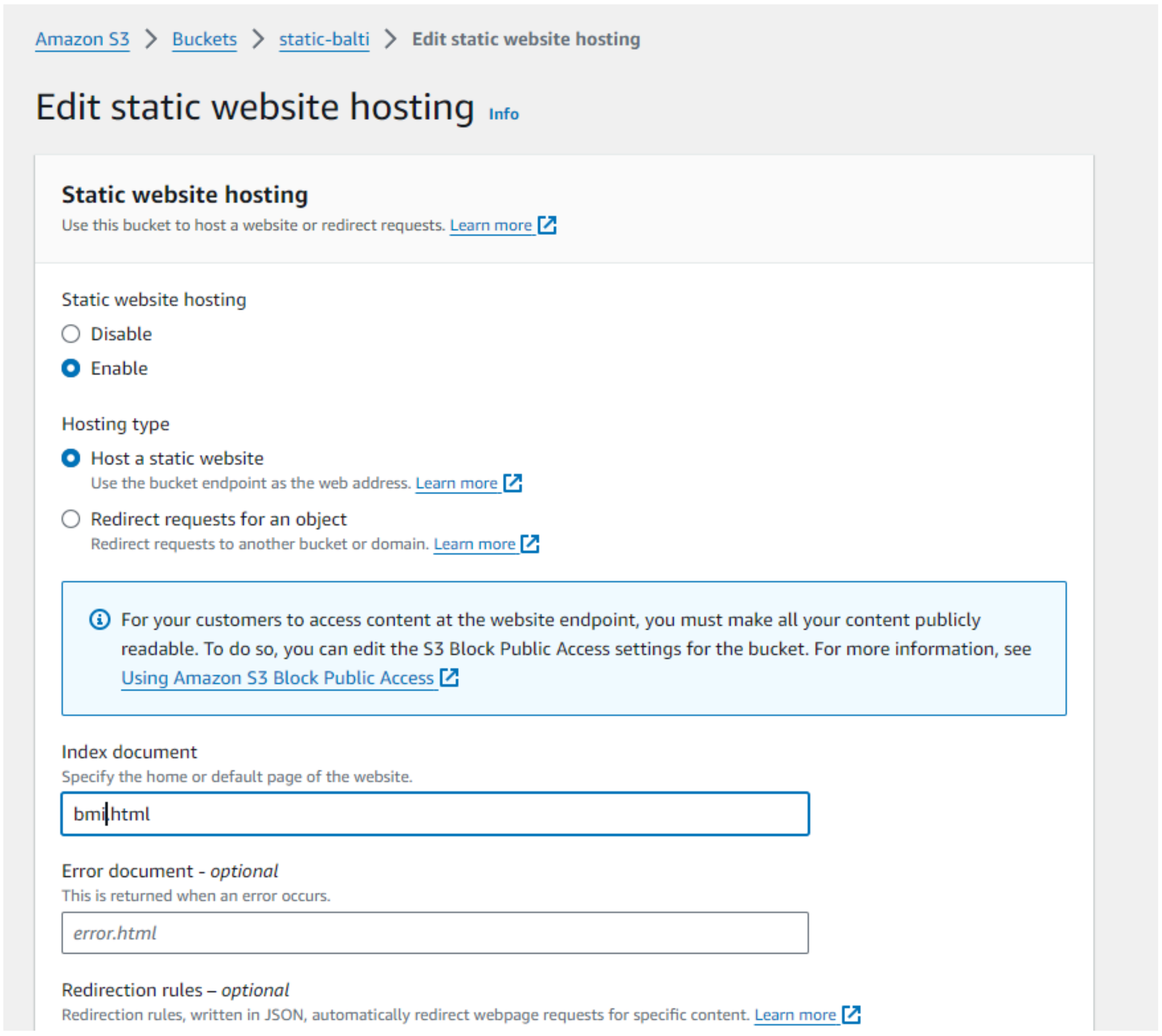
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|

No objects

You don't have any objects in this bucket.


