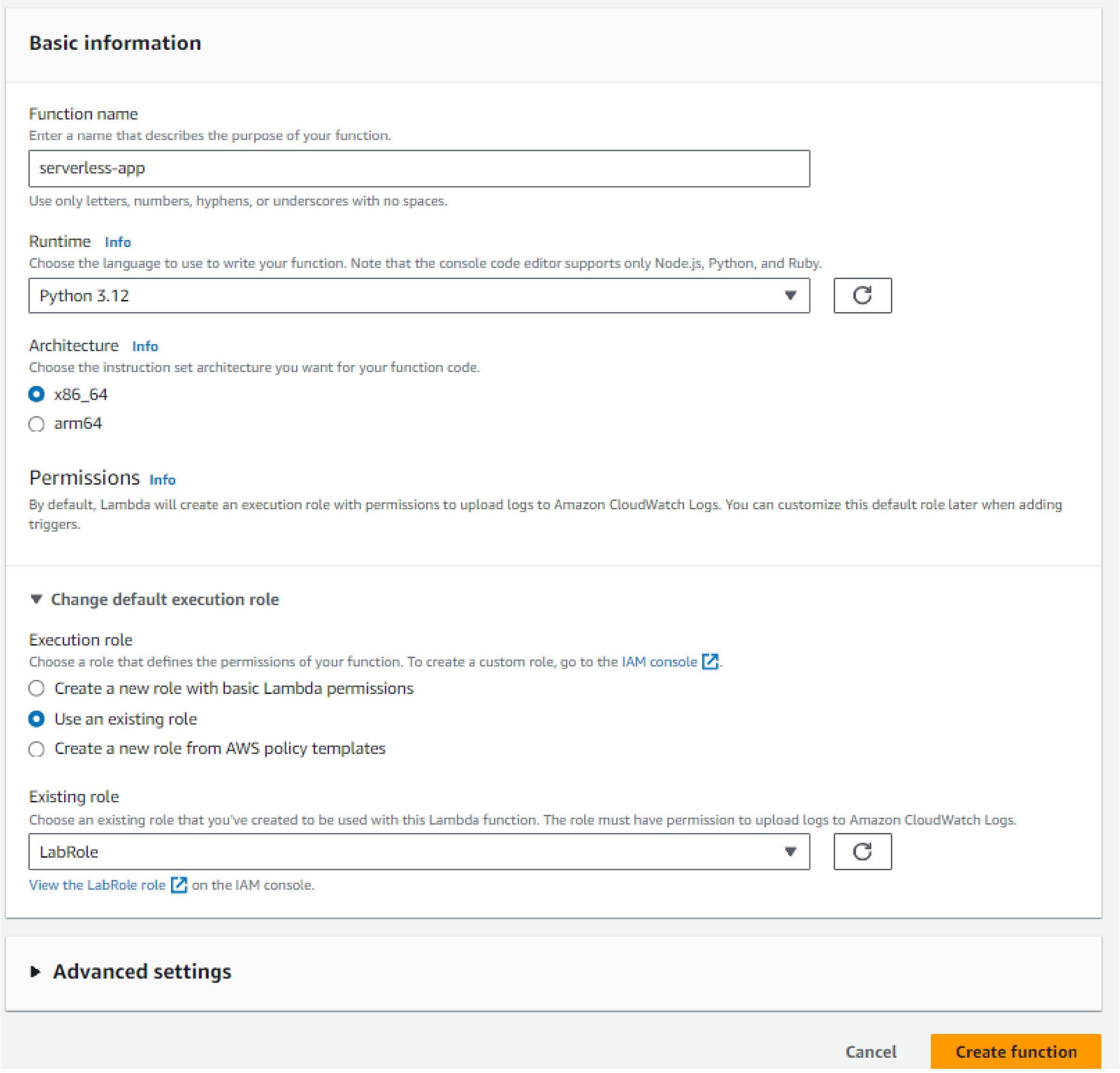
Building Serverless Web Application

Create a Lambda function using existing role that is LabRole



Basic information

Function name
Enter a name that describes the purpose of your function.
serverless-app
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

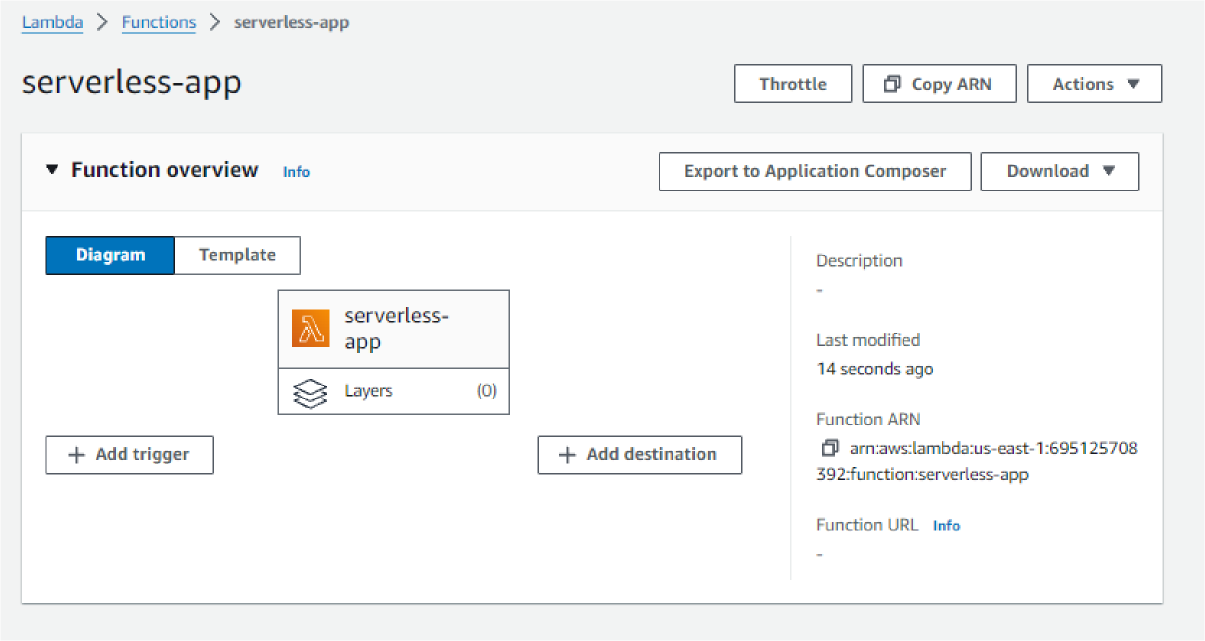
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

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.



Lambda > Functions > serverless-app

serverless-app

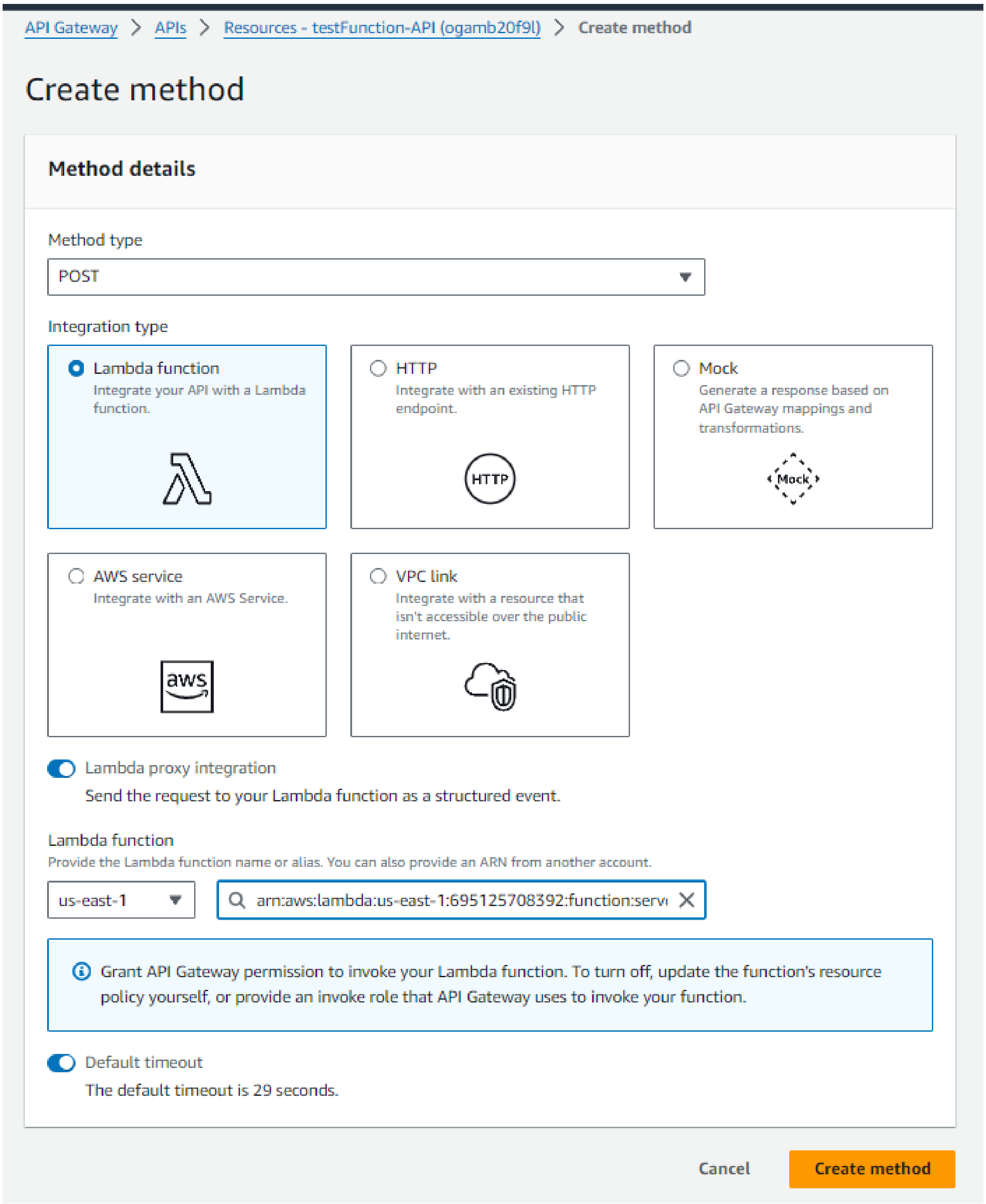
▼ Function overview Info

Description
-

Last modified
14 seconds ago

Function ARN
arn:aws:lambda:us-east-1:695125708392:function:serverless-app

Function URL Info
-



API Gateway > APIs > Resources - testFunction-API (ogamb20f9l) > 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.

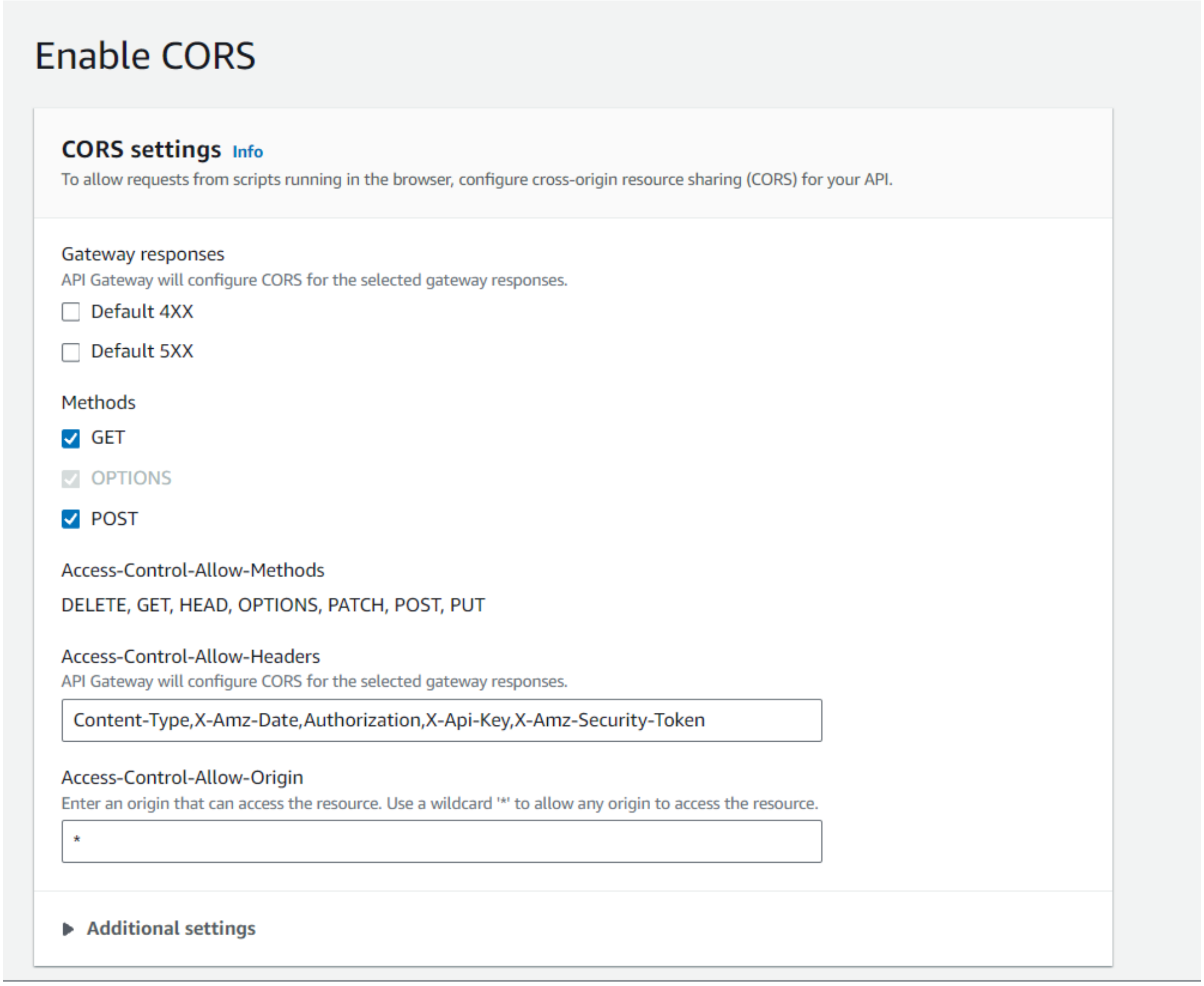
us-east-1

arn:aws:lambda:us-east-1:695125708392:function:serv

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.

Next step is to create DynamoDB



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

Methods

☑ GET

☑ OPTIONS

☑ POST

Access-Control-Allow-Methods

DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT

Access-Control-Allow-Headers

API Gateway will configure CORS for the selected gateway responses.

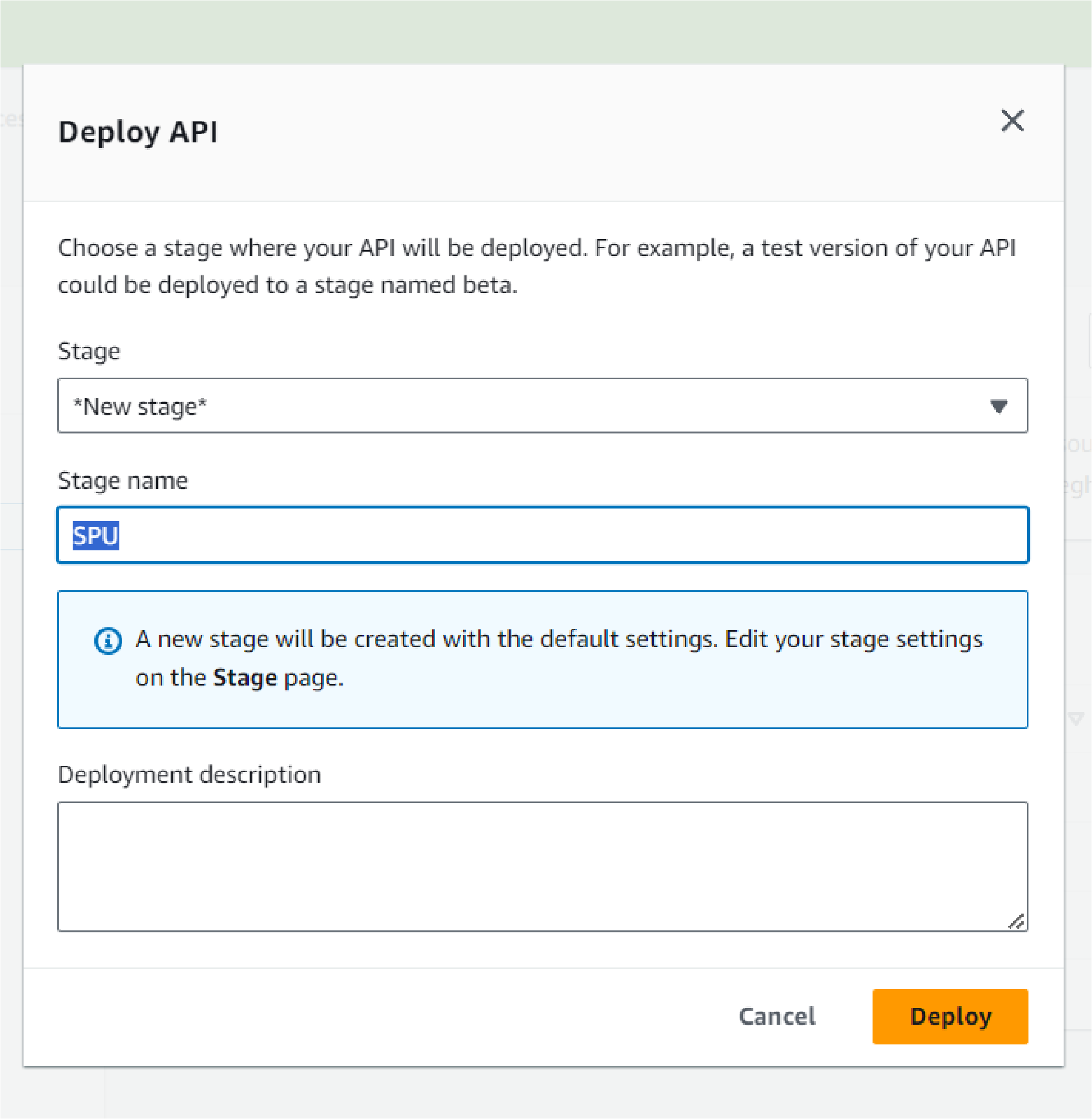
Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token

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



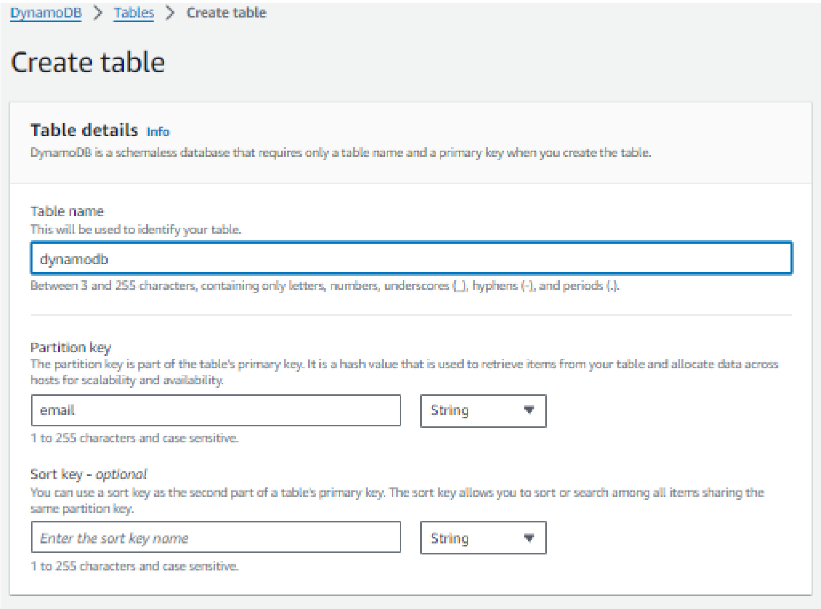
Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Stage: *New stage*

Stage name: SPU

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

Deployment description



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.

dynamodb

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.

email | String

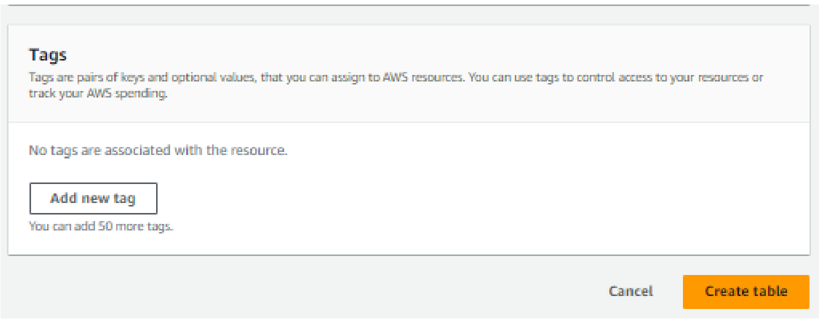
1 to 255 characters and case sensitive.

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.

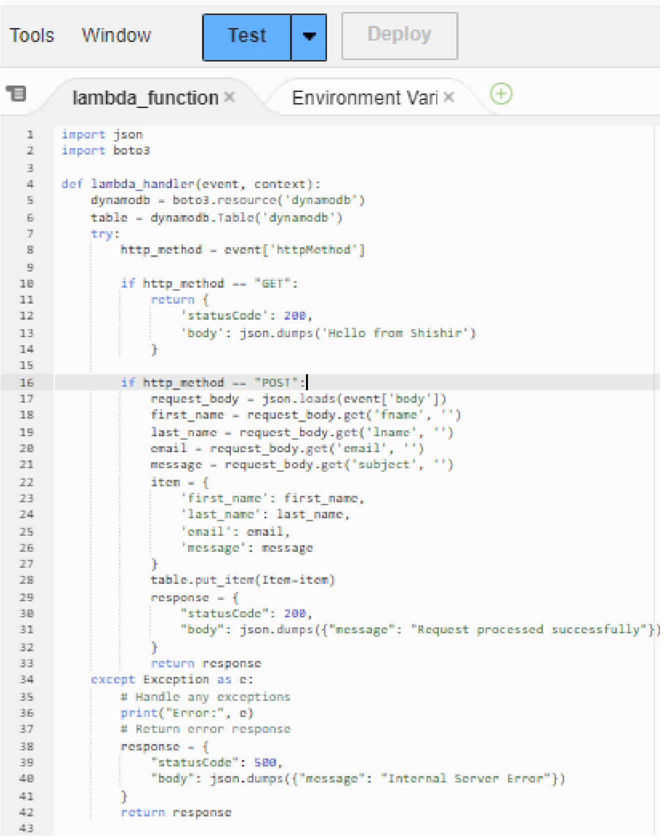


Tags

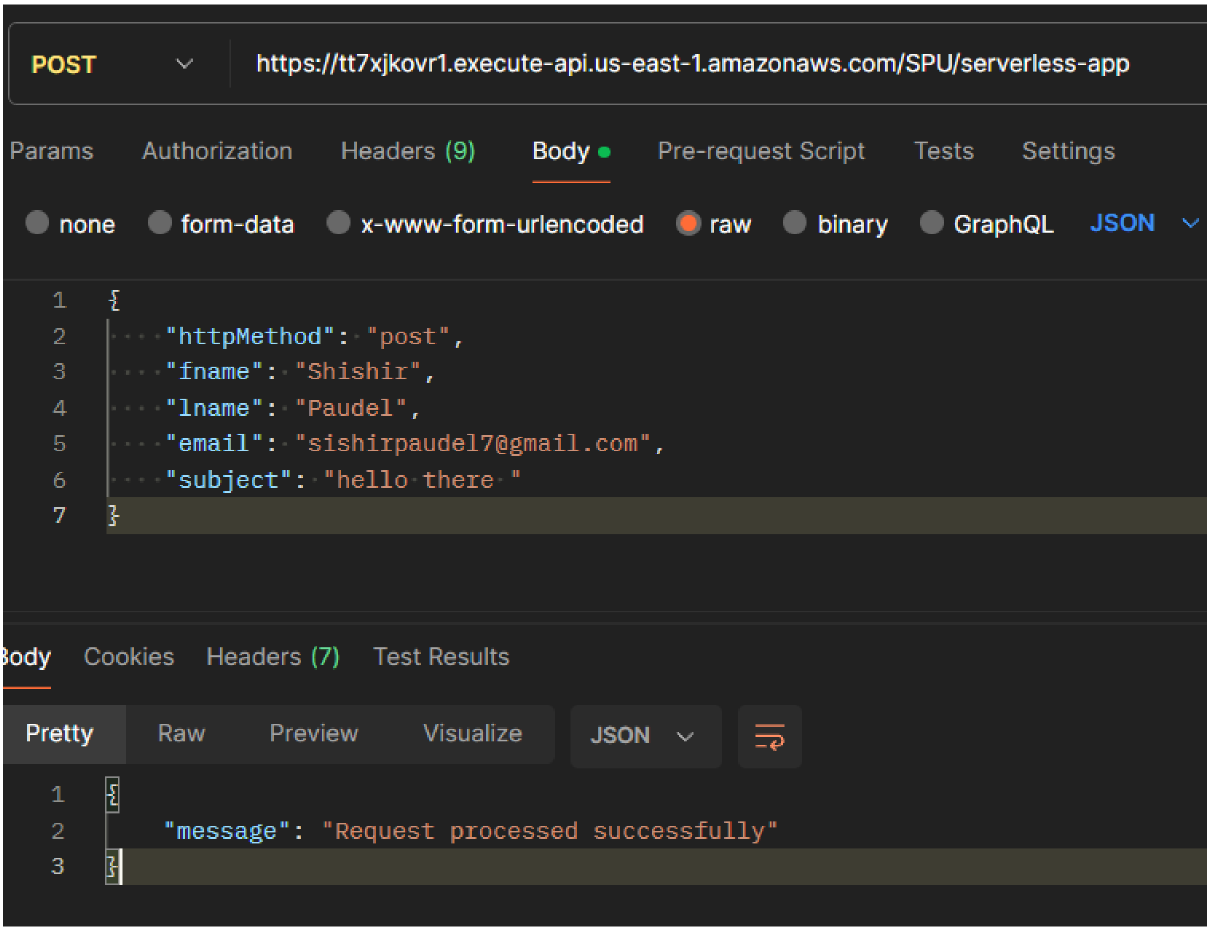
Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

You can add 50 more tags.



```
import json
import boto3

def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('dynamodb')
    try:
        http_method = event['httpMethod']

        if http_method == "GET":
            return {
                'statusCode': 200,
                'body': json.dumps('Hello from Shishir')
            }

        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
            }
            table.put_item(Item=item)
            response = {
                "statusCode": 200,
                "body": json.dumps({"message": "Request processed successfully"})
            }
            return response
    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

