Building a Serverless Web Application

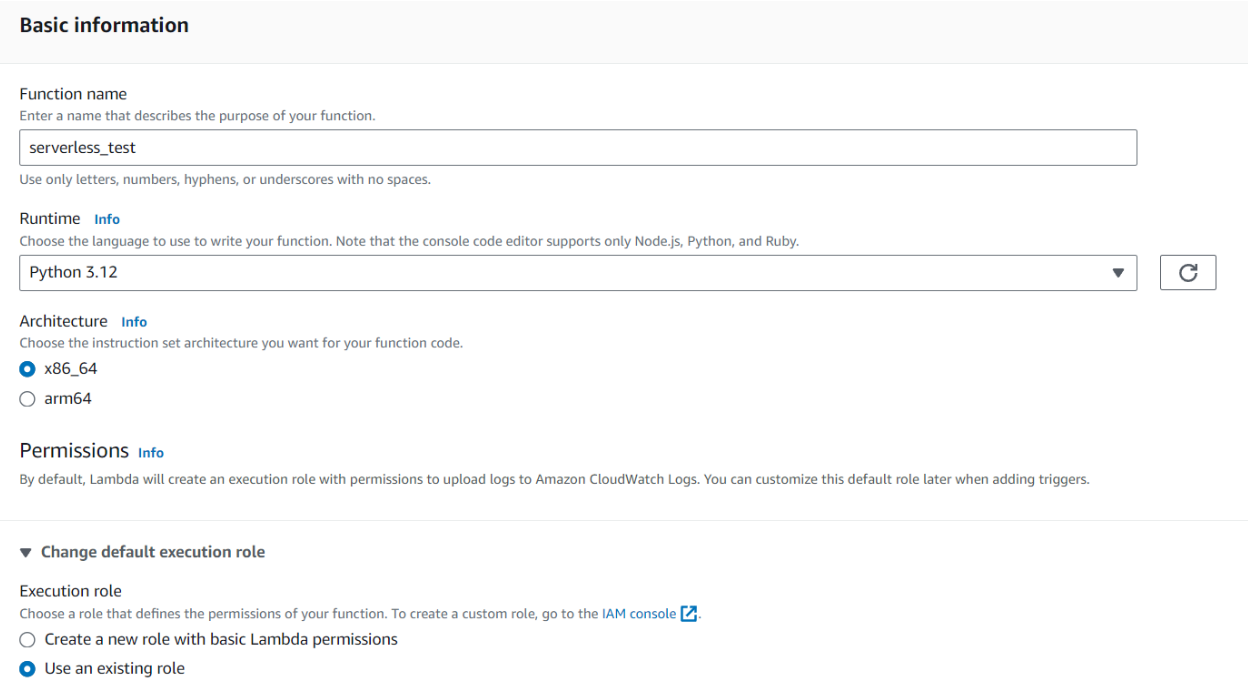
Objective: Create a serverless web application using AWS Lambda, API Gateway, S3, and DynamoDB.

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 of all, we have to create a function by selecting Lambda from services.



Basic information

Function name
Enter a name that describes the purpose of your function.
serverless_test
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

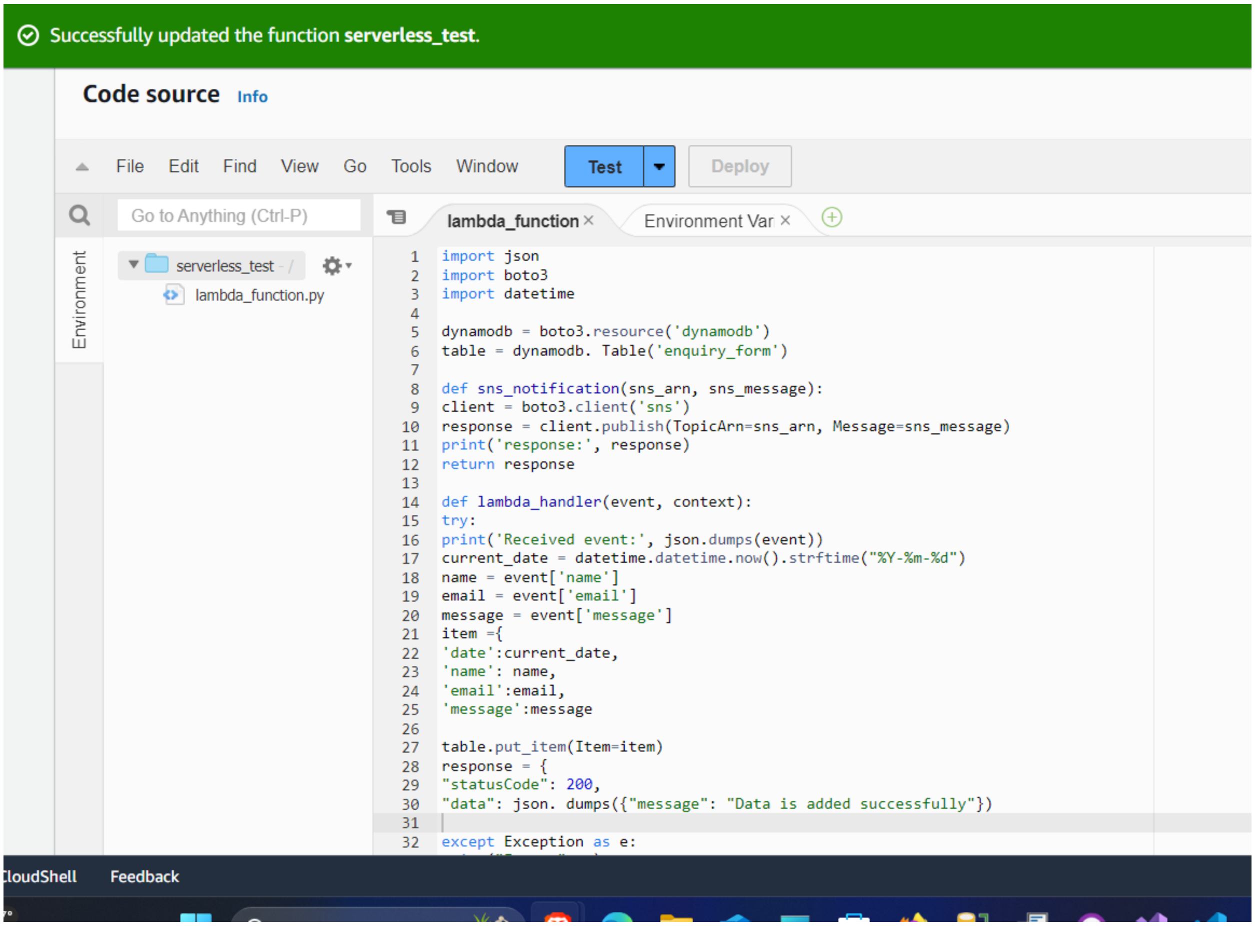
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

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

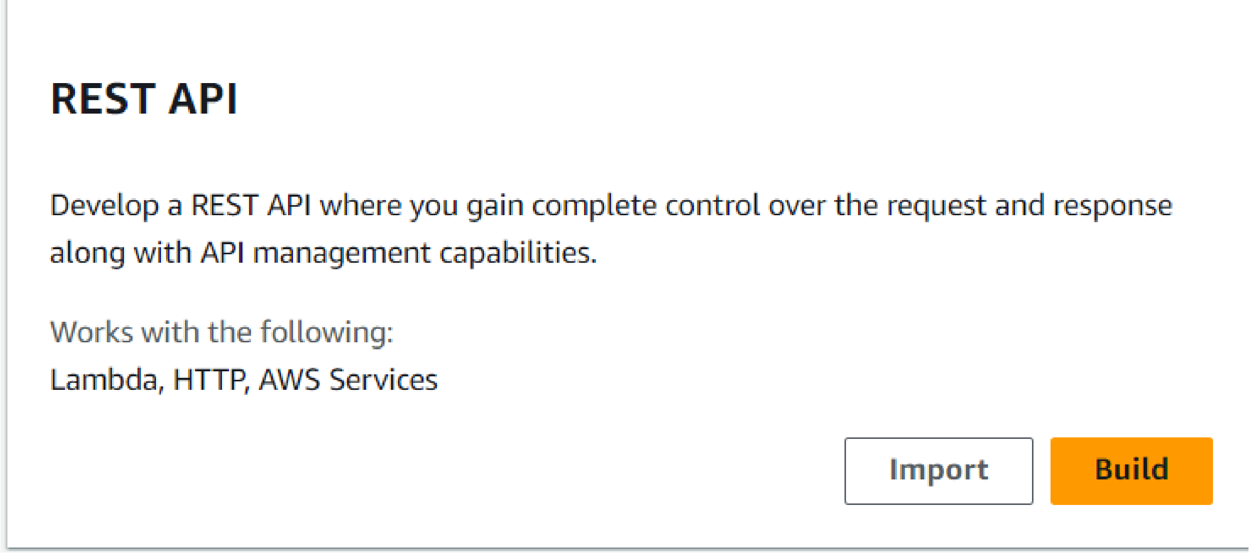
2. Then use a code and deploy that.