Amazon S3 > Buckets > static-balti > 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.

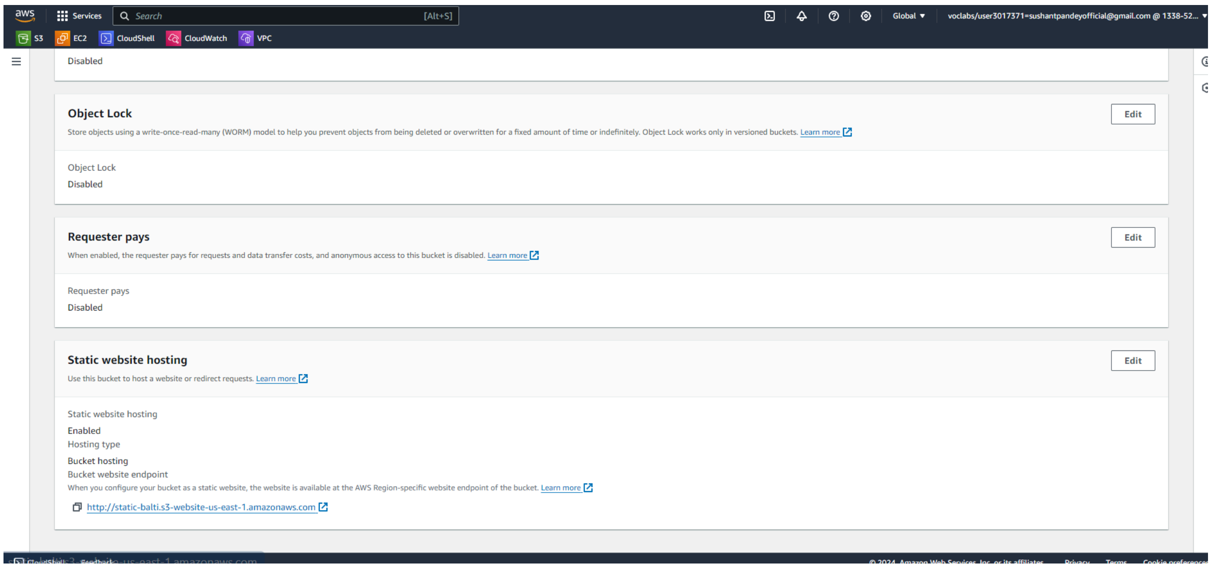
bmi.html

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

12. Find the link and click to check if working



Disabled

**Object Lock**

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. Learn more

Object Lock
Disabled

**Requester pays**

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more

Requester pays
Disabled

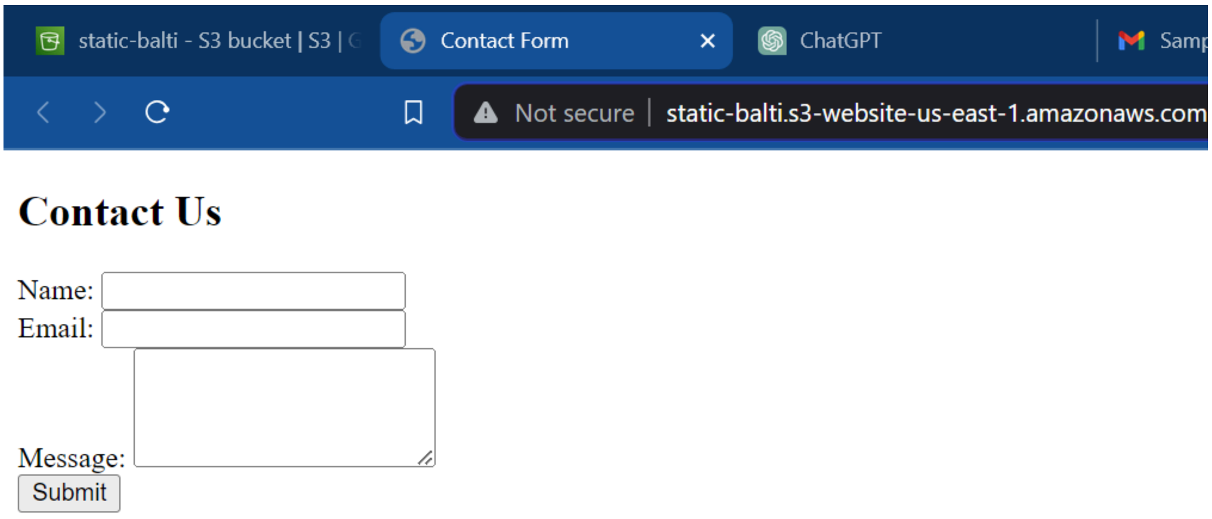
**Static website hosting**

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://static-balti.s3-website-us-east-1.amazonaws.com

aws
Services
Search [Alt+S]
Global
voclabs/user3017371=sushantpandeyofficial@gmail.com @ 1338-52...
S3
EC2
CloudShell
CloudWatch
VPC