```

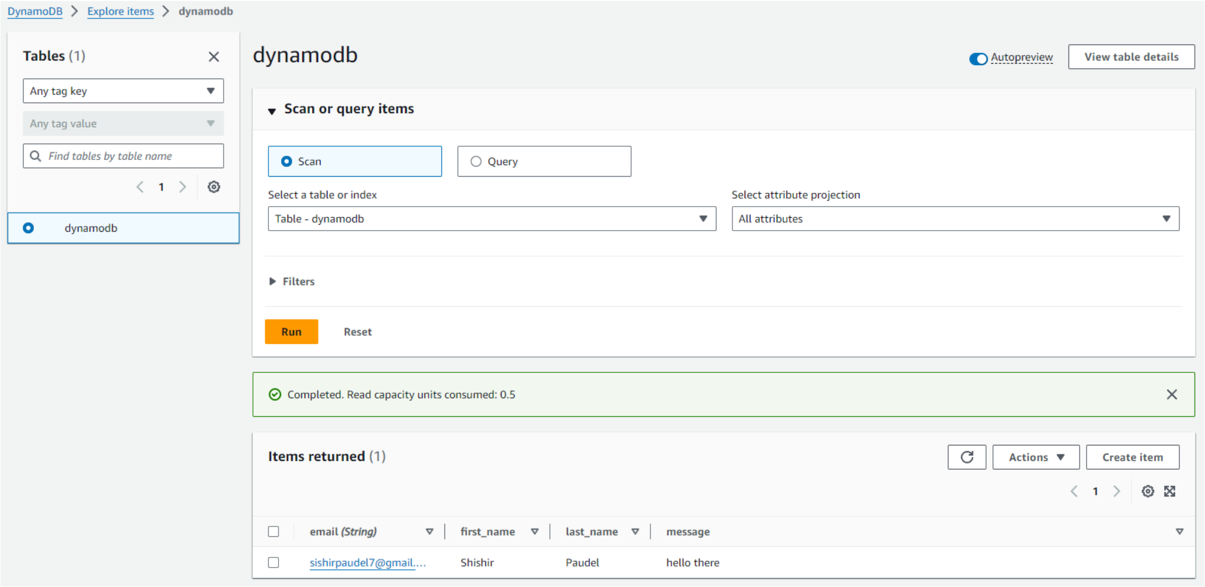


POST https://tt7xjkovr1.execute-api.us-east-1.amazonaws.com/SPU/serverless-app

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "httpMethod": "post",
    "fname": "Shishir",
    "lname": "Paudel",
    "email": "sishirpaudel7@gmail.com",
    "subject": "hello there "
}
```

```
{
    "message": "Request processed successfully"
}
```



dynamodb

● Scan ○ Query

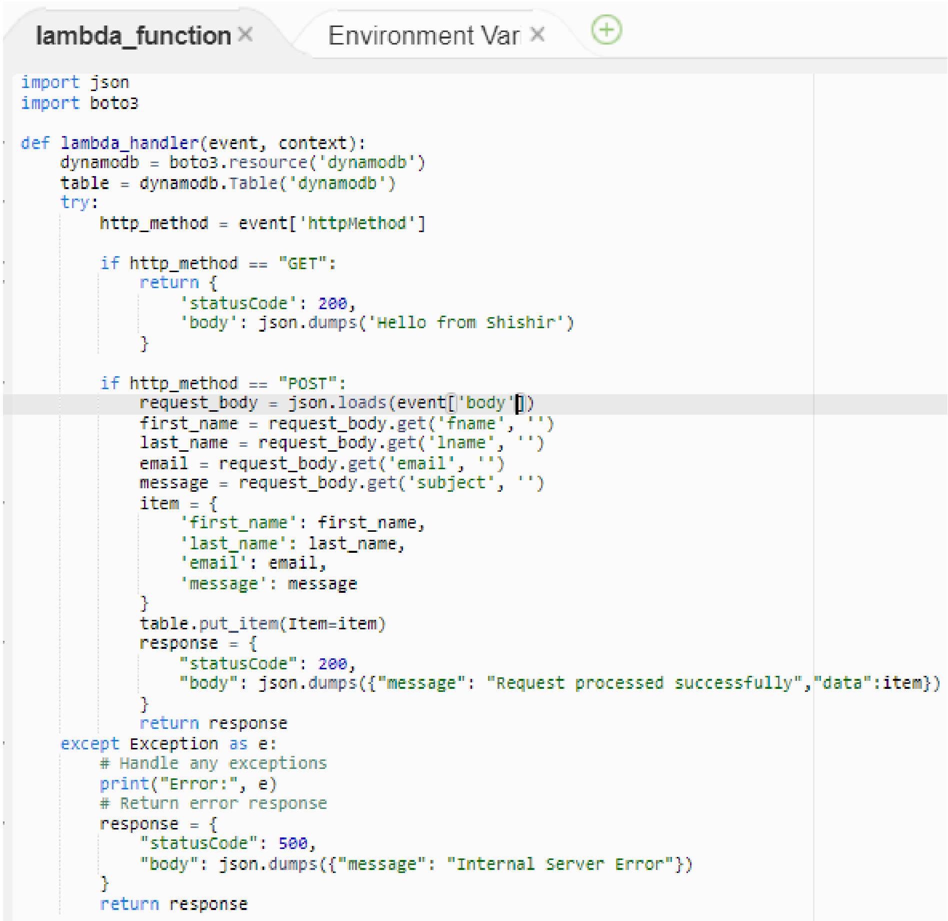
Select a table or index: Table - dynamodb

Select attribute projection: All attributes

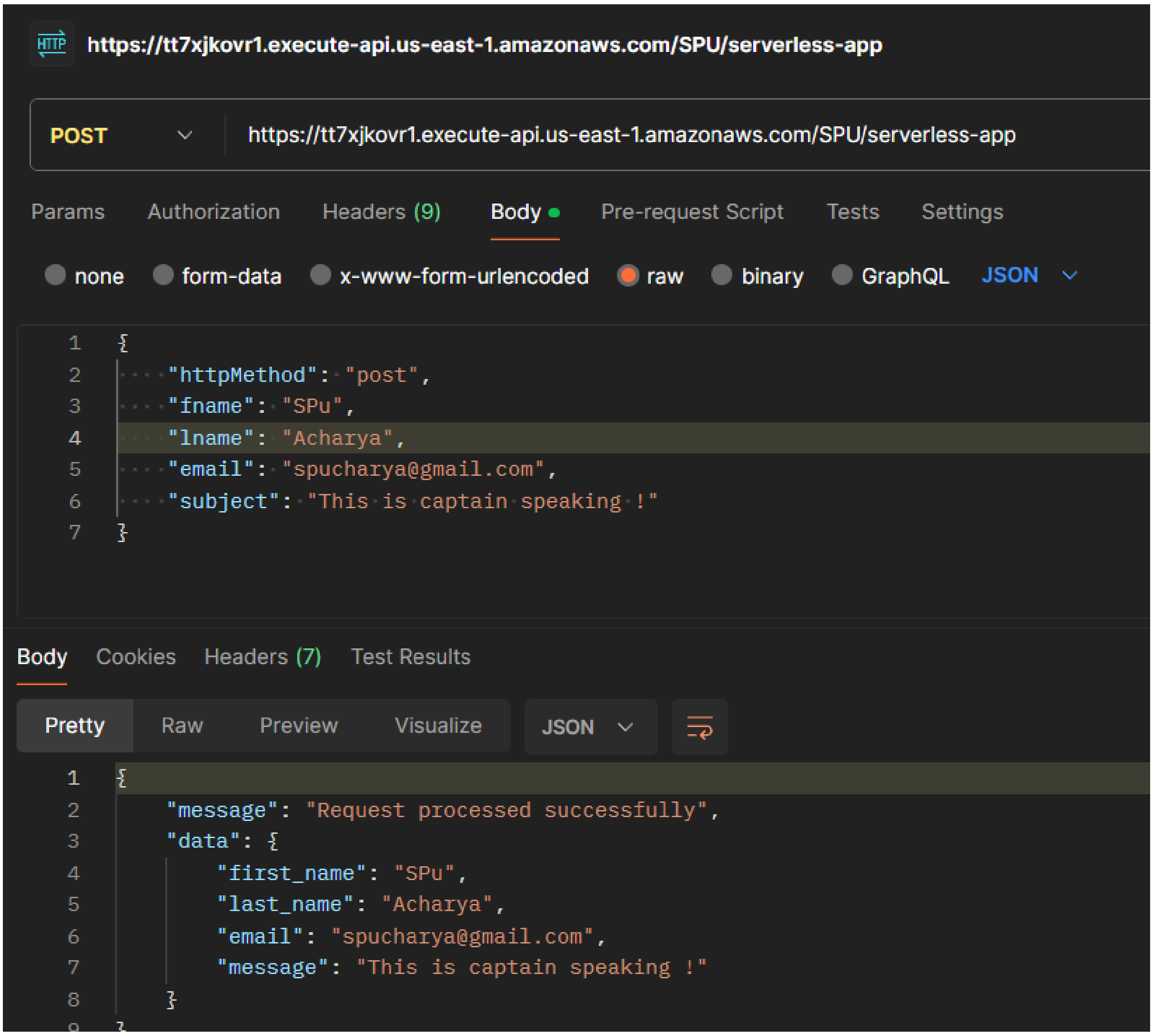
Completed. Read capacity units consumed: 0.5

Items returned (1)

□	email (String)	first_name	last_name	message
□	sishirpaudel7@gmail....	Shishir	Paudel	hello there



```python
import json
import boto3

def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('dynamodb')
    try:
        http_method = event['httpMethod']

        if http_method == "GET":
            return {
                'statusCode': 200,
                'body': json.dumps('Hello from Shishir')
            }

        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
            }
            table.put_item(Item=item)
            response = {
                "statusCode": 200,
                "body": json.dumps({"message": "Request processed successfully","data":item})
            }
            return response
    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
```



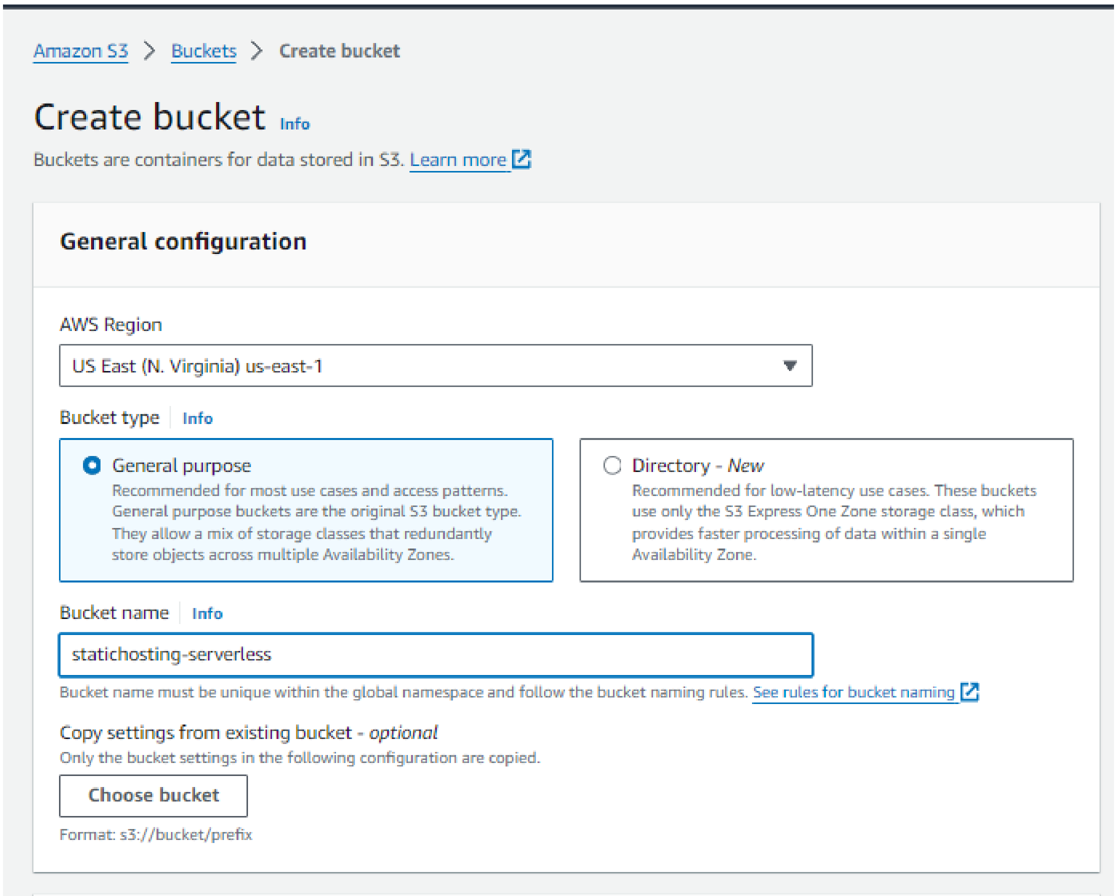
https://tt7xjkovr1.execute-api.us-east-1.amazonaws.com/SPU/serverless-app
POST
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON

POST https://tt7xjkovr1.execute-api.us-east-1.amazonaws.com/SPU/serverless-app

```
{
    "httpMethod": "post",
    "fname": "SPu",
    "lname": "Acharya",
    "email": "spucharya@gmail.com",
    "subject": "This is captain speaking !"
}
```

```
{
    "message": "Request processed successfully",
    "data": {
        "first_name": "SPu",
        "last_name": "Acharya",
        "email": "spucharya@gmail.com",
        "message": "This is captain speaking !"
    }
}
```

Create a S3 bucket to host a static html website



Amazon S3 > Buckets > Create bucket

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

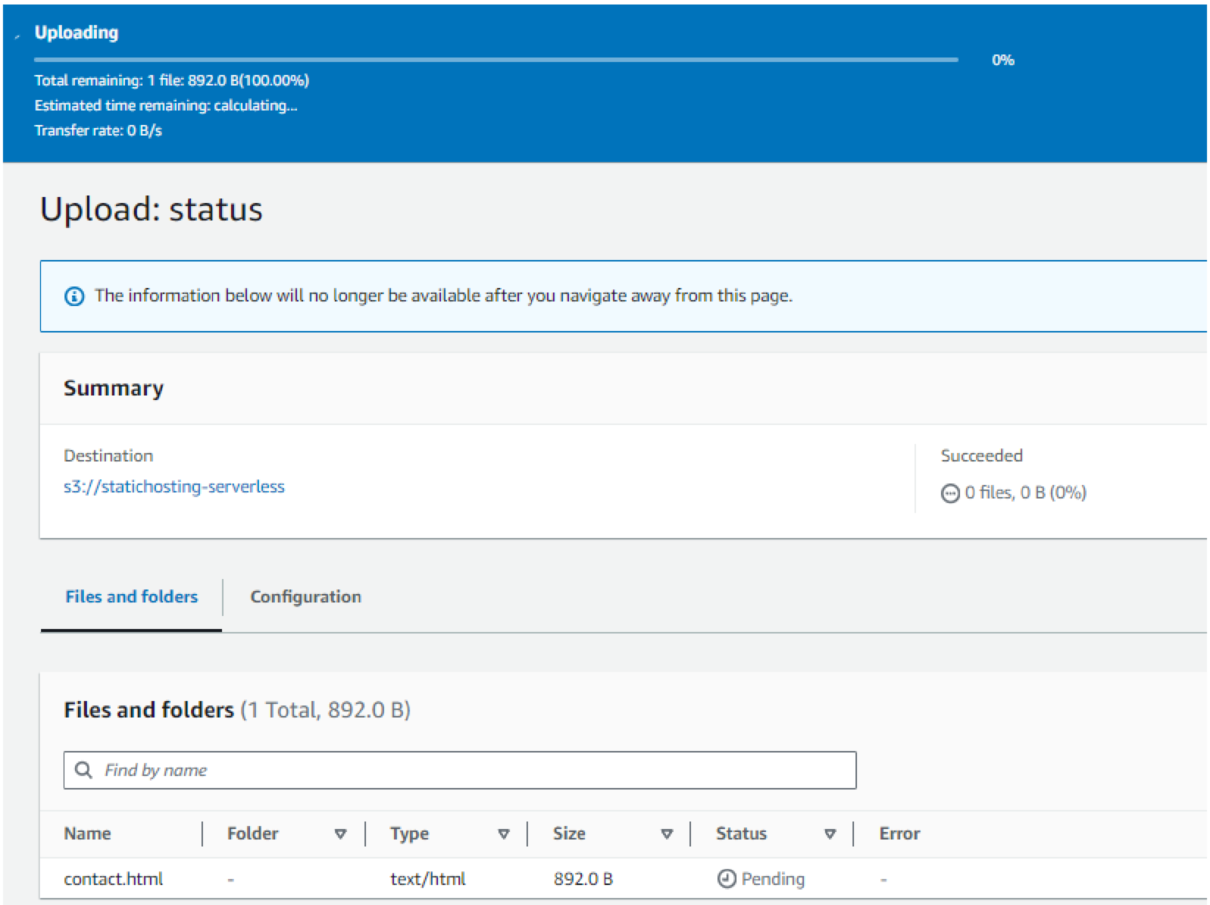
statichosting-serverless

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



Uploading

Total remaining: 1 file: 892.0 B(100.00%)
Estimated time remaining: calculating...
Transfer rate: 0 B/s

0%

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

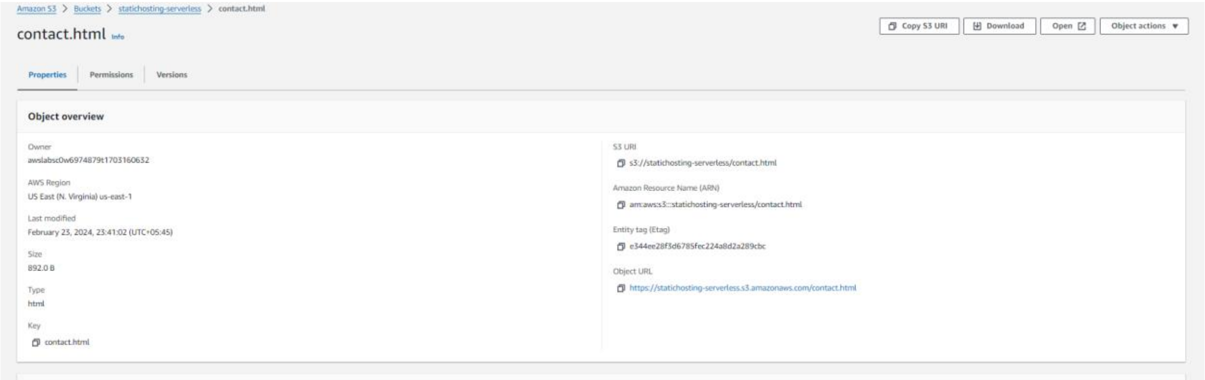
Destination
s3://statichosting-serverless

Succeeded
0 files, 0 B (0%)

Files and folders (1 Total, 892.0 B)

Name	Folder	Type	Size	Status	Error
contact.html	-	text/html	892.0 B	Pending	-

Using object url I tried to run the site , but gave me this error.



contact.html Info

Object overview

Owner
awslabsc0w6974879t1703160632

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

Last modified
February 23, 2024, 23:41:02 (UTC+05:45)

Size
892.0 B

Type
html

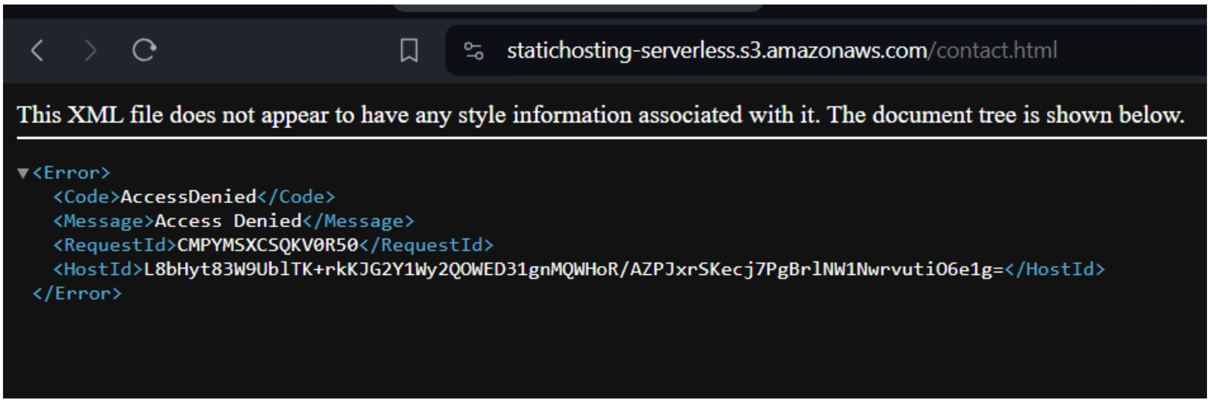
Key
contact.html

S3 URI
s3://statichosting-serverless/contact.html

Amazon Resource Name (ARN)
arn:aws:s3:::statichosting-serverless/contact.html

Entity tag (Etag)
e344ee28f3d6785fec224a8d2a289cbc

Object URL
https://statichosting-serverless.s3.amazonaws.com/contact.html

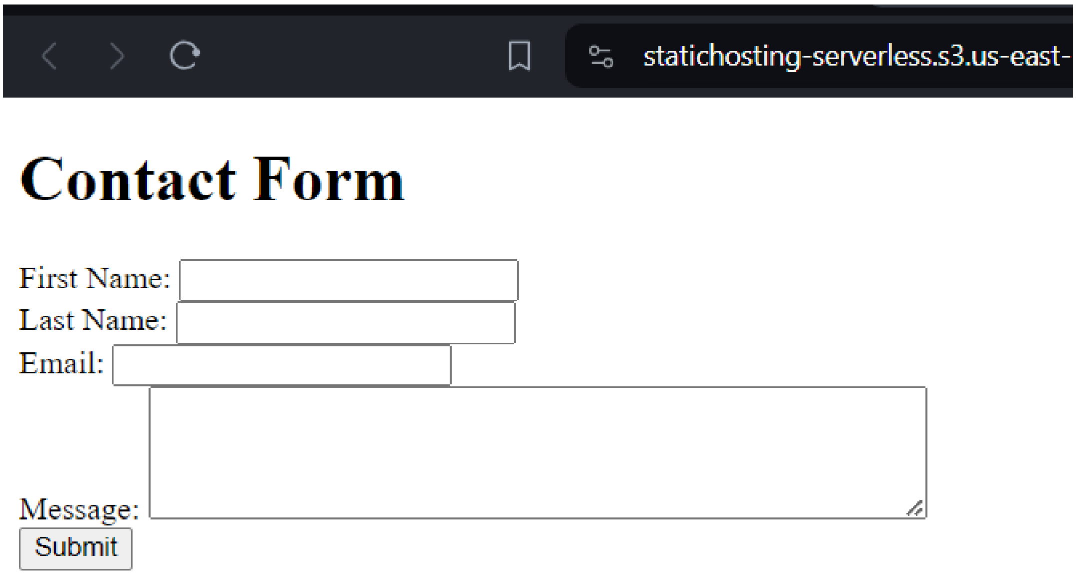


statichosting-serverless.s3.amazonaws.com/contact.html

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>CMPYMSXCSQKV0R50</RequestId>
   <HostId>L8bHyt83W9UblTK+rkKJG2Y1Wy2QOWED31gnMQWHoR/AZPJxrSKecj7PgBrlNW1Nwrvuti06e1g=</HostId>
 </Error>
```

Static Hosting using s3 bucket was successful , when I clicked the open tab from top right corner.



statichosting-serverless.s3.us-east-

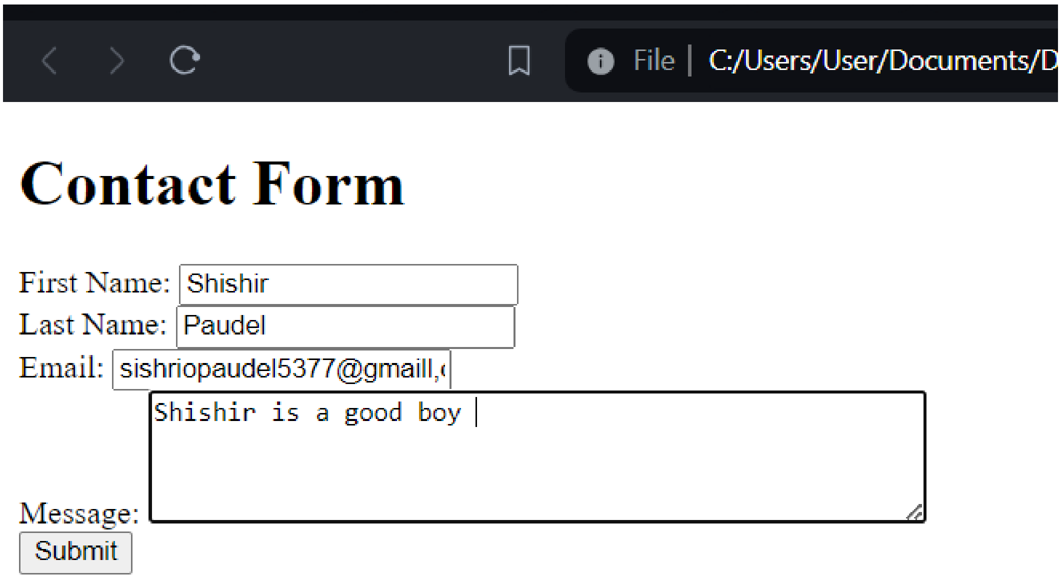
Contact Form

First Name:

Last Name:

Email:

Message:



