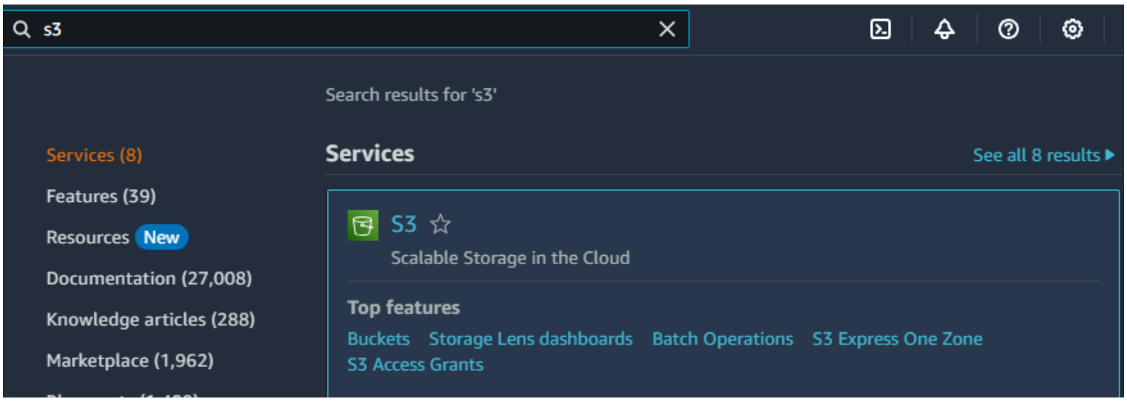
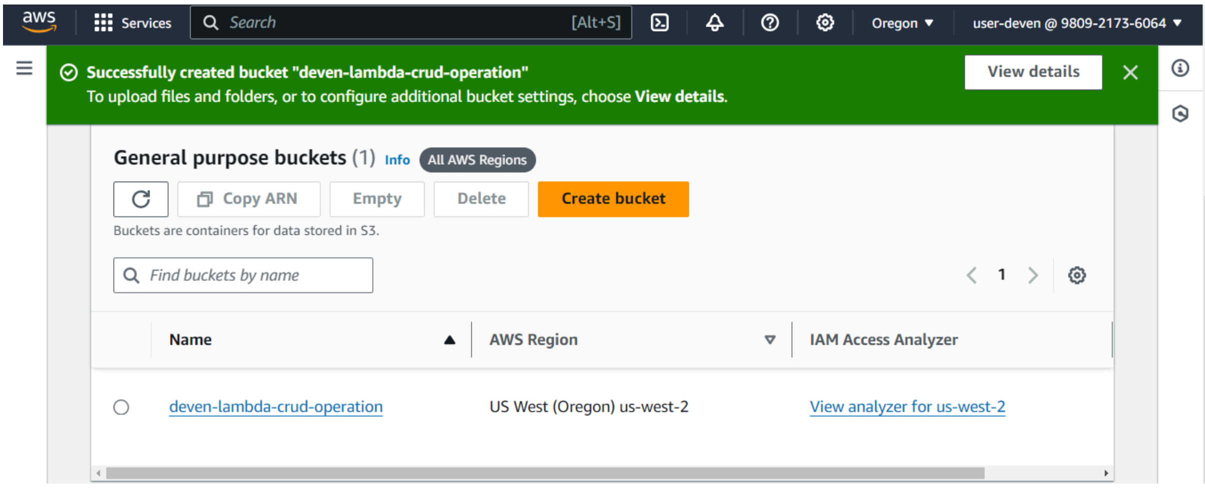
# Performing CRUD (Create, Read, Update, Delete) operations in AWS S3 using AWS Lambda

## Setting Up

1. **Create an S3 Bucket**
   - Sign in to the AWS Management Console.
   - Navigate to the S3 service.



   - Click “Create bucket” and follow the prompts to create a new bucket.