Edit
Edit
Edit

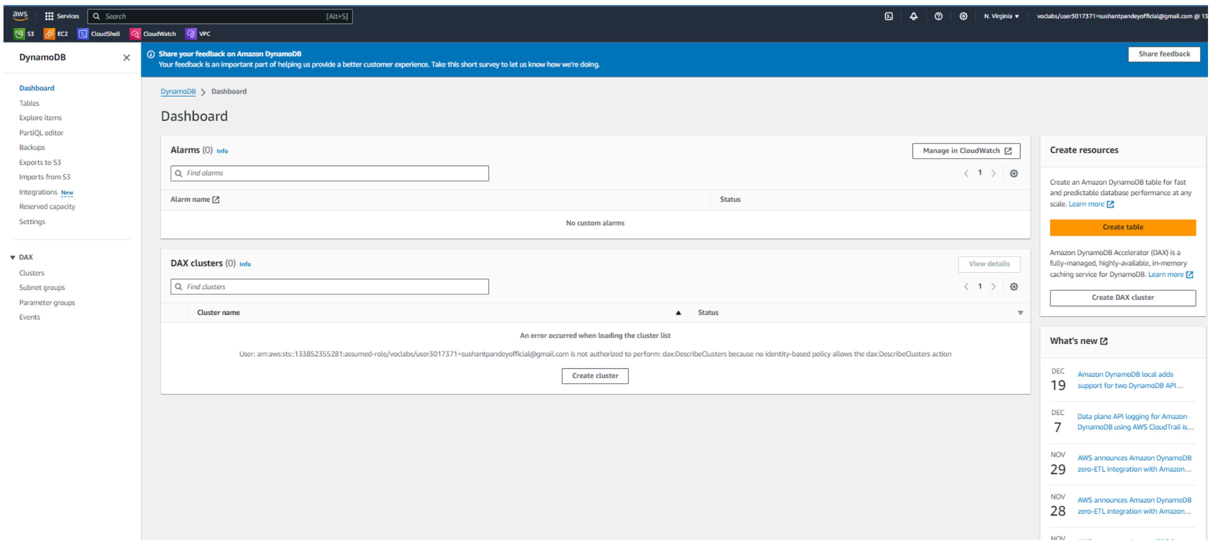


static-balti - S3 bucket | S3 | G
Contact Form
ChatGPT
Samp
Not secure
static-balti.s3-website-us-east-1.amazonaws.com

**Contact Us**

Name:
Email:
Message:
Submit

13. Go to DynamoDb to create tables



**Share your feedback on Amazon DynamoDB**
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Dashboard

Alarms (0) Info

| Alarm name | Status |
|---|---|
| No custom alarms | |

DAX clusters (0) Info

| Cluster name | Status |
|---|---|

An error occurred when loading the cluster list

User: arn:aws:sts::133852355281:assumed-role/voclabs/user3017371=sushantpandeyofficial@gmail.com is not authorized to perform: dax:DescribeClusters because no identity-based policy allows the dax:DescribeClusters action