Successfully updated the function **serverless_test**.

```
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:
```

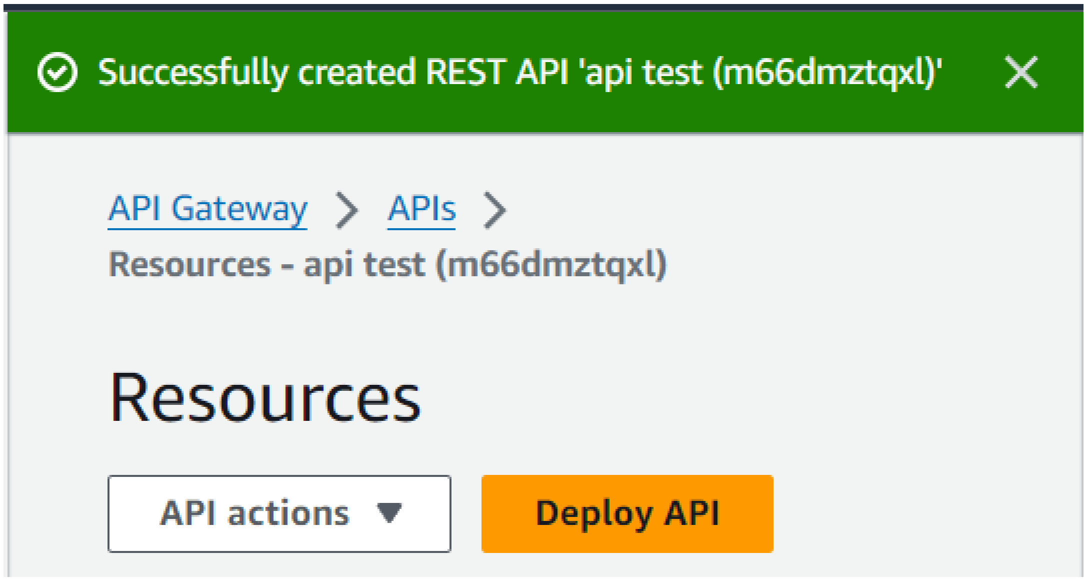
3. Go to API gateway and build and new REST API



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

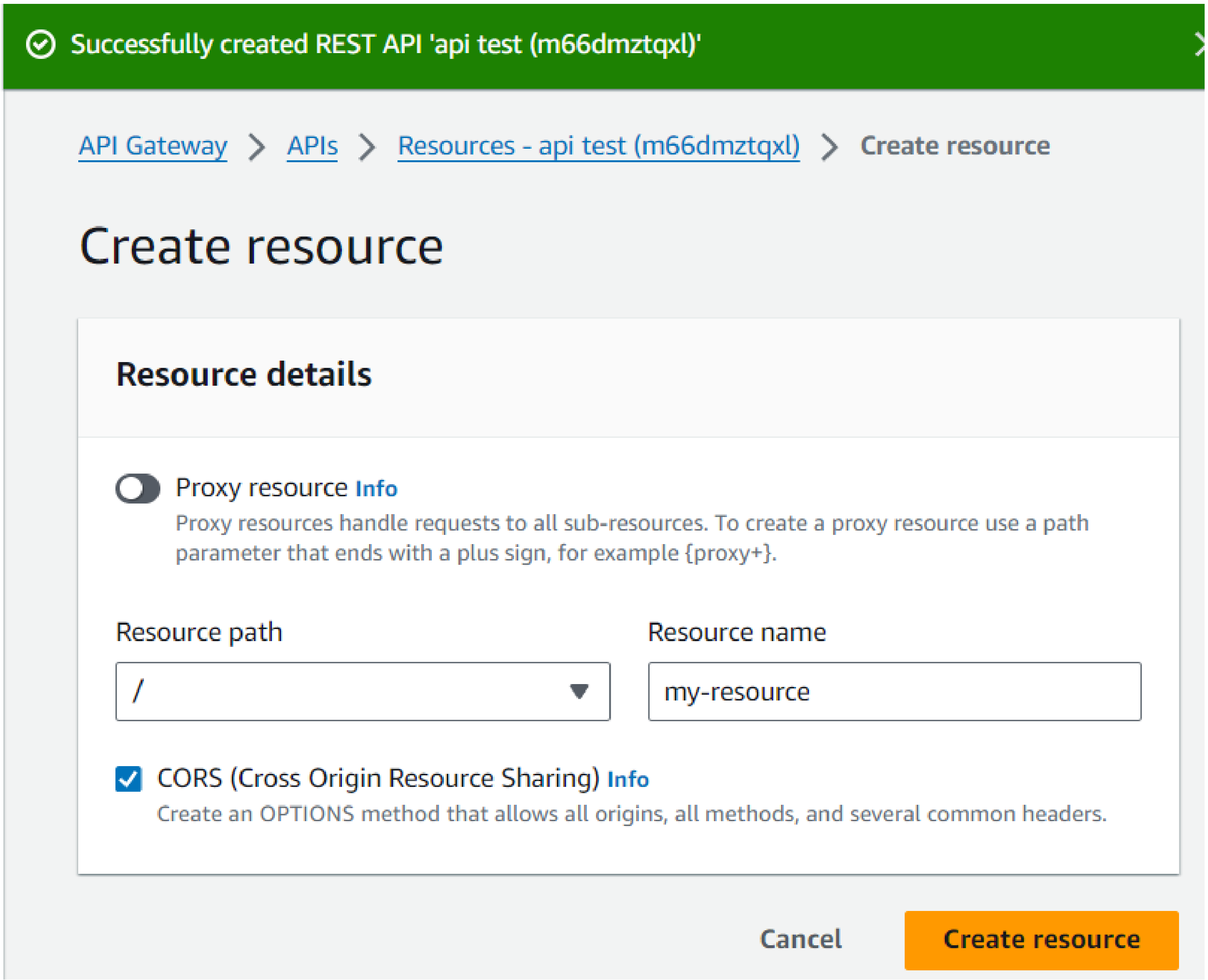


Successfully created REST API 'api test (m66dmztqxl)'

API Gateway > APIs >
Resources - api test (m66dmztqxl)

Resources

4. Now create a new resource for the apis



Successfully created REST API 'api test (m66dmztqxl)'

API Gateway > APIs > Resources - api test (m66dmztqxl) > 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+}.

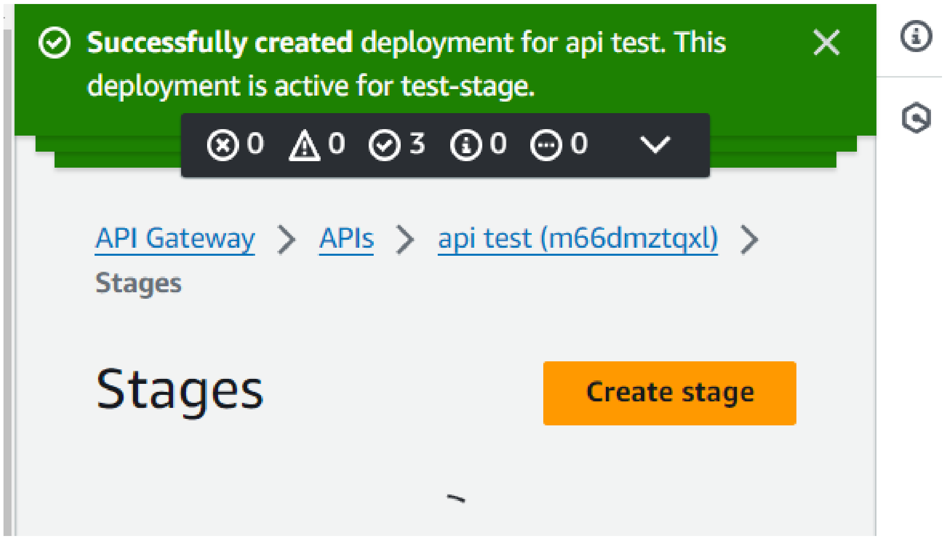
Resource path: /

Resource name: my-resource

☑ CORS (Cross Origin Resource Sharing) Info

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

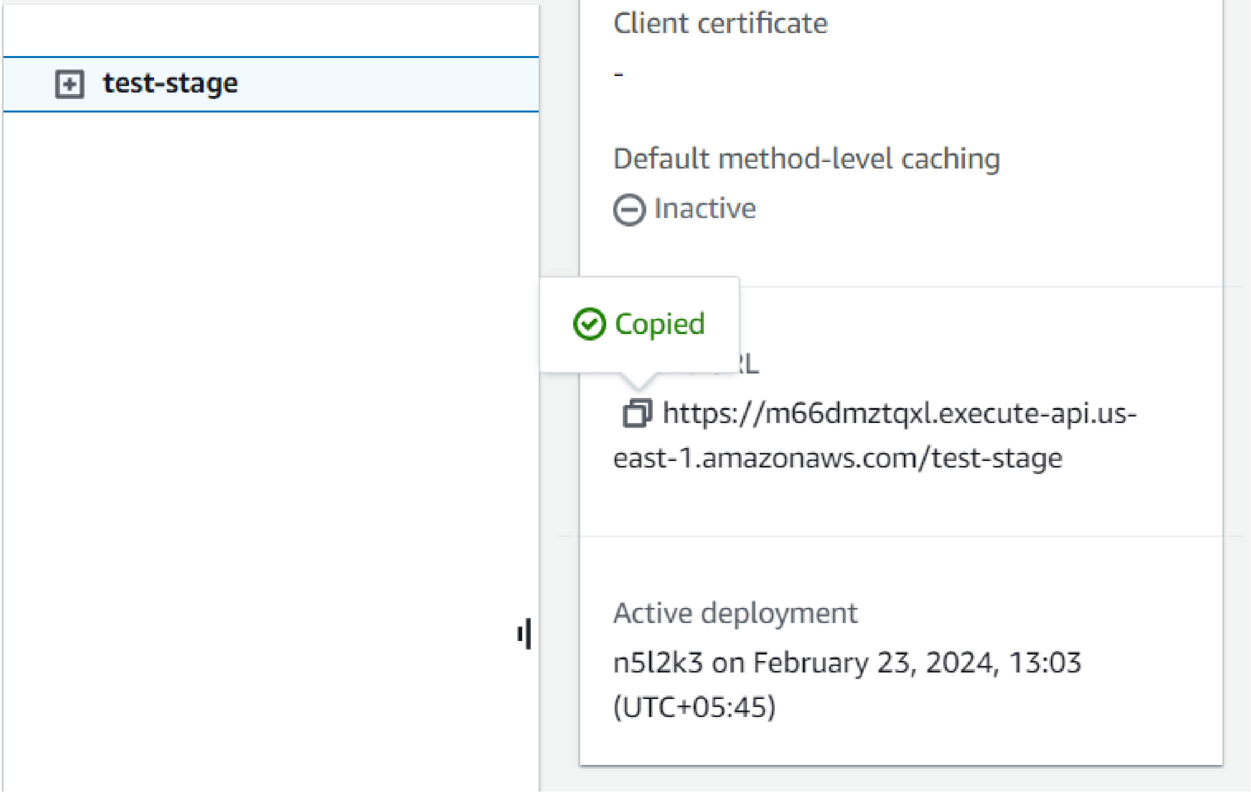
5. Then create a new stage too.



Successfully created deployment for api test. This deployment is active for test-stage.

