1. Building a Serverless Web Application

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

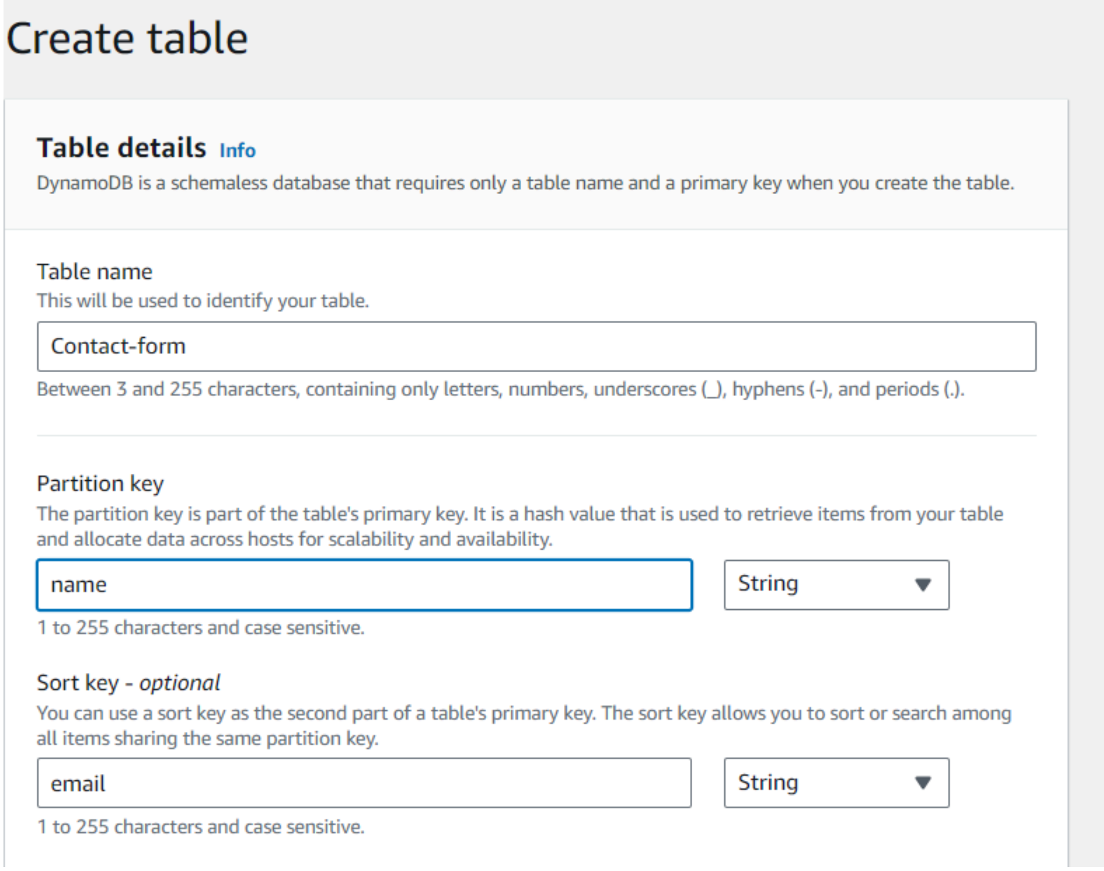
DynamoDB.

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) Go to Dynamo DB> Table > 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.

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.

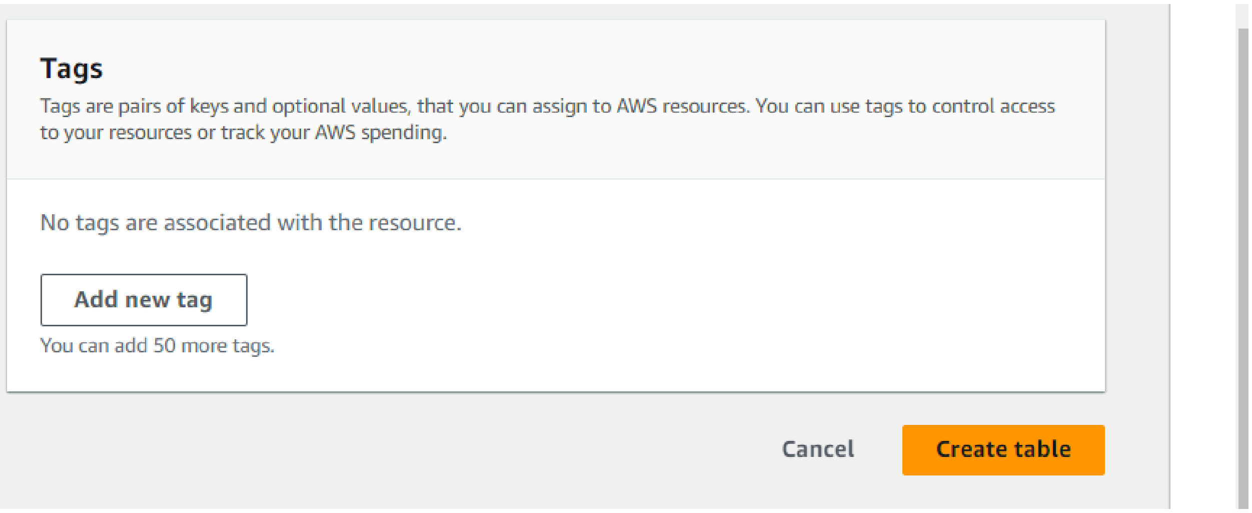
Sort key - *optional*

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.

email | String ▼

1 to 255 characters and case sensitive.

2) Create Table



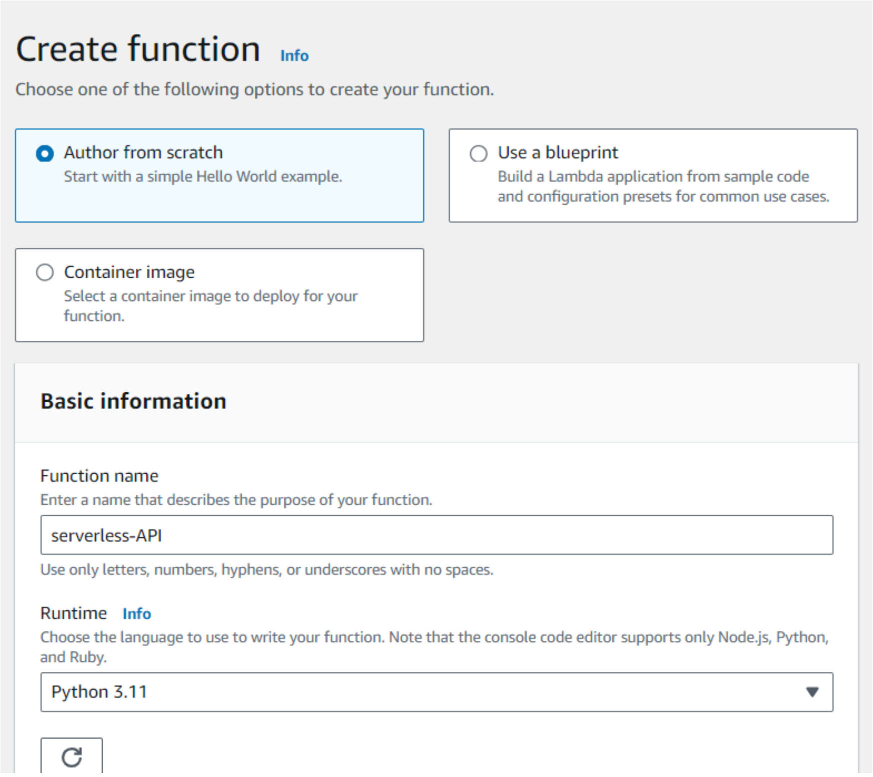
Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

You can add 50 more tags.

3) Go to Lambda function and 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
Enter a name that describes the purpose of your function.

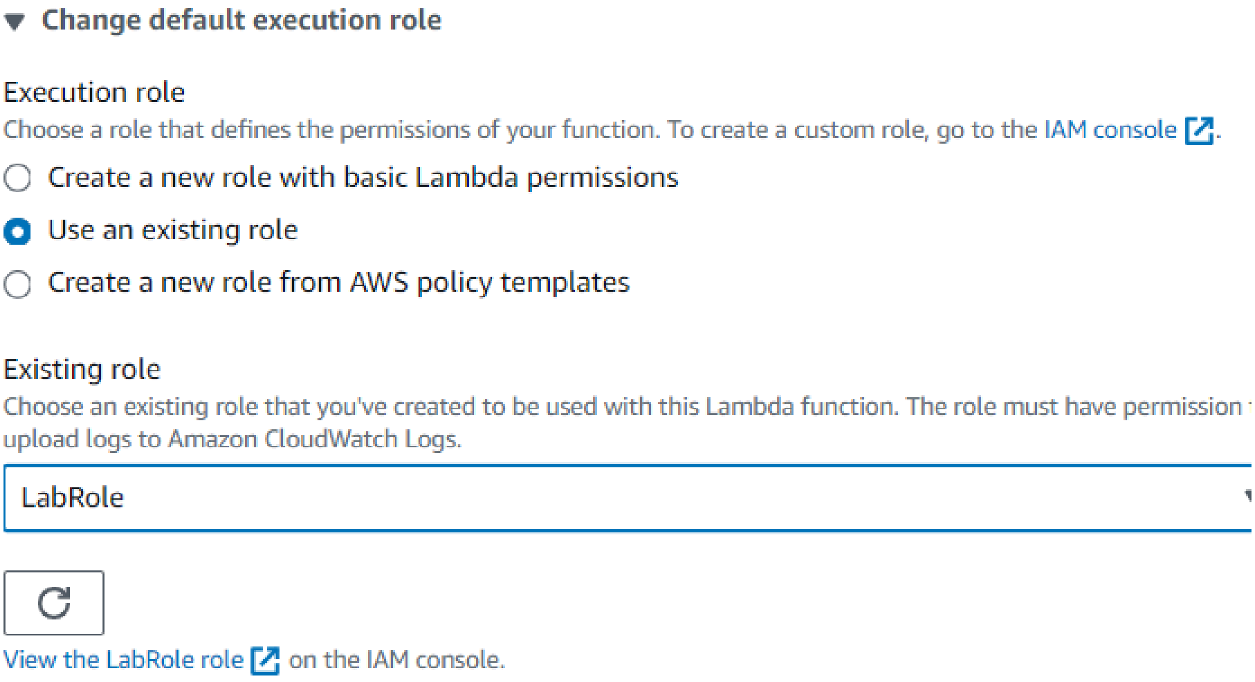
serverless-API

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 3.11

4) Lambda function is created using existing user role.