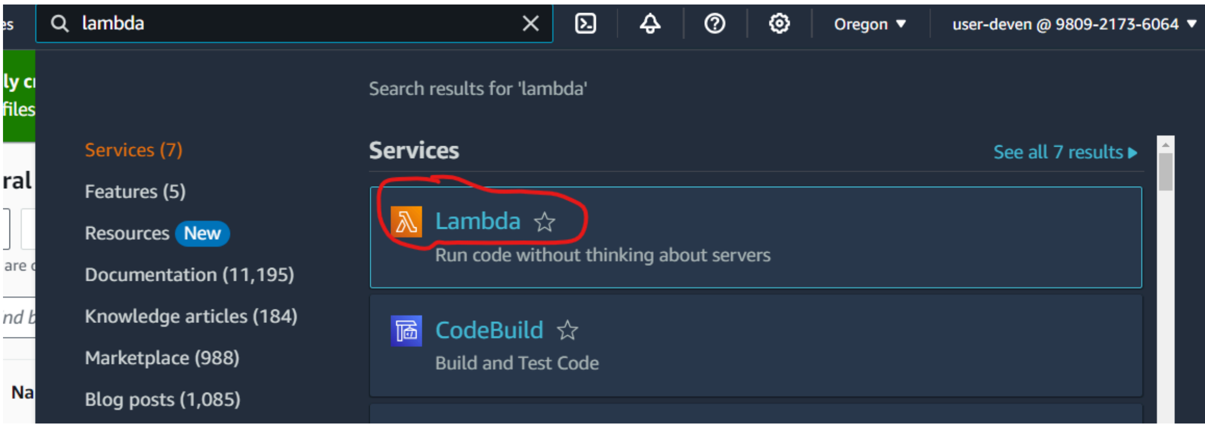


Successfully created bucket "deven-lambda-crud-operation"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Buckets are containers for data stored in S3.

| Name | AWS Region | IAM Access Analyzer |
|---|---|---|
| ○ deven-lambda-crud-operation | US West (Oregon) us-west-2 | View analyzer for us-west-2 |

2. **Create a Lambda Function**
   - Go to the Lambda service in the AWS Management Console.

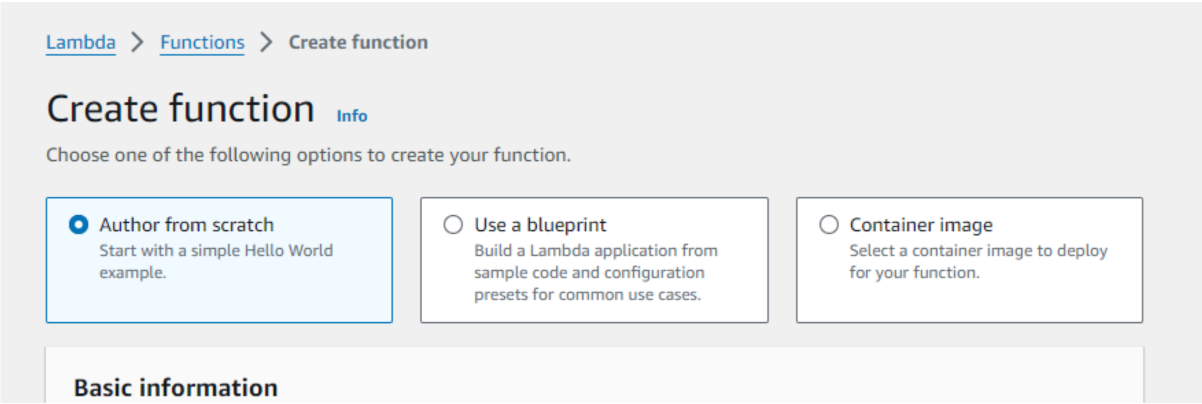


Search results for 'lambda'

Lambda: Run code without thinking about servers

CodeBuild: Build and Test Code

- Click “Create function” and choose “Author from scratch.”