API Gateway > APIs > api test (m66dmztqxl) > Stages

Stages



test-stage

Client certificate

-

Default method-level caching

⊖ Inactive

✓ Copied

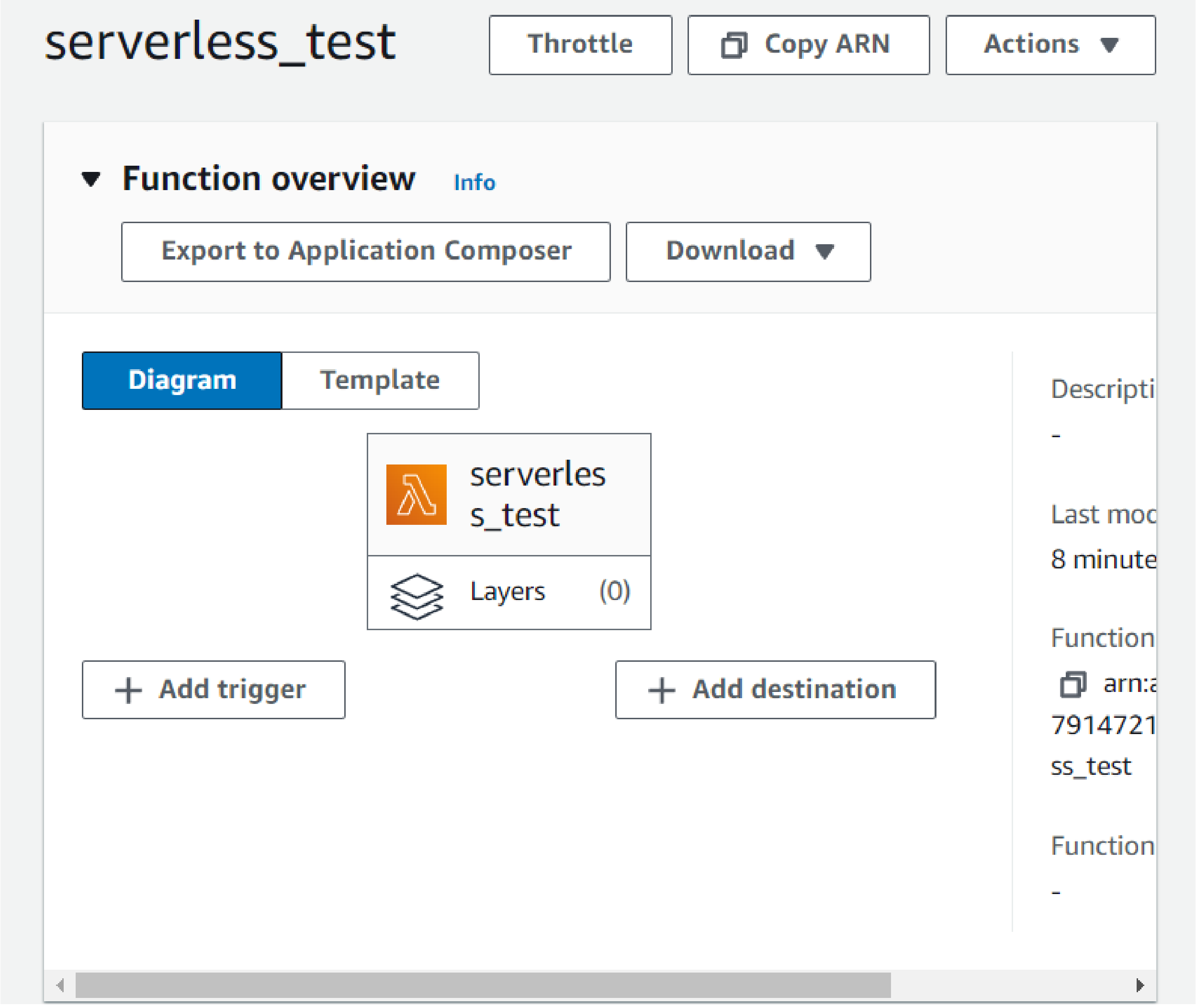
L

https://m66dmztqxl.execute-api.us-east-1.amazonaws.com/test-stage

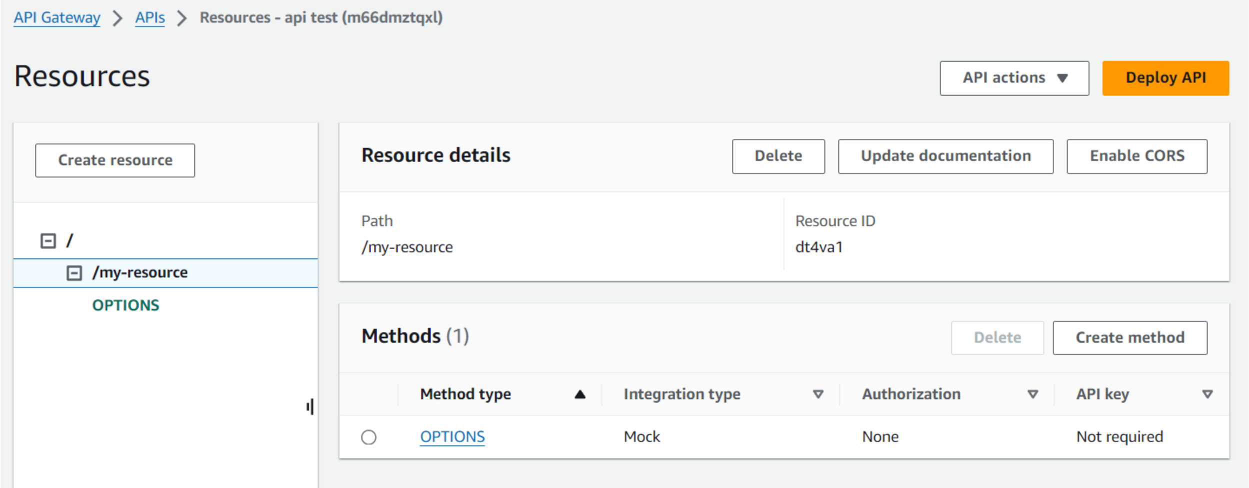
Active deployment

n5l2k3 on February 23, 2024, 13:03 (UTC+05:45)

6. Api gateway is added to the lambda function



7. Go to resources of your APIs and then also create a new post method

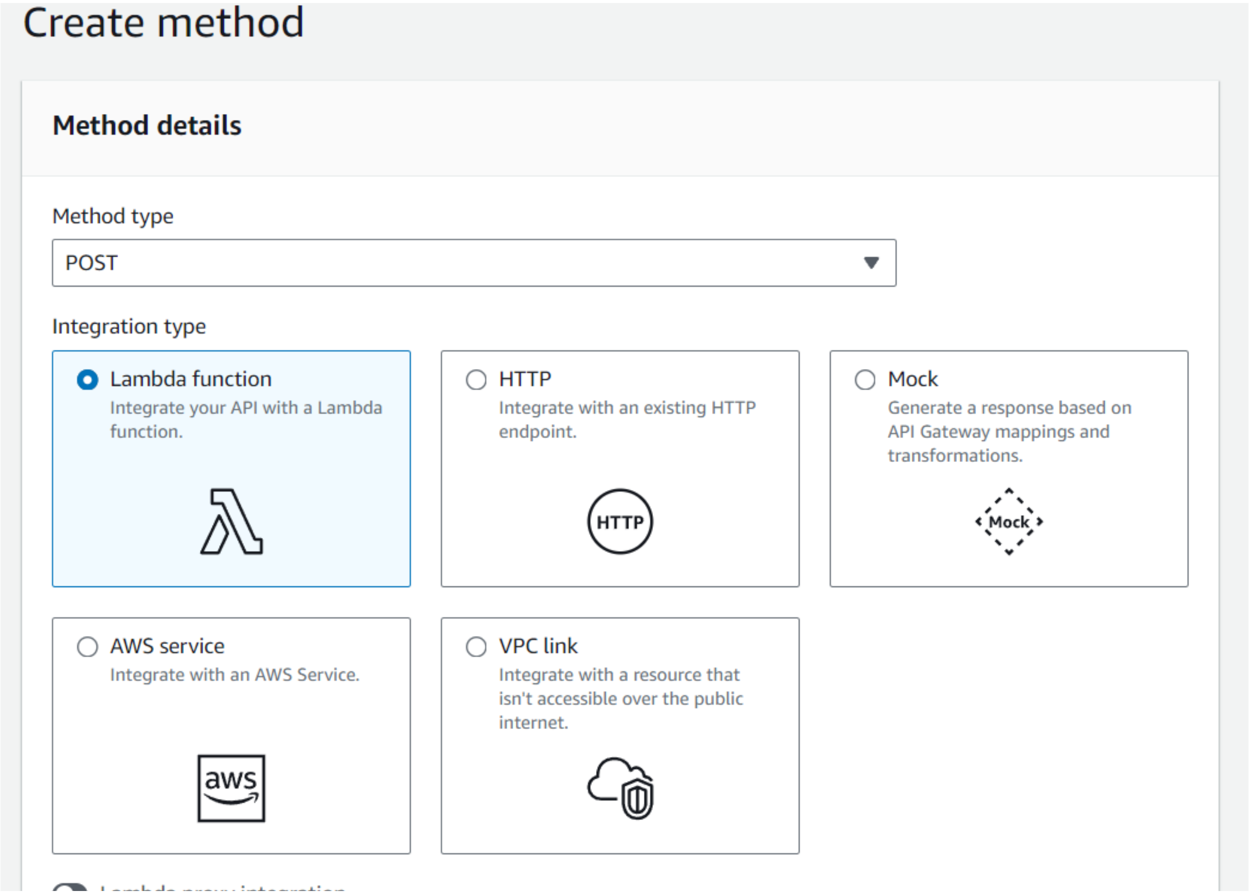


Path
/my-resource

Resource ID
dt4va1

	Method type ▲	Integration type ▽	Authorization ▽	API key ▽
○	OPTIONS	Mock	None	Not required

8. Creating a new post method.



Create method

Method details

Method type

POST

Integration type

● Lambda function
Integrate your API with a Lambda function.