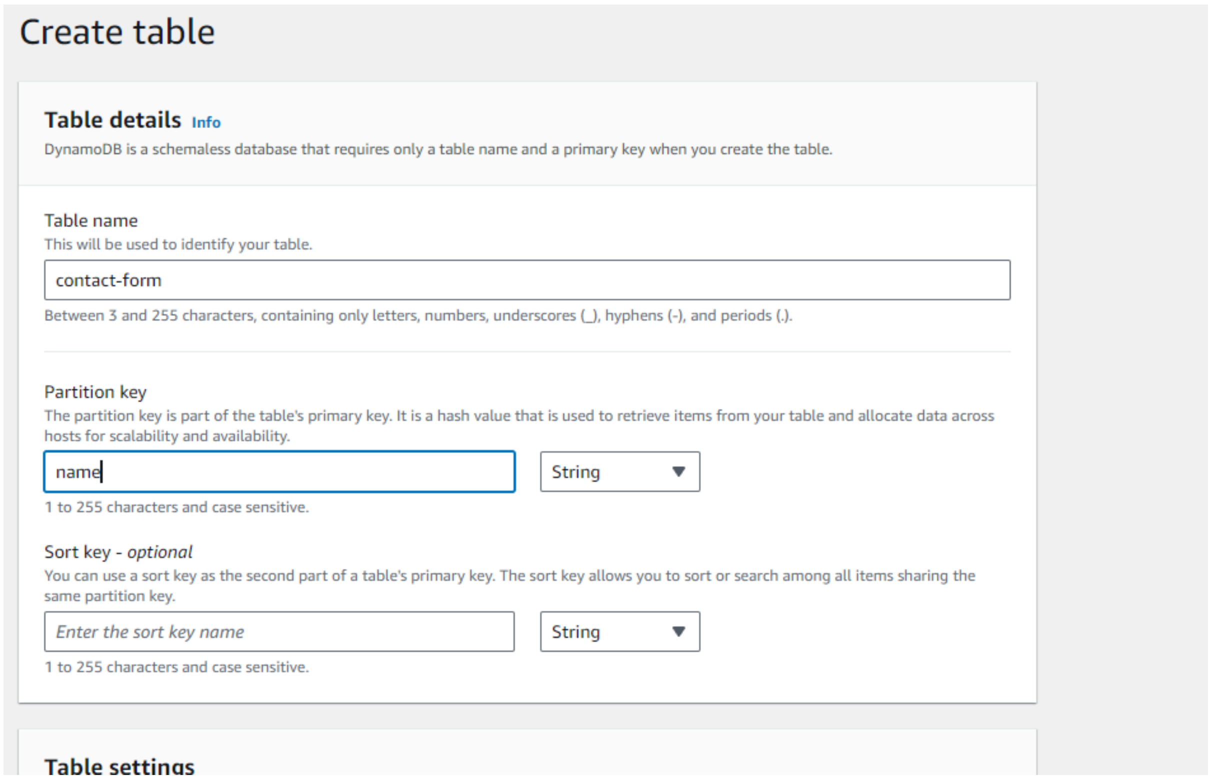
Create resources

Create an Amazon DynamoDB table for fast and predictable database performance at any scale. Learn more

Amazon DynamoDB Accelerator (DAX) is a fully-managed, highly-available, in-memory caching service for DynamoDB. Learn more

What's new

| | |
|---|---|
| DEC 19 | Amazon DynamoDB local adds support for two DynamoDB API… |
| DEC 7 | Data plane API logging for Amazon DynamoDB using AWS CloudTrail is… |
| NOV 29 | AWS announces Amazon DynamoDB zero-ETL integration with Amazon… |
| NOV 28 | AWS announces Amazon DynamoDB zero-ETL integration with Amazon… |



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.

contact-form

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.