File | C:/Users/User/Documents/D

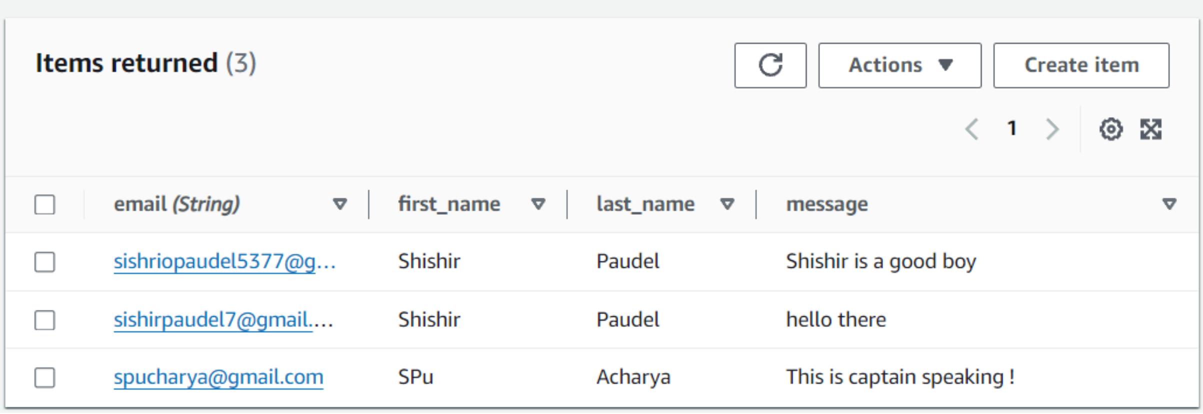
Contact Form

First Name: Shishir

Last Name: Paudel

Email: sishriopaudel5377@gmaill,(

Message: Shishir is a good boy

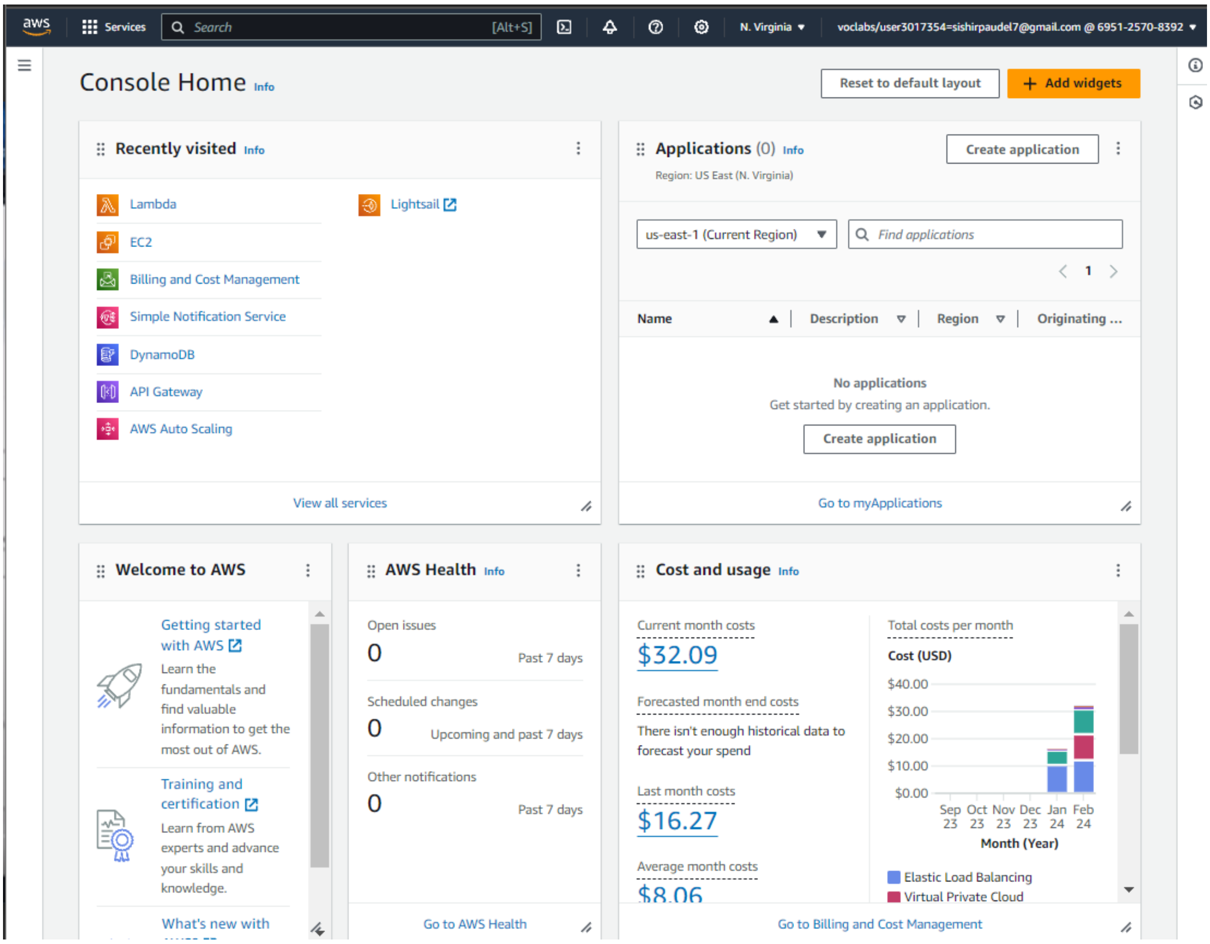


Items returned (3)

□	email *(String)*	first_name	last_name	message
□	sishriopaudel5377@g…	Shishir	Paudel	Shishir is a good boy
□	sishirpaudel7@gmail.…	Shishir	Paudel	hello there
□	spucharya@gmail.com	SPu	Acharya	This is captain speaking !

Lab Work using : Sns Dynamo DB , S3 bucket to upload a file

Creating a Lambda



aws | Services | Search [Alt+S] | N. Virginia ▾ | voclabs/user3017354=sishirpaudel7@gmail.com @ 6951-2570-8392 ▾

Console Home Info

Reset to default layout | + Add widgets

Recently visited Info

- Lambda
- EC2
- Billing and Cost Management
- Simple Notification Service
- DynamoDB
- API Gateway
- AWS Auto Scaling
- Lightsail

View all services

Applications (0) Info | Create application

Region: US East (N. Virginia)

us-east-1 (Current Region) ▾ | Find applications

< 1 >

Name ▲	Description ▽	Region ▽	Originating ...

No applications

Get started by creating an application.

Create application

Go to myApplications

Welcome to AWS

Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification

Learn from AWS experts and advance your skills and knowledge.

What's new with

AWS Health Info

Open issues

0 Past 7 days

Scheduled changes

0 Upcoming and past 7 days

Other notifications

0 Past 7 days

Go to AWS Health

Cost and usage Info

Current month costs

$32.09

Forecasted month end costs

There isn't enough historical data to forecast your spend

Last month costs

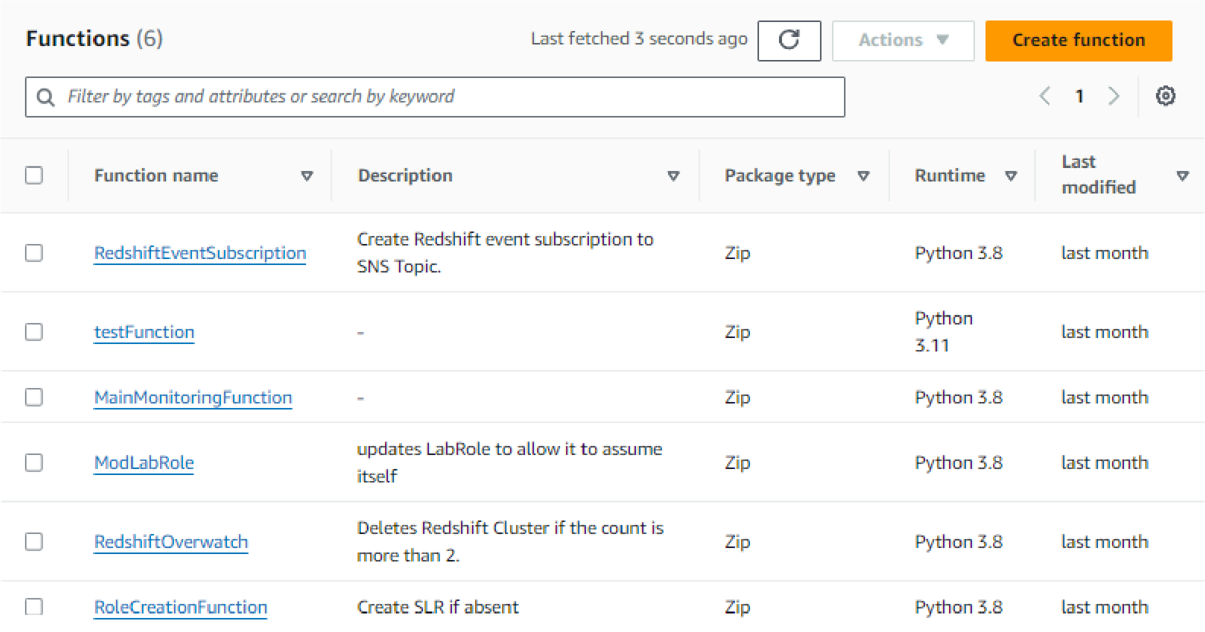
$16.27

Average month costs

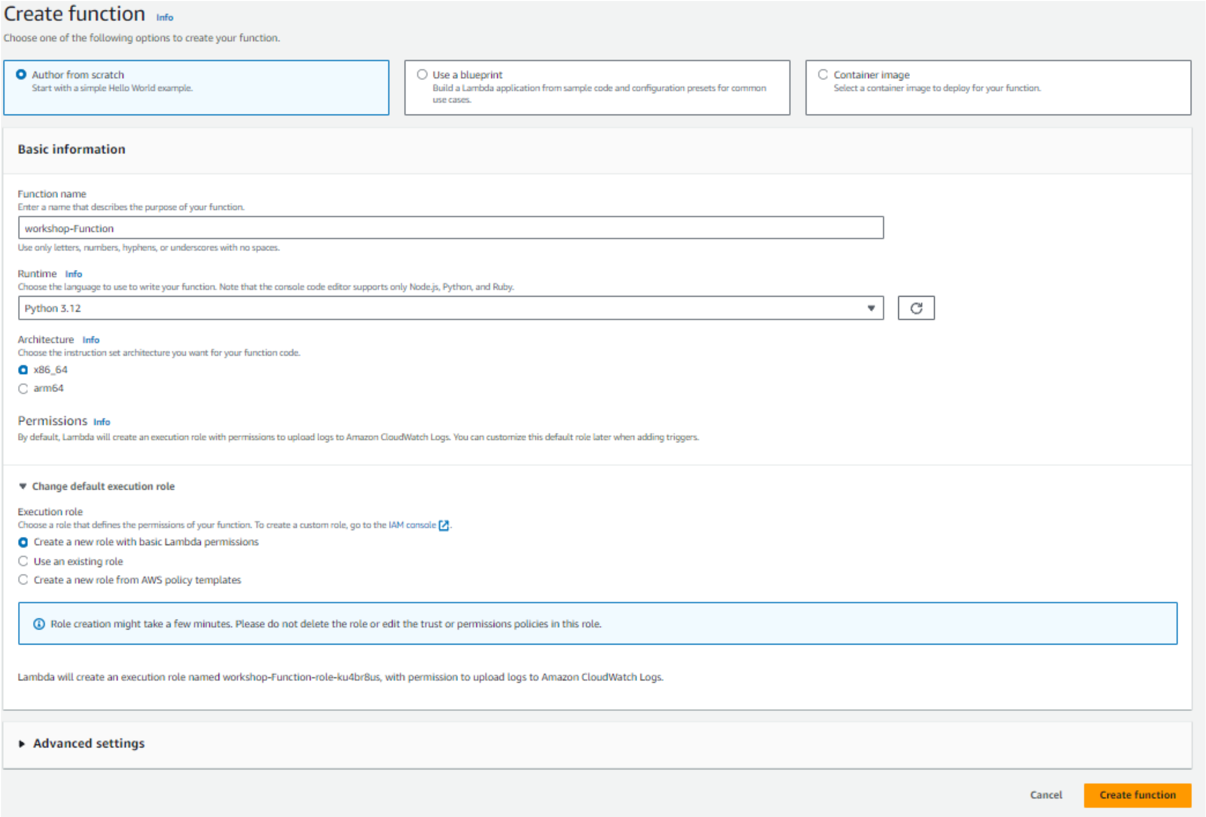
$8.06

Total costs per month

Go to Billing and Cost Management


Functions (6)
Last fetched 3 seconds ago
Actions ▼
Create function
Filter by tags and attributes or search by keyword
1

□	Function name	Description	Package type	Runtime	Last modified
□	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	last month
□	testFunction	-	Zip	Python 3.11	last month
□	MainMonitoringFunction	-	Zip	Python 3.8	last month
□	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	last month
□	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	last month
□	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	last month



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.
workshop-Function
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

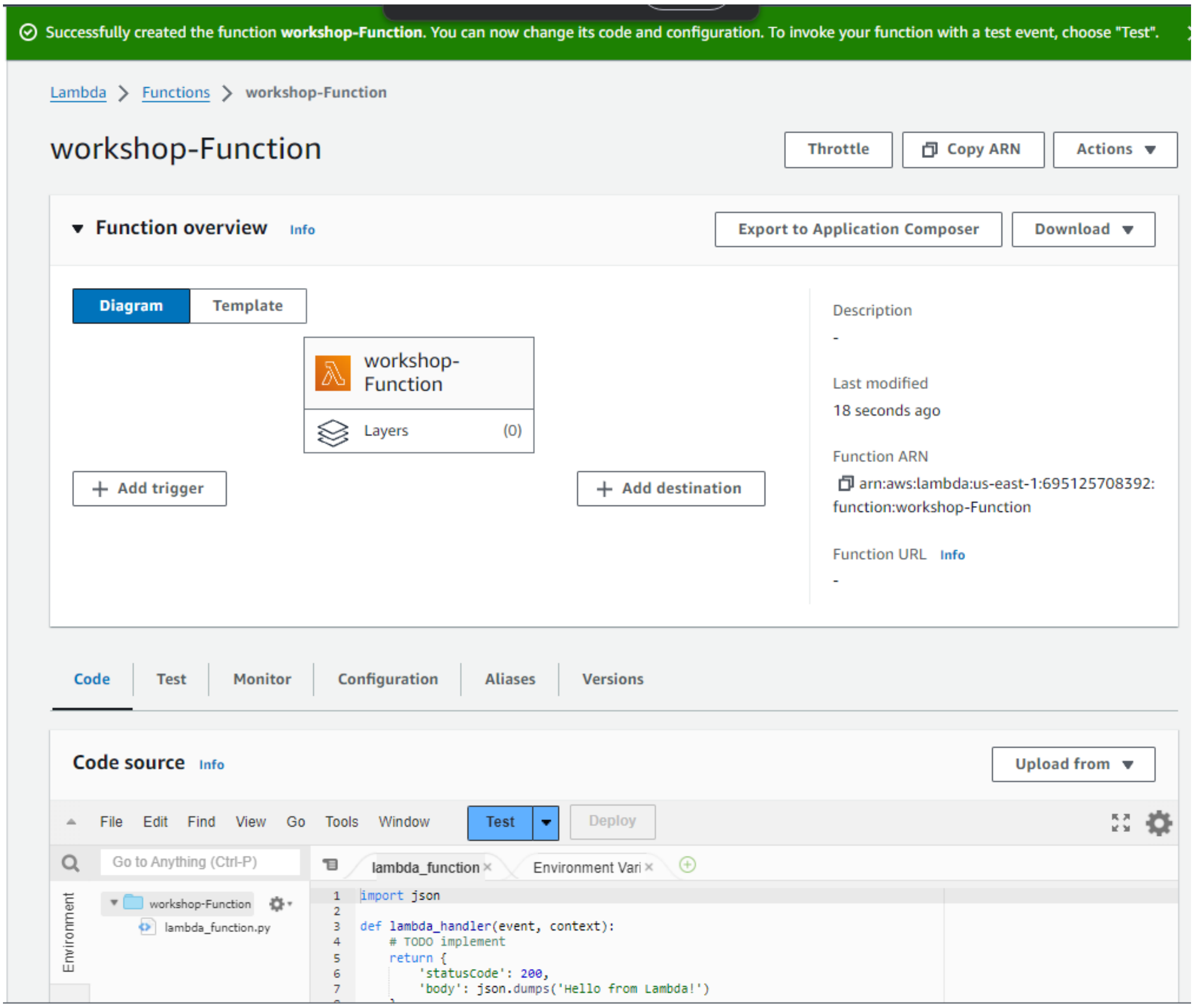
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named workshop-Function-role-ku4br8us, with permission to upload logs to Amazon CloudWatch Logs.



Successfully created the function **workshop-Function**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > workshop-Function

workshop-Function

Function overview Info

Description

-

Last modified

18 seconds ago

Function ARN

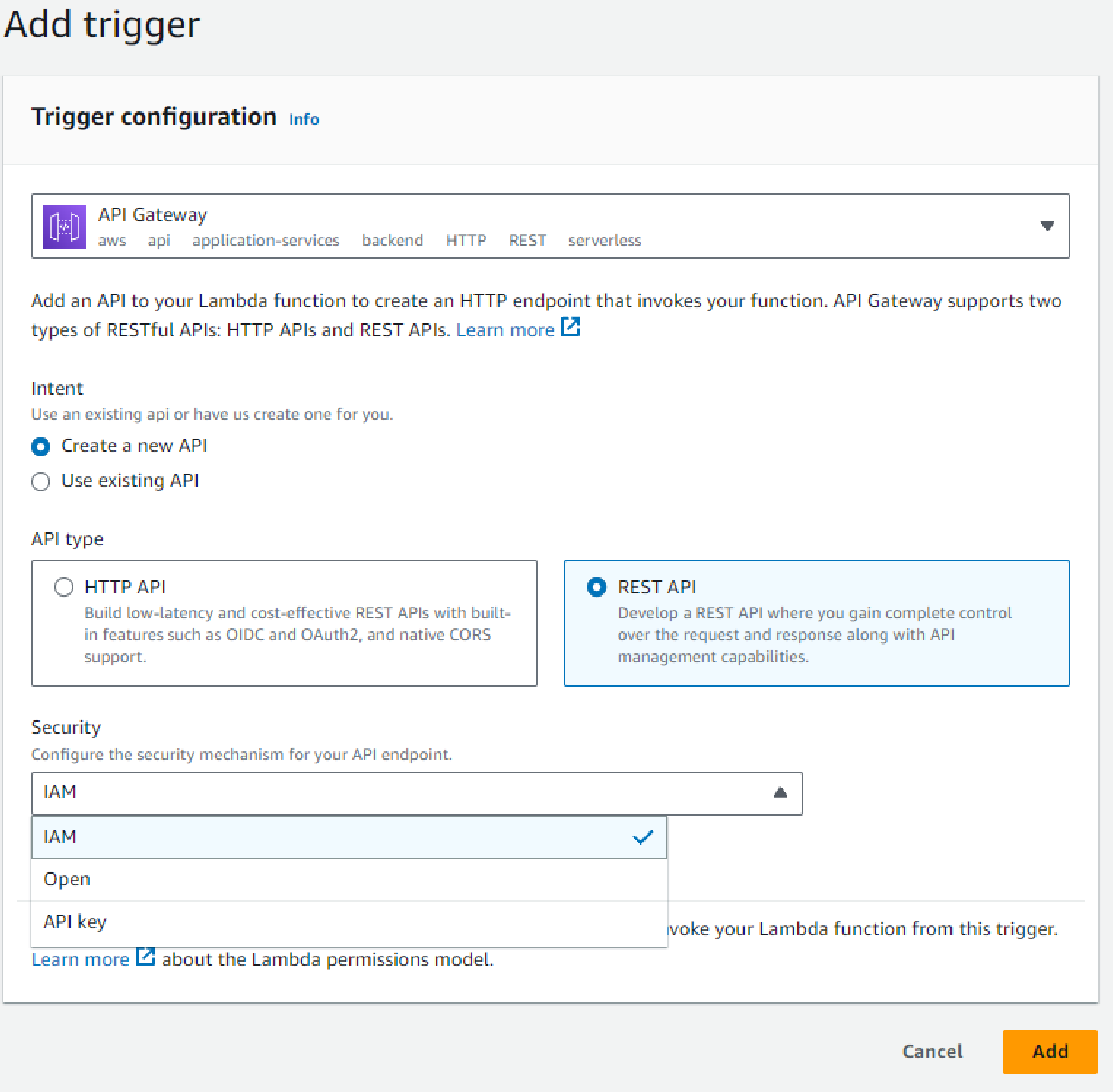
arn:aws:lambda:us-east-1:695125708392:function:workshop-Function