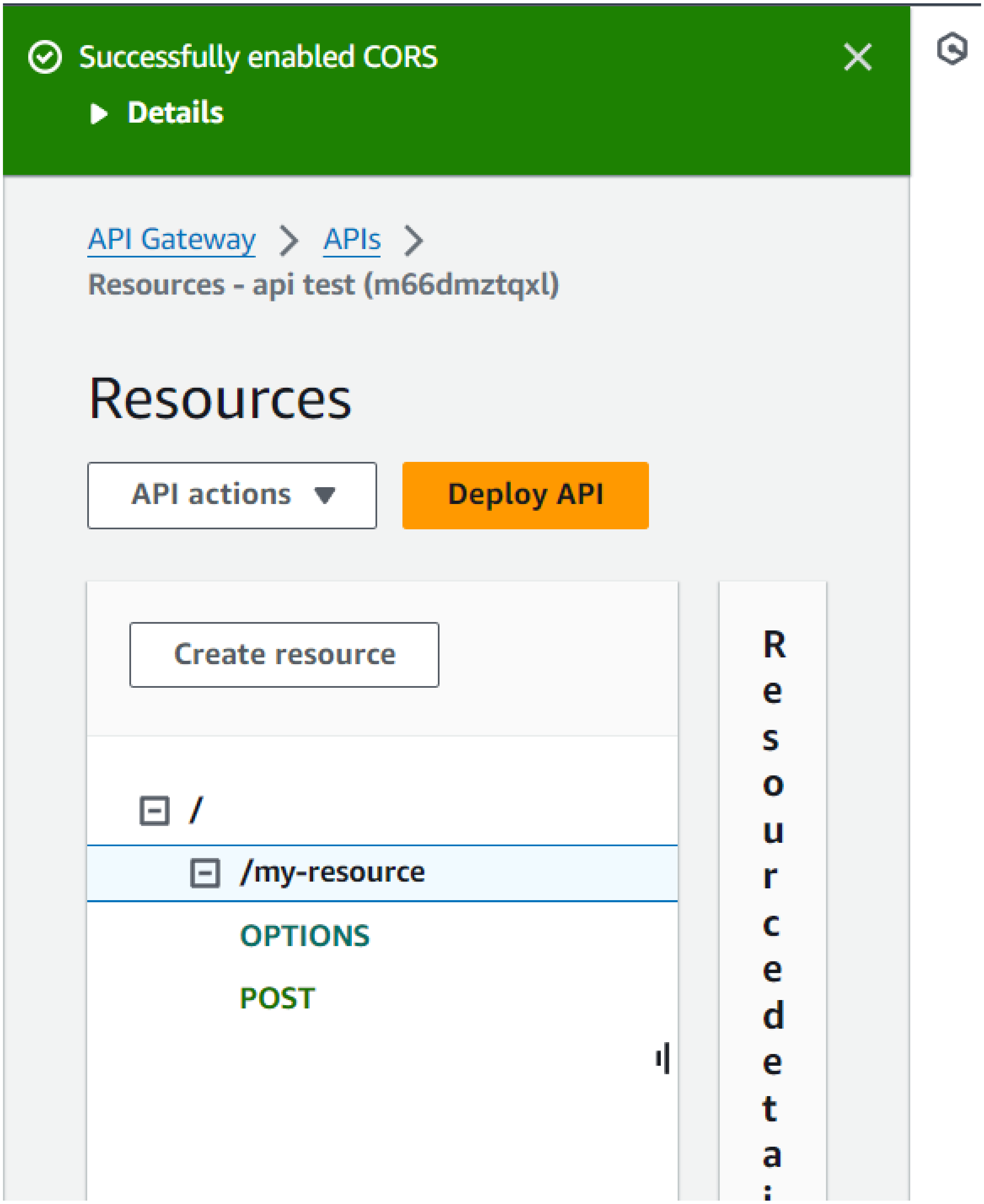
○ HTTP
Integrate with an existing HTTP endpoint.

○ 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.

9. Select enable cors option and after successfully creating that we get screen like this.

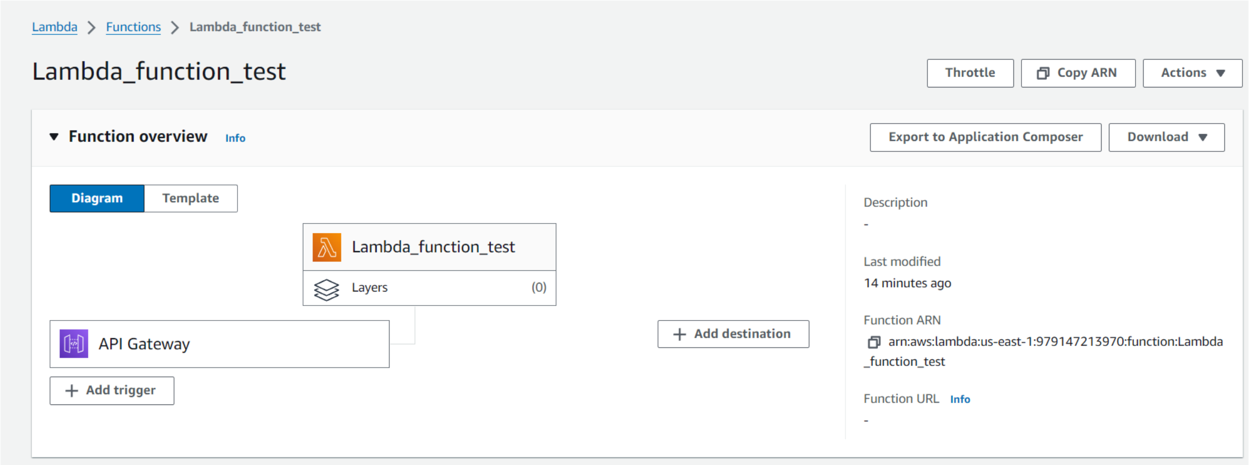


Successfully enabled CORS

API Gateway > APIs > Resources - api test (m66dmztqxl)

Resources

10.You can see that the lambda function is triggered now with api test.



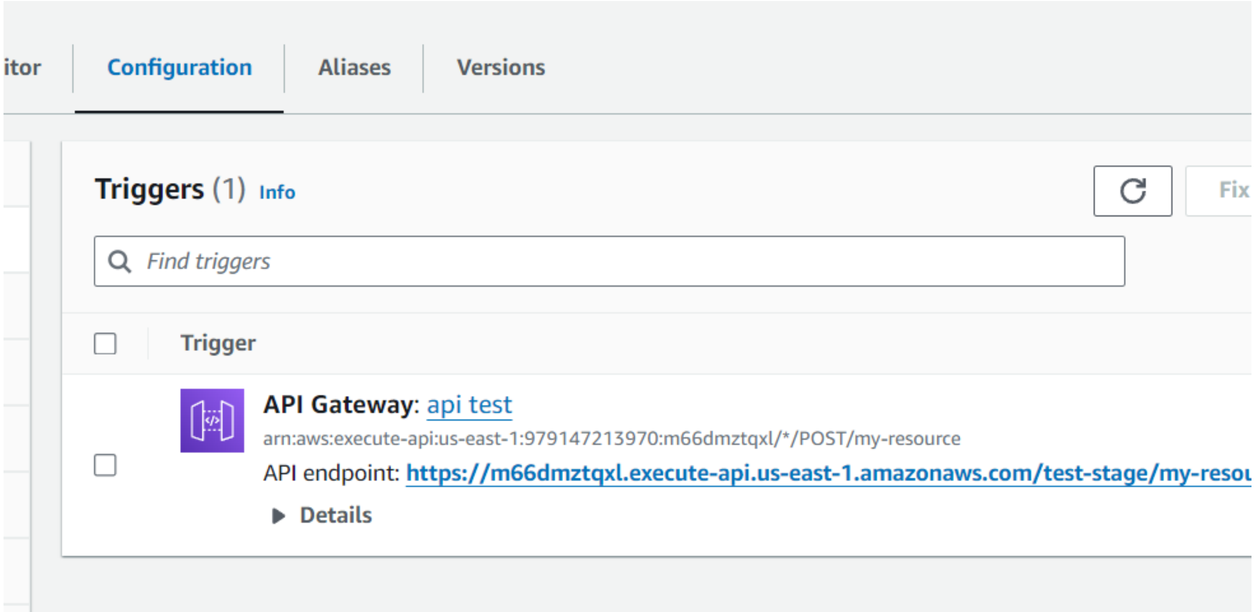
Lambda_function_test

Description
-

Last modified
14 minutes ago

Function ARN
arn:aws:lambda:us-east-1:979147213970:function:Lambda_function_test

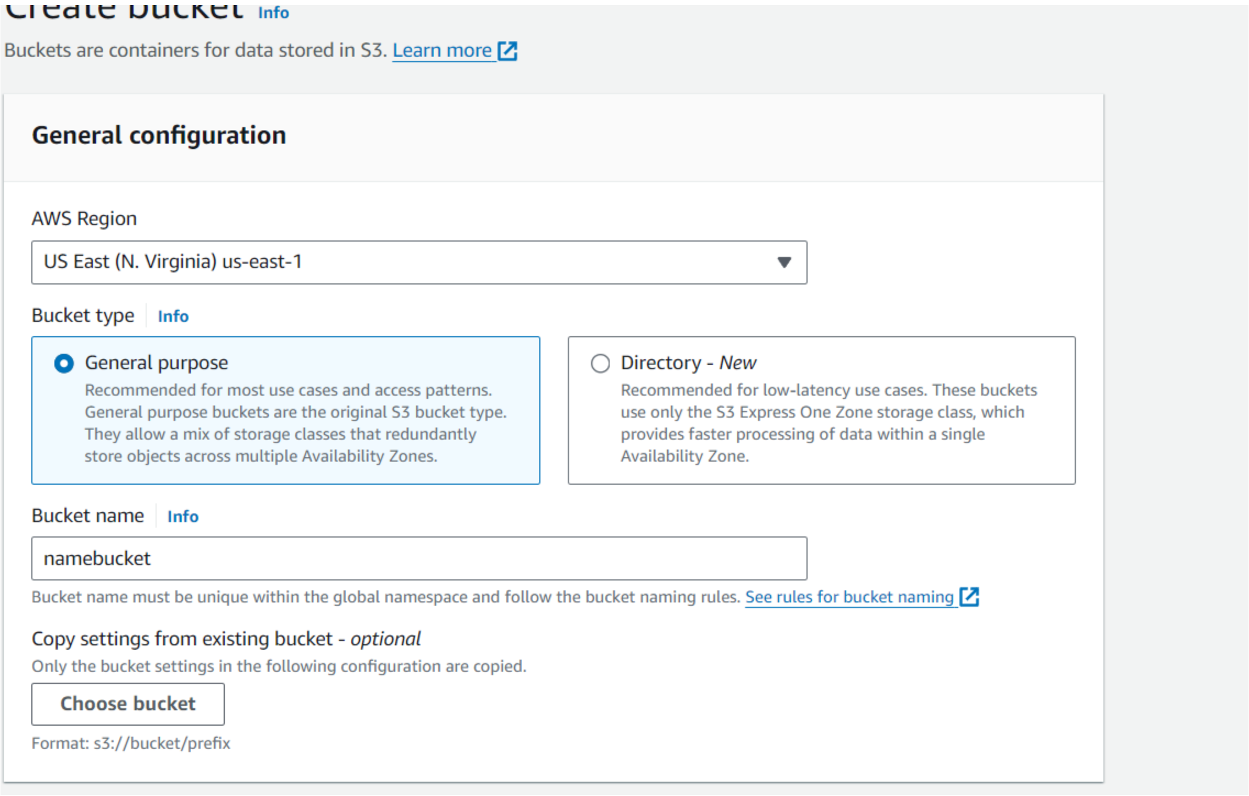
Function URL Info
-



Triggers (1)