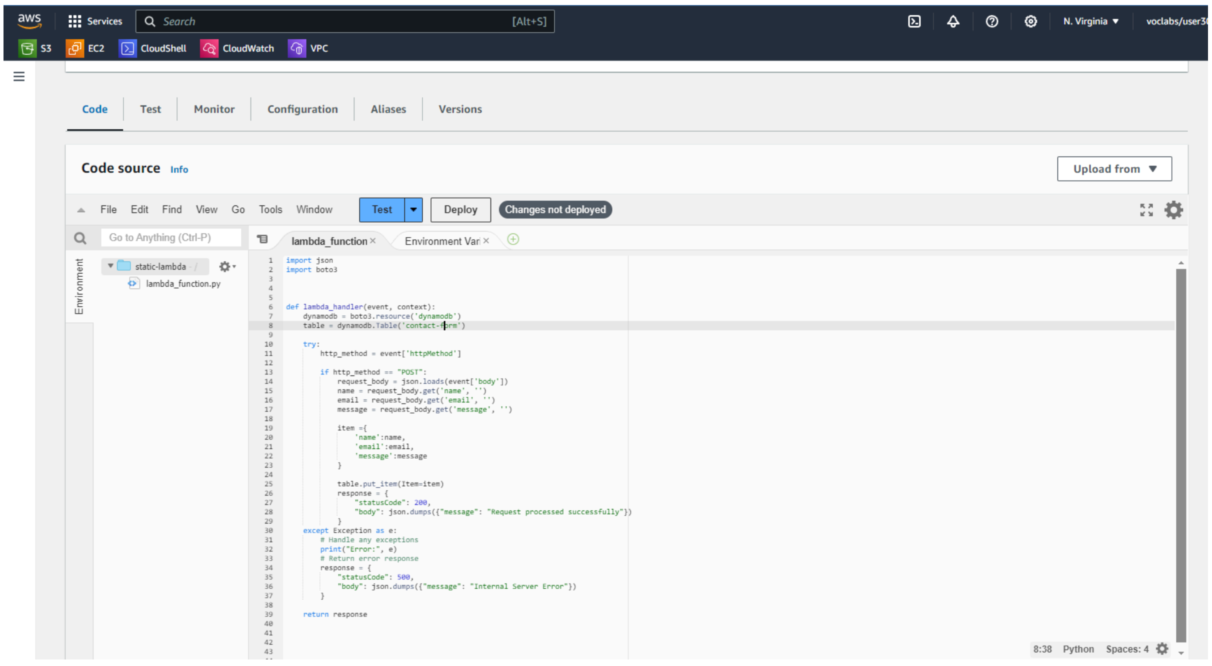
Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