Function URL Info

-

Code source Info

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
```



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

API type

○ HTTP API
Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

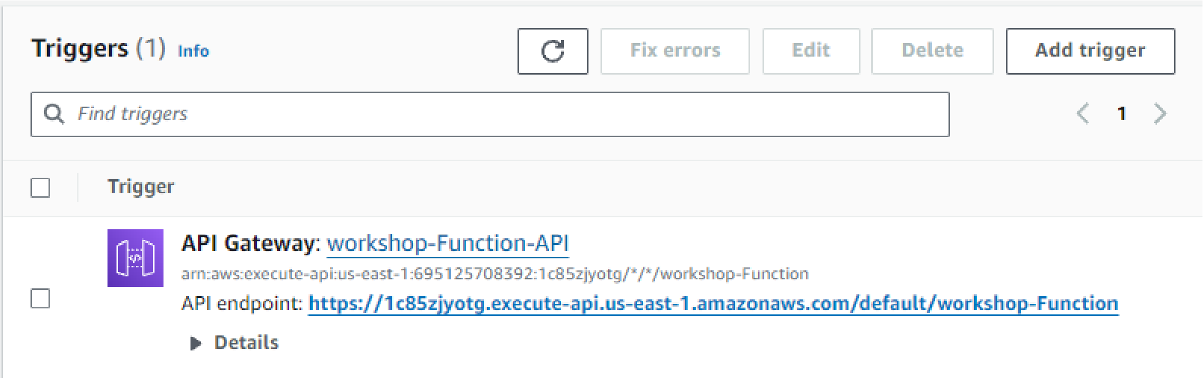
● REST API
Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Security
Configure the security mechanism for your API endpoint.

IAM

IAM ✓
Open
API key

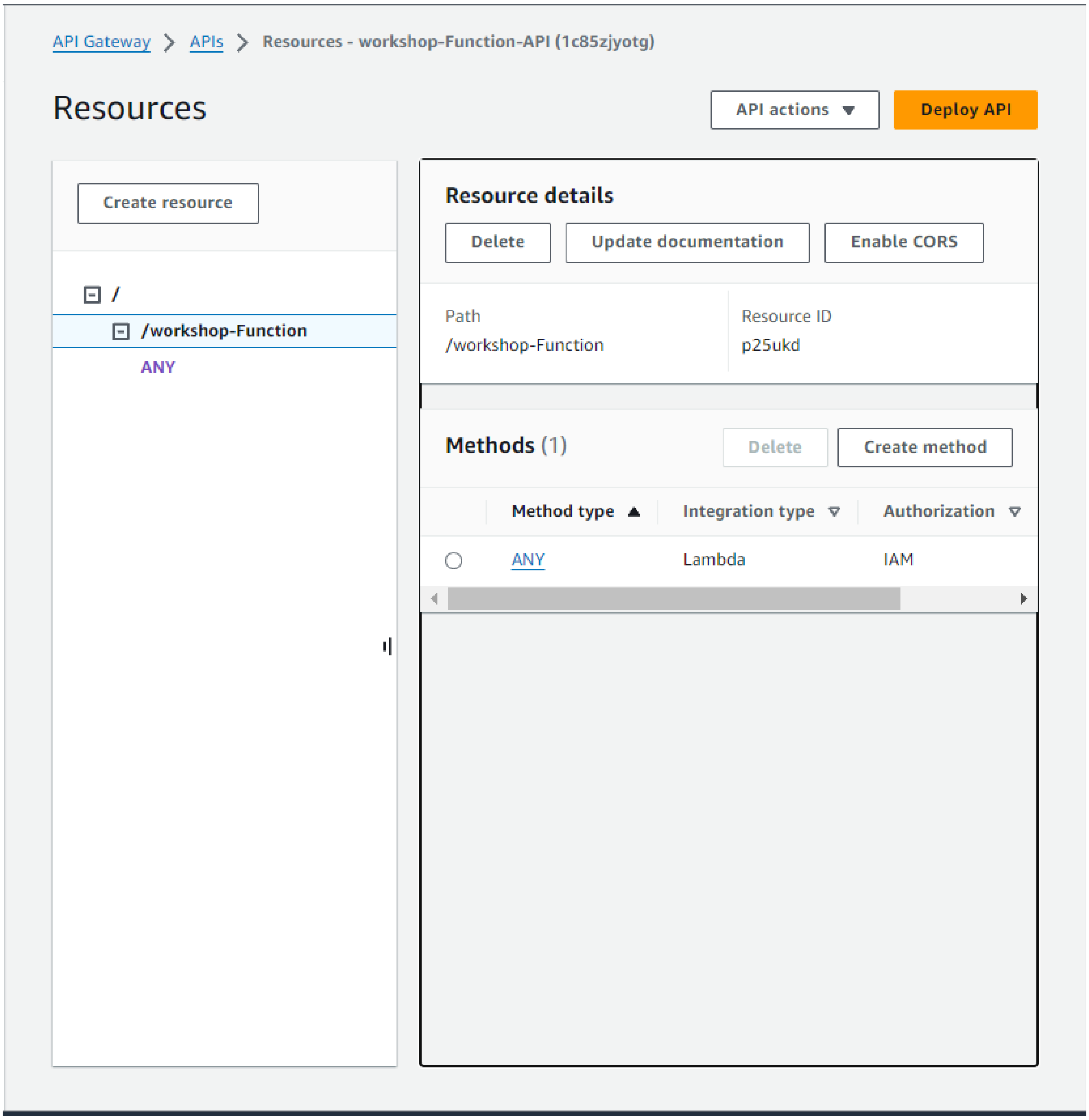
voke your Lambda function from this trigger.
Learn more about the Lambda permissions model.



Triggers (1) Info

□ Trigger

□ API Gateway: workshop-Function-API
arn:aws:execute-api:us-east-1:695125708392:1c85zjyotg/*/*/workshop-Function
API endpoint: https://1c85zjyotg.execute-api.us-east-1.amazonaws.com/default/workshop-Function
▶ Details



Resources

/
/workshop-Function
ANY

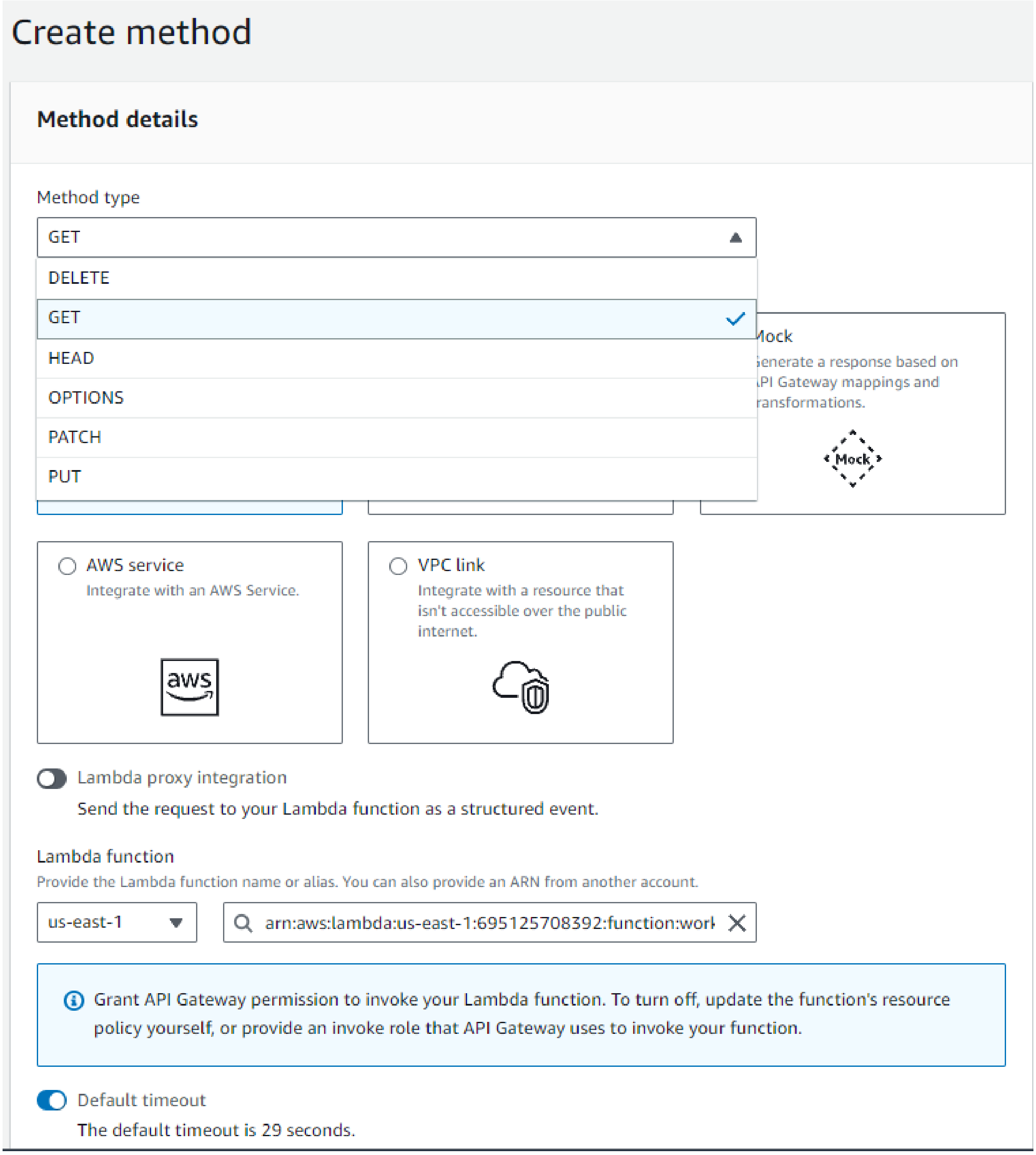
Resource details

Path
/workshop-Function

Resource ID
p25ukd

Methods (1)

	Method type ▲	Integration type ▽	Authorization ▽
○	ANY	Lambda	IAM



Create method

Method details

Method type

GET ▲

DELETE

GET ✓

HEAD

OPTIONS

PATCH

PUT

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.

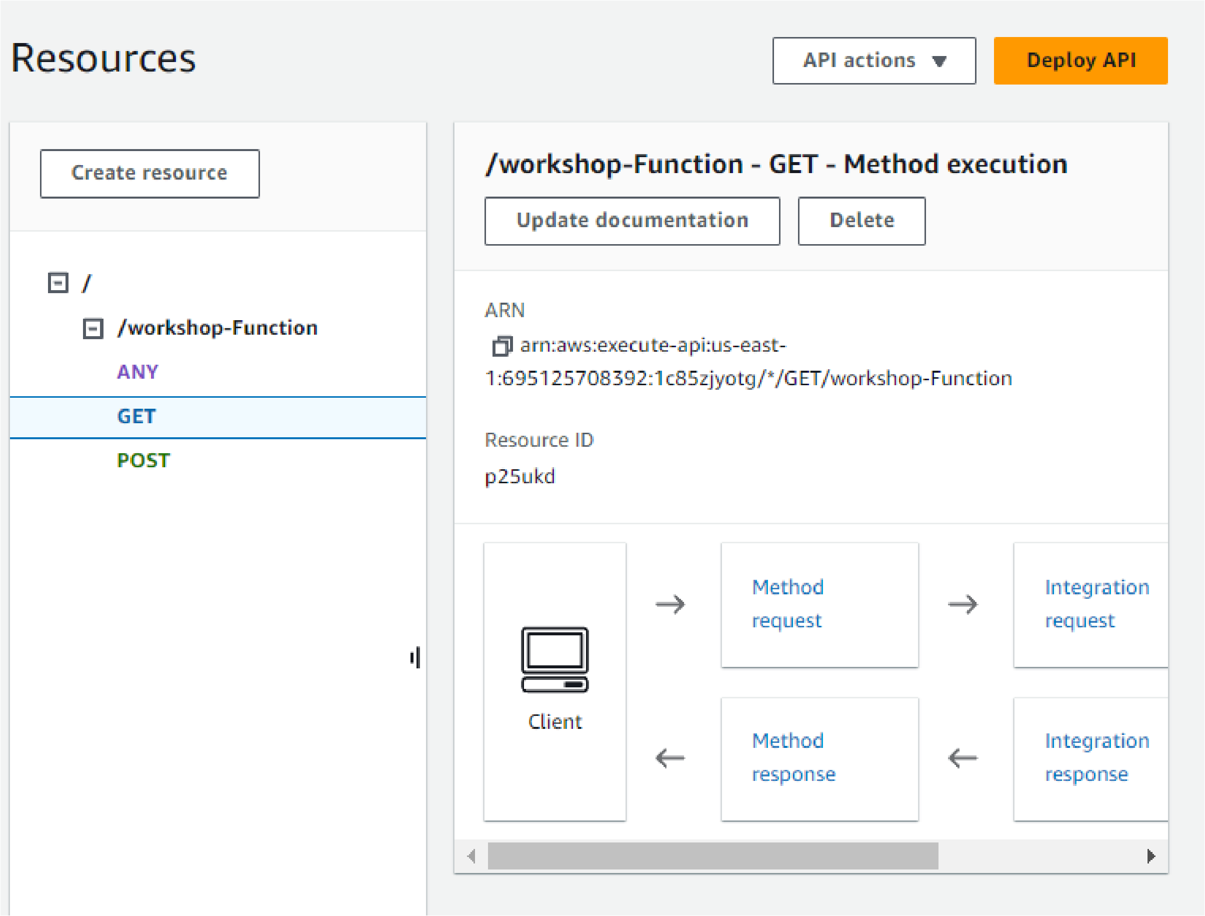
Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 ▼

arn:aws:lambda:us-east-1:695125708392:function:work ✕

ⓘ 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.



Resources

API actions ▼ | Deploy API

Create resource

- /
  - /workshop-Function
    - ANY
    - GET
    - POST

/workshop-Function - GET - Method execution

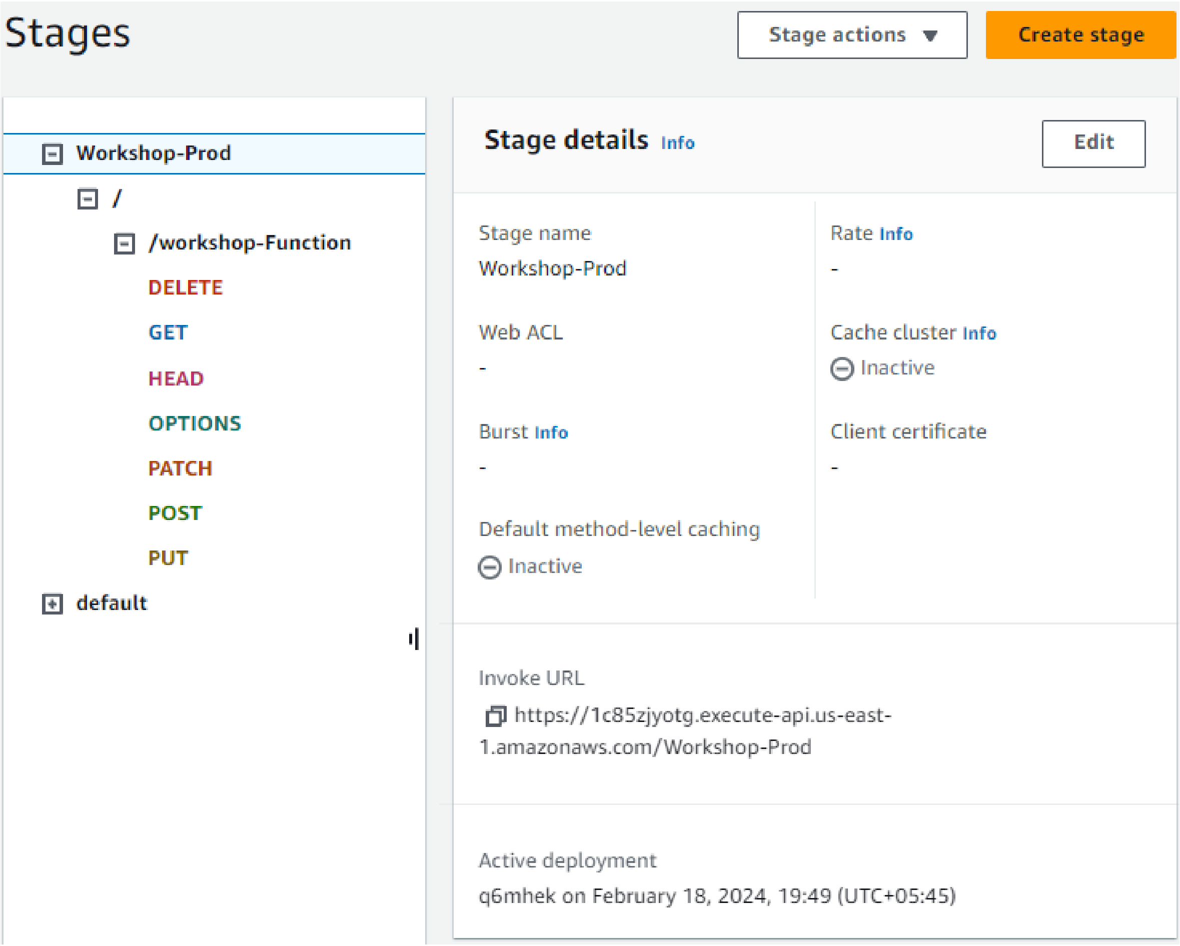
Update documentation | Delete

ARN

arn:aws:execute-api:us-east-1:695125708392:1c85zjyotg/*/GET/workshop-Function

Resource ID

p25ukd



Stage name
Workshop-Prod

Rate Info
-

Web ACL
-

Cache cluster Info
⊖ Inactive

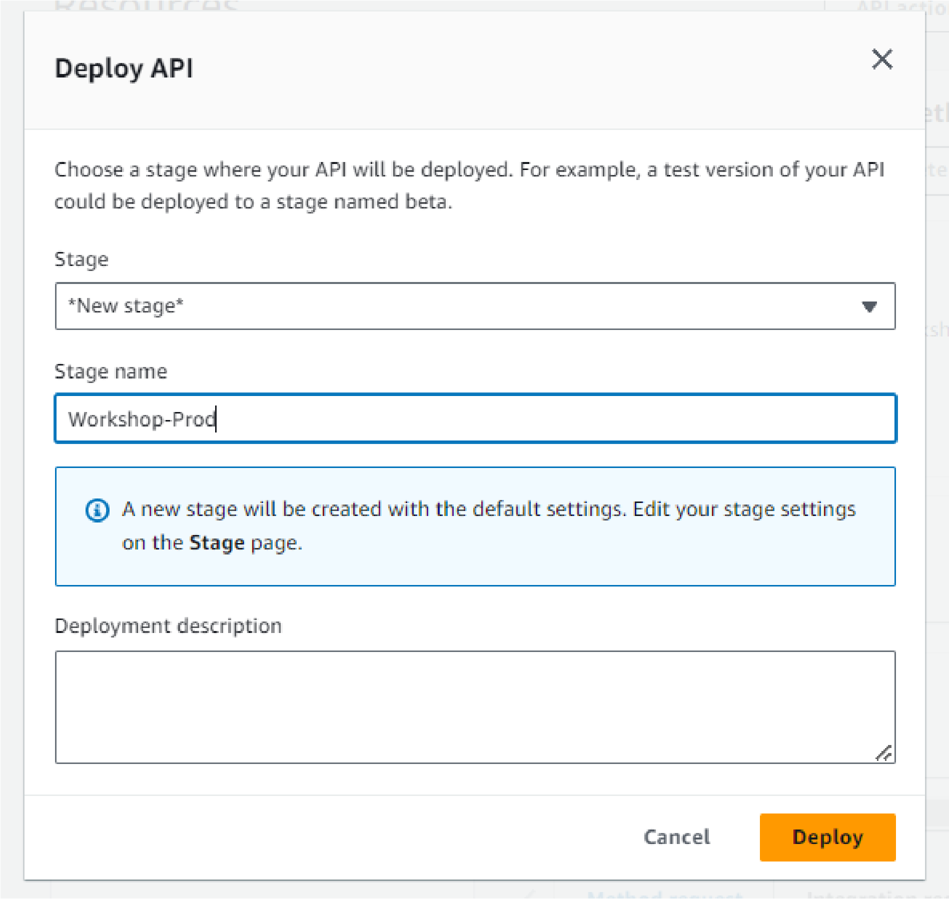
Burst Info
-

Client certificate
-

Default method-level caching
⊖ Inactive

Invoke URL
https://1c85zjyotg.execute-api.us-east-1.amazonaws.com/Workshop-Prod

Active deployment
q6mhek on February 18, 2024, 19:49 (UTC+05:45)



Deploy API

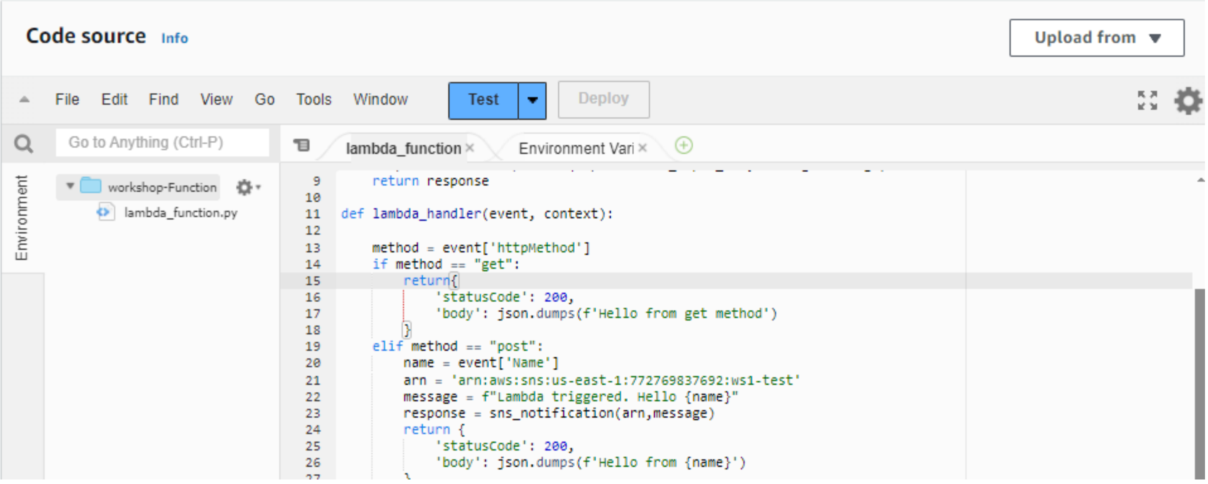
Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Stage: *New stage*

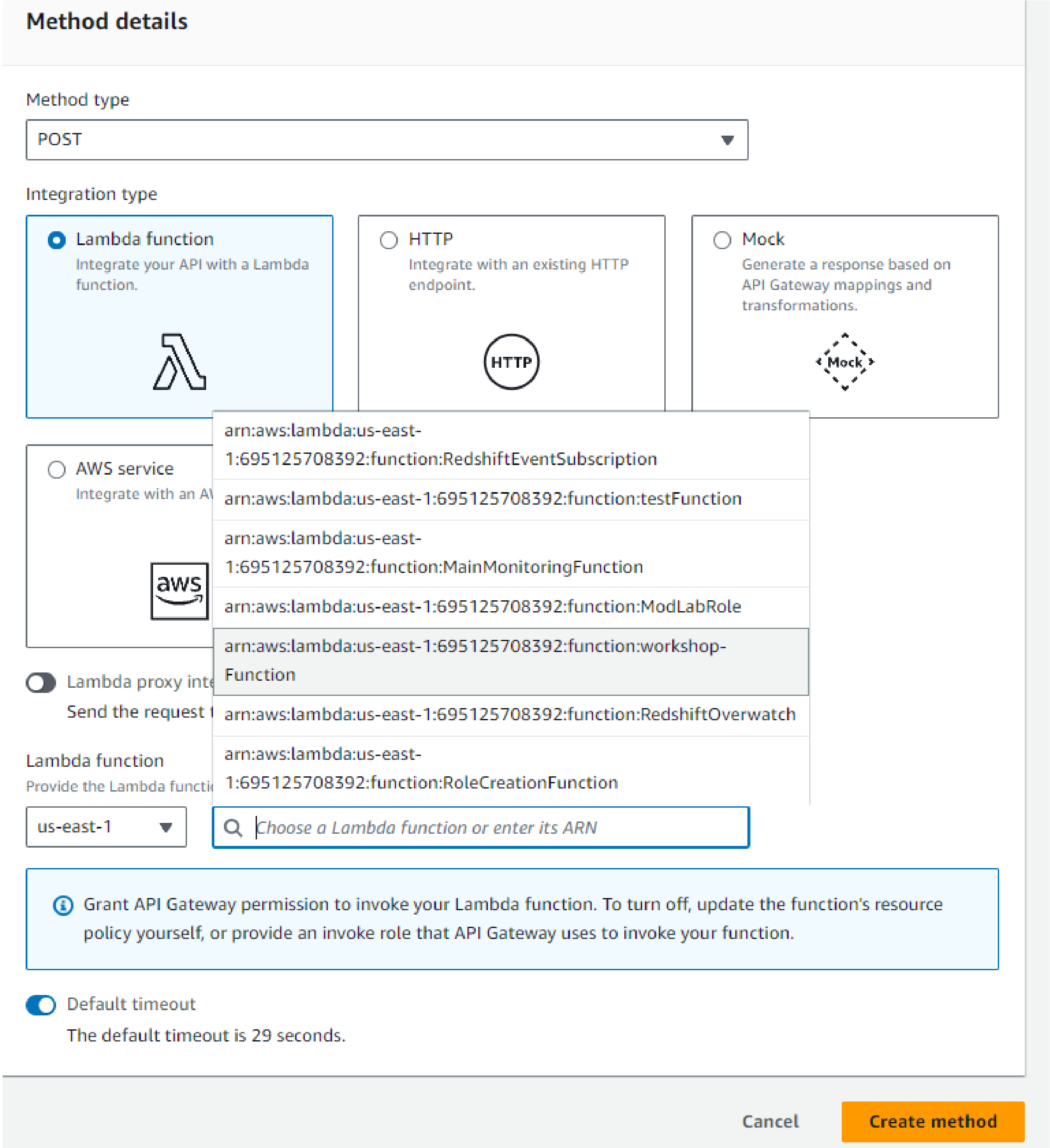
Stage name: Workshop-Prod

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

Deployment description:



```
    return response

def lambda_handler(event, context):