▼ 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

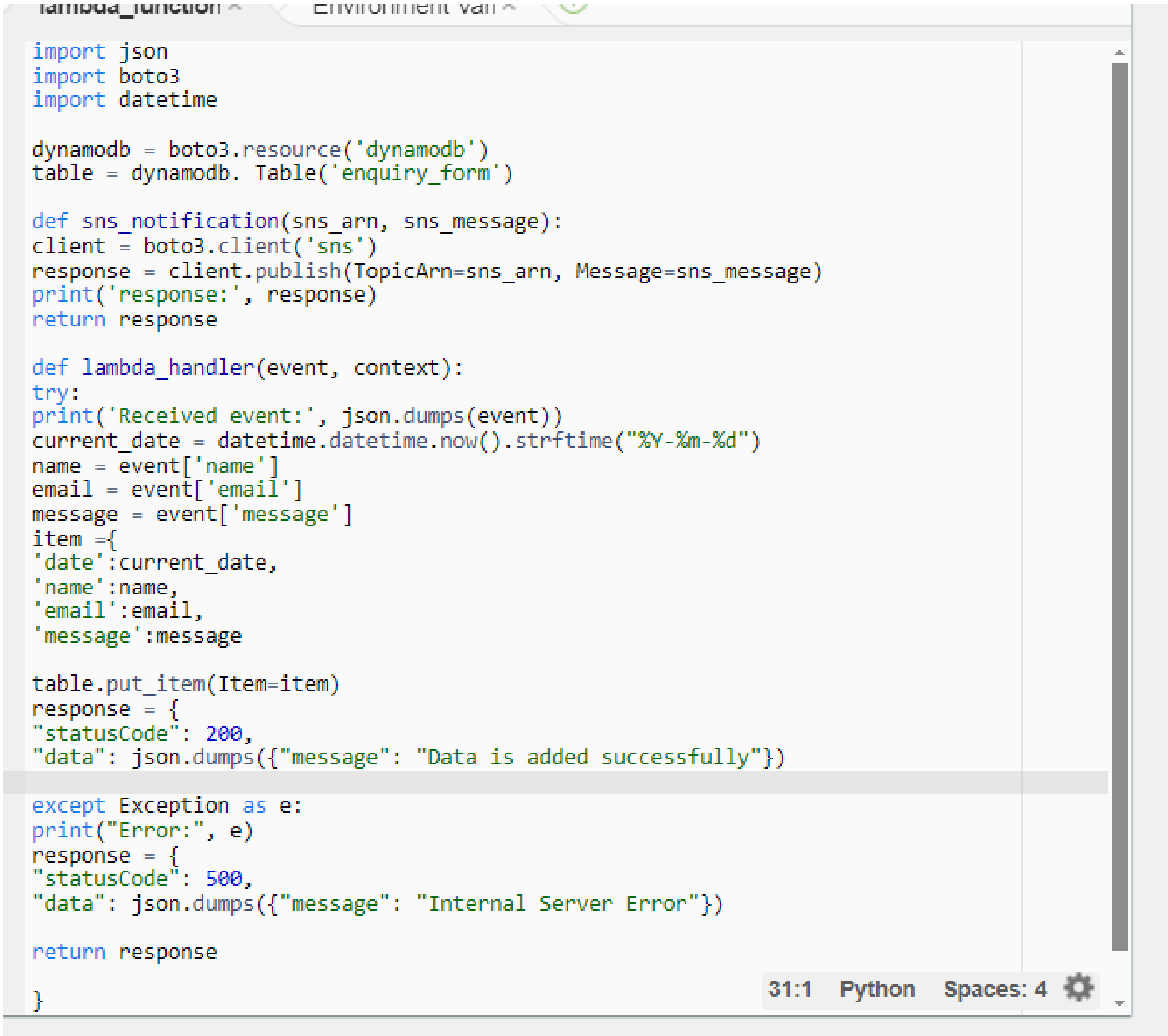
○ Create a new role from AWS policy templates

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

LabRole

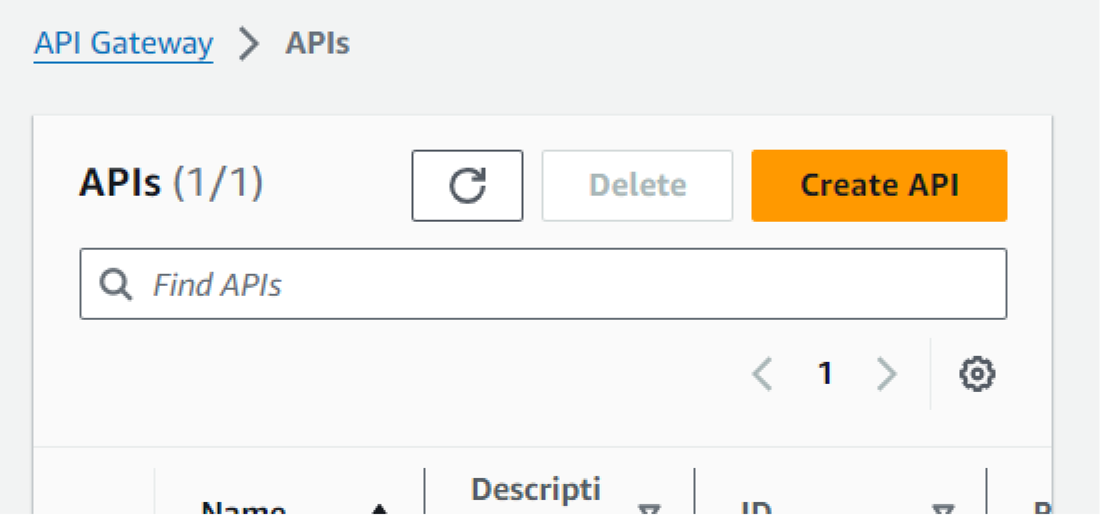
View the LabRole role on the IAM console.

5) **Post request implementation is done and deploy the code.**

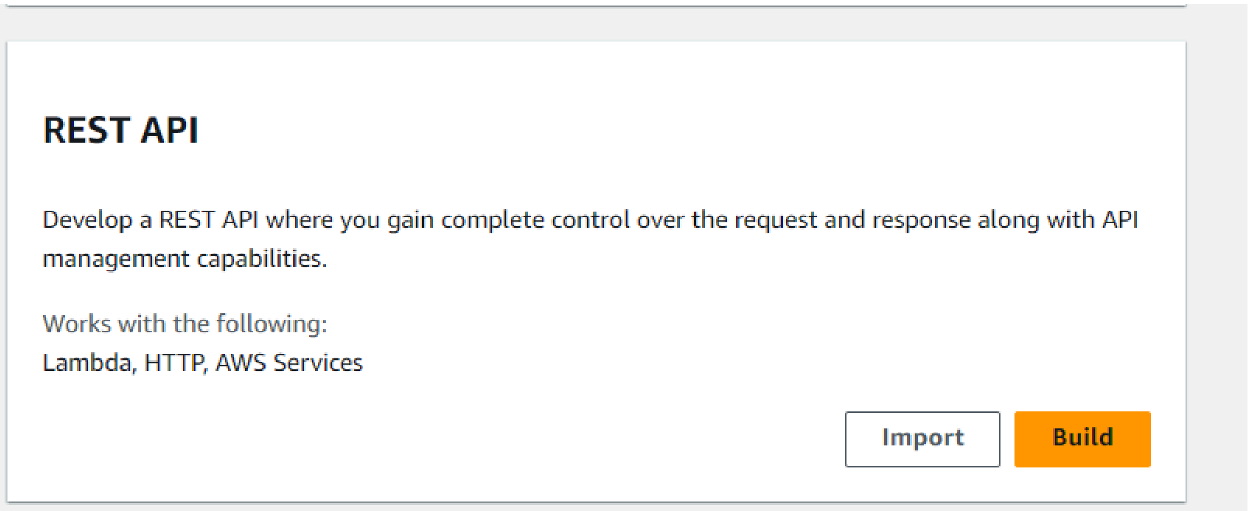


```
import json
import boto3
import datetime

dynamodb = boto3.resource('dynamodb')
table = dynamodb. Table('enquiry_form')

def sns_notification(sns_arn, sns_message):
client = boto3.client('sns')
response = client.publish(TopicArn=sns_arn, Message=sns_message)
print('response:', response)
return response

def lambda_handler(event, context):
try:
print('Received event:', json.dumps(event))
current_date = datetime.datetime.now().strftime("%Y-%m-%d")
name = event['name']
email = event['email']
message = event['message']
item ={
'date':current_date,
'name':name,
'email':email,
'message':message

table.put_item(Item=item)
response = {
"statusCode": 200,
"data": json.dumps({"message": "Data is added successfully"})

except Exception as e:
print("Error:", e)
response = {
"statusCode": 500,
"data": json.dumps({"message": "Internal Server Error"})

return response

}
```

6) **Go to API Gateway and Create API.**



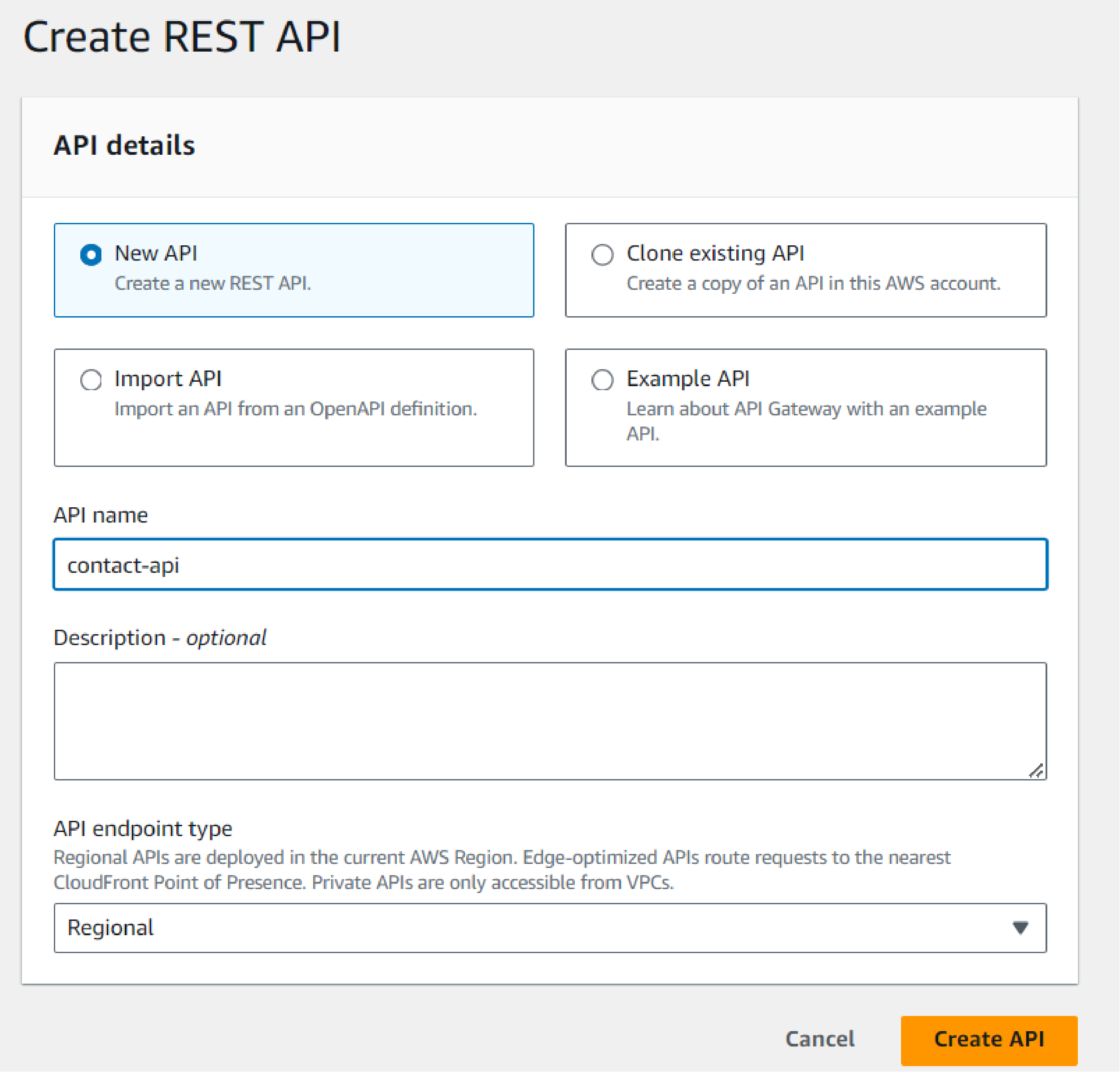
7) REST API configuration is added.


REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:
Lambda, HTTP, AWS Services

8) Then Add the required configuration for 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

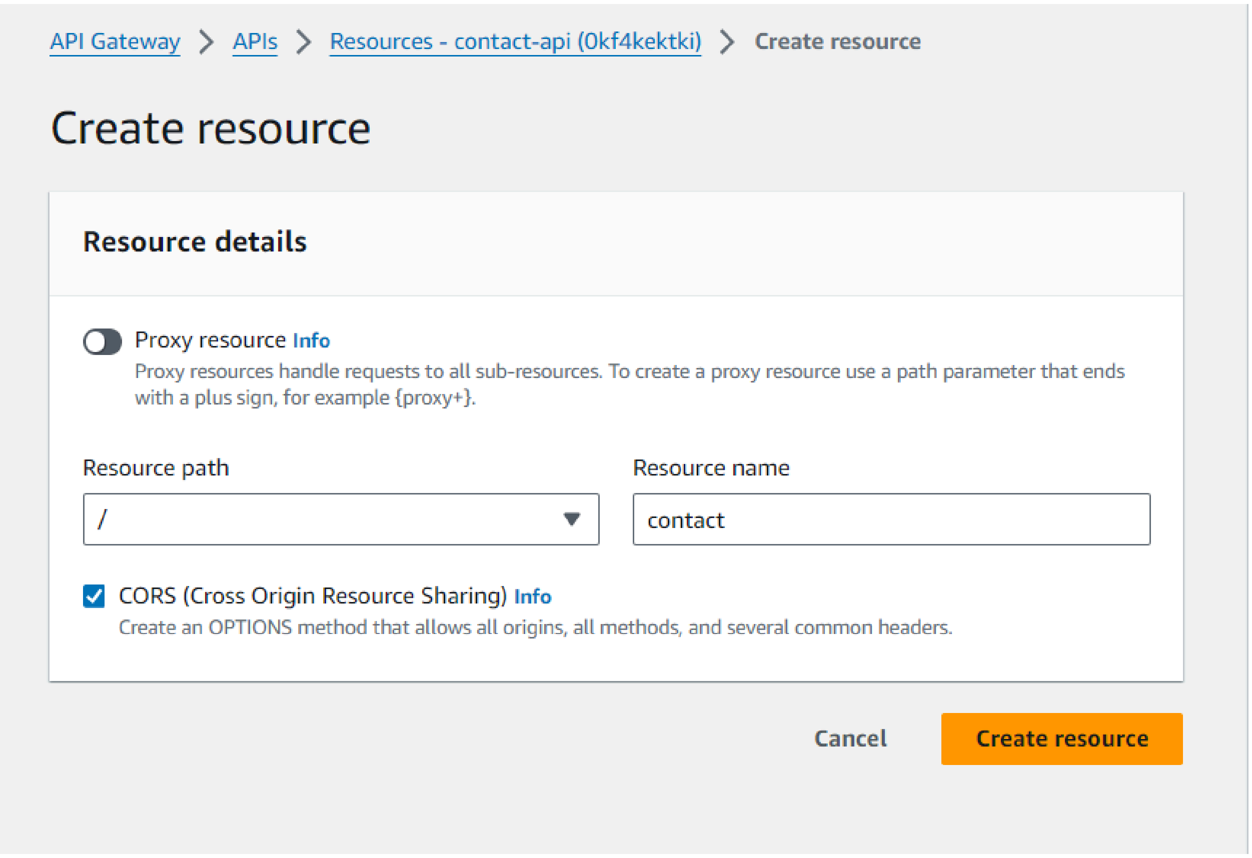
contact-api

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

9) Creating Resource



API Gateway > APIs > Resources - contact-api (0kf4kektki) > 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+}.

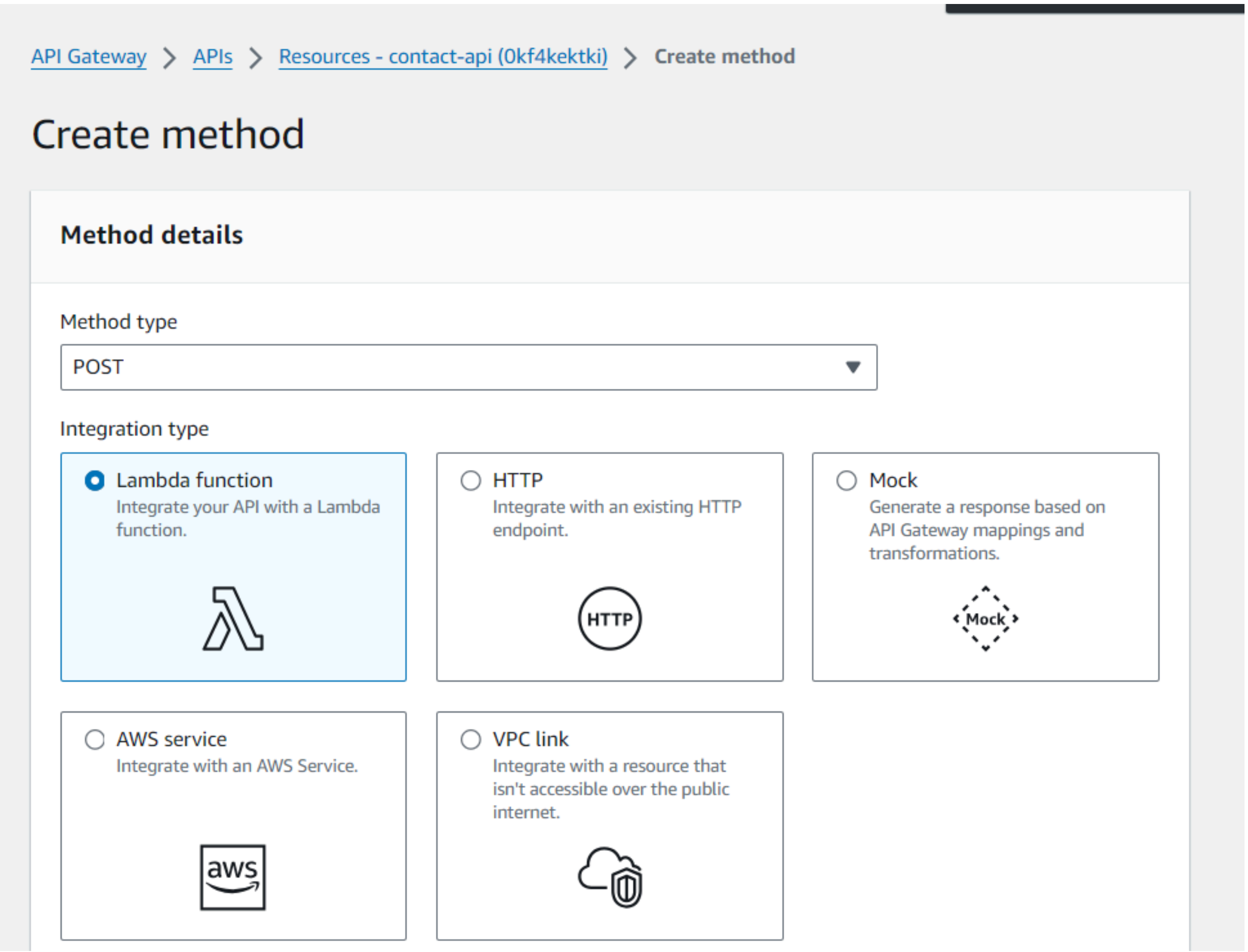
Resource path: /

Resource name: contact

☑ CORS (Cross Origin Resource Sharing) Info

Create an OPTIONS method that allows all origins, all methods, and several common headers.

10) Crete a POST Method and add recently created lambda function in the configuration.



API Gateway > APIs > Resources - contact-api (0kf4kektki) > Create method

Create method

Method details

Method type

POST

Integration type

● Lambda function
Integrate your API with a Lambda function.

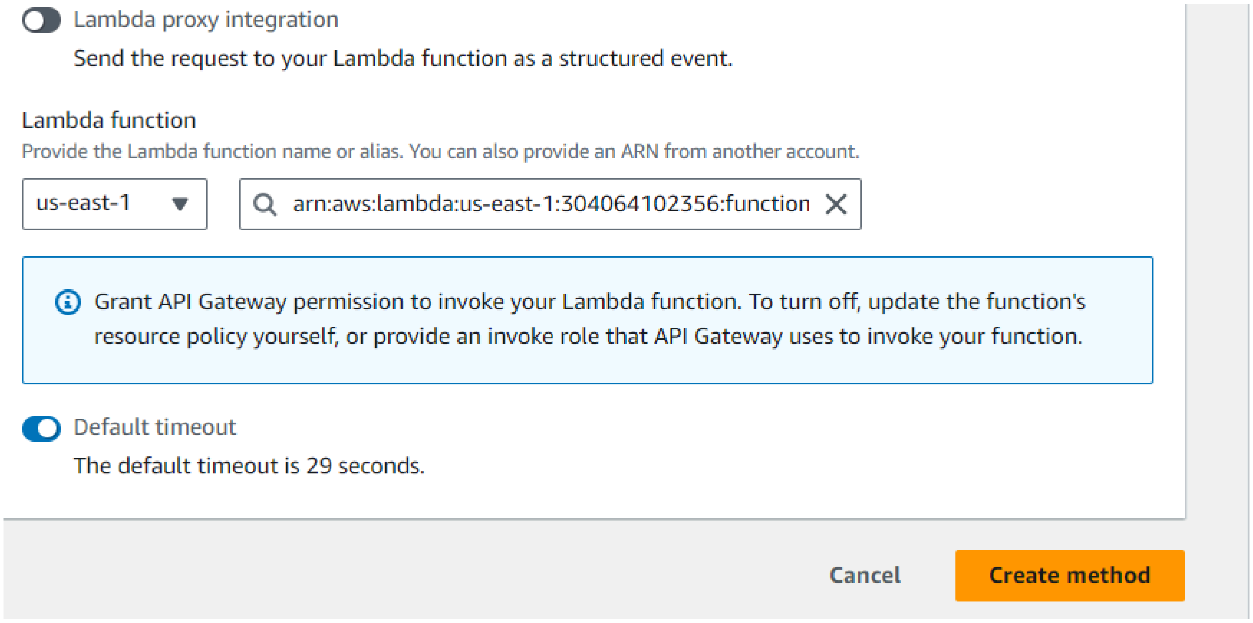
○ HTTP
Integrate with an existing HTTP endpoint.

○ 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.

11) Create method



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.

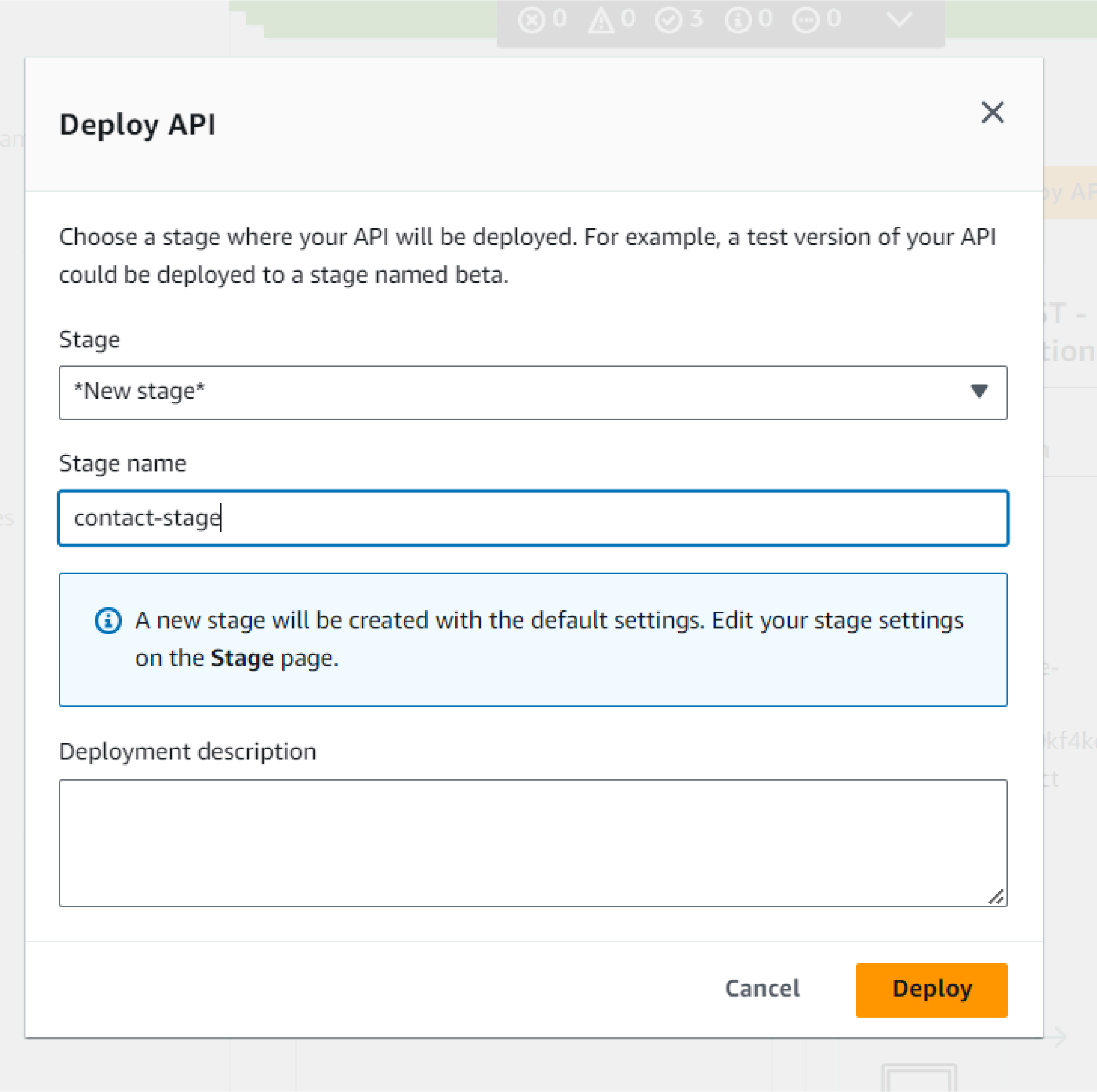
us-east-1 ▼ arn:aws:lambda:us-east-1:304064102356:function

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

The default timeout is 29 seconds.