□ API Gateway: api test
arn:aws:execute-api:us-east-1:979147213970:m66dmztqxl/*/POST/my-resource
API endpoint: https://m66dmztqxl.execute-api.us-east-1.amazonaws.com/test-stage/my-resou

10. Create a new s3 bucket and configure it accordingly.



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

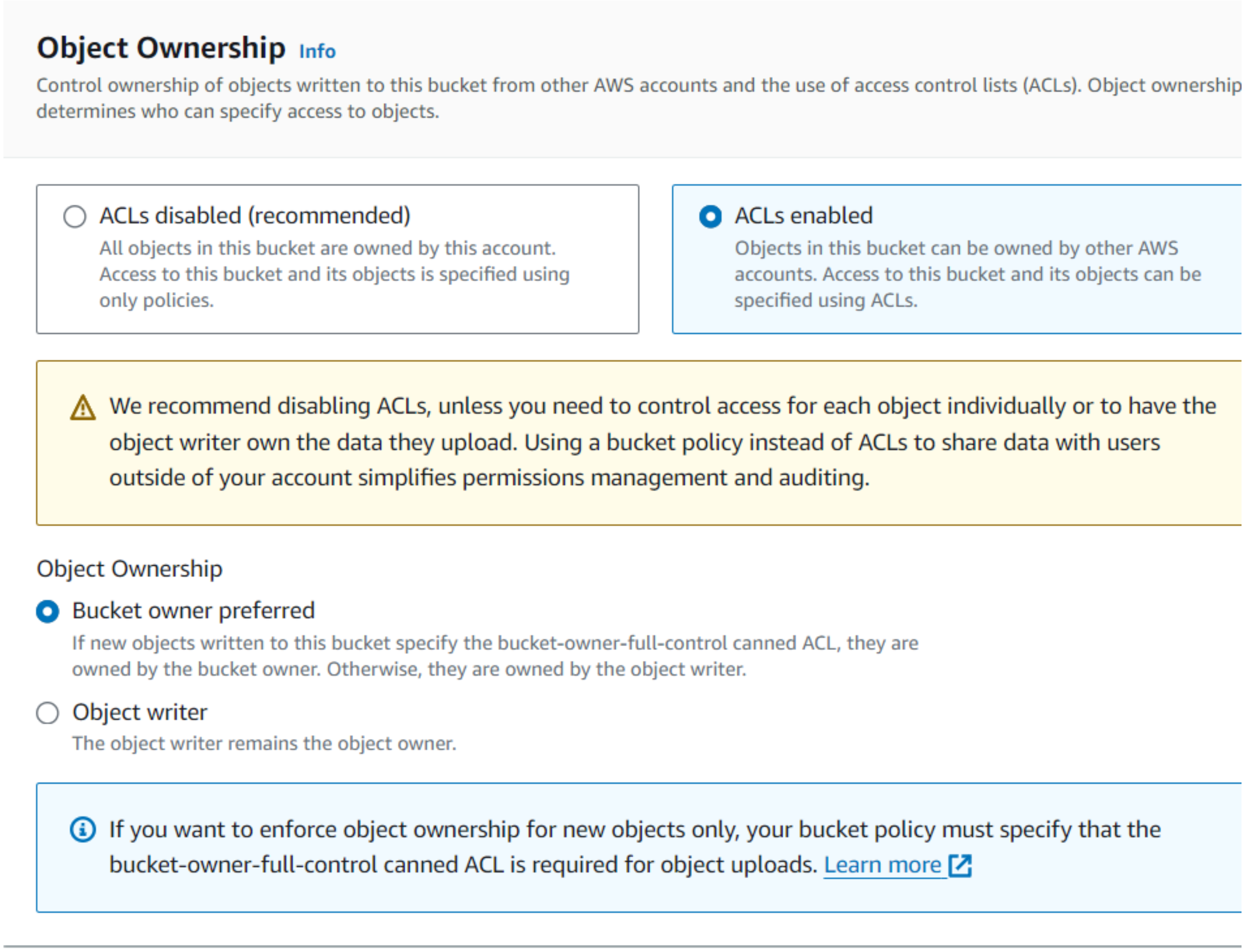
namebucket

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.

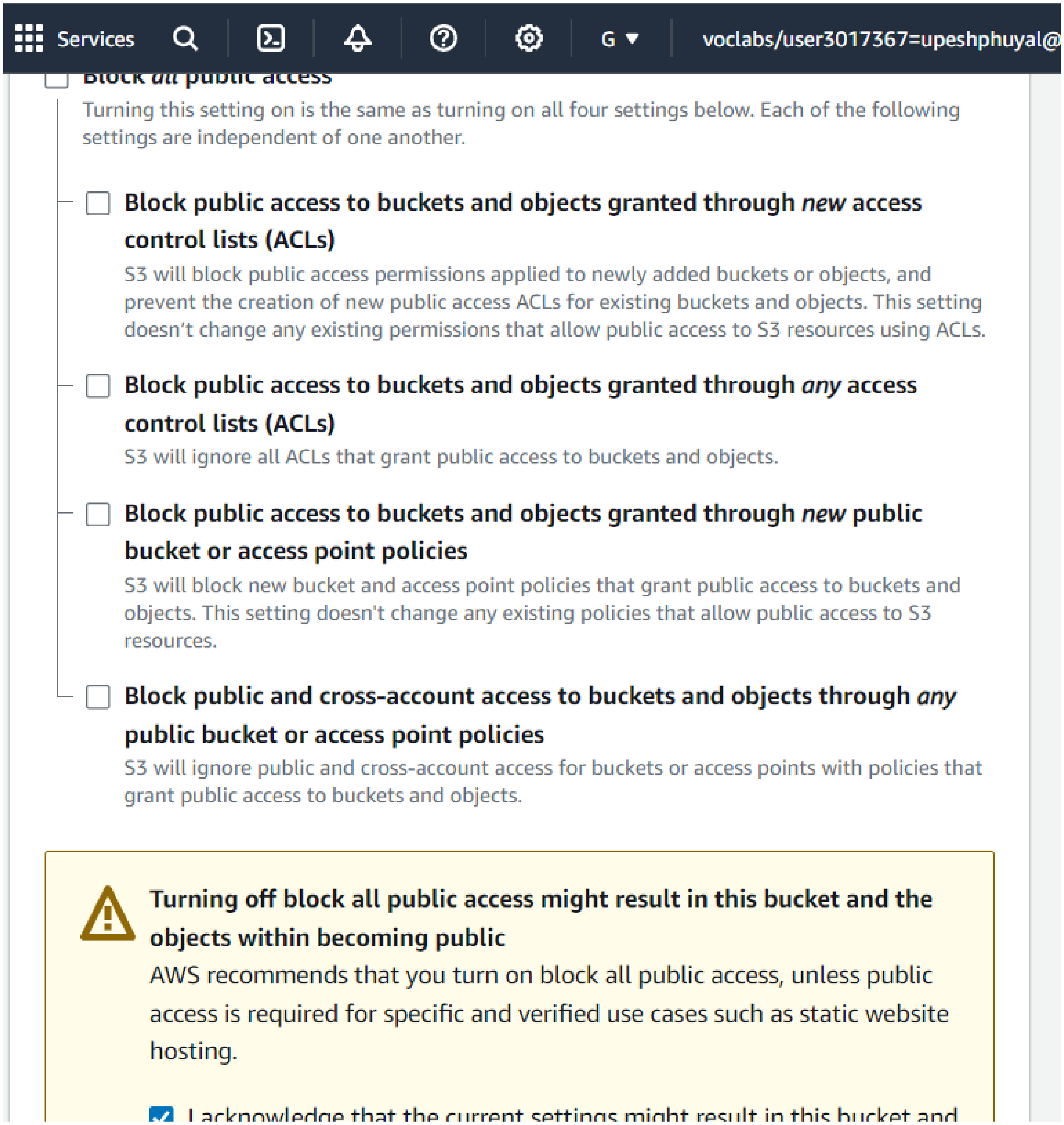
⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

● Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

○ Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. 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.

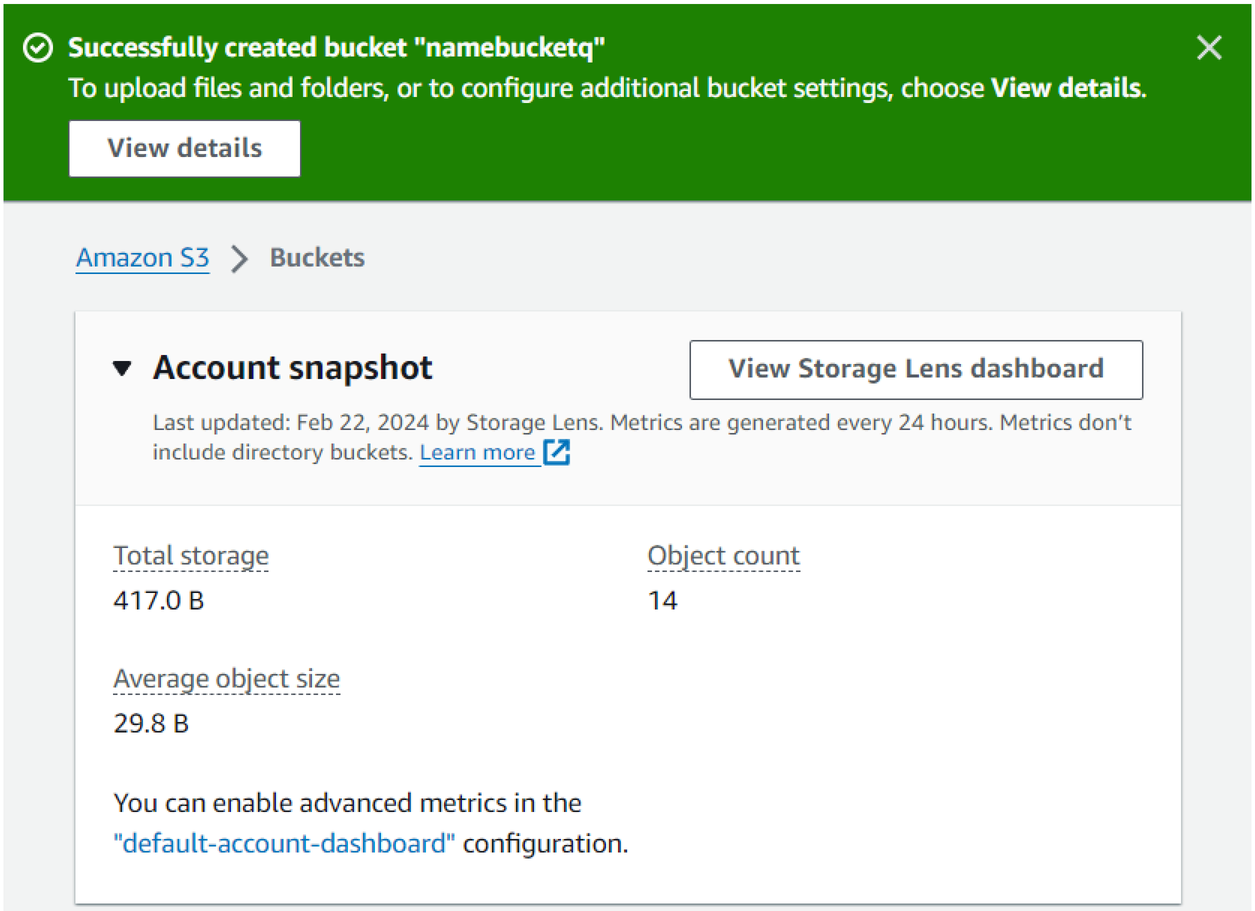
□ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