## Create function

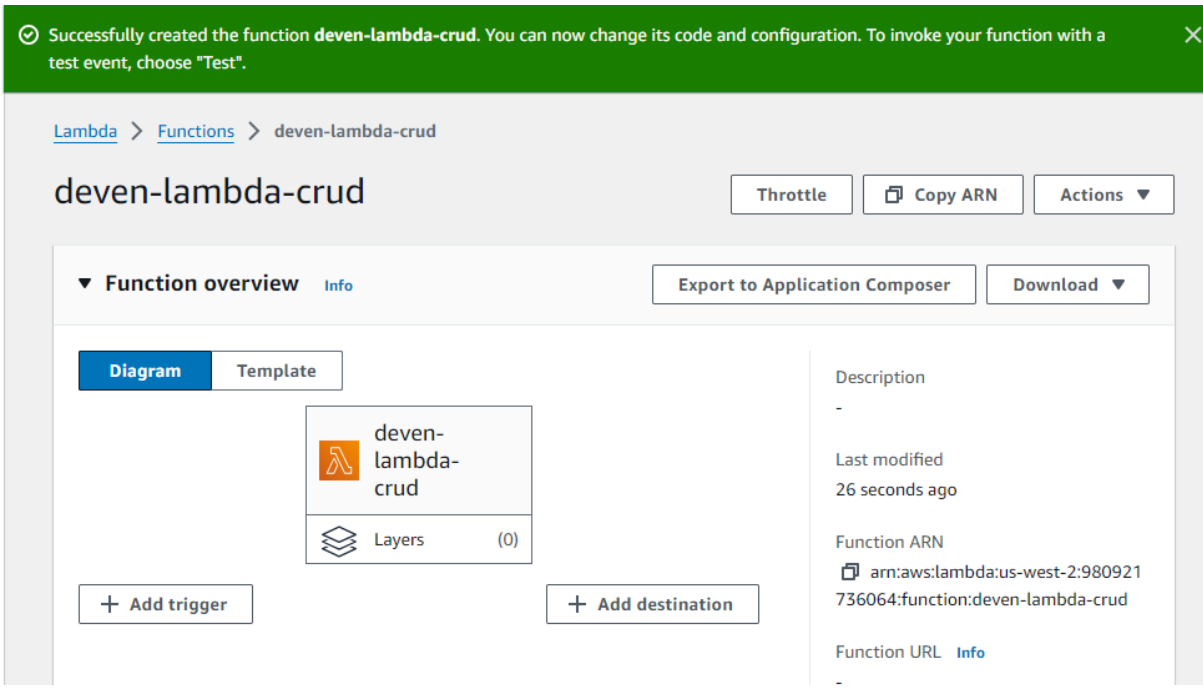
Choose one of the following options to create your function.

● 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.

- Configure the function (name, runtime, role, etc.).


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

Description: -

Last modified: 26 seconds ago

Function ARN: arn:aws:lambda:us-west-2:980921736064:function:deven-lambda-crud

Function URL: -

**Attach IAM Permissions**
Ensure that your Lambda function's execution role has the necessary permissions to interact with S3. You can attach a policy like the following to the role:

Json code

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::deven-lambda-crud-operation",
        "arn:aws:s3:::deven-lambda-crud-operation/*"
      ]
    }
  ]
}
```

## CRUD Operations with Lambda and S3

### 1. Create (Upload a File to S3)

code

```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    bucket_name = 'deven-lambda-crud-operation'
    object_key = 'demofile.txt'
    file_content = 'Hello my name is deven'

    try:
        s3.put_object(Bucket=bucket_name, Key=object_key,
Body=file_content)
        return {
            'statusCode': 200,
            'body': json.dumps('File uploaded successfully!')
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }
```

### 2. Read (Retrieve a File from S3)

code

```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    bucket_name = 'deven-lambda-crud-operation'
    object_key = 'demofile.txt'

    try:
        response = s3.get_object(Bucket=bucket_name, Key=object_key)
        file_content = response['Body'].read().decode('utf-8')
        return {
            'statusCode': 200,
            'body': json.dumps(file_content)
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }
```

### 3. Update (Overwrite an Existing File in S3)

**Note:** S3 does not have a direct update operation. Instead, you overwrite an existing file.

code

```
import json
import boto3
```

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

def lambda_handler(event, context):
    bucket_name = 'deven-lambda-crud-operation'
    object_key = 'demofile.txt'
    new_content = 'This is updated content'

    try:
        s3.put_object(Bucket=bucket_name, Key=object_key,
Body=new_content)
        return {
            'statusCode': 200,
            'body': json.dumps('File updated successfully!')
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }
```

### 4. Delete (Remove a File from S3)

code

```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    bucket_name = 'deven-lambda-crud-operation'
    object_key = 'demofile.txt'

    try:
        s3.delete_object(Bucket=bucket_name, Key=object_key)
        return {
            'statusCode': 200,
            'body': json.dumps('File deleted successfully!')
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps(f'Error: {str(e)}')
        }
```