12) **Deploy the API using a new stage.**



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

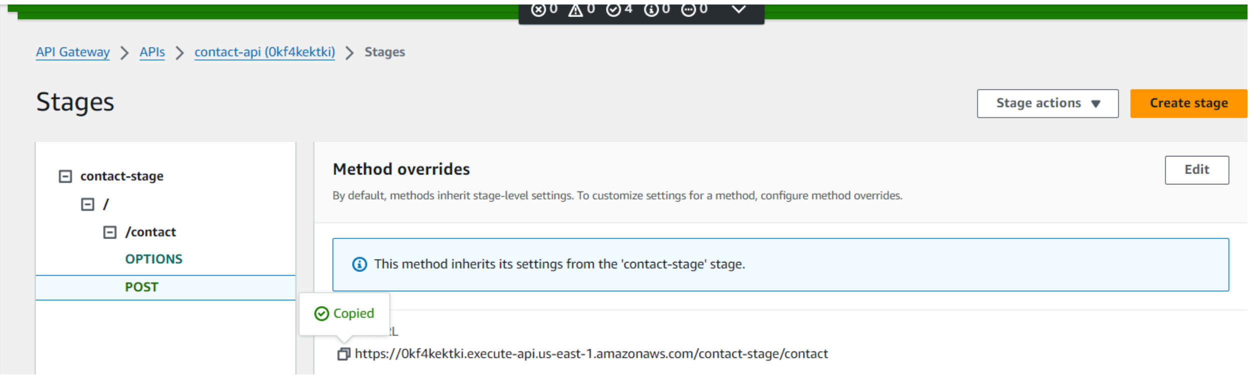
Stage name

contact-stage

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

Deployment description

13) **Copy URL of post method from the created stage.**



Stages

Method overrides

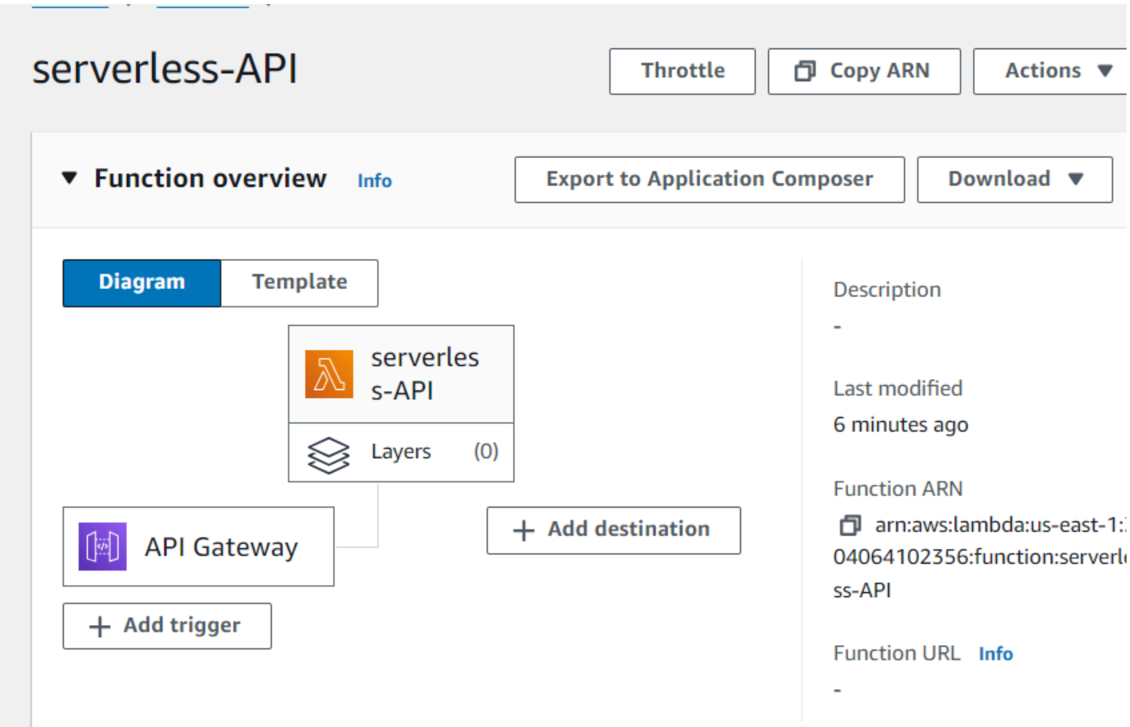
By default, methods inherit stage-level settings. To customize settings for a method, configure method overrides.

ⓘ This method inherits its settings from the 'contact-stage' stage.

L

https://0kf4kektki.execute-api.us-east-1.amazonaws.com/contact-stage/contact

14) We can now observe that API Gateway is added in lambda function



serverless-API

Description

-

Last modified

6 minutes ago

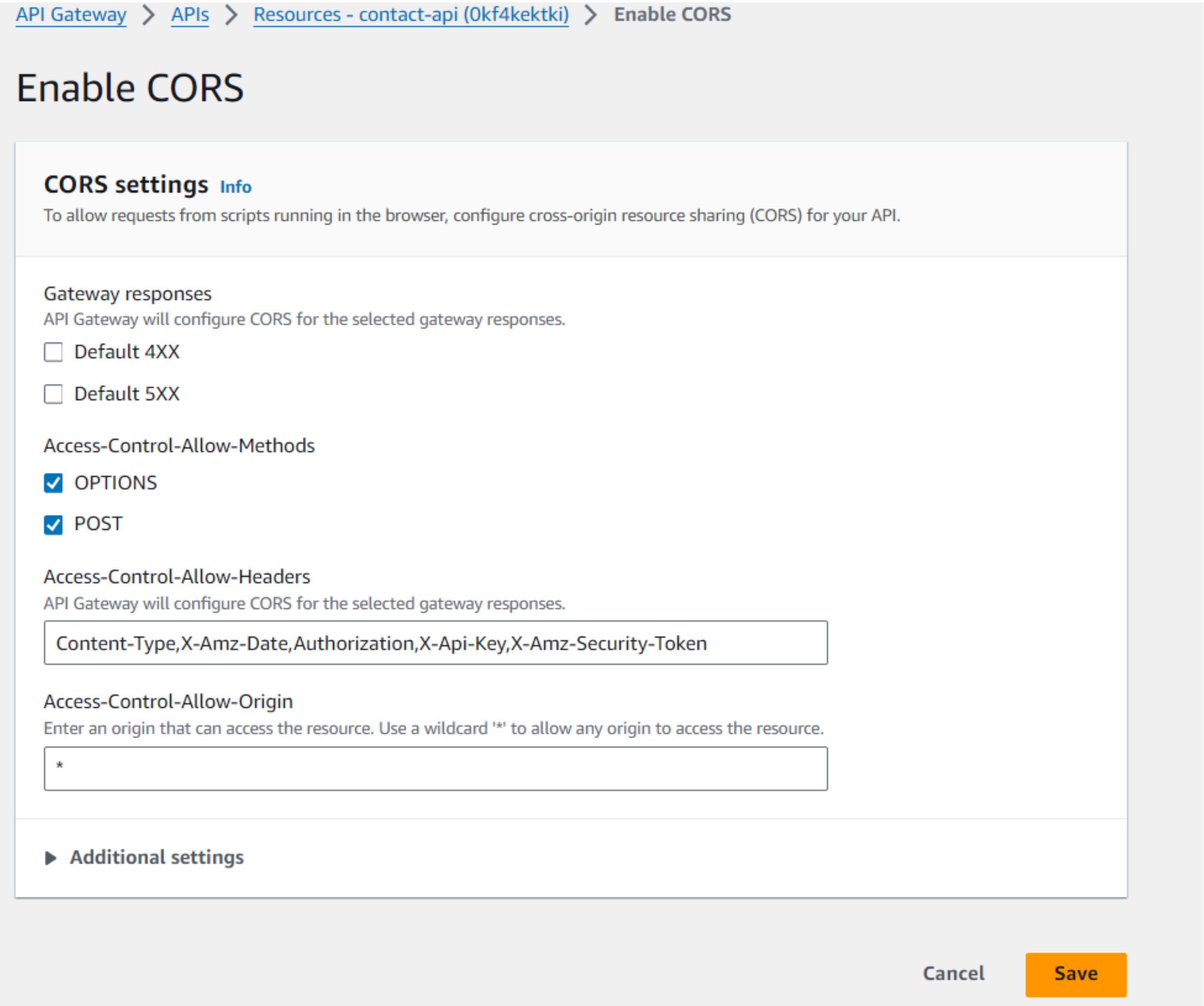
Function ARN

arn:aws:lambda:us-east-1::
04064102356:function:serverl
ss-API

Function URL

-

15) Enable CORS


API Gateway > APIs > Resources - contact-api (0kf4kektki) > 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

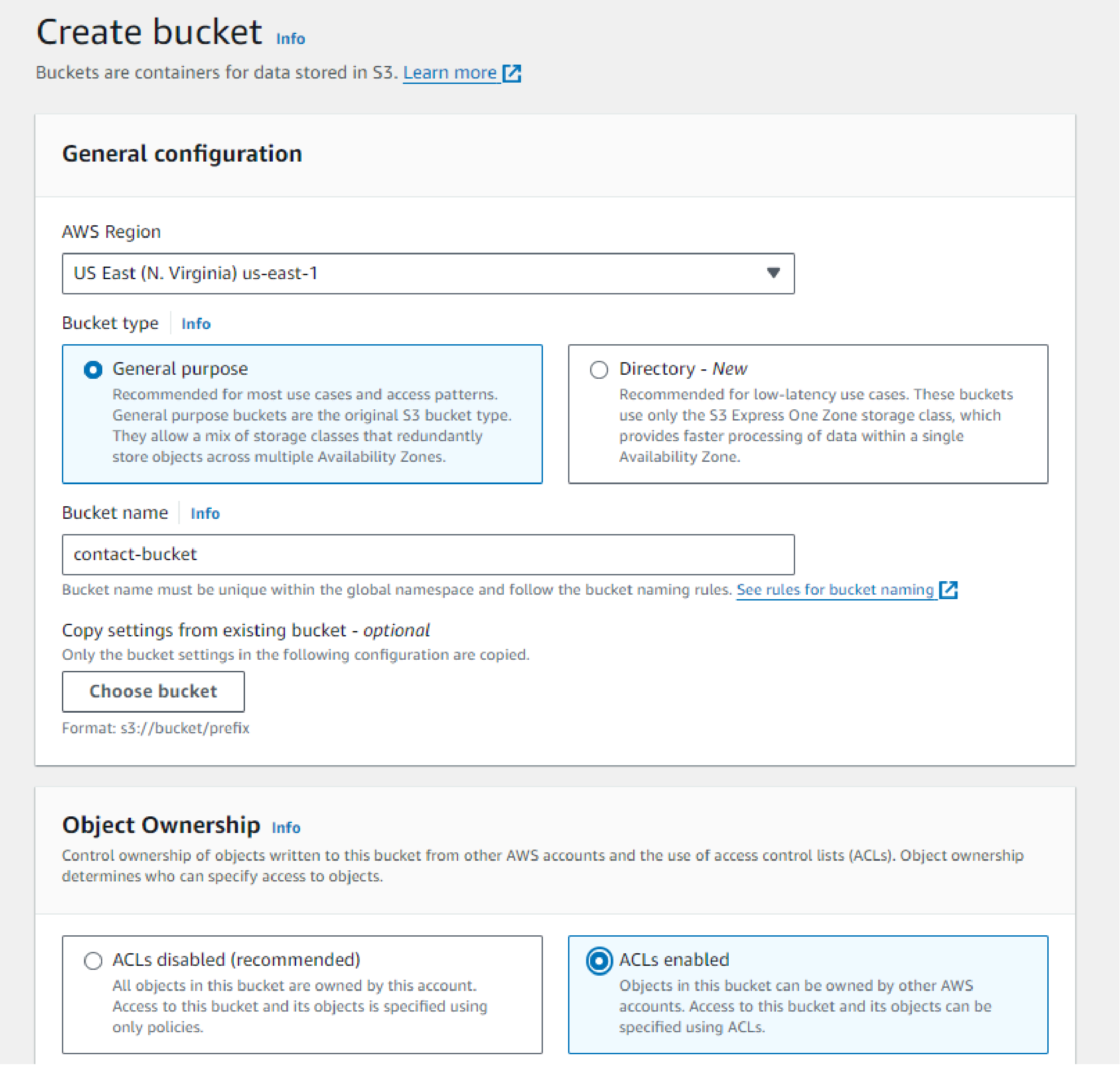
Access-Control-Allow-Origin

Enter an origin that can access the resource. Use a wildcard '*' to allow any origin to access the resource.