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



Successfully created bucket "namebucketq"

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

Account snapshot

Last updated: Feb 22, 2024 by Storage Lens. Metrics are generated every 24 hours. Metrics don't include directory buckets. Learn more

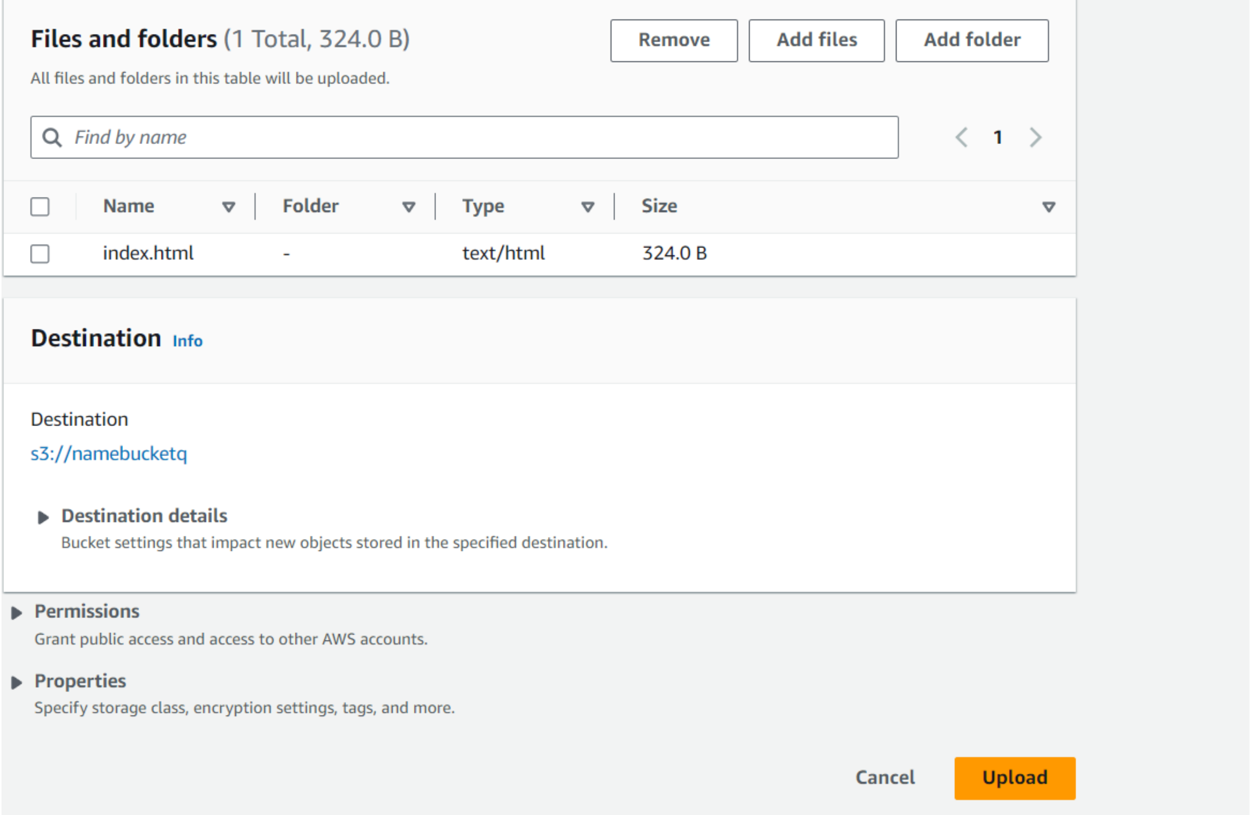
Total storage
417.0 B

Object count
14

Average object size
29.8 B

You can enable advanced metrics in the "default-account-dashboard" configuration.

11. Upload the html file



Files and folders (1 Total, 324.0 B)

All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size
□	index.html	-	text/html	324.0 B

Destination

Destination

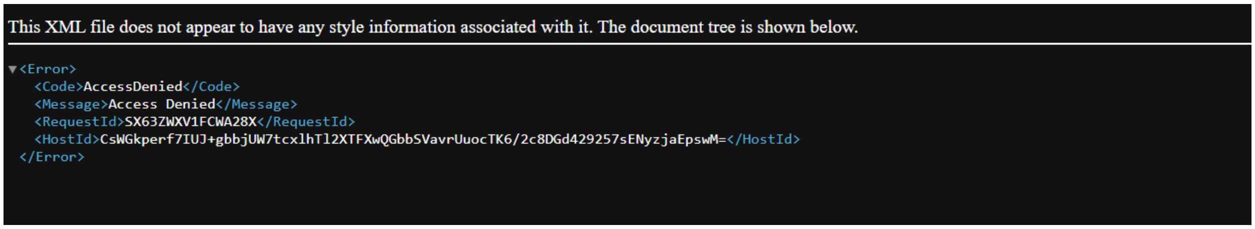
s3://namebucketq

▶ **Destination details**
Bucket settings that impact new objects stored in the specified destination.

▶ **Permissions**
Grant public access and access to other AWS accounts.

▶ **Properties**
Specify storage class, encryption settings, tags, and more.

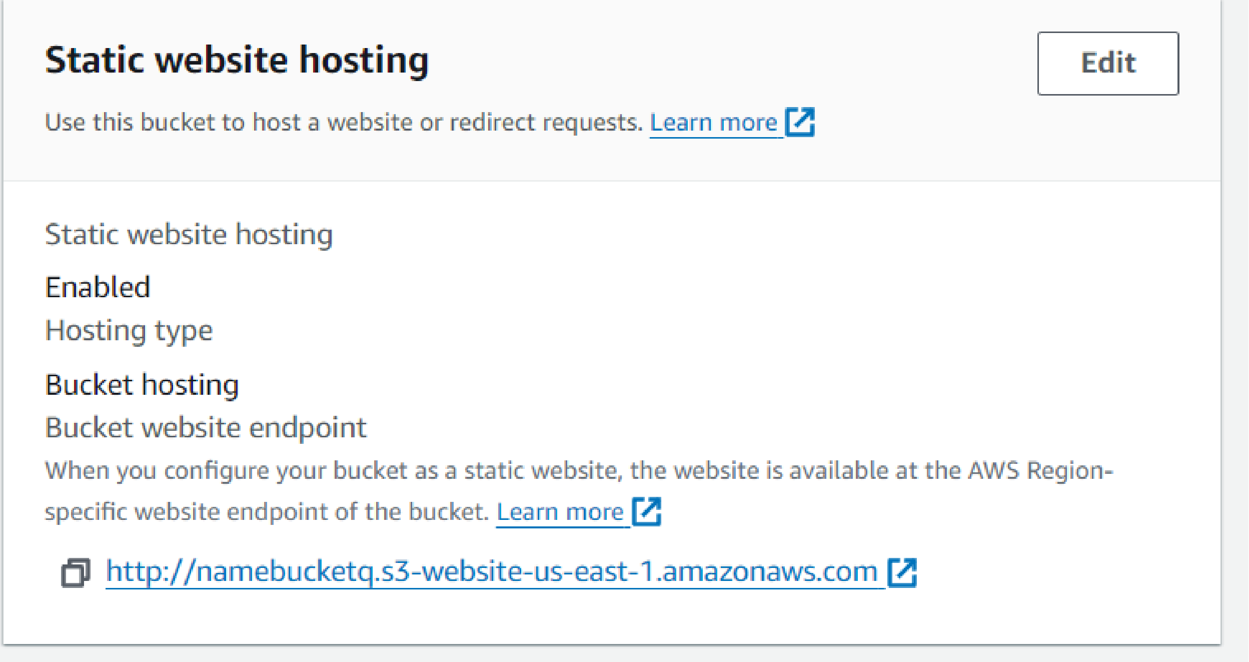
12. Since static web hosting is off it didn’t work .