    method = event['httpMethod']
    if method == "get":
        return{
            'statusCode': 200,
            'body': json.dumps(f'Hello from get method')
        }
    elif method == "post":
        name = event['Name']
        arn = 'arn:aws:sns:us-east-1:772769837692:ws1-test'
        message = f"Lambda triggered. Hello {name}"
        response = sns_notification(arn,message)
        return {
            'statusCode': 200,
            'body': json.dumps(f'Hello from {name}')
```



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 A\

arn:aws:lambda:us-east-1:695125708392:function:RedshiftEventSubscription
arn:aws:lambda:us-east-1:695125708392:function:testFunction
arn:aws:lambda:us-east-1:695125708392:function:MainMonitoringFunction
arn:aws:lambda:us-east-1:695125708392:function:ModLabRole
arn:aws:lambda:us-east-1:695125708392:function:workshop-Function
arn:aws:lambda:us-east-1:695125708392:function:RedshiftOverwatch
arn:aws:lambda:us-east-1:695125708392:function:RoleCreationFunction

Lambda proxy inte
Send the request t

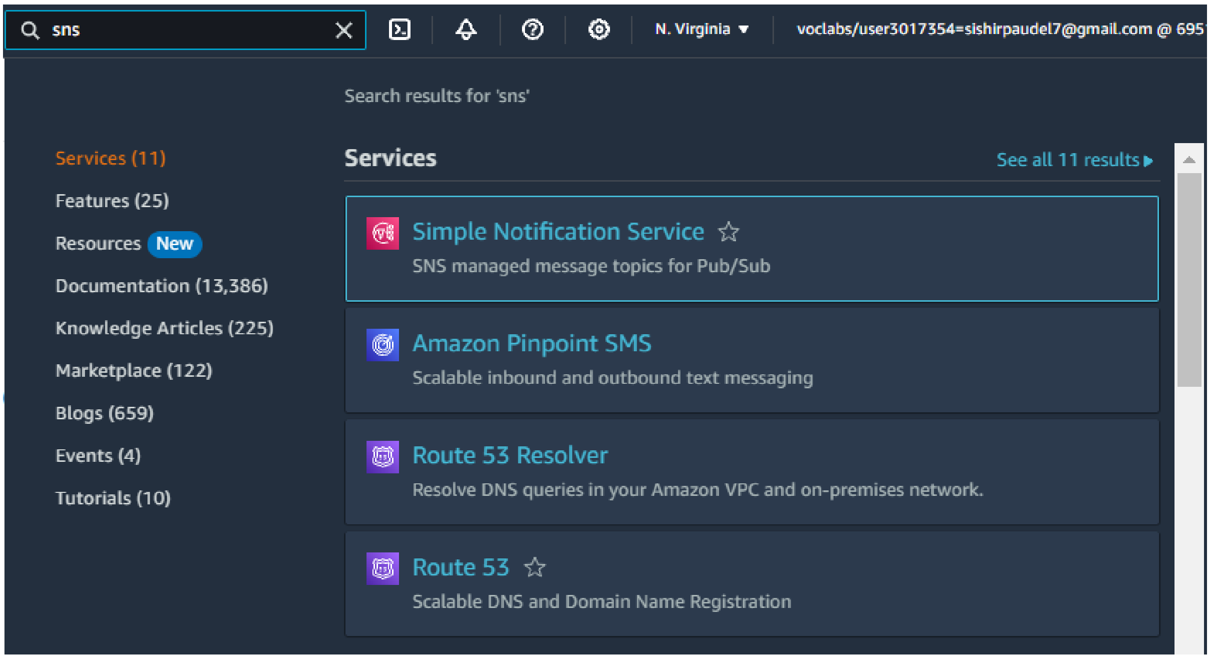
Lambda function
Provide the Lambda functi

us-east-1 ▼ | Choose a Lambda function or enter its ARN

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.

SNS



sns

N. Virginia ▼ voclabs/user3017354=sishirpaudel7@gmail.com @ 695

Search results for 'sns'

- Services (11)
- Features (25)
- Resources New
- Documentation (13,386)
- Knowledge Articles (225)
- Marketplace (122)
- Blogs (659)
- Events (4)
- Tutorials (10)

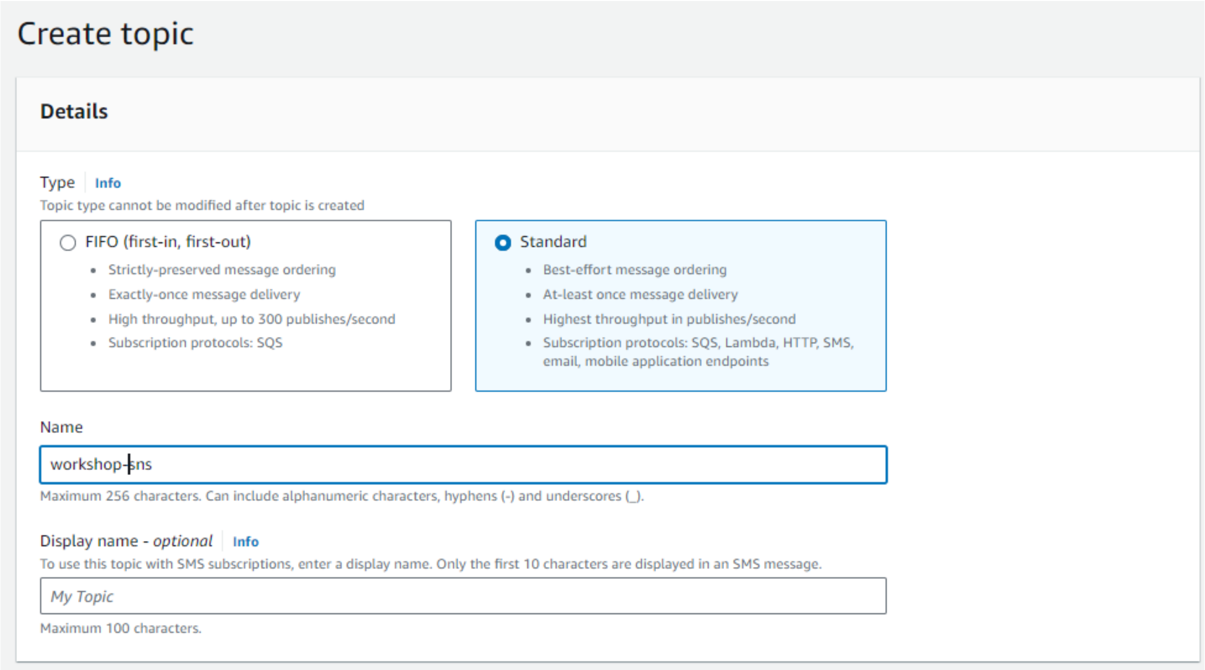
Services See all 11 results ▶

Simple Notification Service ☆
SNS managed message topics for Pub/Sub

Amazon Pinpoint SMS
Scalable inbound and outbound text messaging

Route 53 Resolver
Resolve DNS queries in your Amazon VPC and on-premises network.

Route 53 ☆
Scalable DNS and Domain Name Registration



Create topic

Details

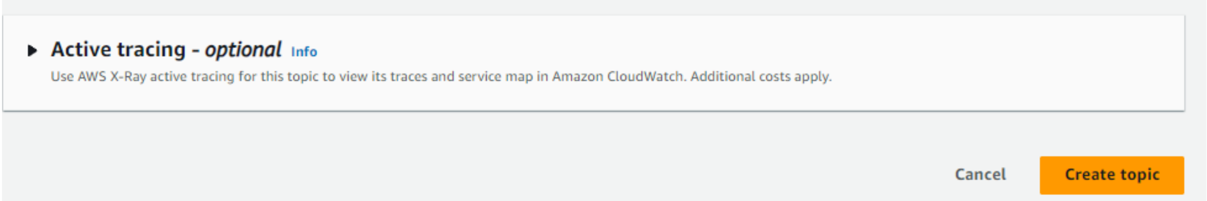
Type Info
Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

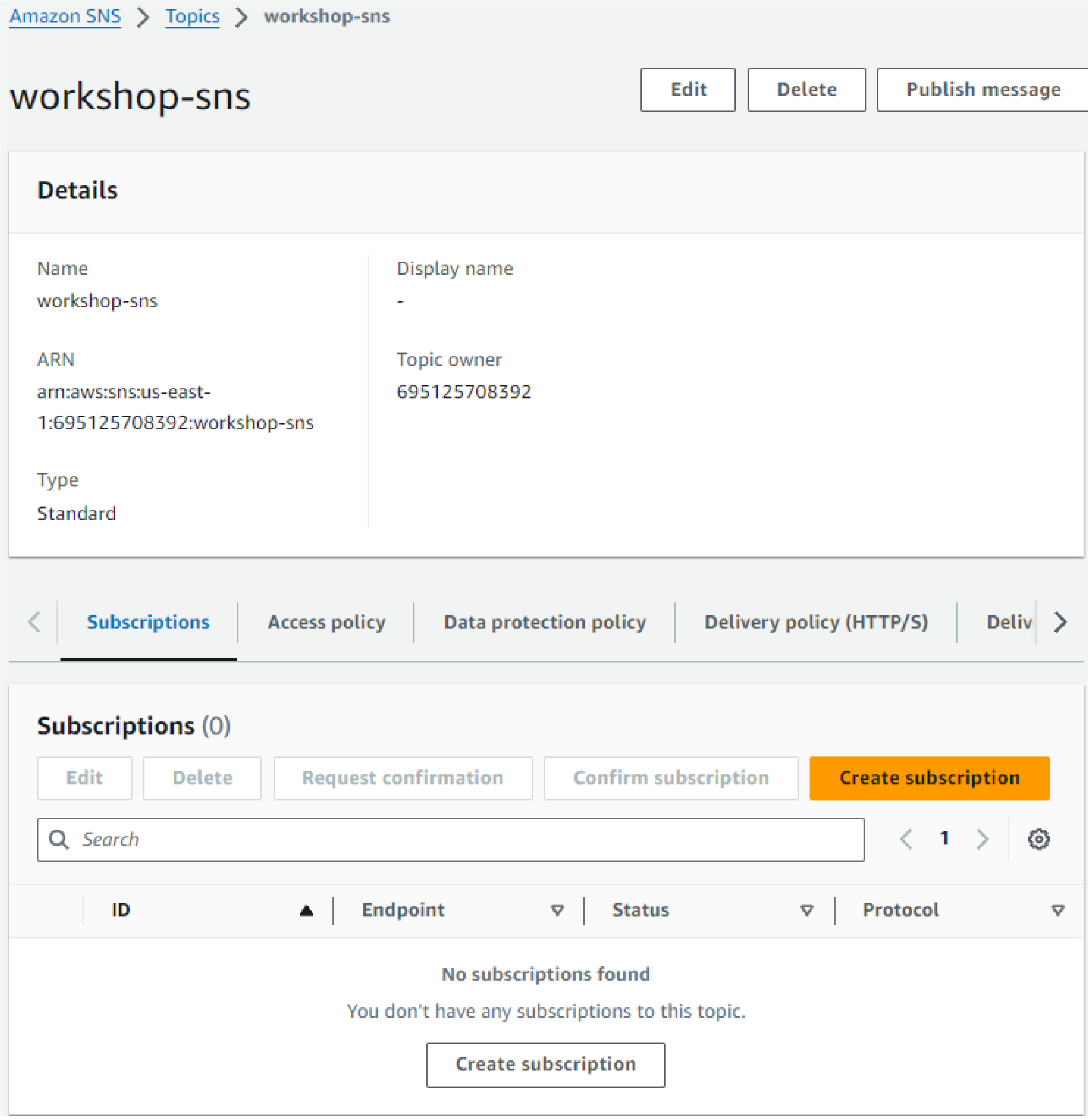
● Standard
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name
workshop-sns
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* Info
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.
My Topic
Maximum 100 characters.



▶ **Active tracing - *optional*** Info
Use AWS X-Ray active tracing for this topic to view its traces and service map in Amazon CloudWatch. Additional costs apply.



Amazon SNS > Topics > workshop-sns

workshop-sns

Details

Name
workshop-sns

ARN
arn:aws:sns:us-east-1:695125708392:workshop-sns

Type
Standard

Display name
-

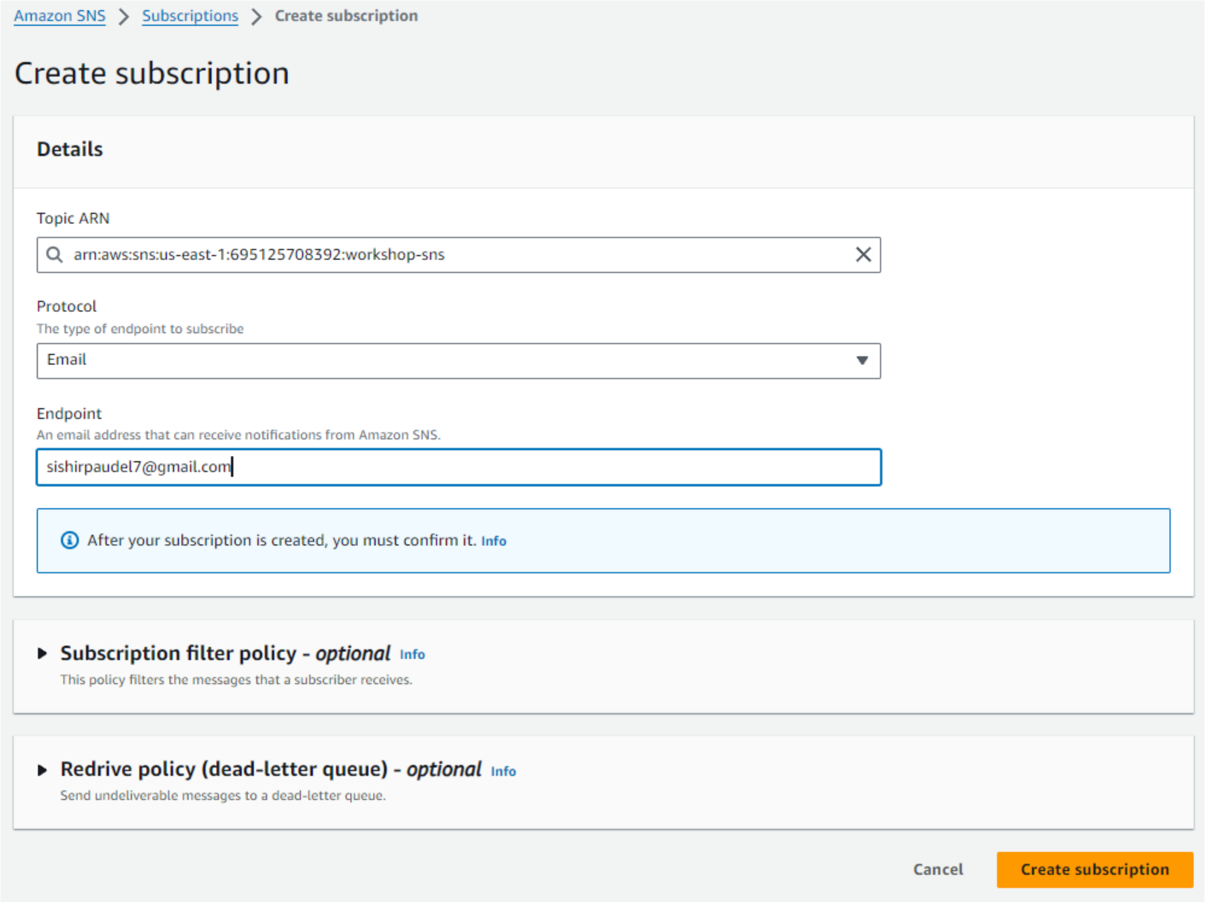
Topic owner
695125708392

Subscriptions (0)

ID	Endpoint	Status	Protocol

No subscriptions found

You don't have any subscriptions to this topic.



Create subscription

Details

Topic ARN: arn:aws:sns:us-east-1:695125708392:workshop-sns

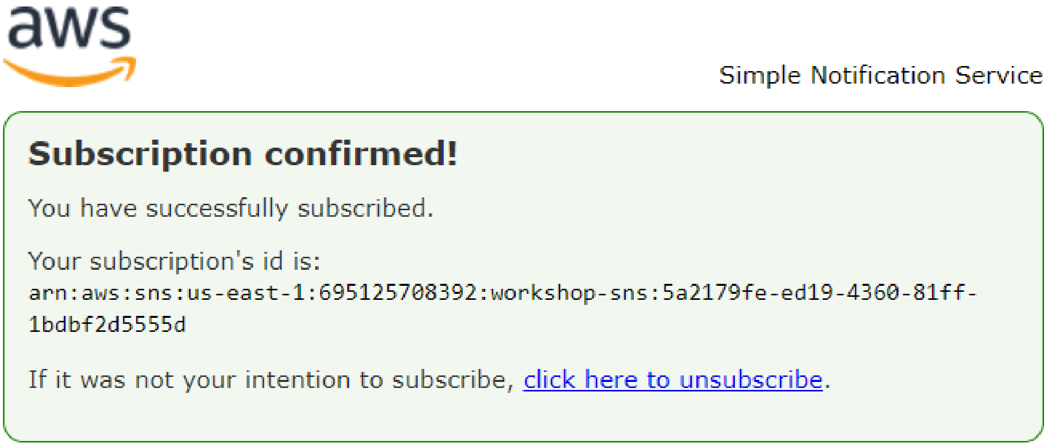
Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
sishirpaudel7@gmail.com

After your subscription is created, you must confirm it. Info

▶ Subscription filter policy - *optional* Info
This policy filters the messages that a subscriber receives.

▶ Redrive policy (dead-letter queue) - *optional* Info
Send undeliverable messages to a dead-letter queue.



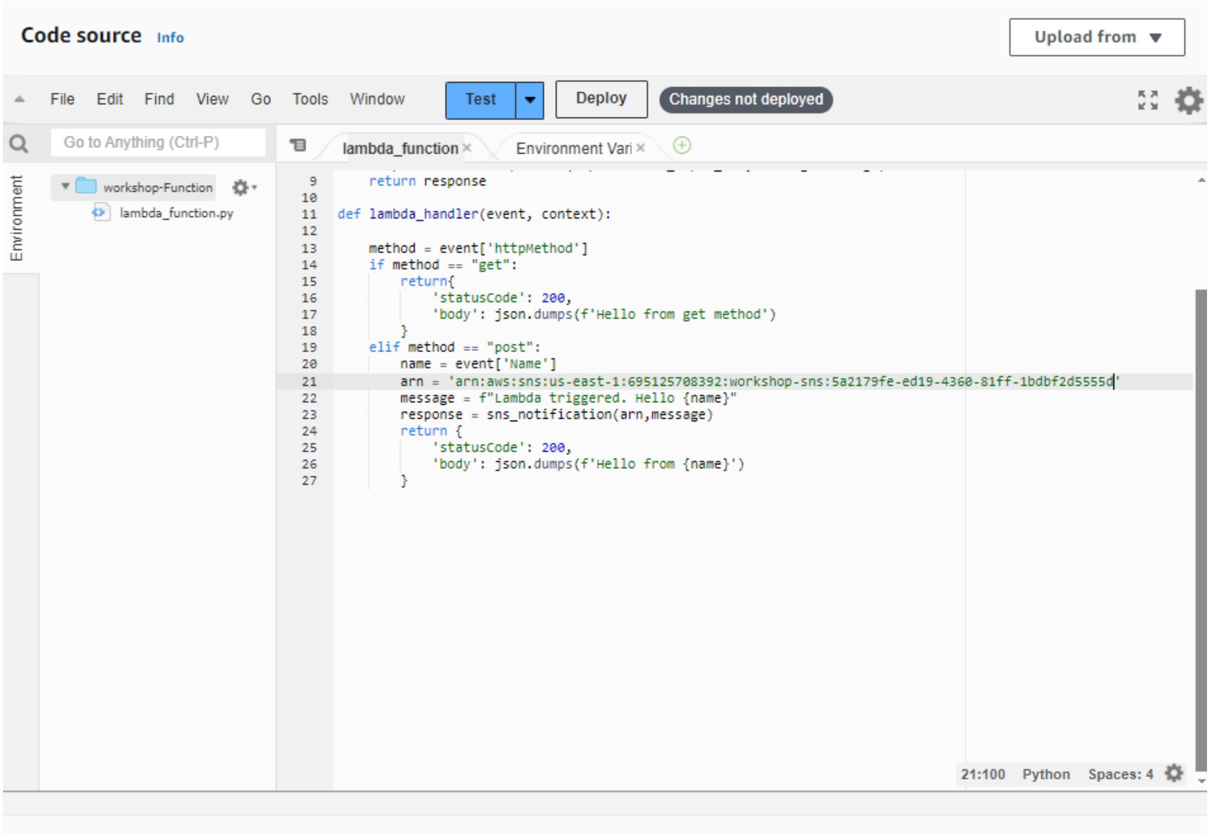
Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

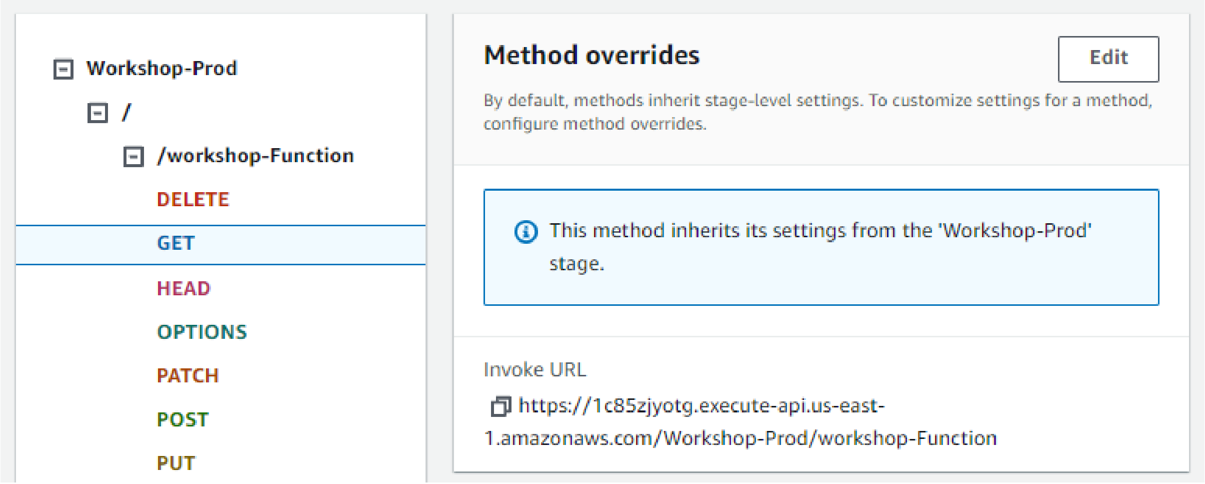
Your subscription's id is:
arn:aws:sns:us-east-1:695125708392:workshop-sns:5a2179fe-ed19-4360-81ff-1bdbf2d5555d

If it was not your intention to subscribe, click here to unsubscribe.



```python
    return response

def lambda_handler(event, context):

    method = event['httpMethod']
    if method == "get":
        return{
            'statusCode': 200,
            'body': json.dumps(f'Hello from get method')
        }
    elif method == "post":
        name = event['Name']
        arn = 'arn:aws:sns:us-east-1:695125708392:workshop-sns:5a2179fe-ed19-4360-81ff-1bdbf2d5555d'
        message = f"Lambda triggered. Hello {name}"
        response = sns_notification(arn,message)
        return {
            'statusCode': 200,
            'body': json.dumps(f'Hello from {name}')
        }
```

Copy the arn from aws sns and paste it into the source code.

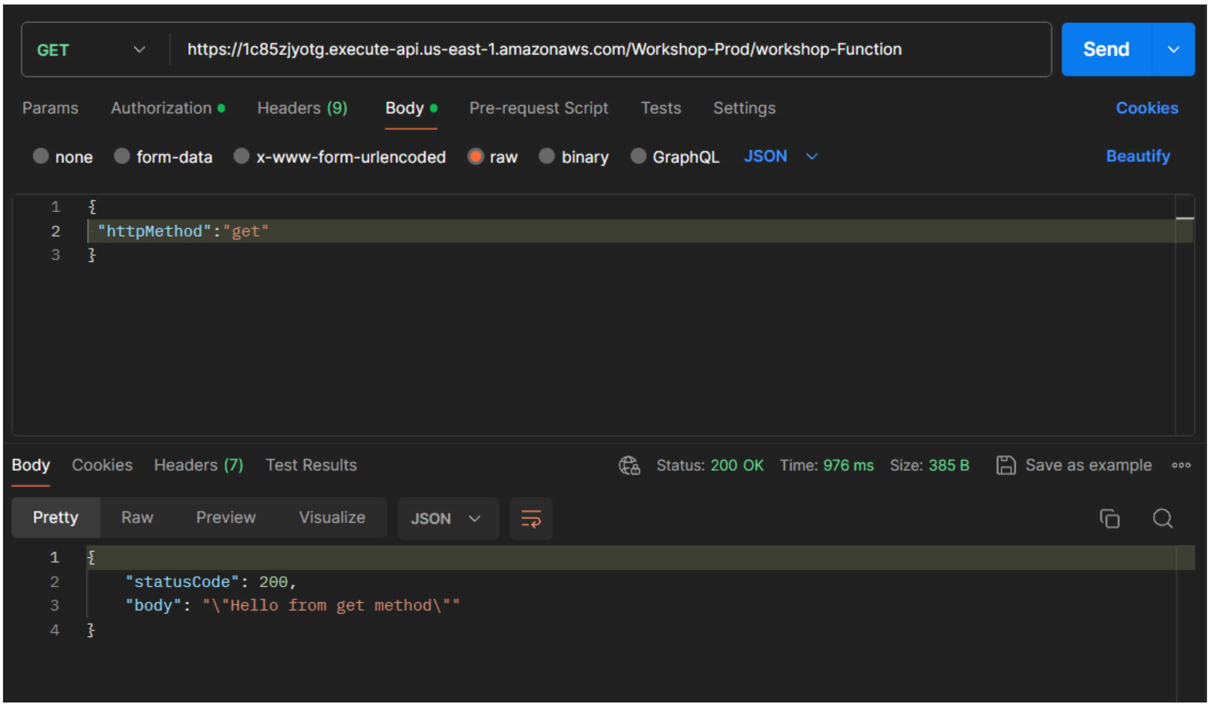


By default, methods inherit stage-level settings. To customize settings for a method, configure method overrides.

This method inherits its settings from the 'Workshop-Prod' stage.

Invoke URL

https://1c85zjyotg.execute-api.us-east-1.amazonaws.com/Workshop-Prod/workshop-Function



GET https://1c85zjyotg.execute-api.us-east-1.amazonaws.com/Workshop-Prod/workshop-Function

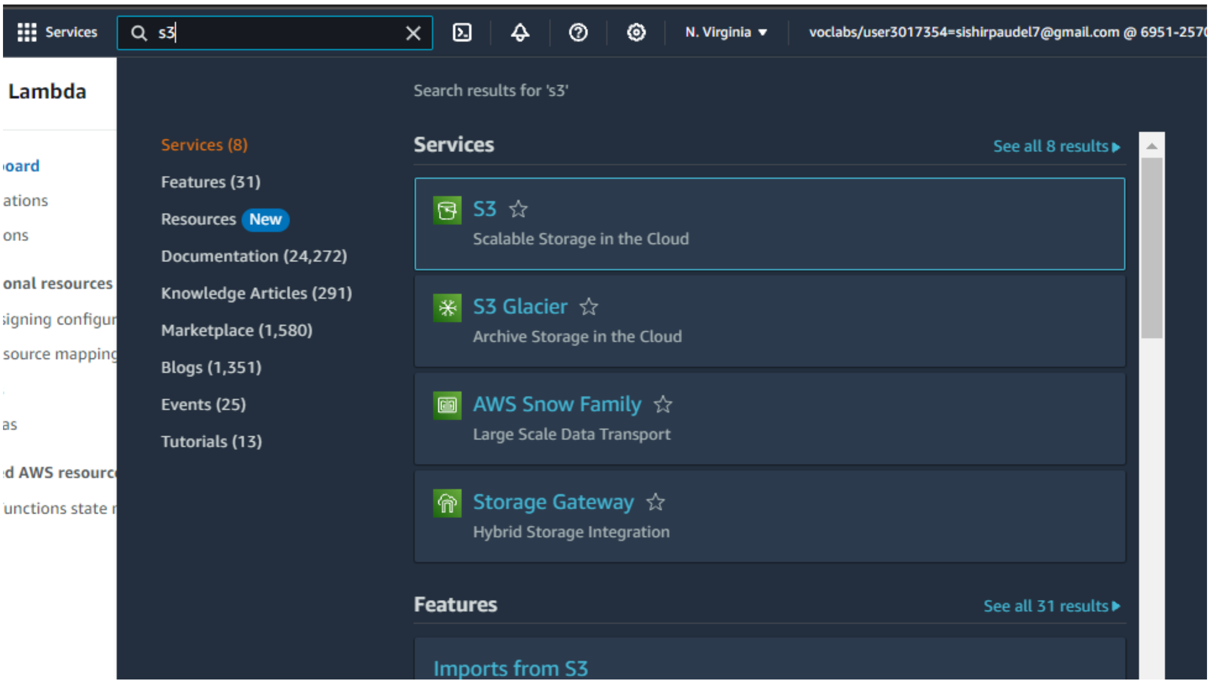
```
{
"httpMethod":"get"
}
```

Status: 200 OK Time: 976 ms Size: 385 B

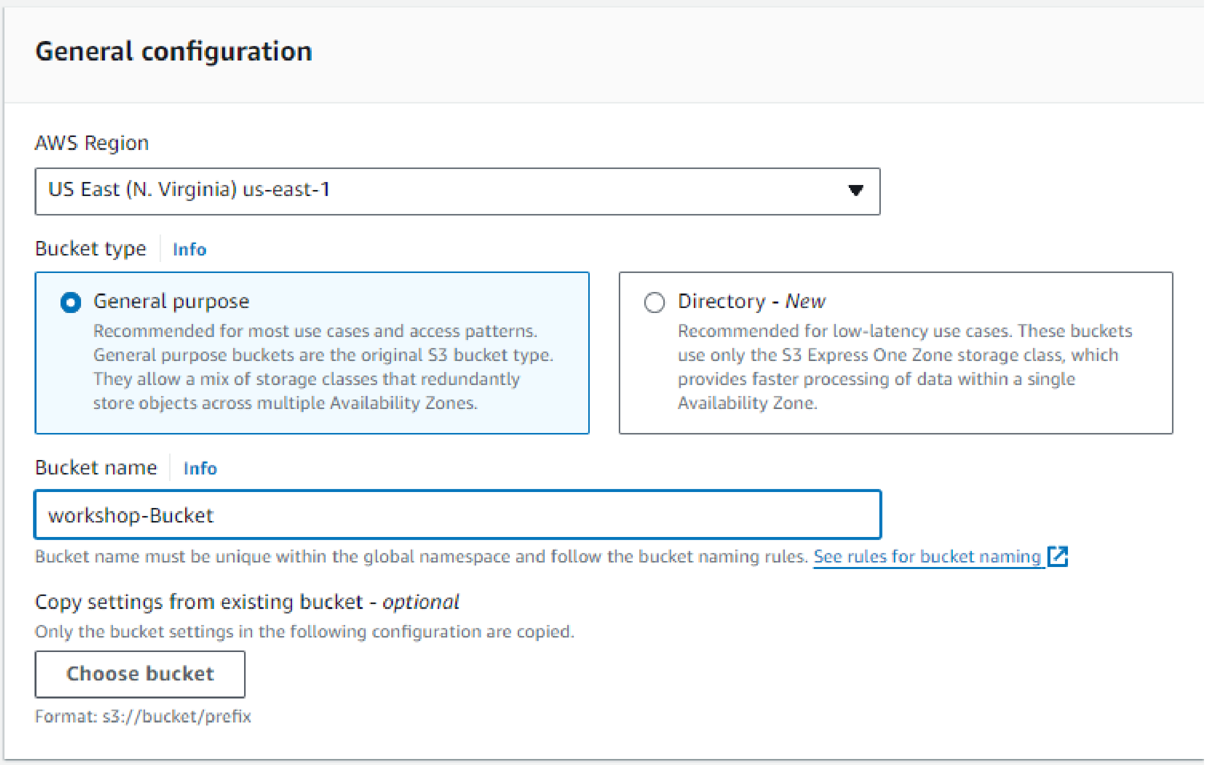
JSON