*

Now for Front end hosting

16) **s3 bucket creation and uploading the html file Adding all the required configuration is S3 bucket. “ACLs enabled” is choosen in object ownership so that bucket is publicy available with the url.**



Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory - *New*
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

contact-bucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

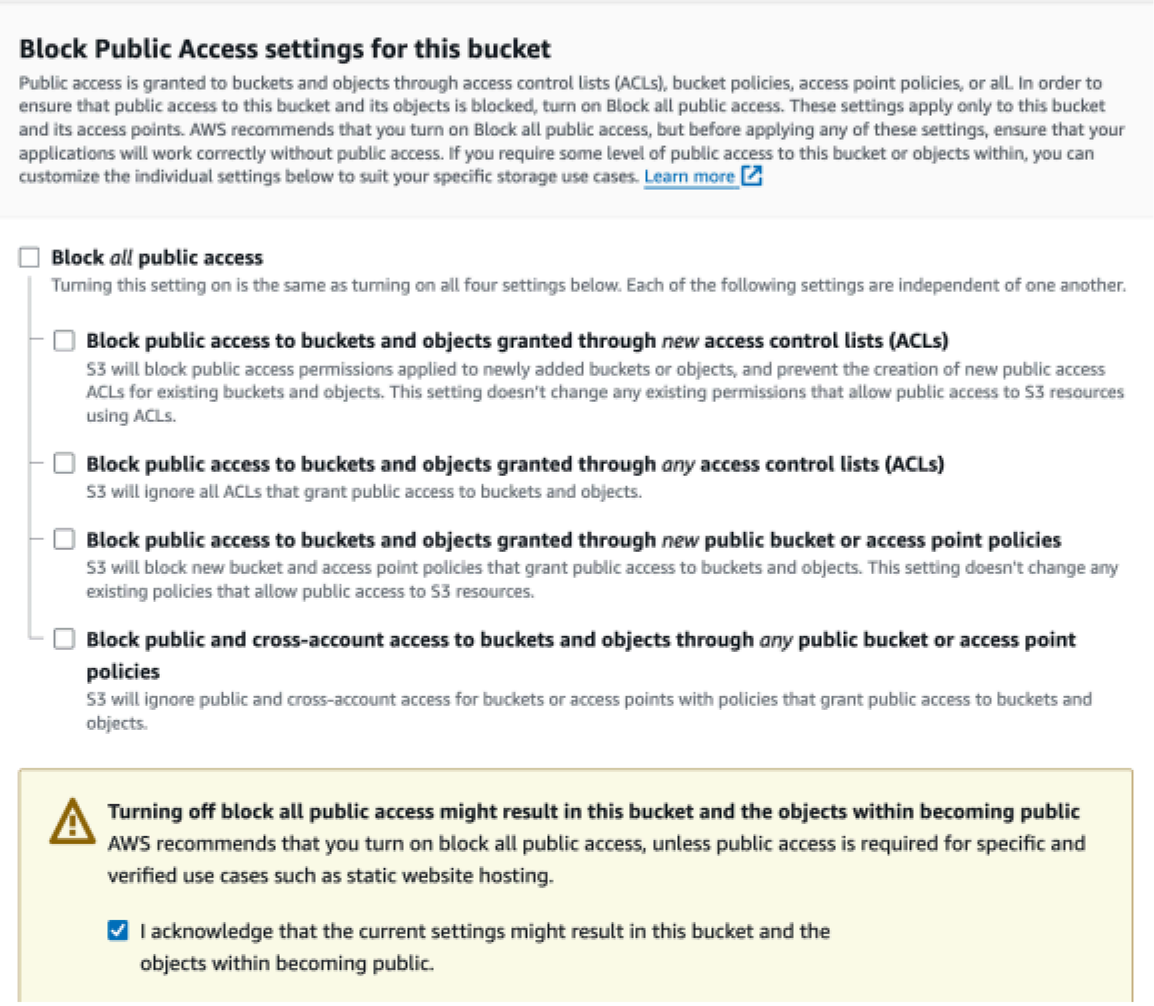
Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

17) Block all public acess is removed for getting access through website.



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

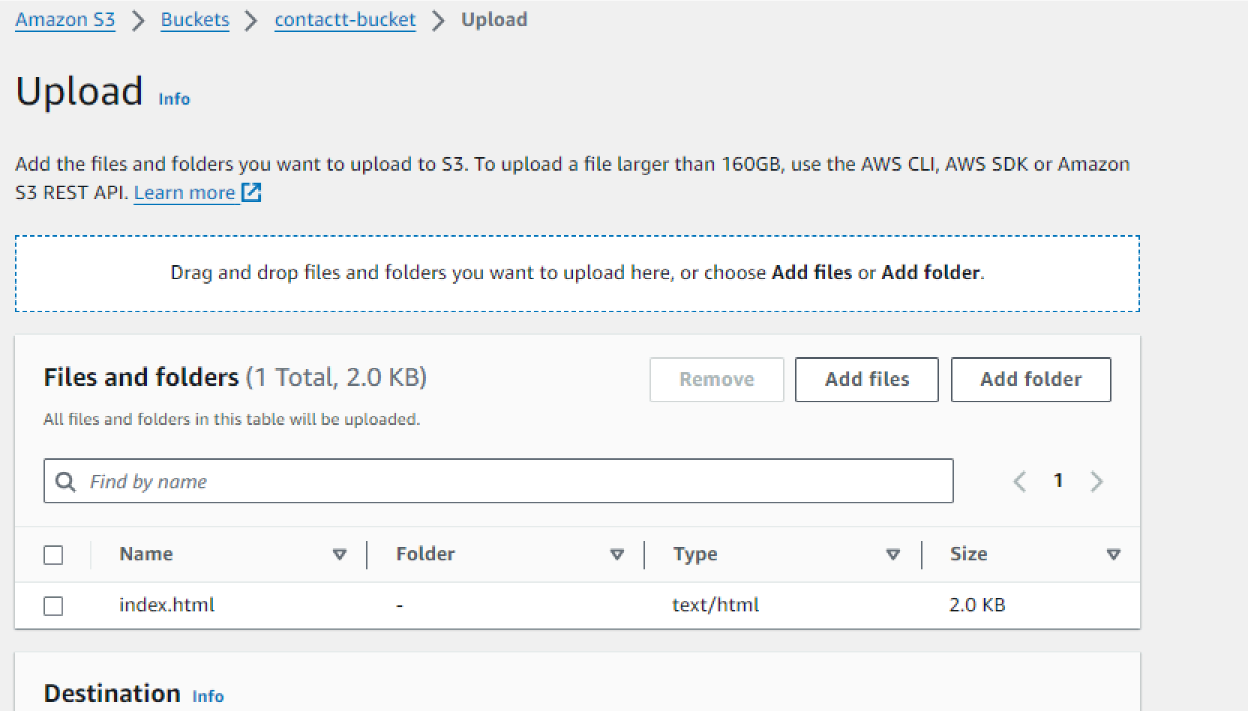
□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☑ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

18) After creation of S3 bucket, Upload html file containing contact form.



Amazon S3 > Buckets > contactt-bucket > 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**.

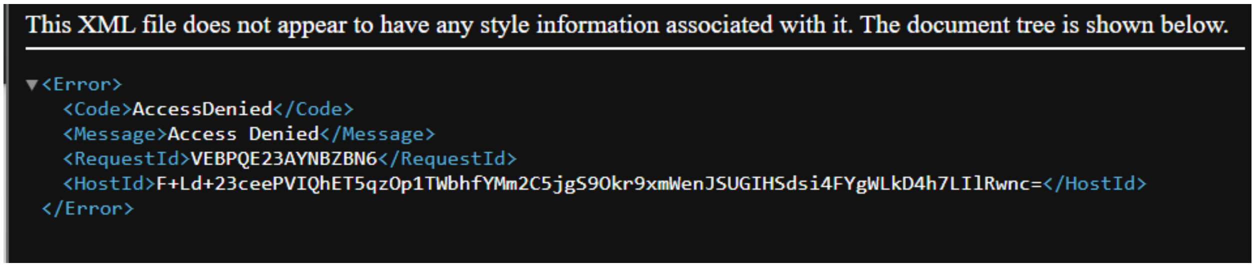
Files and folders (1 Total, 2.0 KB)

All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size
□	index.html	-	text/html	2.0 KB