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>SX63ZWXV1FCWA28X</RequestId>
   <HostId>CsWGkperf7IUJ+gbbjUW7tcxlhTl2XTFXwQGbbSVavrUuocTK6/2c8DGd429257sENyzjaEpswM=</HostId>
 </Error>
```

13. After this we turn on the static hosting and it should be made public.



Static website hosting

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

Static website hosting

Enabled

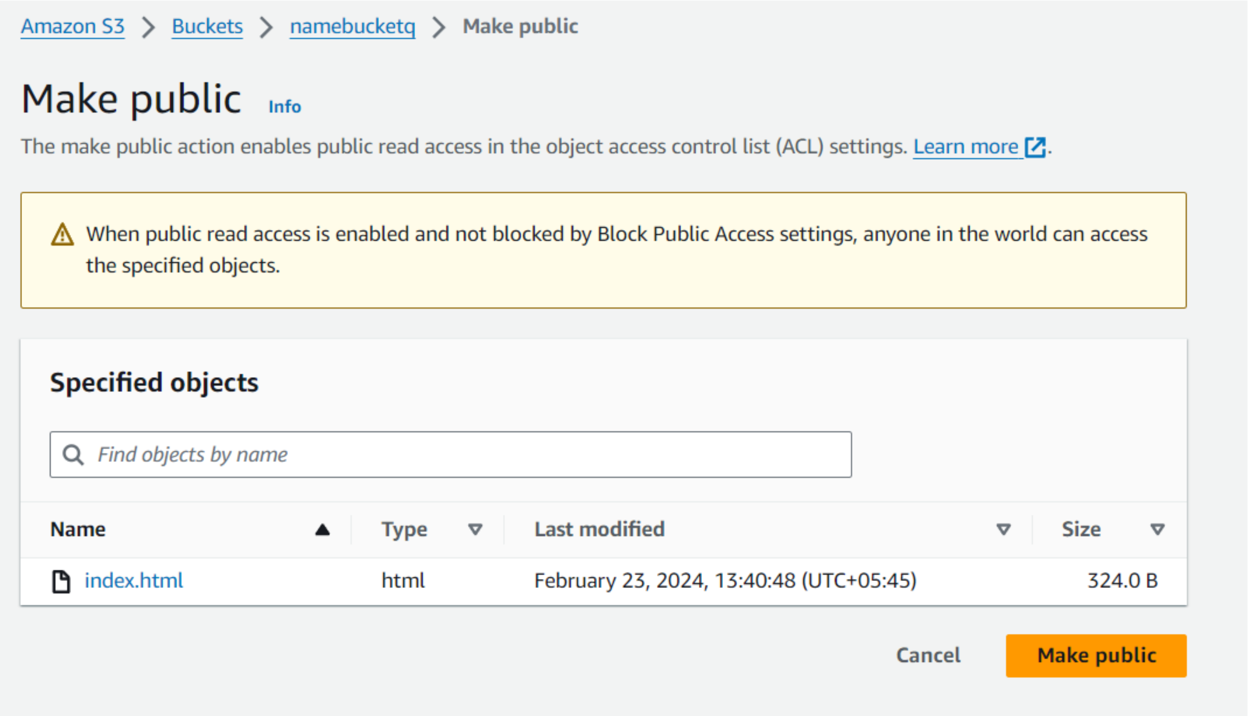
Hosting type

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



Amazon S3 > Buckets > namebucketq > Make public

Make public

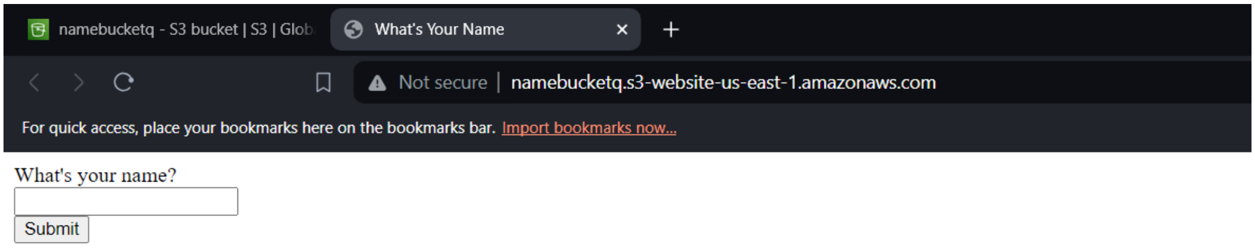
The make public action enables public read access in the object access control list (ACL) settings. Learn more.

⚠ 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

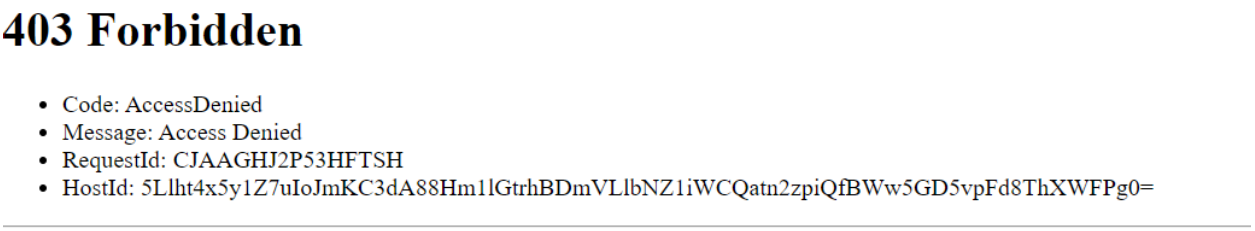
Name ▲	Type ▽	Last modified ▽	Size ▽
index.html	html	February 23, 2024, 13:40:48 (UTC+05:45)	324.0 B

14. Then after we check the dns we get the idex.html file loaded from the aws server.
15. Since I got many issues that I couldn't get the labda triggered. So I created every thing again from new labda function .


namebucketq - S3 bucket | S3 | Glob
What's Your Name
Not secure
namebucketq.s3-website-us-east-1.amazonaws.com
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

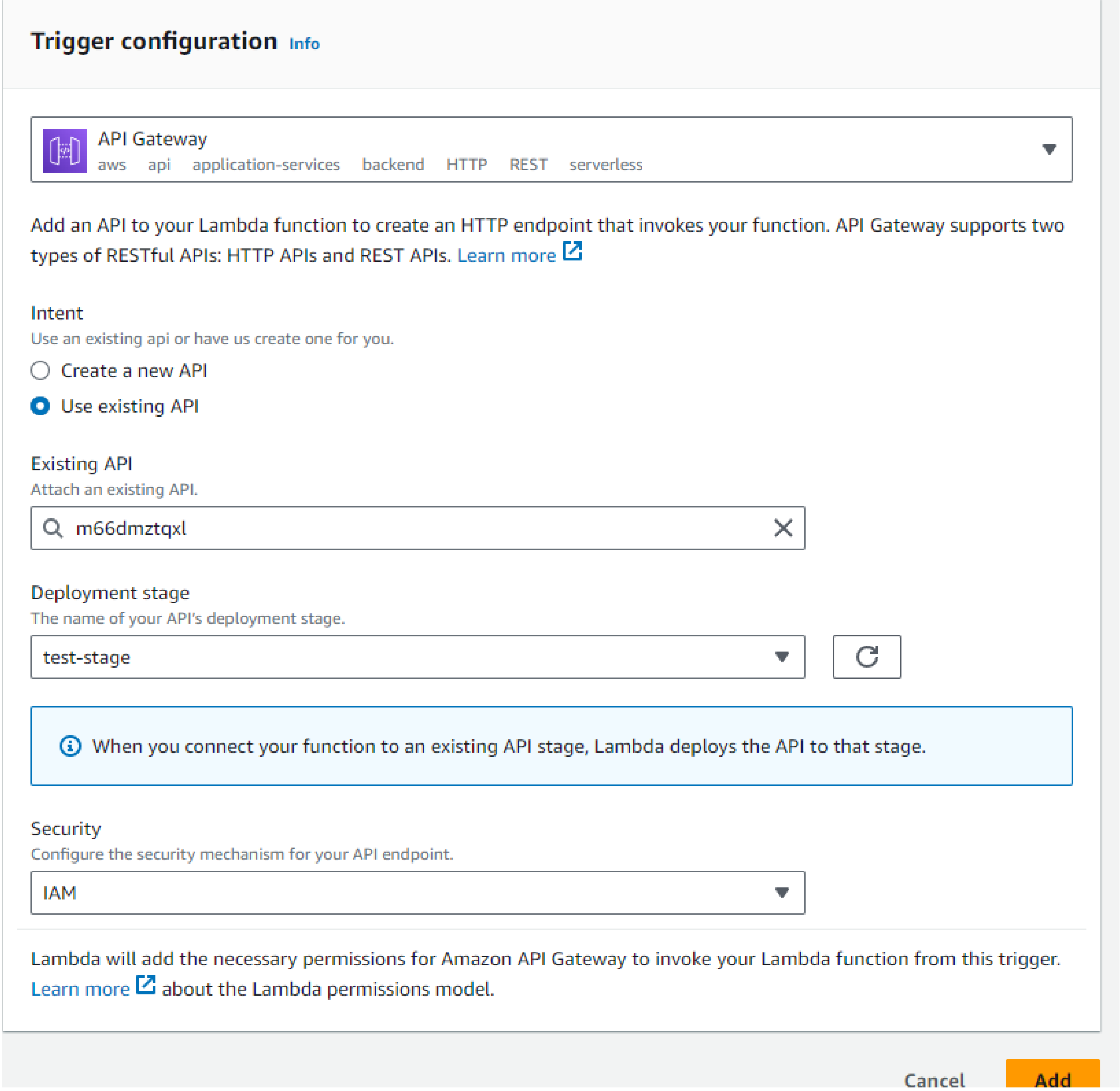
What's your name?

Submit



403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: CJAAGHJ2P53HFTSH
- HostId: 5Llht4x5y1Z7uIoJmKC3dA88Hm1lGtrhBDmVLlbNZ1iWCQatn2zpiQfBWw5GD5vpFd8ThXWFPg0=



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.

m66dmztqxl

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

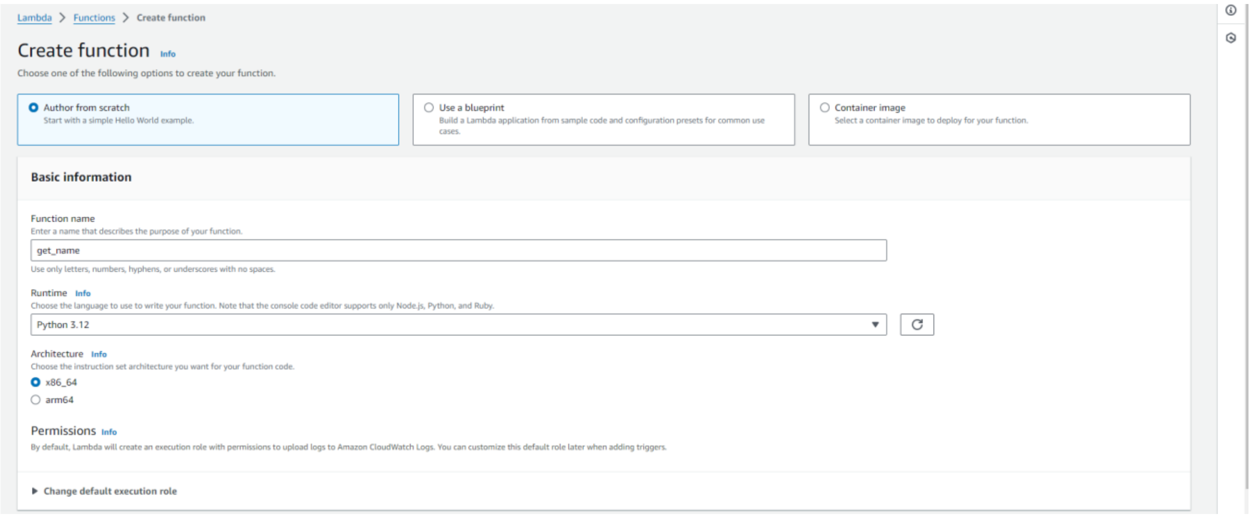
test-stage

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.

IAM

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.



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
Enter a name that describes the purpose of your function.

get_name

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

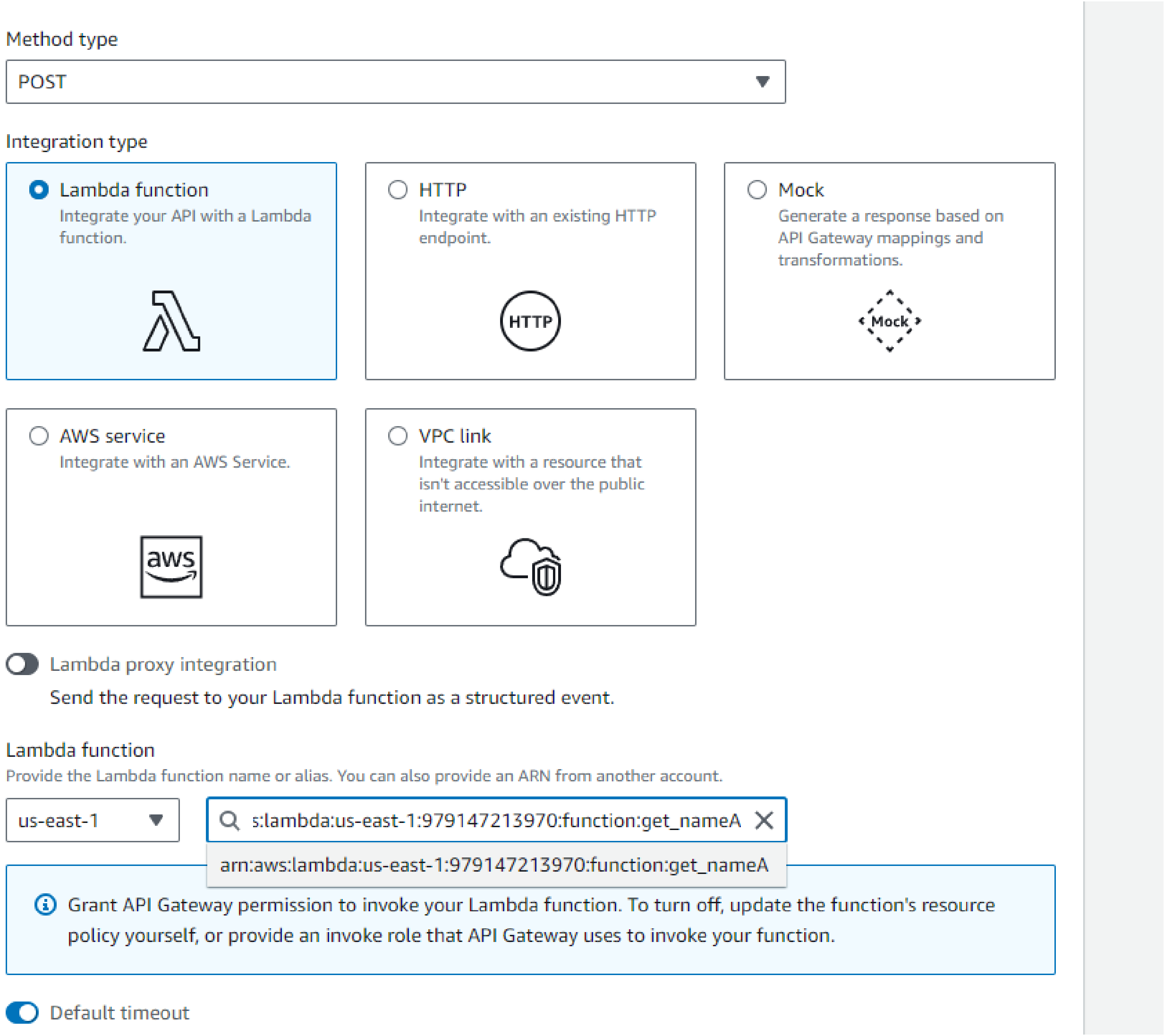
Architecture Info
Choose the instruction set architecture you want for your function code.

● 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



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

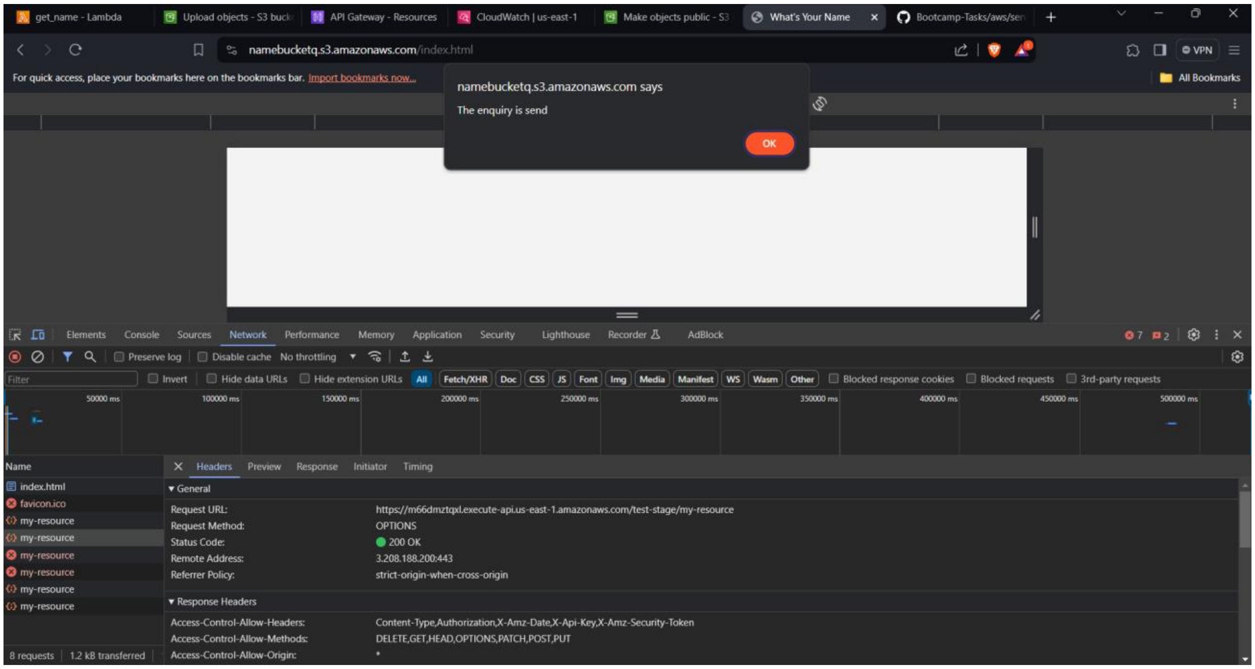
s:lambda:us-east-1:979147213970:function:get_nameA

arn:aws:lambda:us-east-1:979147213970:function:get_nameA

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

16. Then finally the Lambda function was triggered and I was able to send the data from html to our server.



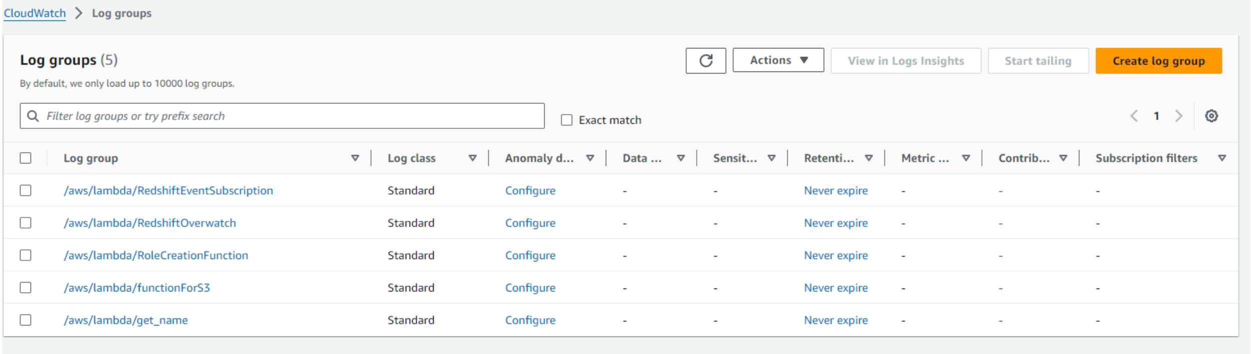
namebucketq.s3.amazonaws.com says

The enquiry is send

OK