*Enter the sort key name* | String

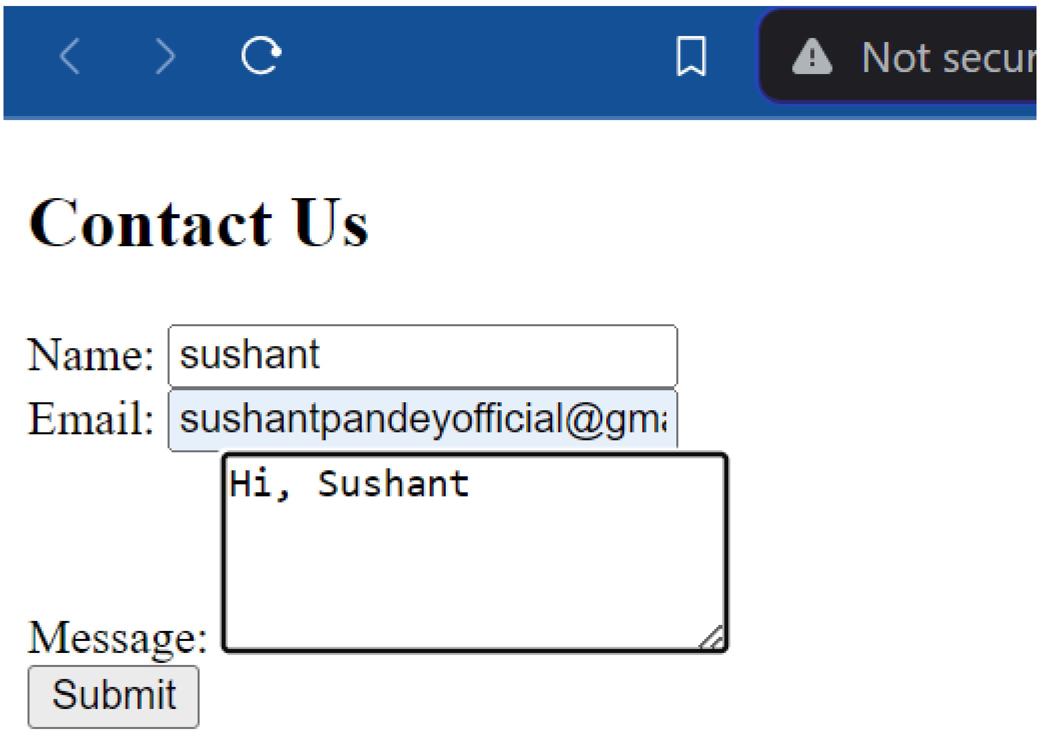
1 to 255 characters and case sensitive.

**Table settings**

14. Write code to receive data in the table in lambda function



```python
import json
import boto3



def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('contact-form')

    try:
        http_method = event['httpMethod']

        if http_method == "POST":
            request_body = json.loads(event['body'])
            name = request_body.get('name', '')
            email = request_body.get('email', '')
            message = request_body.get('message', '')

            item ={
                'name':name,
                'email':email,
                'message':message
            }

            table.put_item(Item=item)
            response = {
                "statusCode": 200,
                "body": json.dumps({"message": "Request processed successfully"})
            }
    except Exception as e:
        # Handle any exceptions
        print("Error:", e)
        # Return error response
        response = {
            "statusCode": 500,
            "body": json.dumps({"message": "Internal Server Error"})
        }

    return response
```