Destination Info

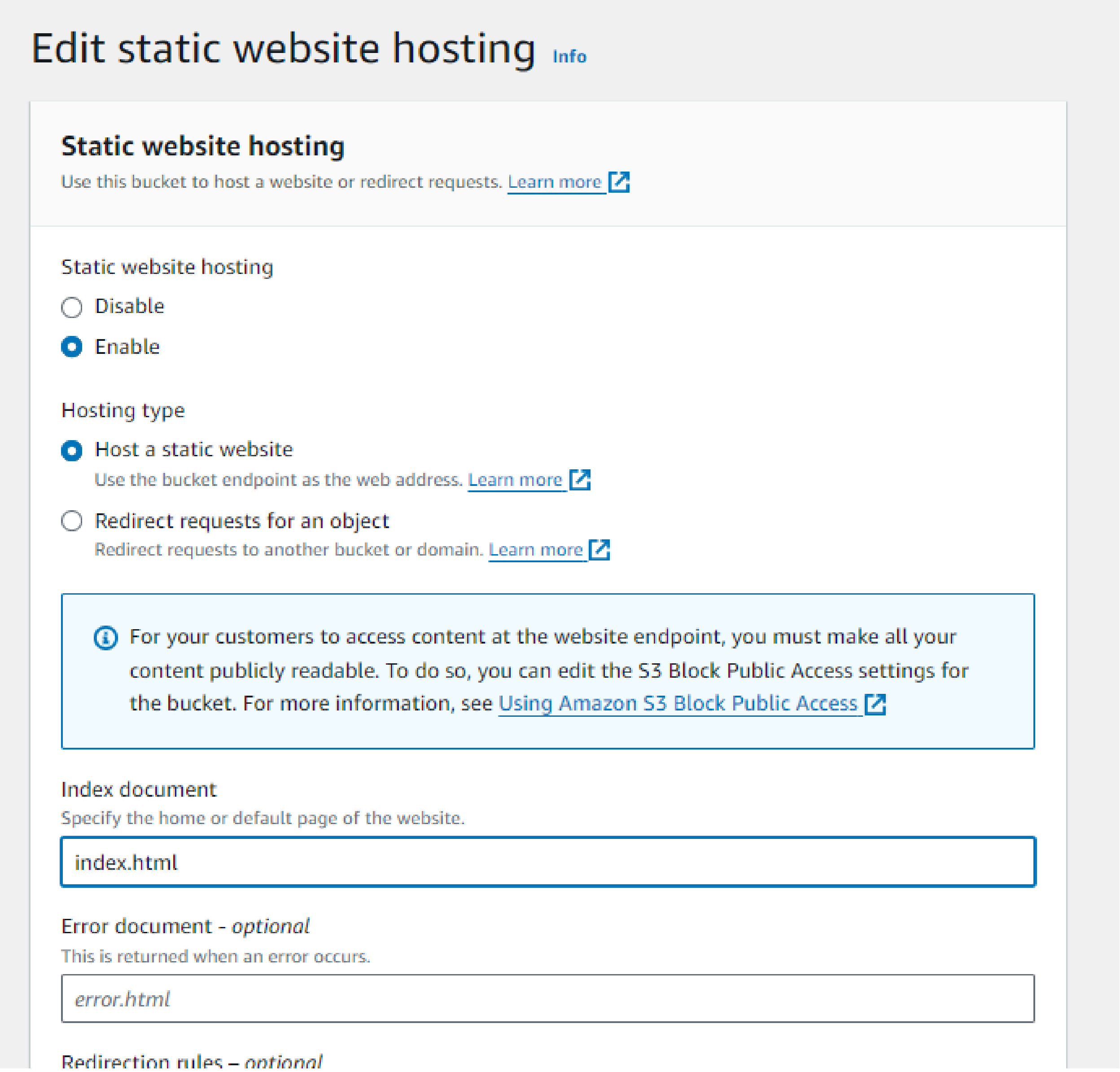
19) Access denied message is shown when the url of uploaded file is browsed.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>VEBPQE23AYNBZBN6</RequestId>
   <HostId>F+Ld+23ceePVIQhET5qzOp1TWbhfYMm2C5jgS9Okr9xmWenJSUGIHSdsi4FYgWLkD4h7LIlRwnc=</HostId>
 </Error>
```

20) So, now the website is hosted from properties tab of created S3 bucket. Initially the static website hosting is disabled and for hosting it’s status is updated to “Enable”



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

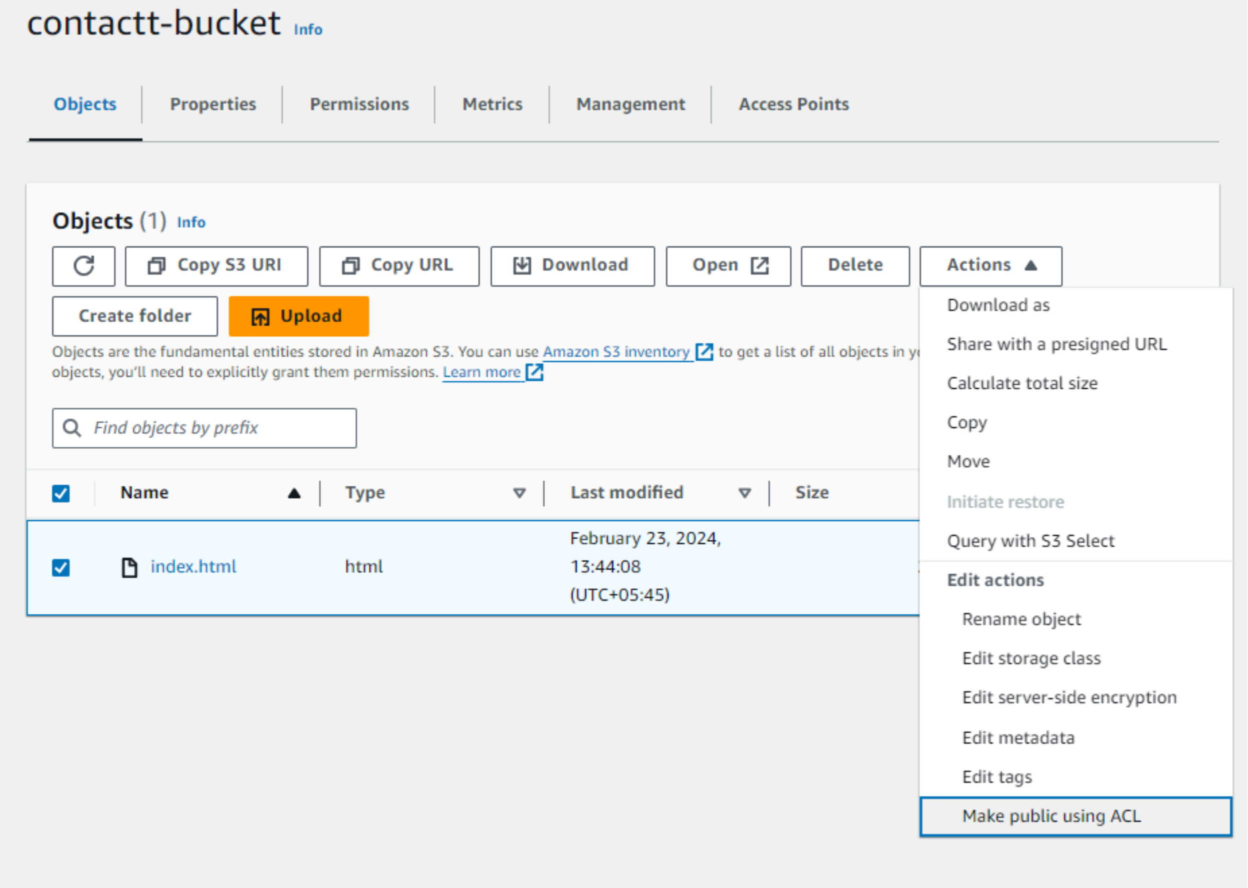
Error document - *optional*

This is returned when an error occurs.

error.html

Redirection rules – *optional*

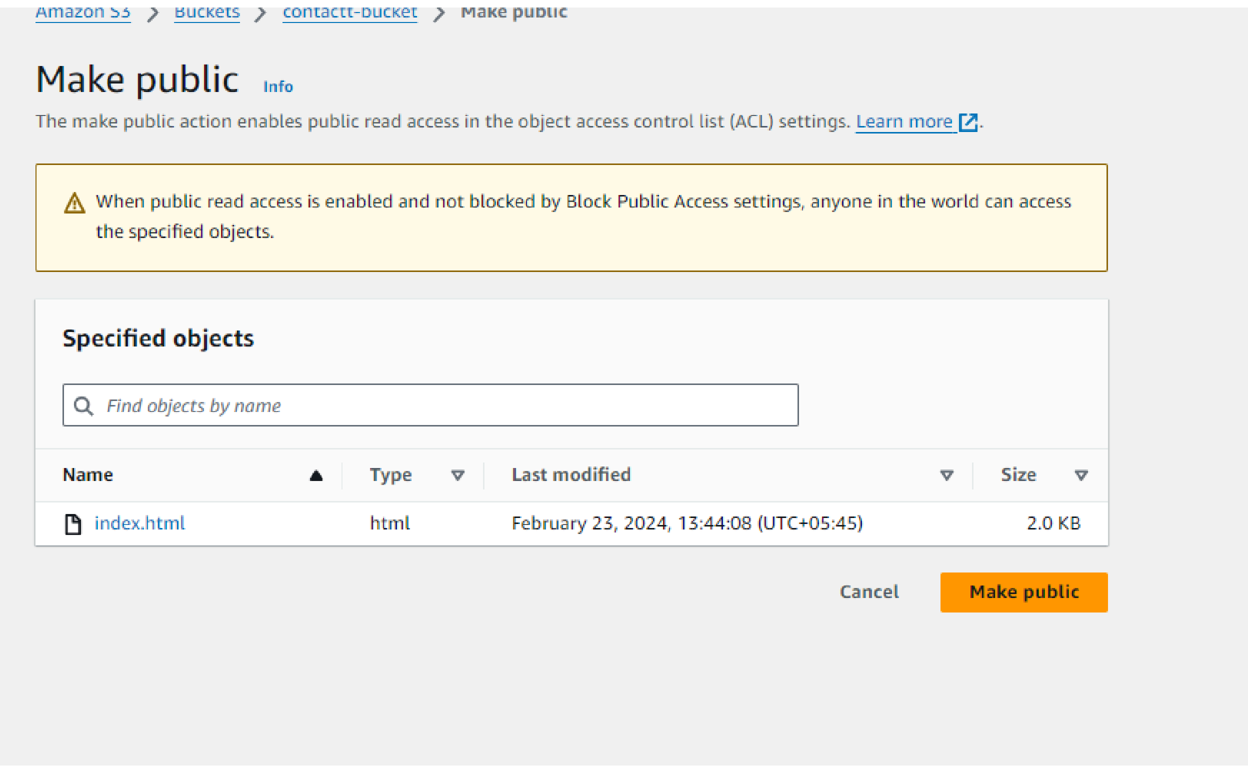
21) After that, the uploded file should give permission to make public using ACL



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

☑	Name ▲	Type ▽	Last modified ▽	Size
☑	index.html	html	February 23, 2024, 13:44:08 (UTC+05:45)	

22) Click on make public



Amazon S3 > Buckets > contactt-bucket > Make public

Make public Info

The make public action enables public read access in the object access control list (ACL) settings. Learn more.

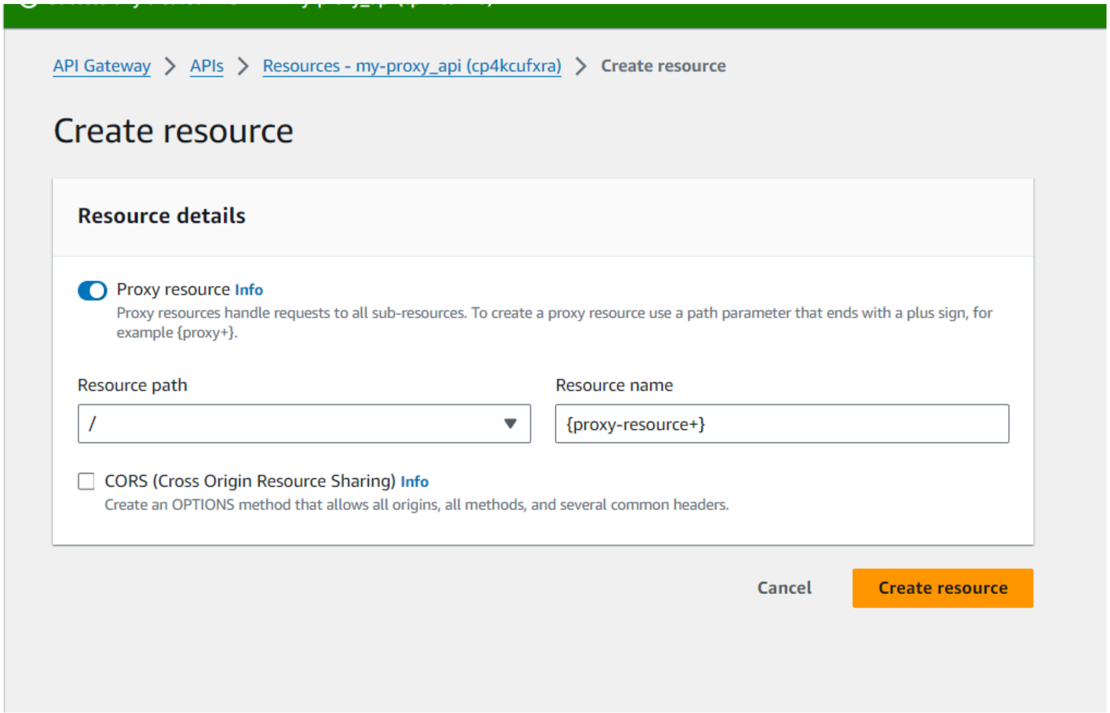
⚠ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

Specified objects

Find objects by name

Name	Type	Last modified	Size
index.html	html	February 23, 2024, 13:44:08 (UTC+05:45)	2.0 KB

Testing phases.. Allocating proxy resource (for payload data)



API Gateway > APIs > Resources - my-proxy_api (cp4kcufxra) > Create resource

Create resource

Resource details

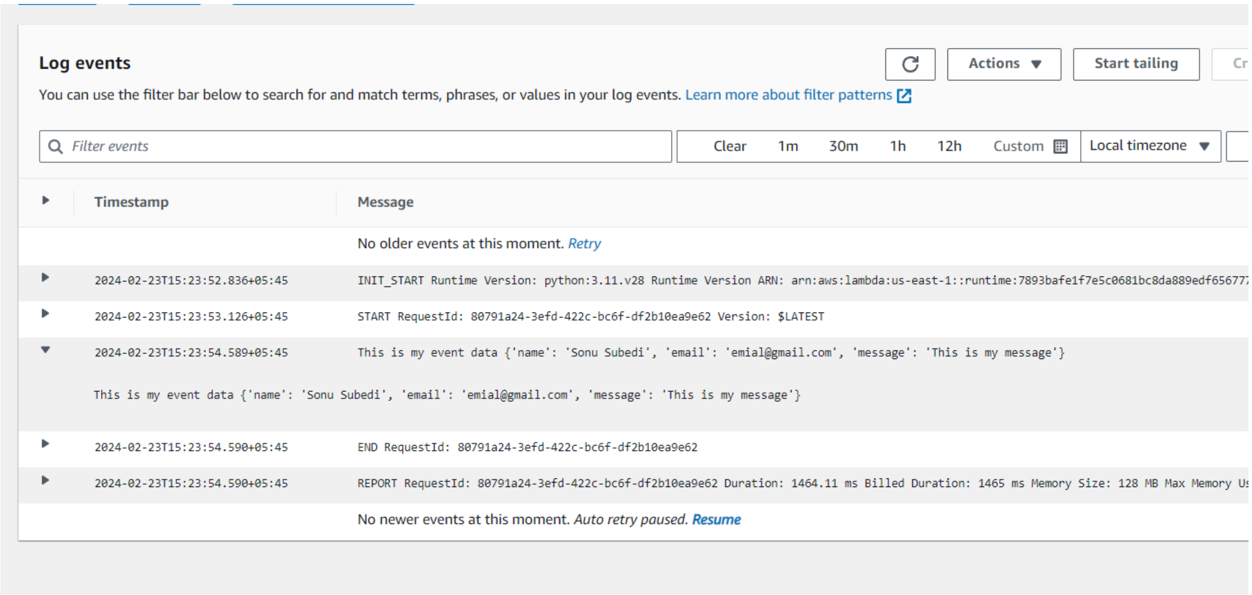
● Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

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.

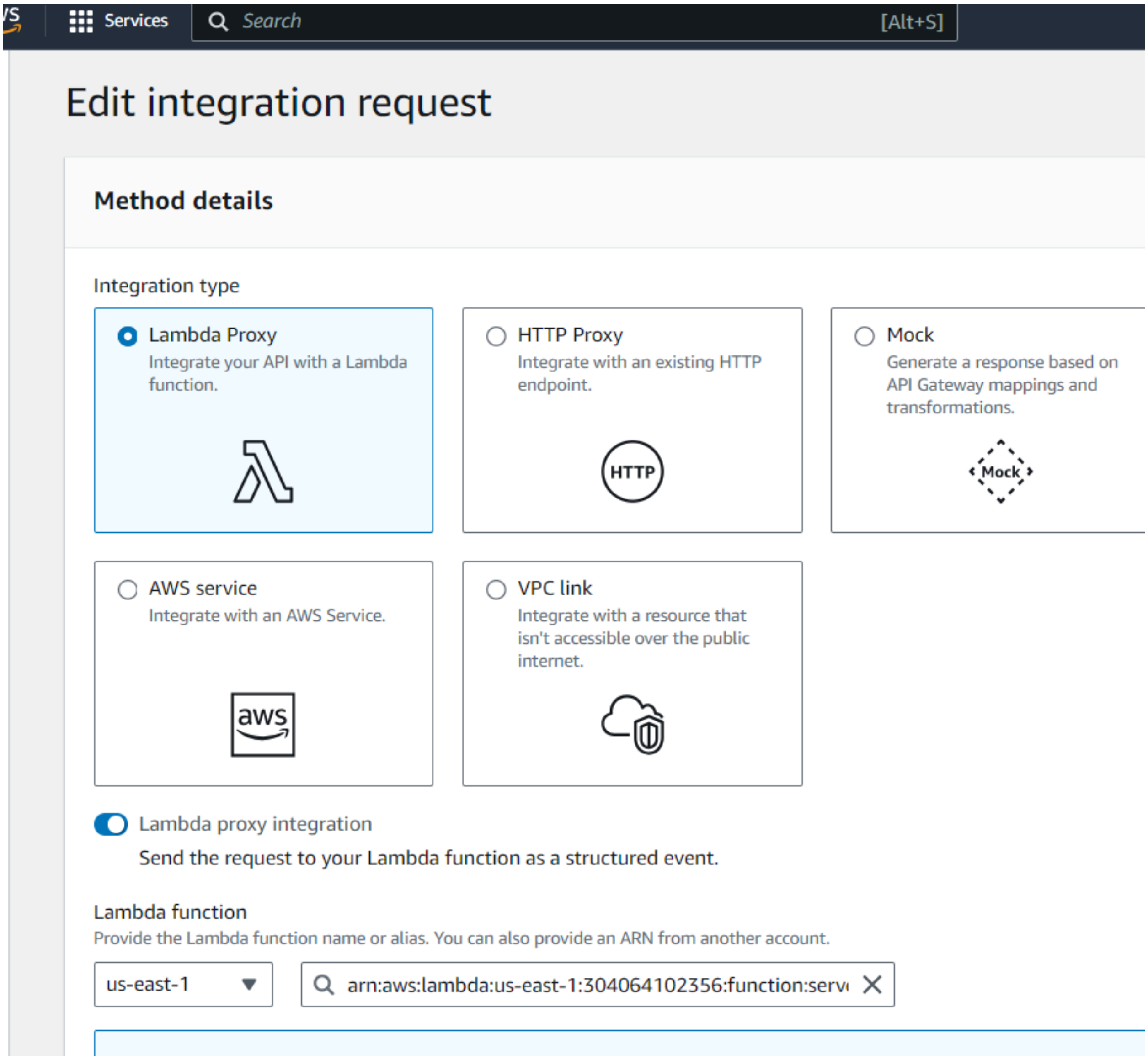
Log events in Cloudwatch can be seen as:



You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                       Message
                                No older events at this moment. Retry
2024-02-23T15:23:52.836+05:45   INIT_START Runtime Version: python:3.11.v28 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:7893bafe1f7e5c0681bc8da889edf65677
2024-02-23T15:23:53.126+05:45   START RequestId: 80791a24-3efd-422c-bc6f-df2b10ea9e62 Version: $LATEST
2024-02-23T15:23:54.589+05:45   This is my event data {'name': 'Sonu Subedi', 'email': 'emial@gmail.com', 'message': 'This is my message'}

This is my event data {'name': 'Sonu Subedi', 'email': 'emial@gmail.com', 'message': 'This is my message'}

2024-02-23T15:23:54.590+05:45   END RequestId: 80791a24-3efd-422c-bc6f-df2b10ea9e62
2024-02-23T15:23:54.590+05:45   REPORT RequestId: 80791a24-3efd-422c-bc6f-df2b10ea9e62 Duration: 1464.11 ms Billed Duration: 1465 ms Memory Size: 128 MB Max Memory Us
                                No newer events at this moment. Auto retry paused. Resume
```