```
Name
index.html
favicon.ico
my-resource
my-resource
my-resource
my-resource
my-resource
my-resource
8 requests | 1.2 kB transferred
```

```
▾ General
Request URL:        https://m66dmztqxl.execute-api.us-east-1.amazonaws.com/test-stage/my-resource
Request Method:     OPTIONS
Status Code:        200 OK
Remote Address:     3.208.188.200:443
Referrer Policy:    strict-origin-when-cross-origin
▾ Response Headers
Access-Control-Allow-Headers:   Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token
Access-Control-Allow-Methods:   DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT
Access-Control-Allow-Origin:    *
```



Log groups (5)

By default, we only load up to 10000 log groups.

□	Log group	Log class	Anomaly d...	Data ...	Sensit...	Retenti...	Metric ...	Contrib...	Subscription filters
□	/aws/lambda/RedshiftEventSubscription	Standard	Configure	-	-	Never expire	-	-	-
□	/aws/lambda/RedshiftOverwatch	Standard	Configure	-	-	Never expire	-	-	-
□	/aws/lambda/RoleCreationFunction	Standard	Configure	-	-	Never expire	-	-	-
□	/aws/lambda/functionForS3	Standard	Configure	-	-	Never expire	-	-	-
□	/aws/lambda/get_name	Standard	Configure	-	-	Never expire	-	-	-