```
{
    "statusCode": 200,
    "body": "\"Hello from get method\""
}
```

S3 Bucket



Create a bucket



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

workshop-Bucket

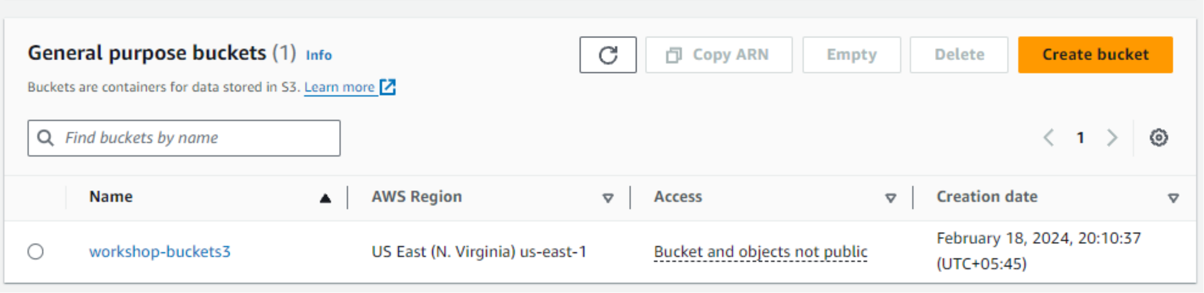
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

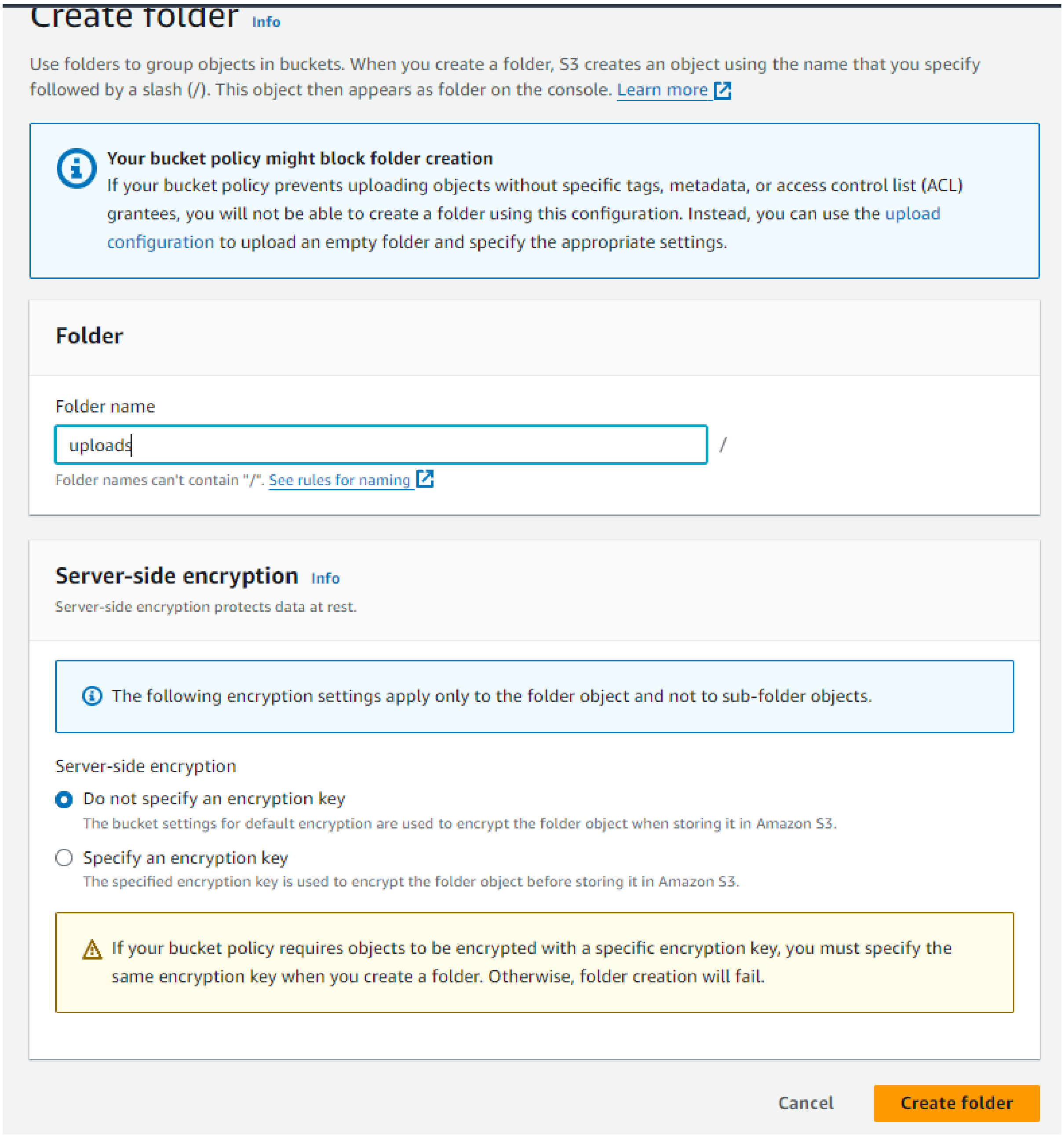
Format: s3://bucket/prefix



General purpose buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name ▲	AWS Region ▽	Access ▽	Creation date ▽
○	workshop-buckets3	US East (N. Virginia) us-east-1	Bucket and objects not public	February 18, 2024, 20:10:37 (UTC+05:45)



Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. Learn more

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.

Folder

Folder name

uploads /

Folder names can't contain "/". See rules for naming

Server-side encryption Info

Server-side encryption protects data at rest.

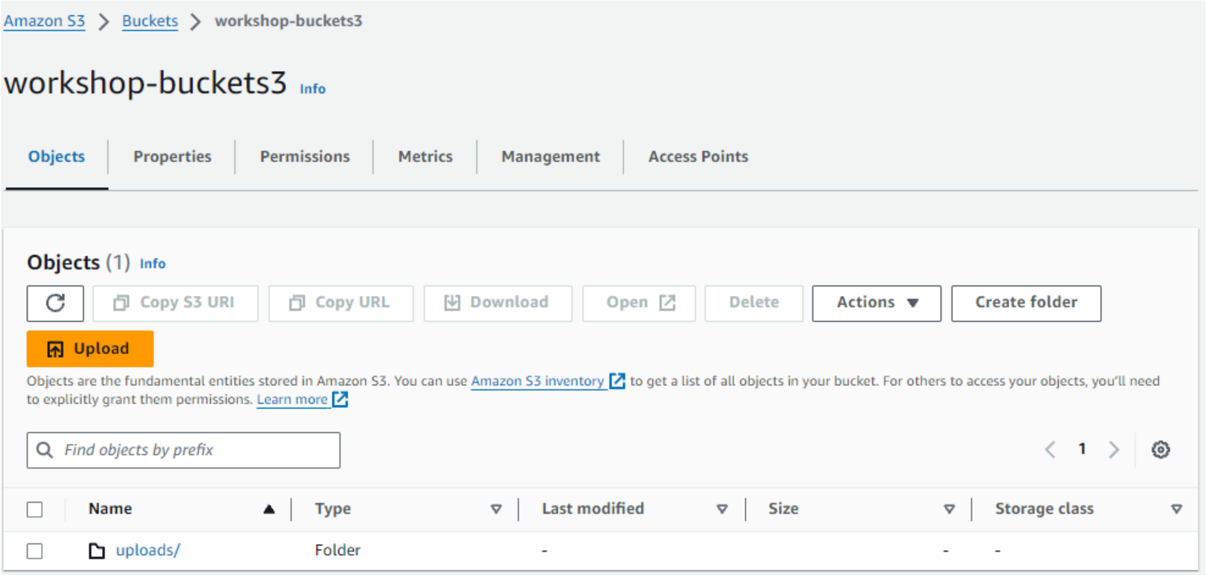
The following encryption settings apply only to the folder object and not to sub-folder objects.

Server-side encryption

● Do not specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

○ Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

⚠ If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail.

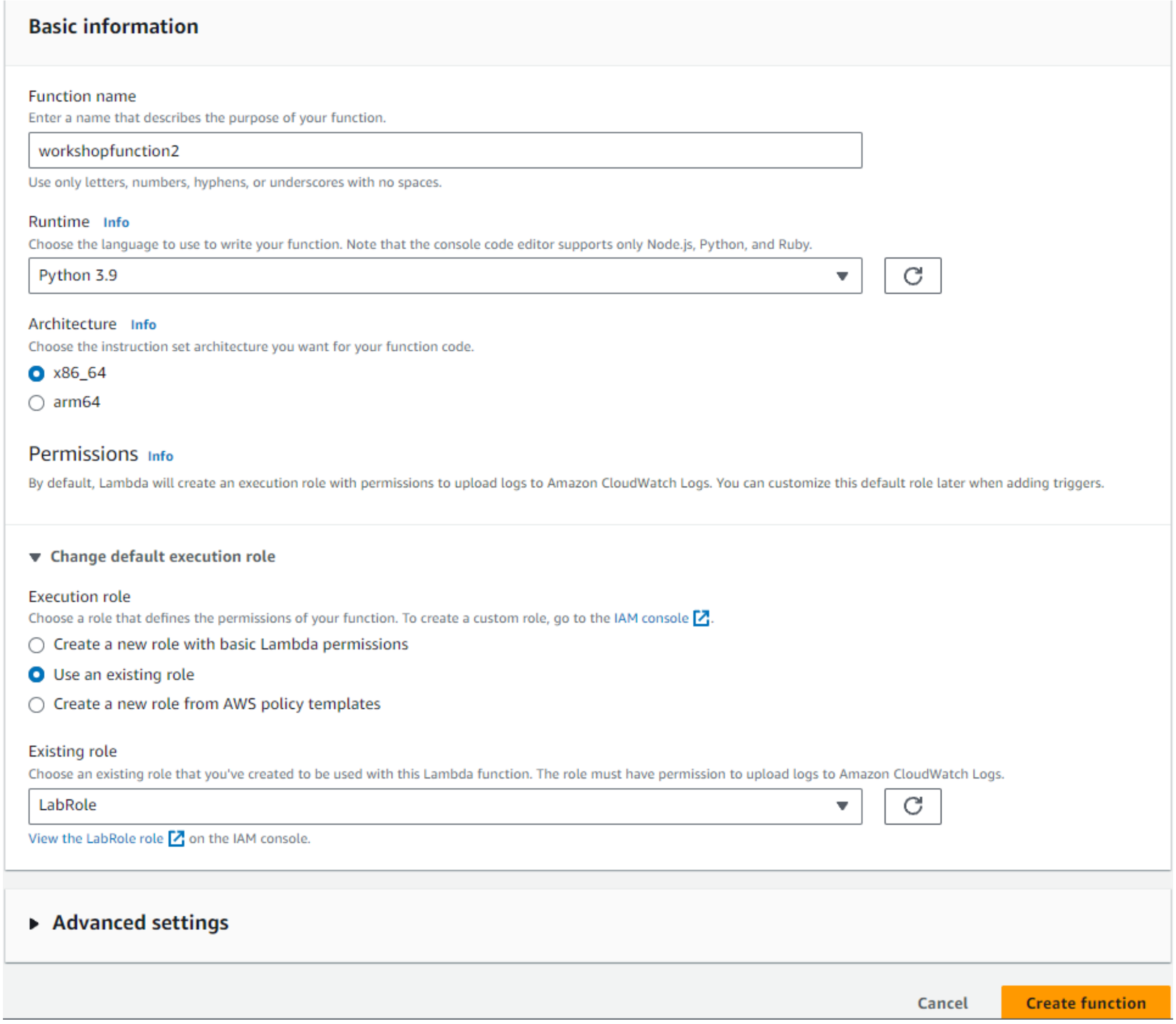


Amazon S3 > Buckets > workshop-buckets3
workshop-buckets3 Info
Objects | Properties | Permissions | Metrics | Management | Access Points
Objects (1) Info
Copy S3 URI | Copy URL | Download | Open | Delete | Actions ▼ | Create folder
Upload

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
□	uploads/	Folder	-	-	-



Basic information

Function name
Enter a name that describes the purpose of your function.
workshopfunction2
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 3.9 ▼

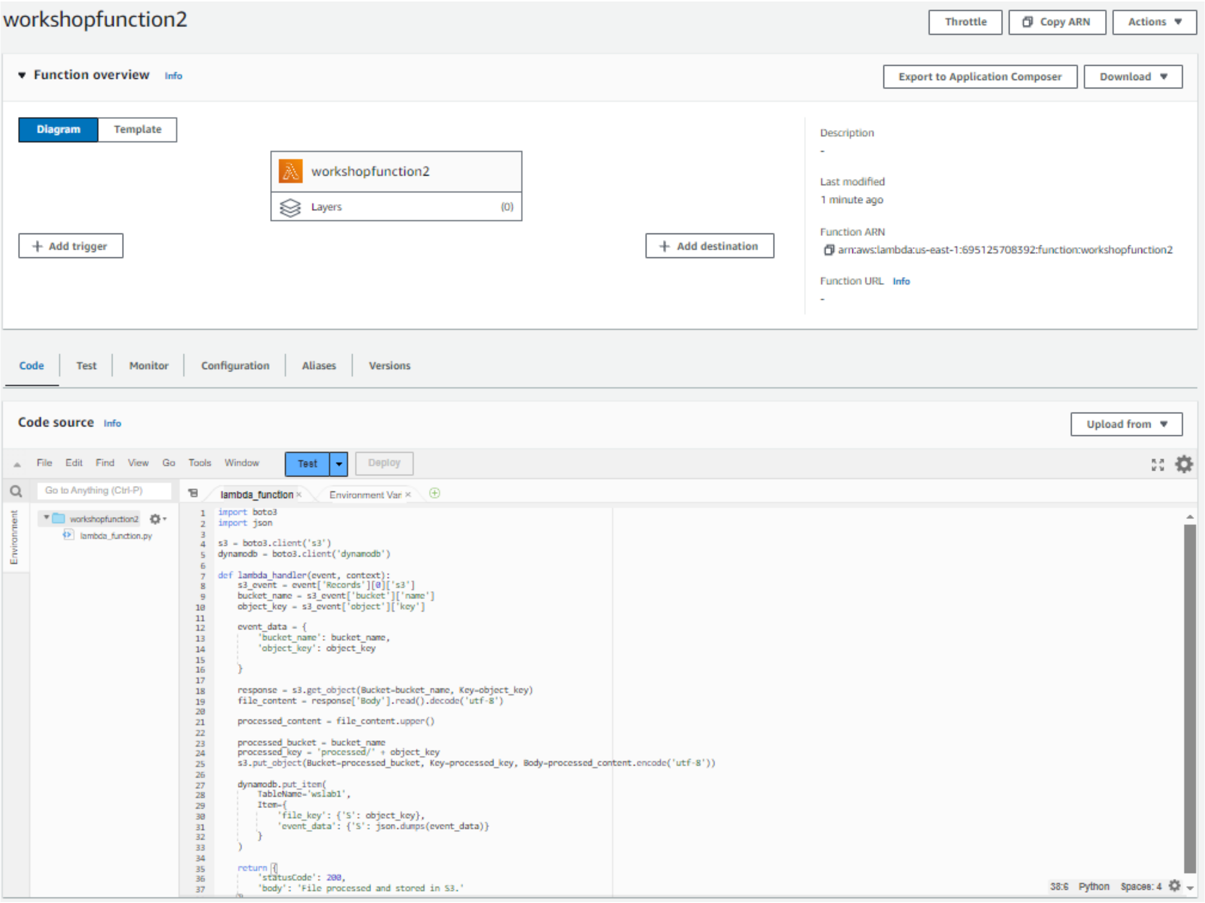
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

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.



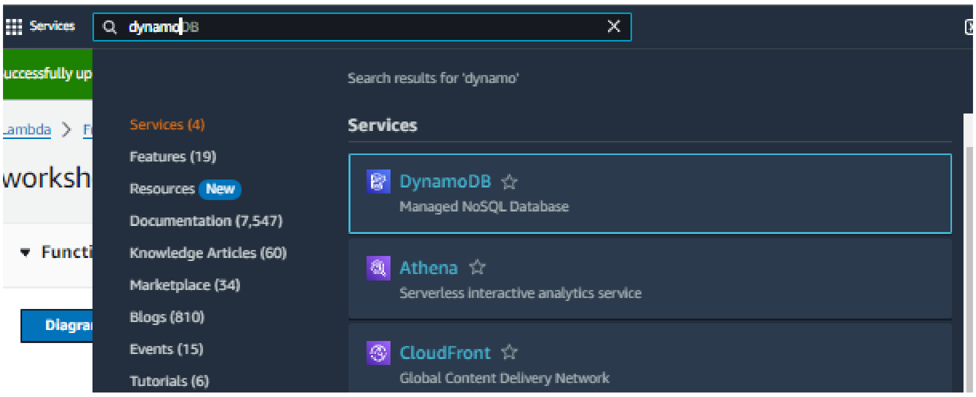
Description
-

Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:695125708392:function:workshopfunction2

Function URL Info
-

```
import boto3
import json

s3 = boto3.client('s3')
dynamodb = boto3.client('dynamodb')

def lambda_handler(event, context):
    s3_event = event['Records'][0]['s3']
    bucket_name = s3_event['bucket']['name']
    object_key = s3_event['object']['key']

    event_data = {
        'bucket_name': bucket_name,
        'object_key': object_key

    }

    response = s3.get_object(Bucket=bucket_name, Key=object_key)
    file_content = response['Body'].read().decode('utf-8')

    processed_content = file_content.upper()

    processed_bucket = bucket_name
    processed_key = 'processed/' + object_key
    s3.put_object(Bucket=processed_bucket, Key=processed_key, Body=processed_content.encode('utf-8'))

    dynamodb.put_item(
        TableName='wslab1',
        Item={
            'file_key': {'S': object_key},
            'event_data': {'S': json.dumps(event_data)}
        }
    )