Creating new Lambda Proxy method



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.

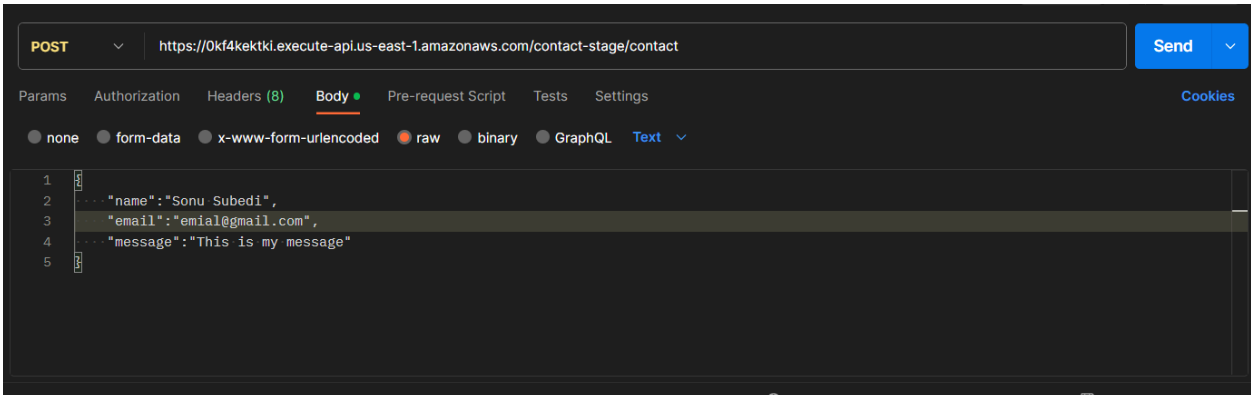
Lambda proxy integration: on
Send the request to your Lambda function as a structured event.

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-1:304064102356:function:serv

Testing in PostMan

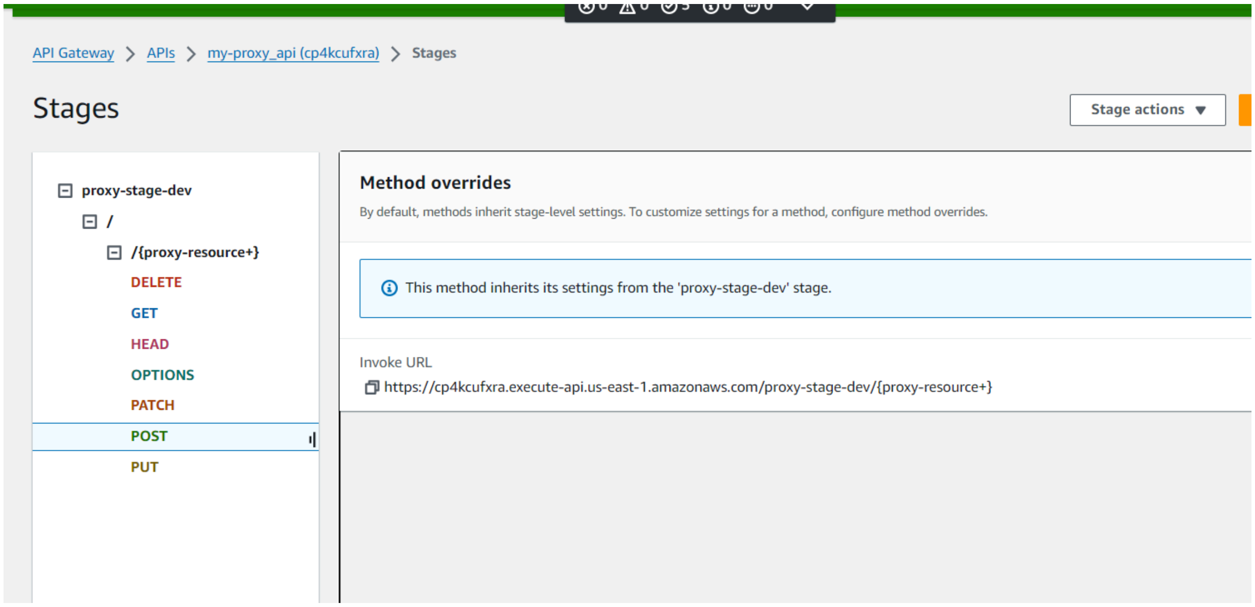


POST https://0kf4kektki.execute-api.us-east-1.amazonaws.com/contact-stage/contact

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL Text

```
{
    "name":"Sonu Subedi",
    "email":"emial@gmail.com",
    "message":"This is my message"
}
```

This are seen in Stages in API’s Gateway



API Gateway > APIs > my-proxy_api (cp4kcufxra) > Stages

Stages

Stage actions ▼

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

Method overrides

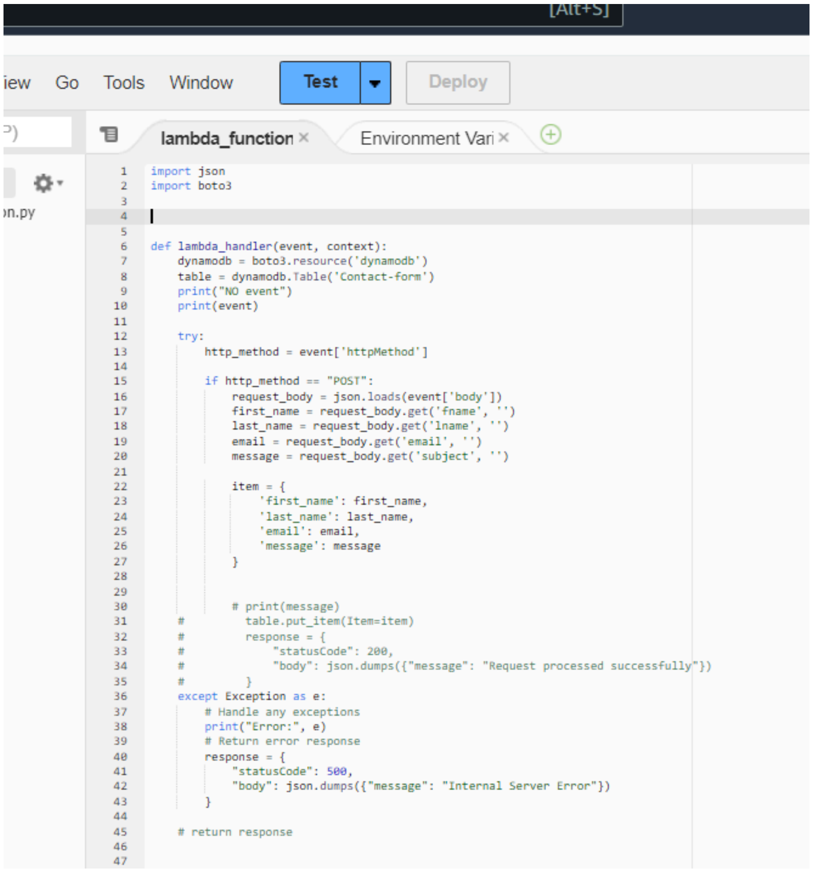
By default, methods inherit stage-level settings. To customize settings for a method, configure method overrides.