Contact Us

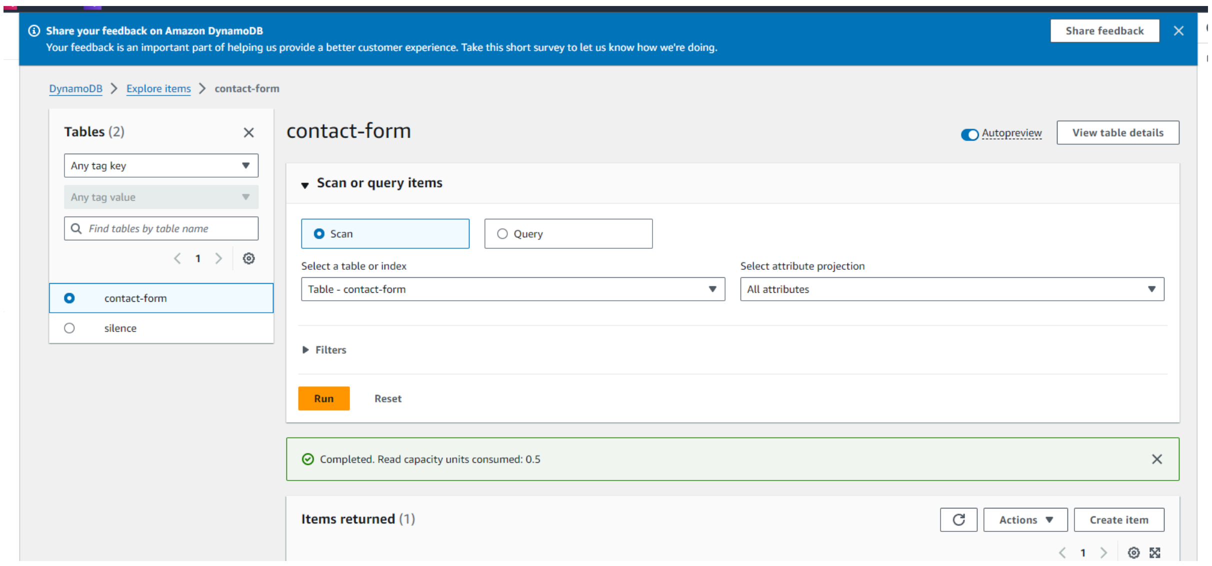
Name: sushant

Email: sushantpandeyofficial@gm

Message: Hi, Sushant

Submit

15. Check if data is updated in table



Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.
Share feedback

DynamoDB > Explore items > contact-form

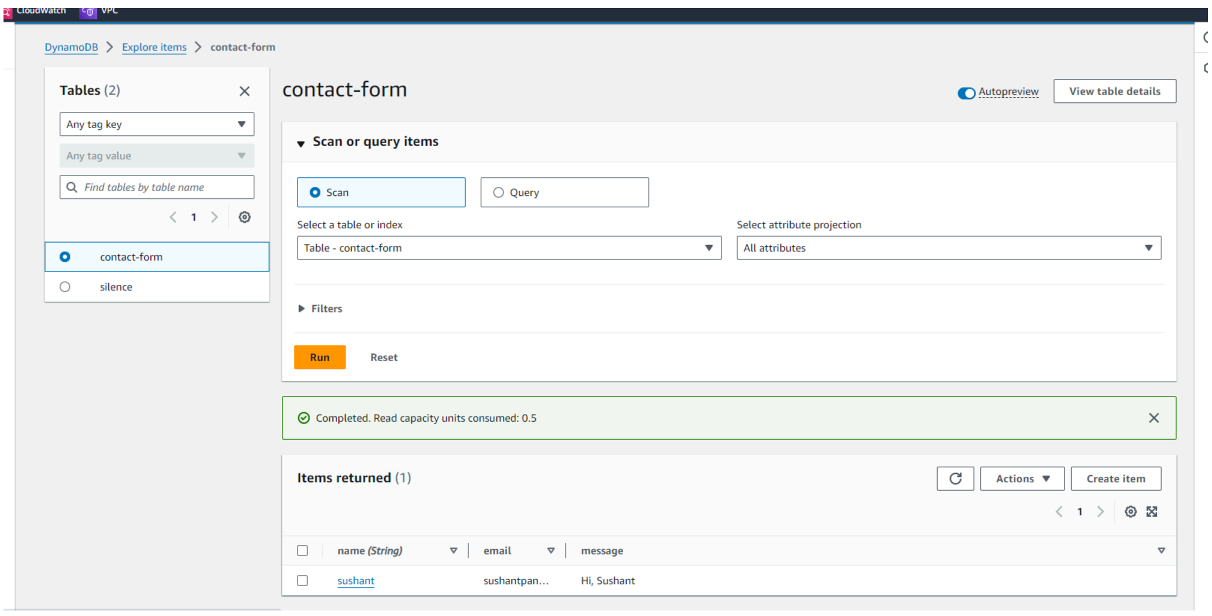
Tables (2)
Any tag key
Any tag value
Find tables by table name
● contact-form
○ silence

contact-form
Autopreview | View table details

Scan or query items
● Scan ○ Query
Select a table or index: Table - contact-form
Select attribute projection: All attributes
Filters
Run | Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)
Actions | Create item



DynamoDB > Explore items > contact-form

Tables (2)
Any tag key
Any tag value
Find tables by table name
● contact-form
○ silence

contact-form
Autopreview | View table details

Scan or query items
● Scan ○ Query
Select a table or index: Table - contact-form
Select attribute projection: All attributes
Filters
Run | Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)
Actions | Create item

| name (String) | email | message |
|---|---|---|
| sushant | sushantpan... | Hi, Sushant |

Conclusion:
Hence, our frontend interacted with the backend. And data was stored in the necessary table.