    return {
        'statusCode': 200,
        'body': 'File processed and stored in S3.'
```



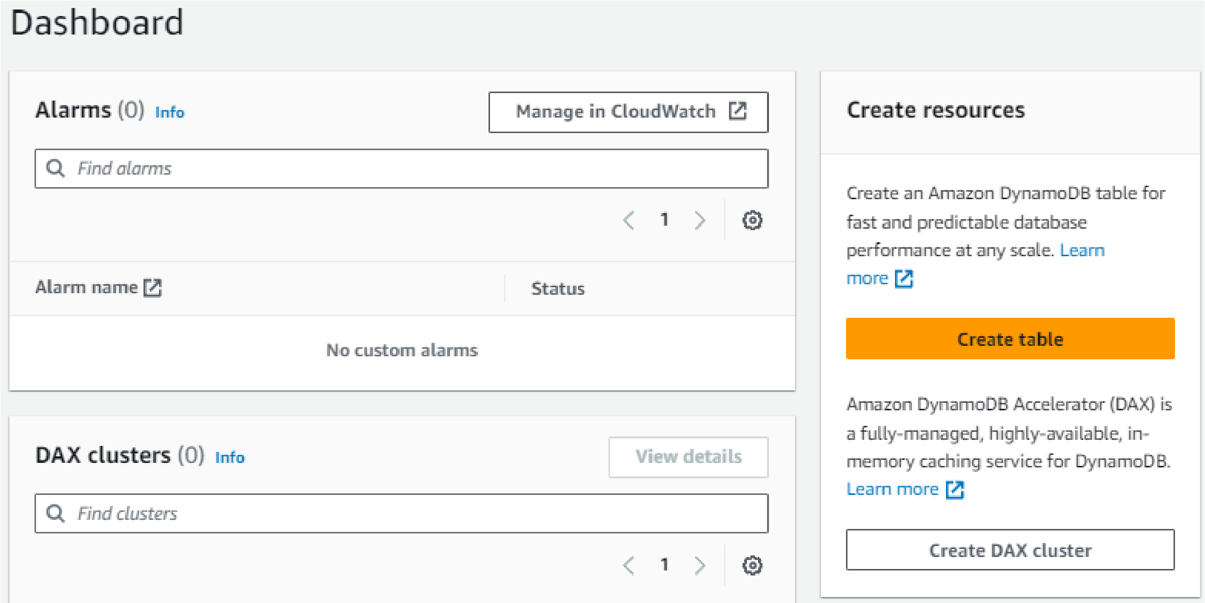
Search results for 'dynamo'

DynamoDB
Managed NoSQL Database

Athena
Serverless interactive analytics service

CloudFront
Global Content Delivery Network

Creating dynamo DB



Dashboard

Alarms (0) Info

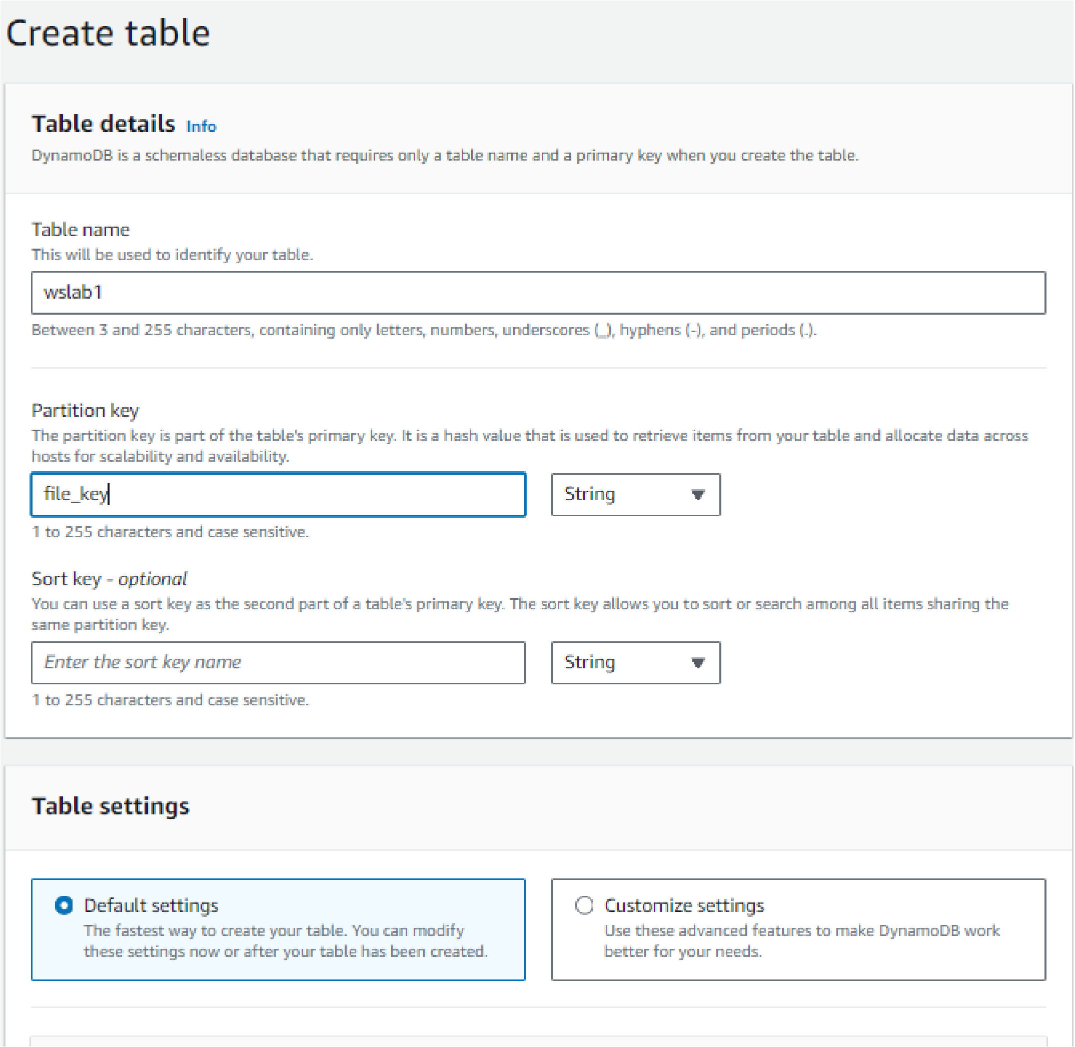
Alarm name	Status
No custom alarms	

DAX clusters (0) Info

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



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.

wslab1

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.

file_key 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.

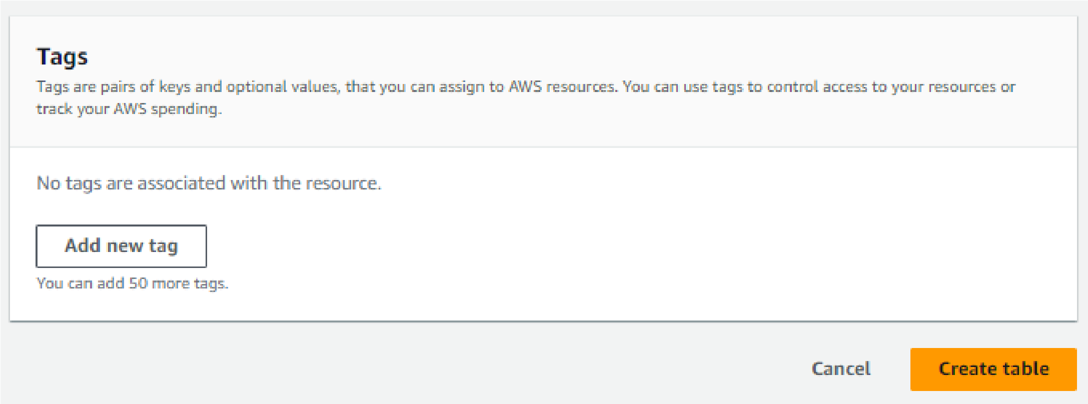
String ▼

1 to 255 characters and case sensitive.

Table settings

● Default settings
The fastest way to create your table. You can modify these settings now or after your table has been created.

○ Customize settings
Use these advanced features to make DynamoDB work better for your needs.

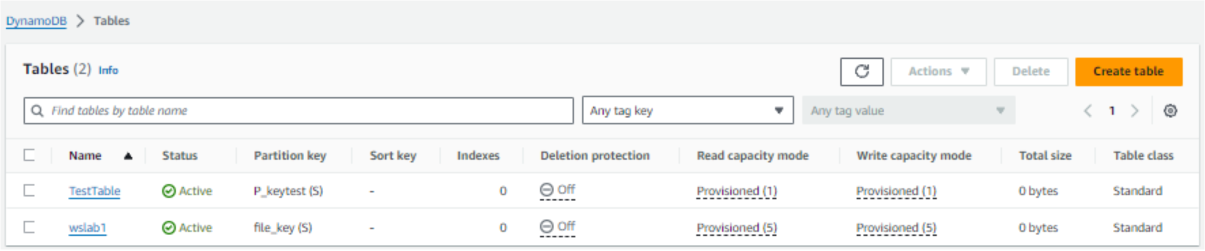


Tags

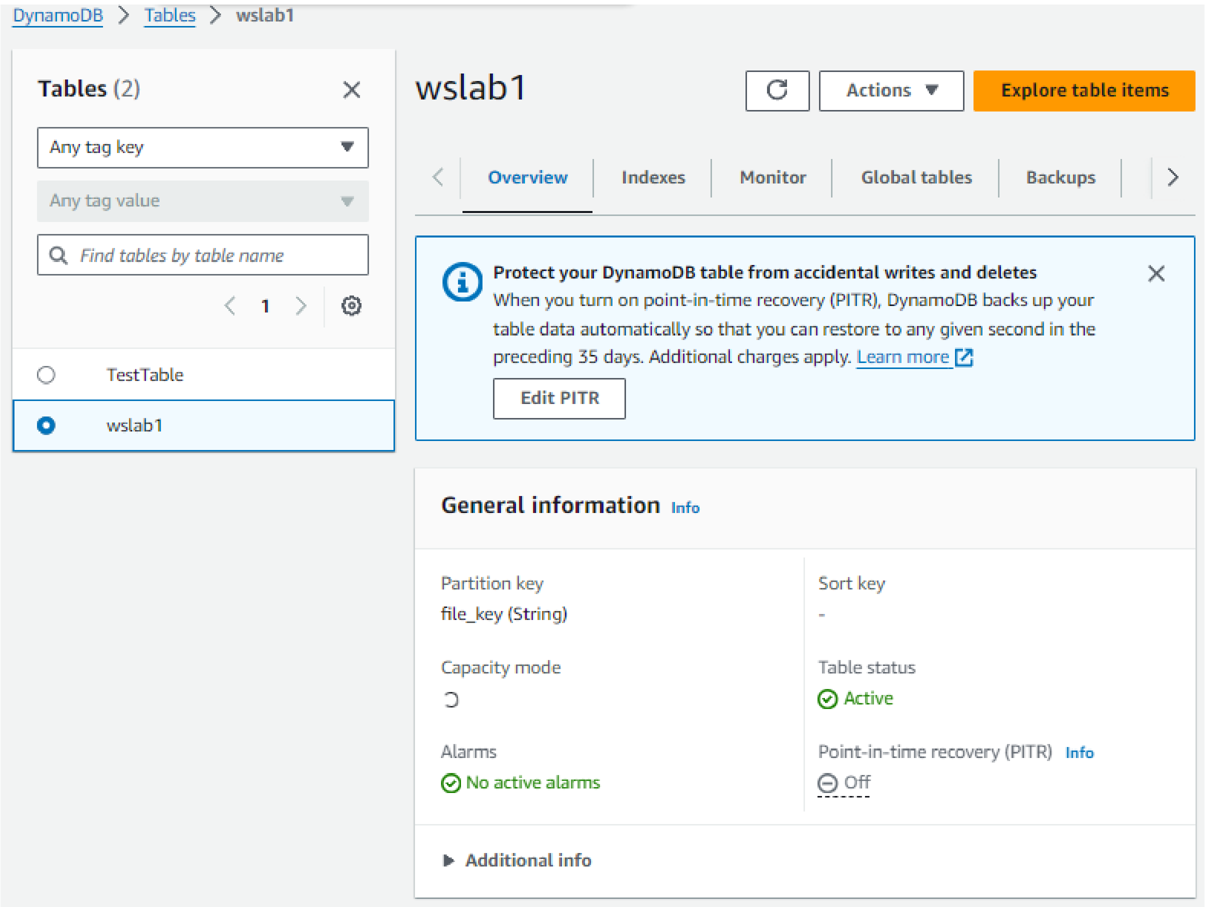
Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

You can add 50 more tags.



☐	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size	Table class
☐	TestTable	✓ Active	P_keytest (S)	-	0	⊖ Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
☐	wslab1	✓ Active	file_key (S)	-	0	⊖ Off	Provisioned (5)	Provisioned (5)	0 bytes	Standard



○ TestTable

● wslab1

wslab1

Protect your DynamoDB table from accidental writes and deletes
When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. Learn more

General information Info

Partition key
file_key (String)

Sort key
-

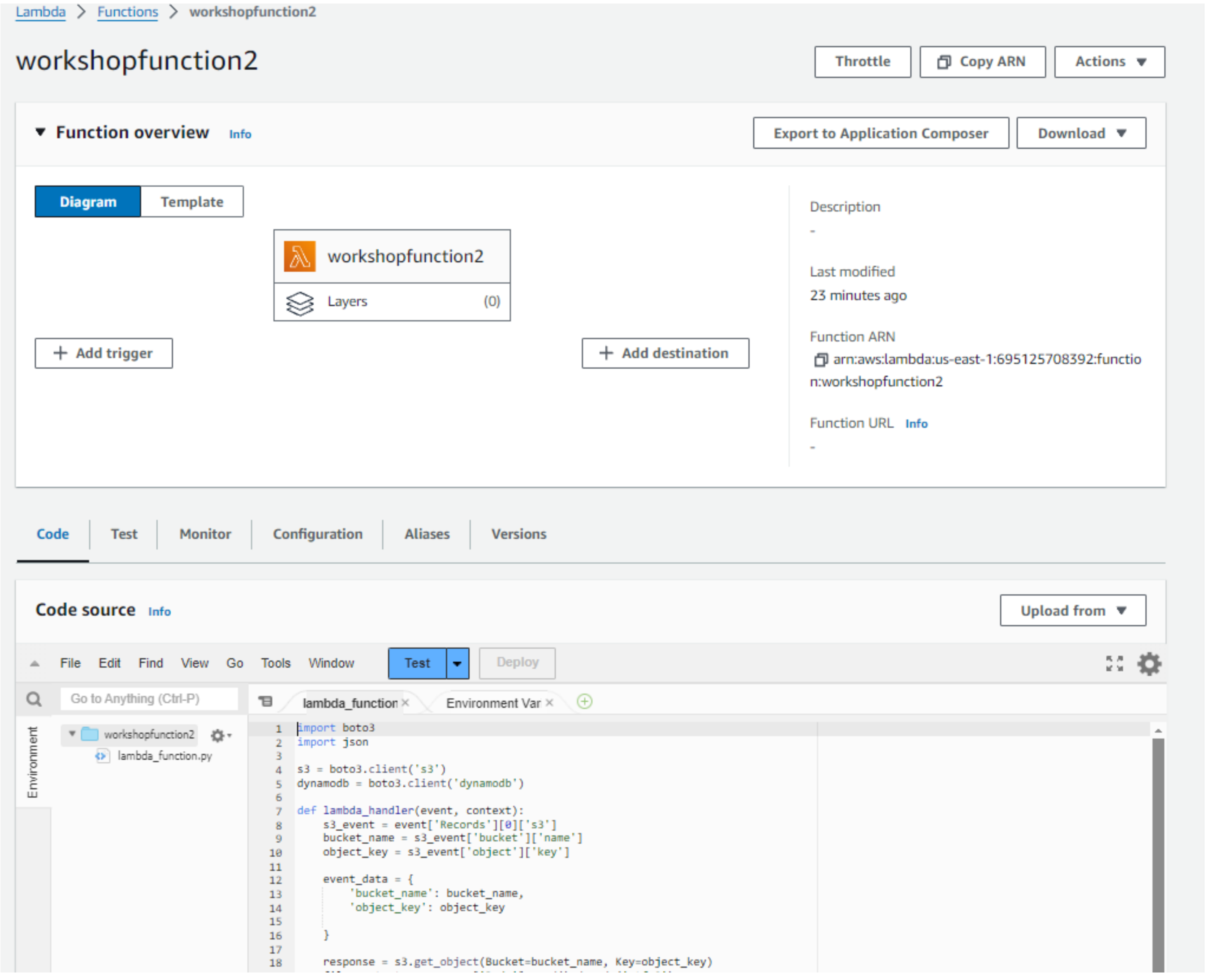
Capacity mode

Table status
✓ Active

Alarms
✓ No active alarms

Point-in-time recovery (PITR) Info
⊖ Off

▶ Additional info



Description

-

Last modified

23 minutes ago

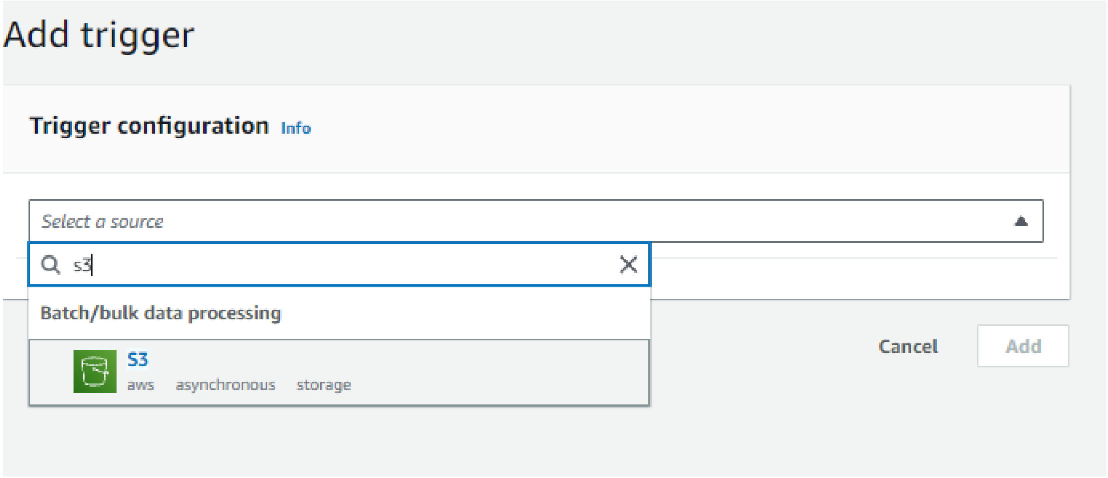
Function ARN

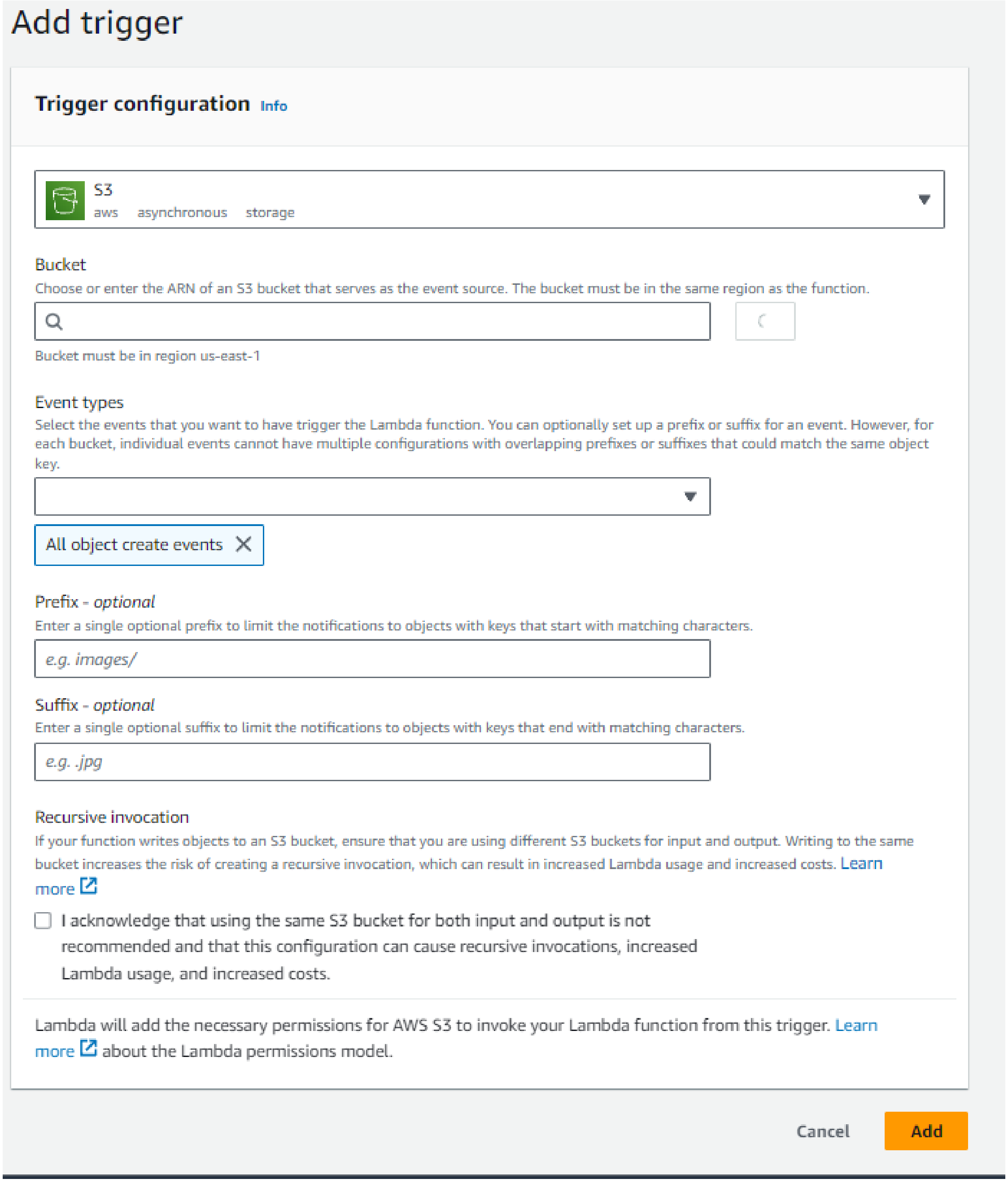
arn:aws:lambda:us-east-1:695125708392:function:workshopfunction2

Function URL Info

-

```
import boto3
import json

s3 = boto3.client('s3')
dynamodb = boto3.client('dynamodb')

def lambda_handler(event, context):
    s3_event = event['Records'][0]['s3']
    bucket_name = s3_event['bucket']['name']
    object_key = s3_event['object']['key']

    event_data = {
        'bucket_name': bucket_name,
        'object_key': object_key

    }

    response = s3.get_object(Bucket=bucket_name, Key=object_key)
```





Add trigger

Trigger configuration Info

S3
aws asynchronous storage

Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Bucket must be in region us-east-1

Event types

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events ✕

Prefix - *optional*

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - *optional*

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

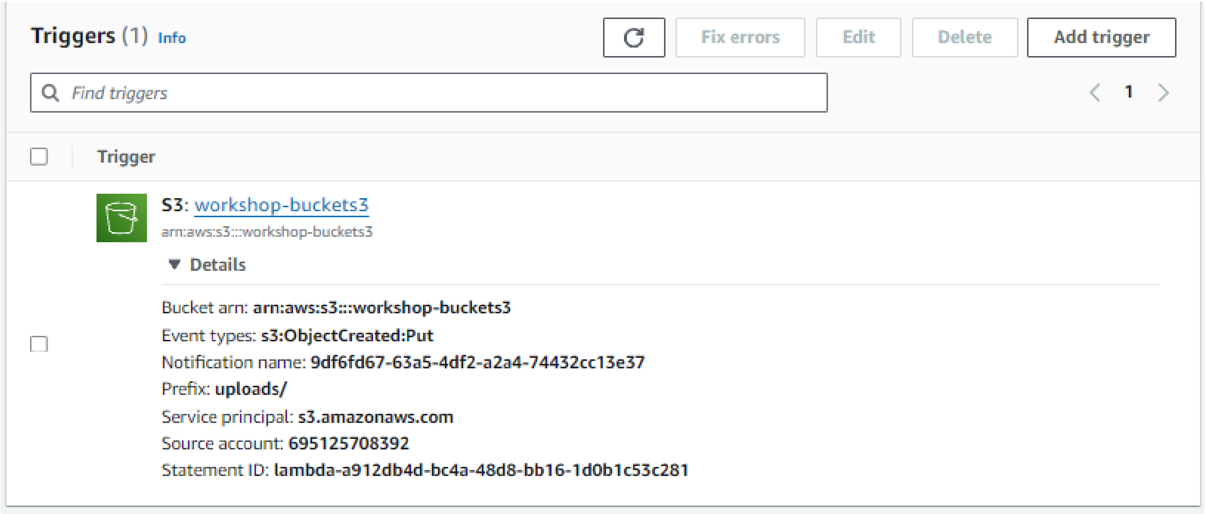
e.g. .jpg

Recursive invocation

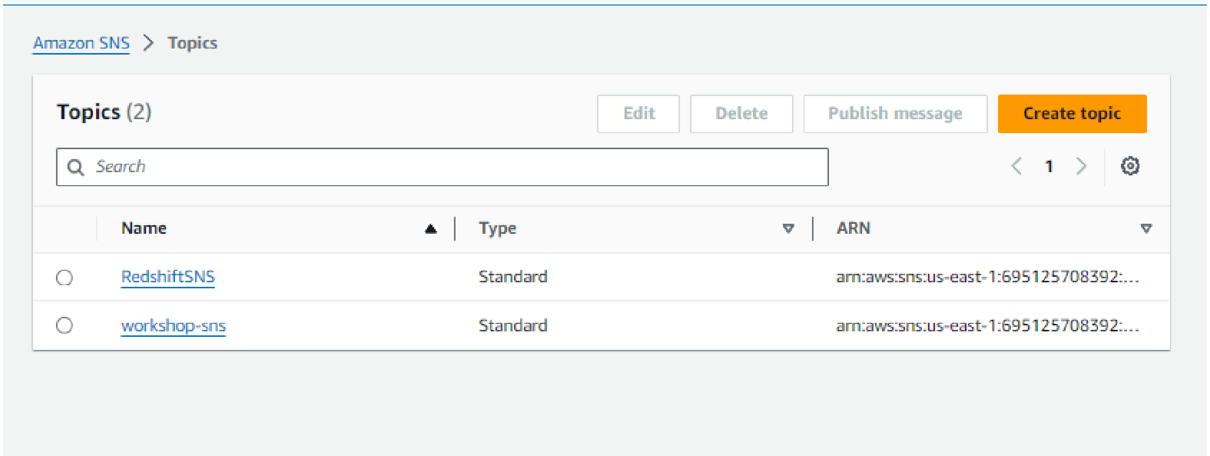
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

□ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

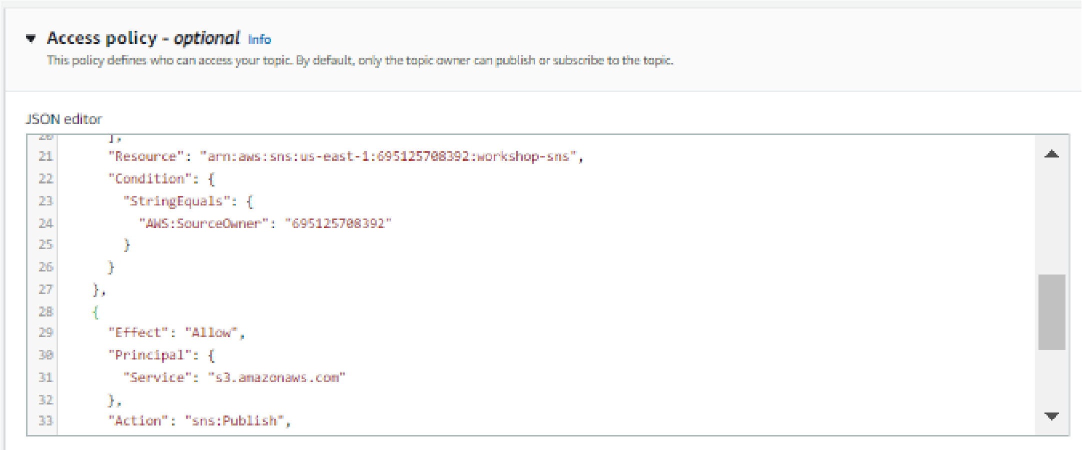
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.



□	Trigger
□	S3: workshop-buckets3 arn:aws:s3:::workshop-buckets3 ▼ Details Bucket arn: **arn:aws:s3:::workshop-buckets3** Event types: **s3:ObjectCreated:Put** Notification name: **9df6fd67-63a5-4df2-a2a4-74432cc13e37** Prefix: **uploads/** Service principal: **s3.amazonaws.com** Source account: **695125708392** Statement ID: **lambda-a912db4d-bc4a-48d8-bb16-1d0b1c53c281**



	Name	Type	ARN
○	RedshiftSNS	Standard	arn:aws:sns:us-east-1:695125708392:...
○	workshop-sns	Standard	arn:aws:sns:us-east-1:695125708392:...

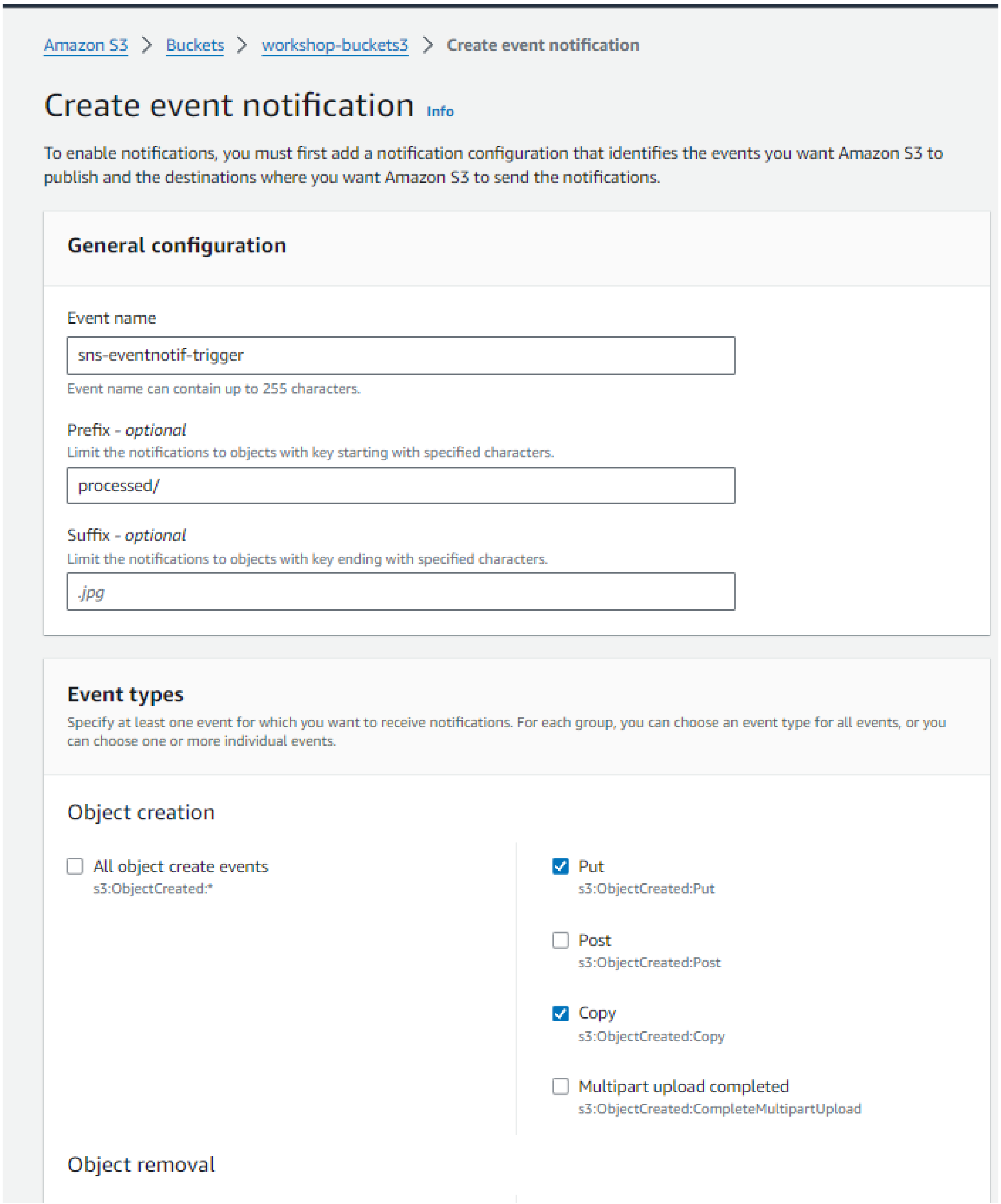


This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

JSON editor