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

Invoke URL

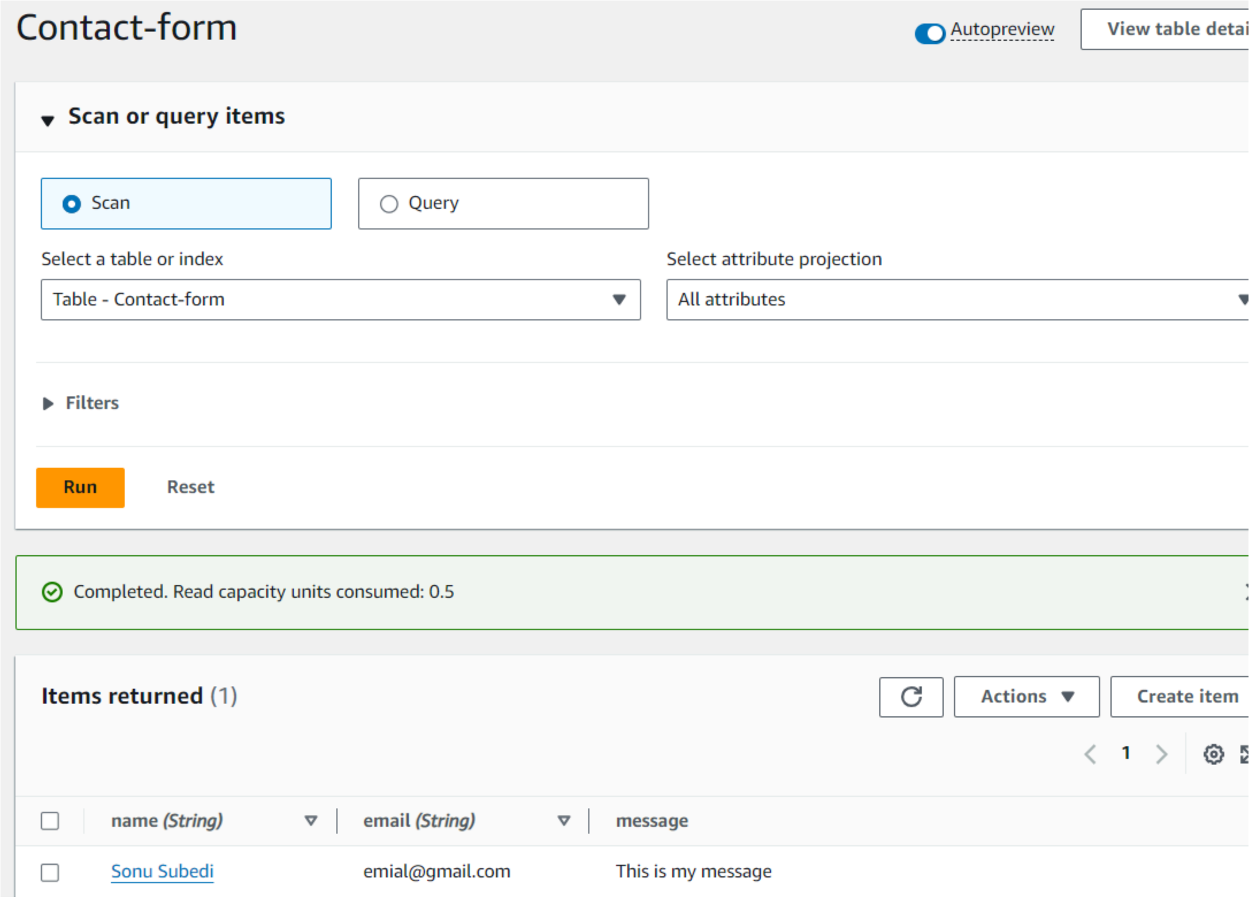
https://cp4kcufxra.execute-api.us-east-1.amazonaws.com/proxy-stage-dev/{proxy-resource+}

This is the code portion of Lambda function.



```python
import json
import boto3


def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Contact-form')
    print("NO event")
    print(event)

    try:
        http_method = event['httpMethod']

        if http_method == "POST":
            request_body = json.loads(event['body'])
            first_name = request_body.get('fname', '')
            last_name = request_body.get('lname', '')
            email = request_body.get('email', '')
            message = request_body.get('subject', '')

            item = {
                'first_name': first_name,
                'last_name': last_name,
                'email': email,
                'message': message
            }


            # print(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


```

Similarly, we can see them in DynamoDb



Select a table or index: Table - Contact-form

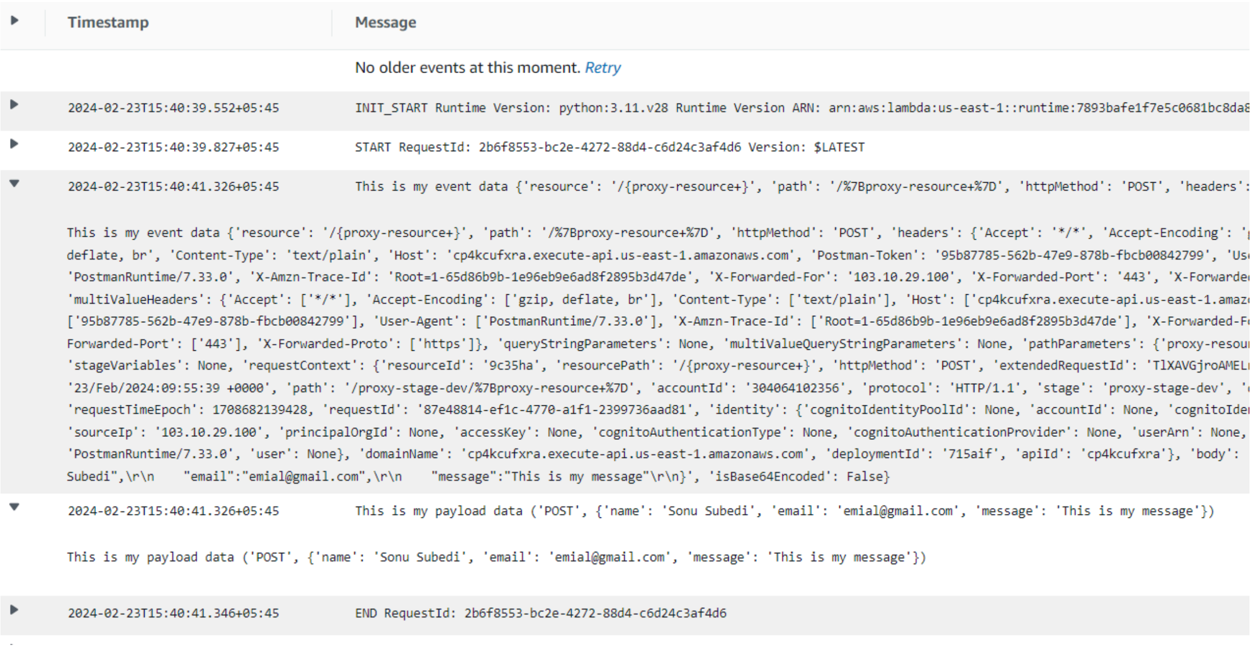
Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (1)

□	name (String)	email (String)	message
□	Sonu Subedi	emial@gmail.com	This is my message

Testing in CloudWatch



```
    Timestamp                          Message
                                       No older events at this moment. Retry
▸   2024-02-23T15:40:39.552+05:45      INIT_START Runtime Version: python:3.11.v28 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:7893bafe1f7e5c0681bc8da
▸   2024-02-23T15:40:39.827+05:45      START RequestId: 2b6f8553-bc2e-4272-88d4-c6d24c3af4d6 Version: $LATEST
▾   2024-02-23T15:40:41.326+05:45      This is my event data {'resource': '/{proxy-resource+}', 'path': '/%7Bproxy-resource+%7D', 'httpMethod': 'POST', 'headers':

    This is my event data {'resource': '/{proxy-resource+}', 'path': '/%7Bproxy-resource+%7D', 'httpMethod': 'POST', 'headers': {'Accept': '*/*', 'Accept-Encoding': '
    deflate, br', 'Content-Type': 'text/plain', 'Host': 'cp4kcufxra.execute-api.us-east-1.amazonaws.com', 'Postman-Token': '95b87785-562b-47e9-878b-fbcb00842799', 'Us
    'PostmanRuntime/7.33.0', 'X-Amzn-Trace-Id': 'Root=1-65d86b9b-1e96eb9e6ad8f2895b3d47de', 'X-Forwarded-For': '103.10.29.100', 'X-Forwarded-Port': '443', 'X-Forwarde
    'multiValueHeaders': {'Accept': ['*/*'], 'Accept-Encoding': ['gzip, deflate, br'], 'Content-Type': ['text/plain'], 'Host': ['cp4kcufxra.execute-api.us-east-1.amaz
    ['95b87785-562b-47e9-878b-fbcb00842799'], 'User-Agent': ['PostmanRuntime/7.33.0'], 'X-Amzn-Trace-Id': ['Root=1-65d86b9b-1e96eb9e6ad8f2895b3d47de'], 'X-Forwarded-F
    Forwarded-Port': ['443'], 'X-Forwarded-Proto': ['https']}, 'queryStringParameters': None, 'multiValueQueryStringParameters': None, 'pathParameters': {'proxy-resou
    'stageVariables': None, 'requestContext': {'resourceId': '9c35ha', 'resourcePath': '/{proxy-resource+}', 'httpMethod': 'POST', 'extendedRequestId': 'TlXAVGjroAMEL
    '23/Feb/2024:09:55:39 +0000', 'path': '/proxy-stage-dev/%7Bproxy-resource+%7D', 'accountId': '304064102356', 'protocol': 'HTTP/1.1', 'stage': 'proxy-stage-dev', '
    'requestTimeEpoch': 1708682139428, 'requestId': '87e48814-ef1c-4770-a1f1-2399736aad81', 'identity': {'cognitoIdentityPoolId': None, 'accountId': None, 'cognitoIde
    'sourceIp': '103.10.29.100', 'principalOrgId': None, 'accessKey': None, 'cognitoAuthenticationType': None, 'cognitoAuthenticationProvider': None, 'userArn': None,
    'PostmanRuntime/7.33.0', 'user': None}, 'domainName': 'cp4kcufxra.execute-api.us-east-1.amazonaws.com', 'deploymentId': '715aif', 'apiId': 'cp4kcufxra'}, 'body':
    Subedi",\r\n    "email":"emial@gmail.com",\r\n    "message":"This is my message"\r\n}', 'isBase64Encoded': False}
▾   2024-02-23T15:40:41.326+05:45      This is my payload data ('POST', {'name': 'Sonu Subedi', 'email': 'emial@gmail.com', 'message': 'This is my message'})

    This is my payload data ('POST', {'name': 'Sonu Subedi', 'email': 'emial@gmail.com', 'message': 'This is my message'})

▸   2024-02-23T15:40:41.346+05:45      END RequestId: 2b6f8553-bc2e-4272-88d4-c6d24c3af4d6
```