```
      ],
      "Resource": "arn:aws:sns:us-east-1:695125708392:workshop-sns",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "695125708392"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sns:Publish",
```

This is only sns code action and conditions, but now we added iam s3 policy to allow , publish in sns.



Amazon S3 > Buckets > workshop-buckets3 > Create event notification

Create event notification

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name

sns-eventnotif-trigger

Event name can contain up to 255 characters.

Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

processed/

Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

.jpg

Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

□ All object create events
s3:ObjectCreated:*

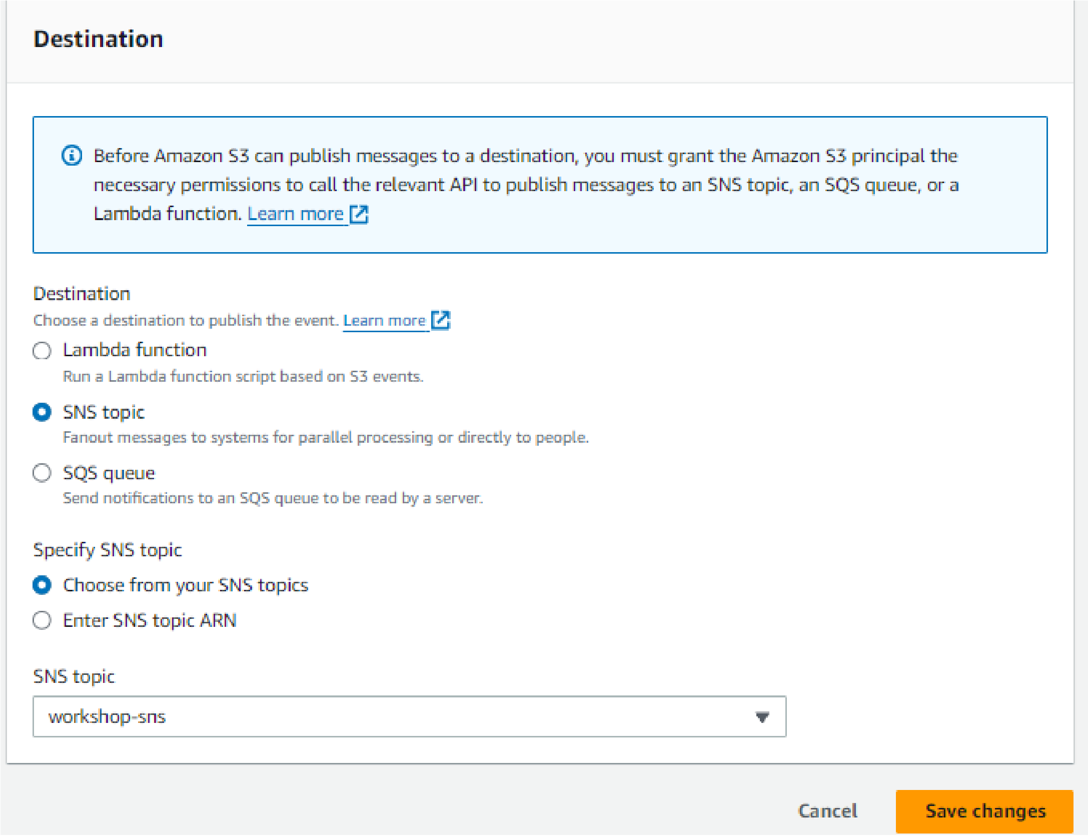
☑ Put
s3:ObjectCreated:Put

□ Post
s3:ObjectCreated:Post

☑ Copy
s3:ObjectCreated:Copy

□ Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

Object removal



Destination

ⓘ Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. Learn more

Destination

Choose a destination to publish the event. Learn more

○ Lambda function
Run a Lambda function script based on S3 events.

● SNS topic
Fanout messages to systems for parallel processing or directly to people.

○ SQS queue
Send notifications to an SQS queue to be read by a server.

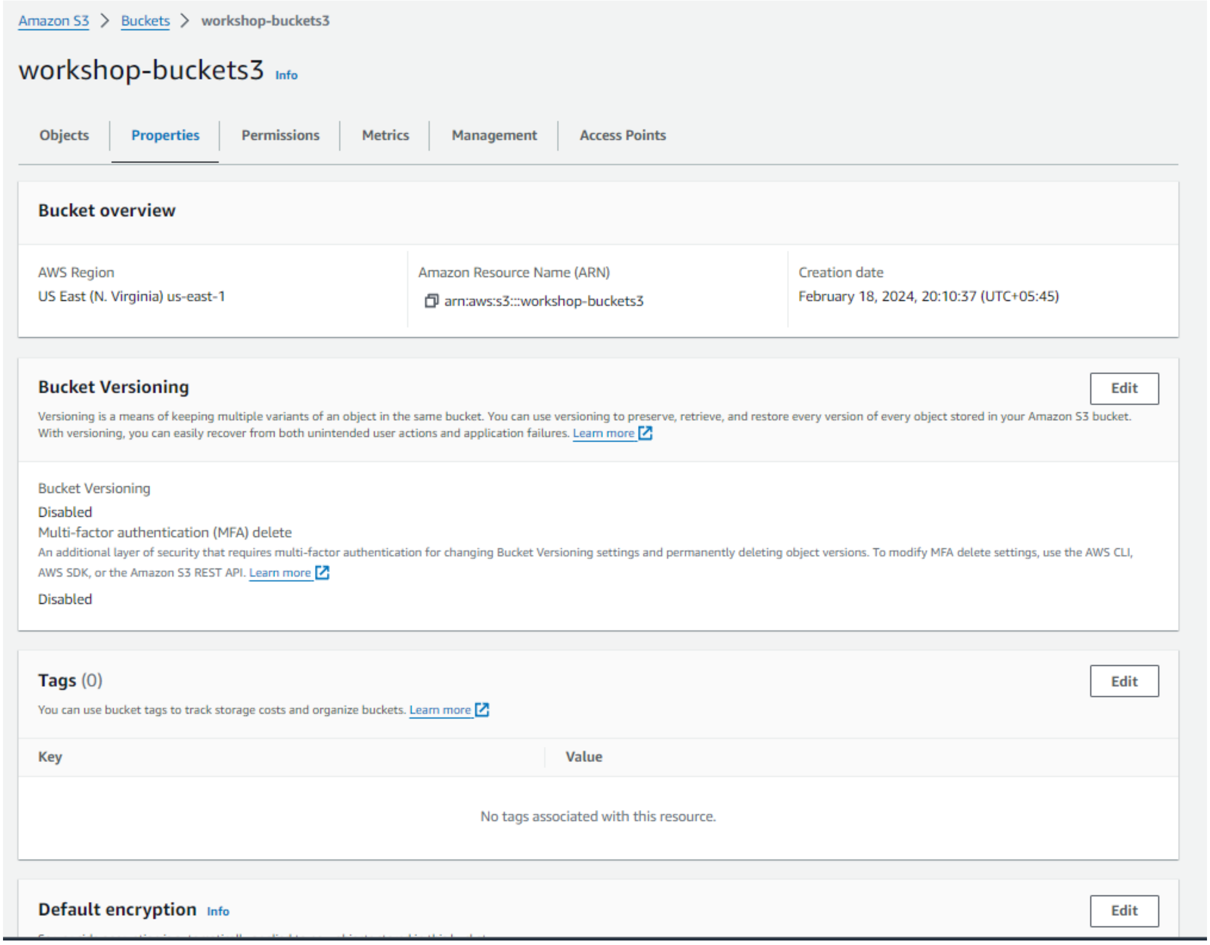
Specify SNS topic

● Choose from your SNS topics

○ Enter SNS topic ARN

SNS topic

workshop-sns ▼



workshop-buckets3 Info

Bucket overview

AWS Region	Amazon Resource Name (ARN)	Creation date
US East (N. Virginia) us-east-1	arn:aws:s3:::workshop-buckets3	February 18, 2024, 20:10:37 (UTC+05:45)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Disabled
Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

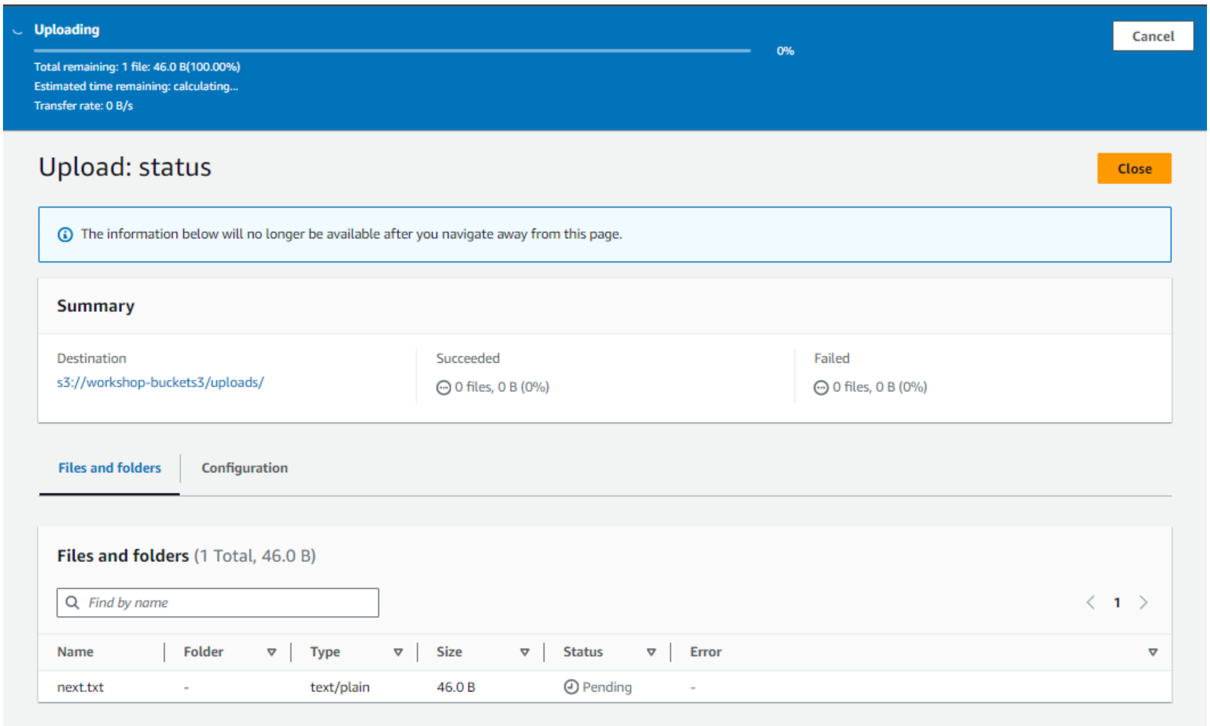
Tags (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value
No tags associated with this resource.	

Default encryption Info

Uploading file in s3 bucket



Uploading

Total remaining: 1 file: 46.0 B(100.00%)
Estimated time remaining: calculating...
Transfer rate: 0 B/s

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

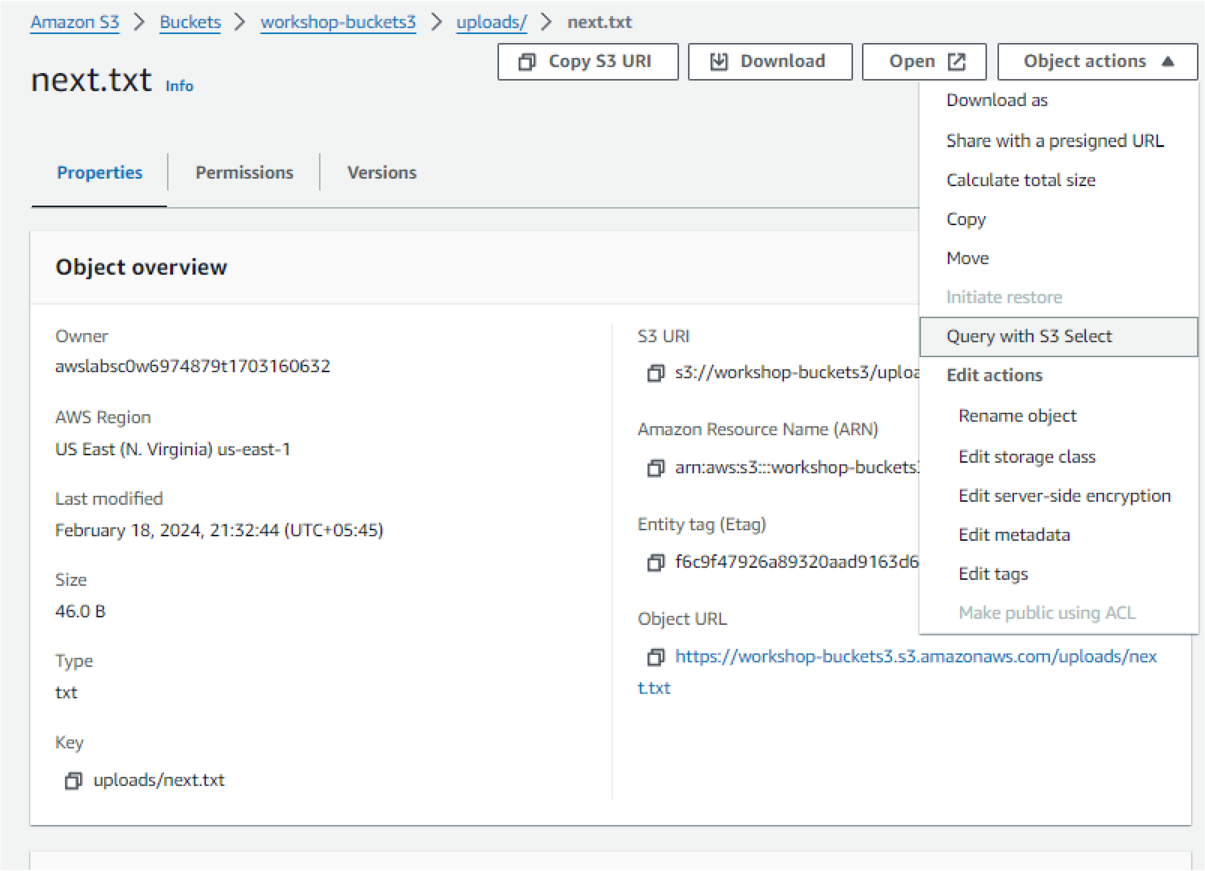
Destination
s3://workshop-buckets3/uploads/

Succeeded
0 files, 0 B (0%)

Failed
0 files, 0 B (0%)

Files and folders (1 Total, 46.0 B)

Name	Folder	Type	Size	Status	Error
next.txt	-	text/plain	46.0 B	Pending	-



Amazon S3 > Buckets > workshop-buckets3 > uploads/ > next.txt

next.txt

Object overview

Owner
awslabsc0w6974879t1703160632

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

Last modified
February 18, 2024, 21:32:44 (UTC+05:45)

Size
46.0 B

Type
txt

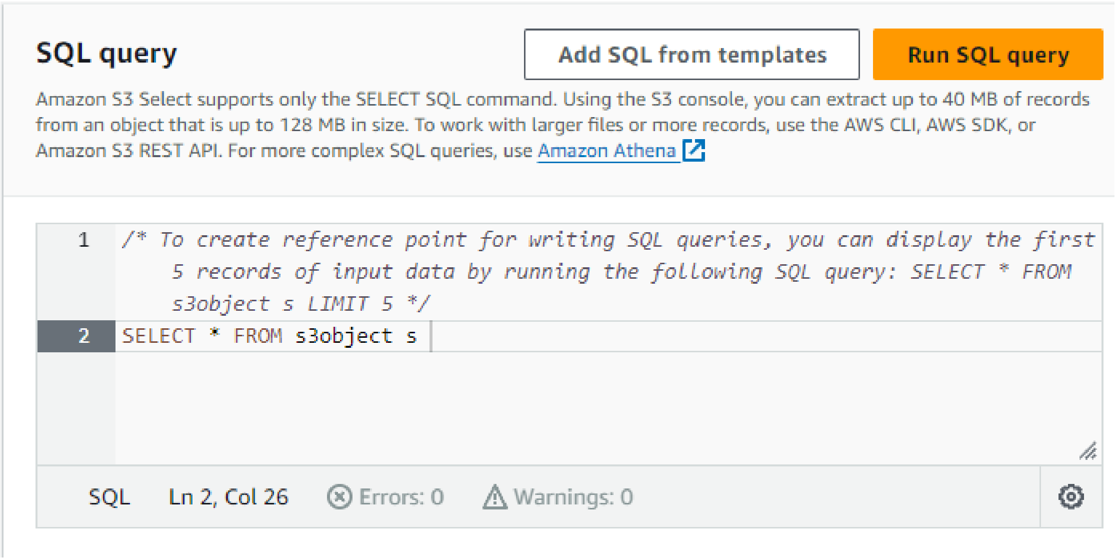
Key
uploads/next.txt

S3 URI
s3://workshop-buckets3/uploa

Amazon Resource Name (ARN)
arn:aws:s3:::workshop-buckets

Entity tag (Etag)
f6c9f47926a89320aad9163d6

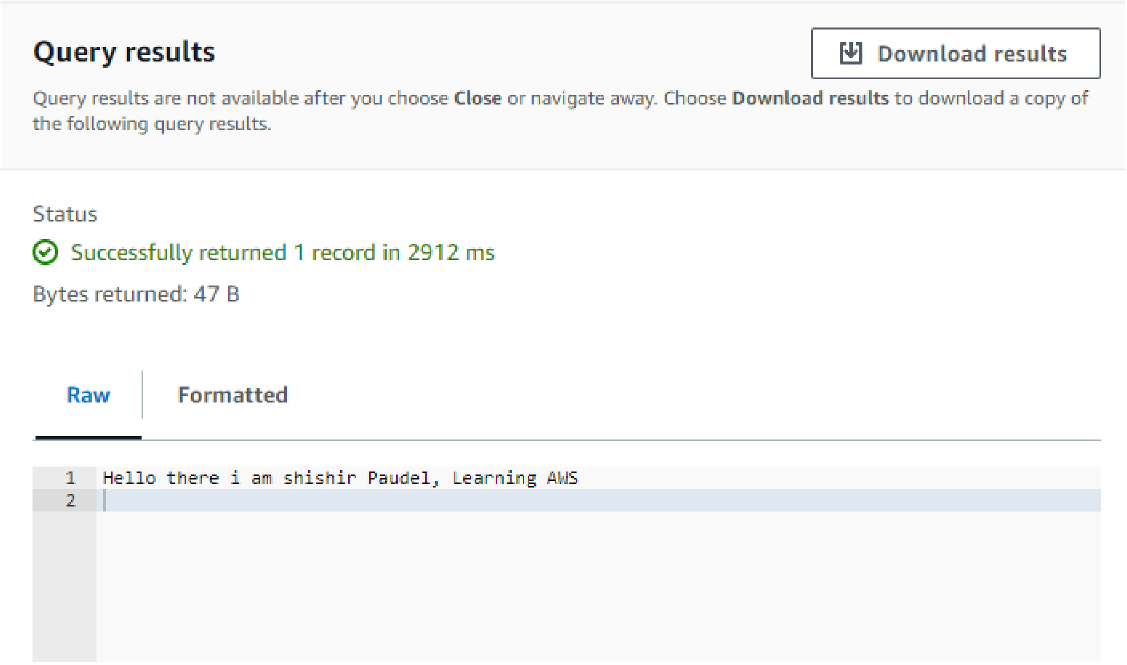
Object URL
https://workshop-buckets3.s3.amazonaws.com/uploads/next.txt



SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first
    5 records of input data by running the following SQL query: SELECT * FROM
    s3object s LIMIT 5 */
SELECT * FROM s3object s 
```



Query results

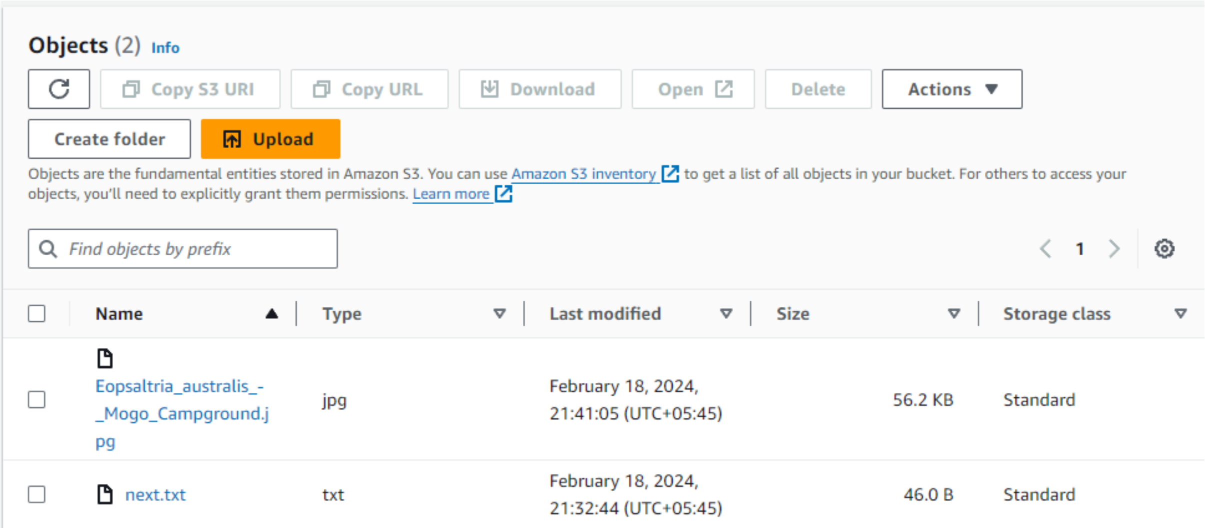
Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 2912 ms

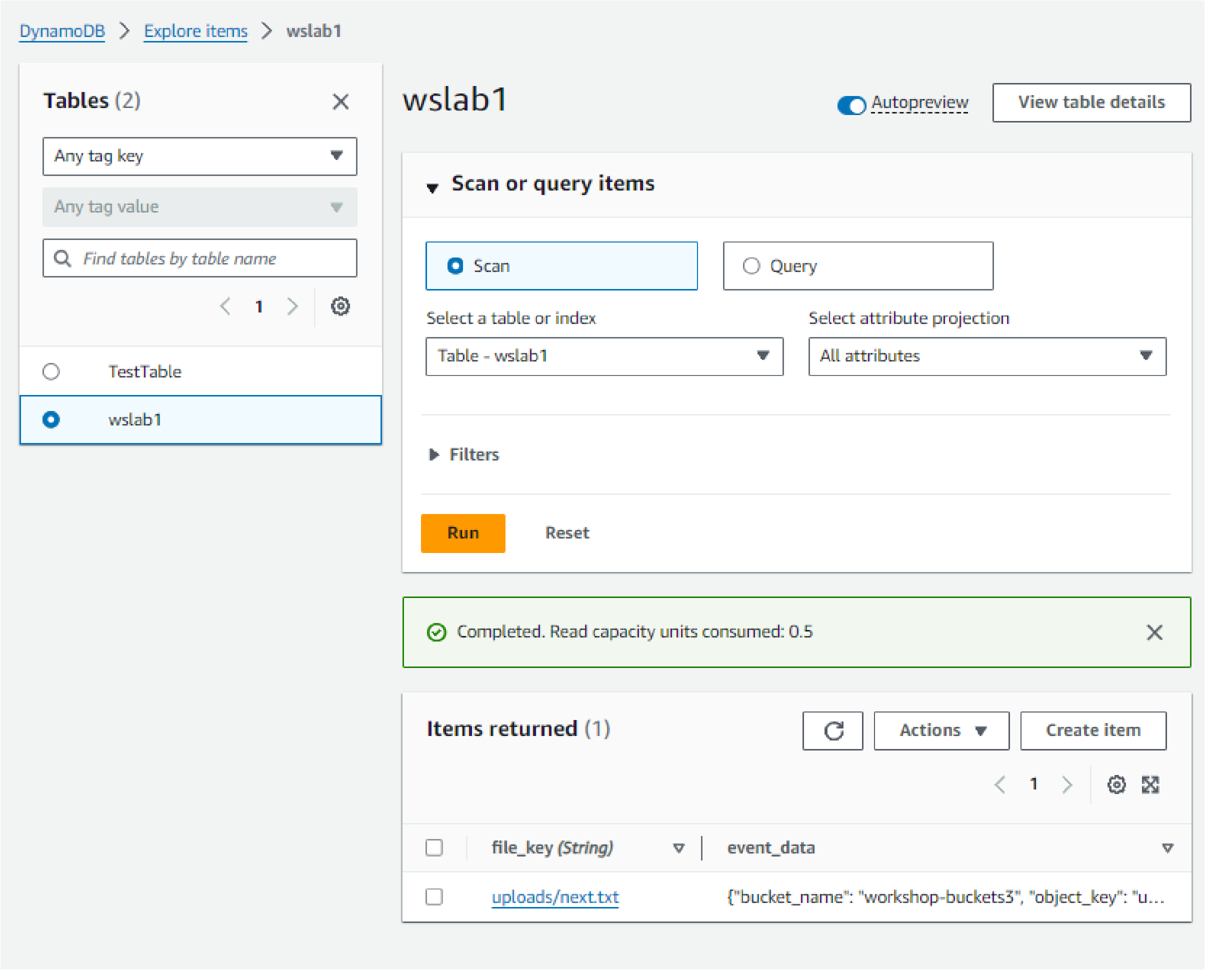
Bytes returned: 47 B

```
Hello there i am shishir Paudel, Learning AWS

```



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

□	Name	Type	Last modified	Size	Storage class
□	Eopsaltria_australis_-_Mogo_Campground.jpg	jpg	February 18, 2024, 21:41:05 (UTC+05:45)	56.2 KB	Standard
□	next.txt	txt	February 18, 2024, 21:32:44 (UTC+05:45)	46.0 B	Standard



● Scan ○ Query

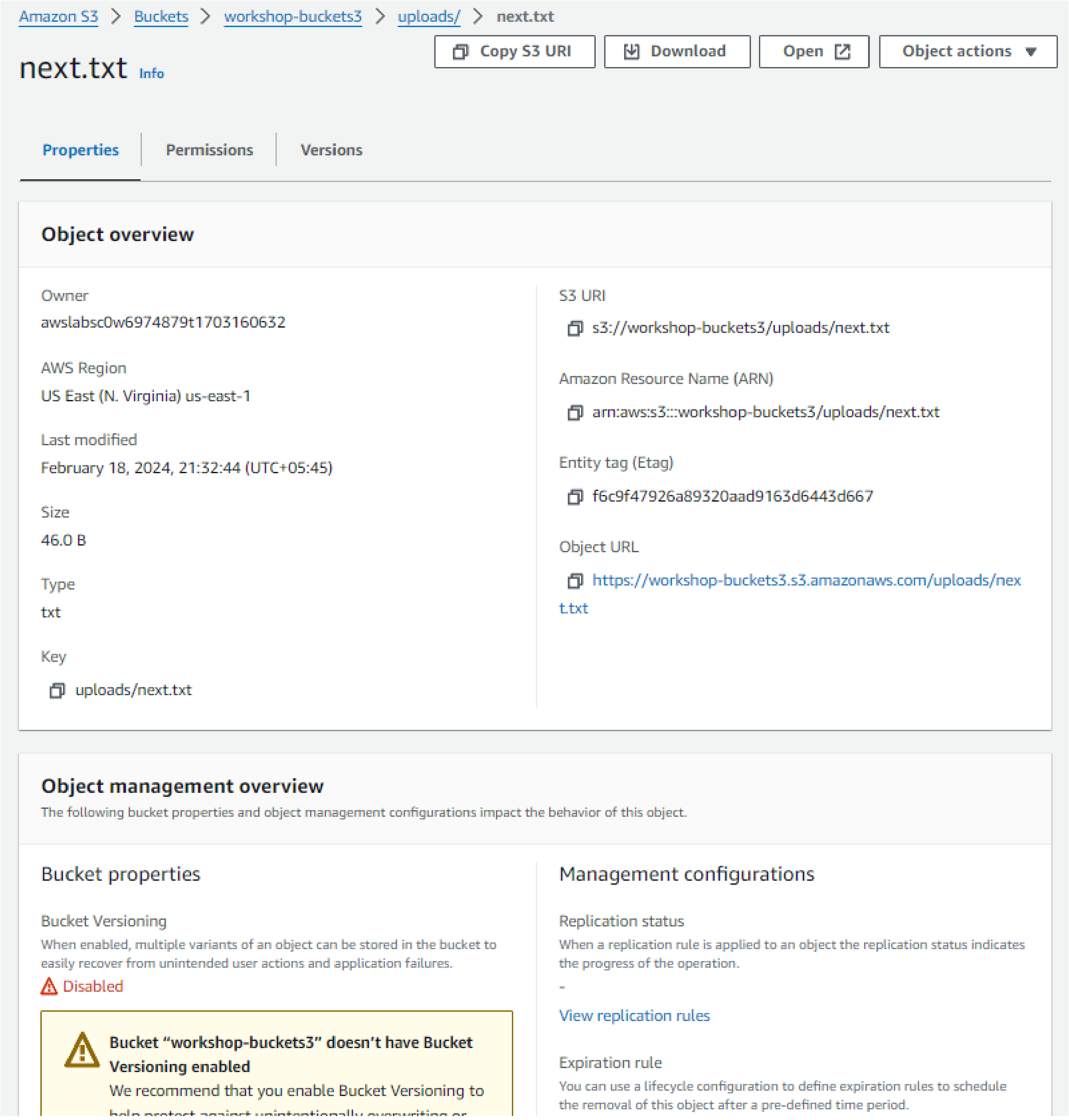
Select a table or index: Table - wslab1

Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Tables (2)
○ TestTable
● wslab1

□	file_key (String)	event_data
□	uploads/next.txt	{"bucket_name": "workshop-buckets3", "object_key": "u…



next.txt

Object overview

Owner
awslabsc0w6974879t1703160632

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

Last modified
February 18, 2024, 21:32:44 (UTC+05:45)

Size
46.0 B

Type
txt

Key
uploads/next.txt

S3 URI
s3://workshop-buckets3/uploads/next.txt

Amazon Resource Name (ARN)
arn:aws:s3:::workshop-buckets3/uploads/next.txt

Entity tag (Etag)
f6c9f47926a89320aad9163d6443d667

Object URL
https://workshop-buckets3.s3.amazonaws.com/uploads/next.txt

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Bucket Versioning
When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended user actions and application failures.
⚠ Disabled

Bucket "workshop-buckets3" doesn't have Bucket Versioning enabled
We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or

Management configurations

Replication status
When a replication rule is applied to an object the replication status indicates the progress of the operation.
-
View replication rules

Expiration rule
You can use a lifecycle configuration to define expiration rules to schedule the removal of this object after a pre-defined time period.

2. Creating a Serverless API****

Objective: Develop a serverless API using AWS Lambda and API Gateway.

Approach:

- **Define API**: Design a simple RESTful API (e.g., for a todo list application).
- **Lambda Functions**: Create Lambda functions for each API method (GET, POST, PUT, DELETE).

- **API Gateway Setup**: Use API Gateway to set up the API endpoints, connecting each endpoint to the corresponding Lambda function.
- **Testing**: Test the API using tools like Postman or AWS API Gateway test functionality.

Goal: Gain hands-on experience in building and deploying a serverless API, understanding the integration between Lambda and API Gateway.