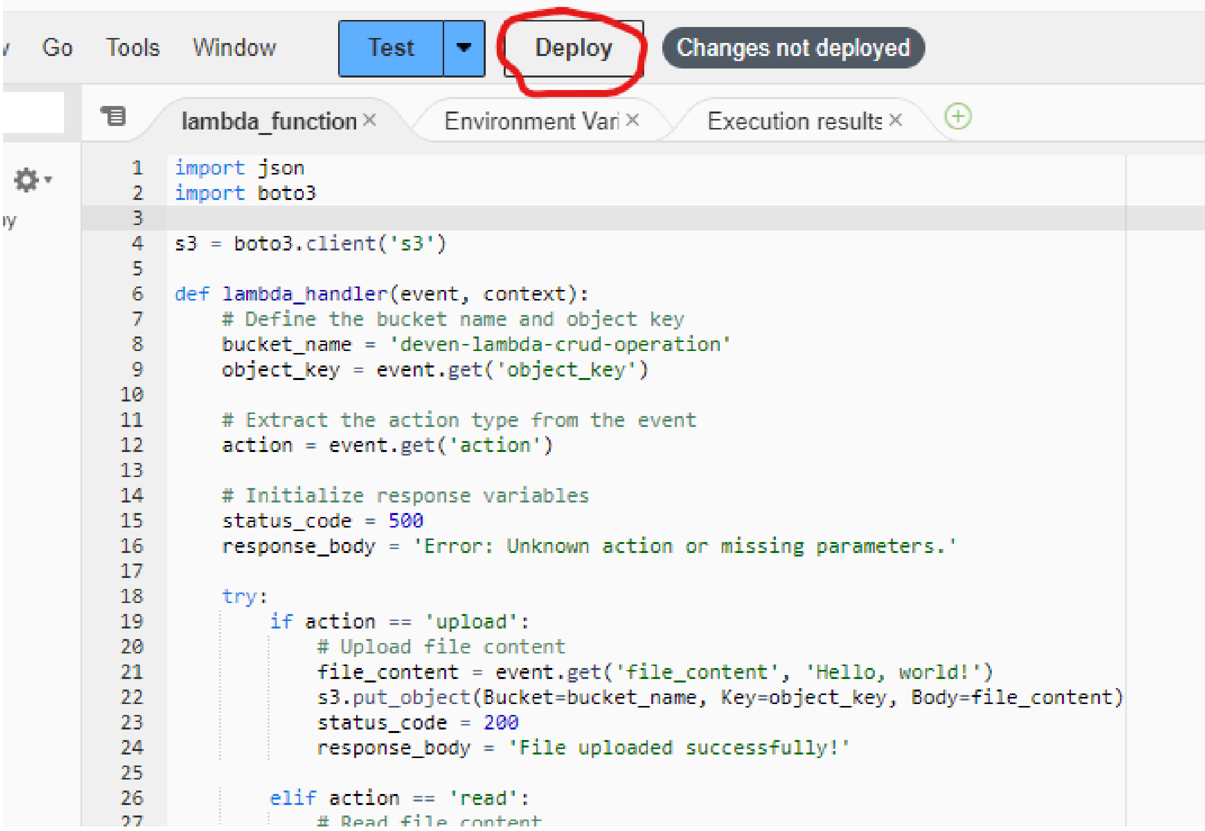
## 5. ALL OPERATION SINGLE FILE

code

```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    # Define the bucket name and object key
    bucket_name = 'deven-lambda-crud-operation'
    object_key = event.get('object_key')

    # Extract the action type from the event
    action = event.get('action')

    # Initialize response variables
    status_code = 500
    response_body = 'Error: Unknown action or missing parameters.'

    try:
        if action == 'upload':
            # Upload file content
            file_content = event.get('file_content', 'Hello,
world!')
            s3.put_object(Bucket=bucket_name, Key=object_key,
Body=file_content)
            status_code = 200
            response_body = 'File uploaded successfully!'

        elif action == 'read':
            # Read file content
            response = s3.get_object(Bucket=bucket_name,
Key=object_key)
            file_content = response['Body'].read().decode('utf-8')
            status_code = 200
            response_body = json.dumps(file_content)

        elif action == 'update':
            # Update file content
            new_content = event.get('new_content', 'Updated
content!')
            s3.put_object(Bucket=bucket_name, Key=object_key,
Body=new_content)
            status_code = 200
            response_body = 'File updated successfully!'

        elif action == 'delete':
            # Delete file
            s3.delete_object(Bucket=bucket_name, Key=object_key)
            status_code = 200
            response_body = 'File deleted successfully!'

        else:
            response_body = 'Error: Invalid action specified.'

    except Exception as e:
        response_body = f'Error: {str(e)}'

    return {
        'statusCode': status_code,
        'body': response_body
    }
```

## Testing Your Lambda Functions

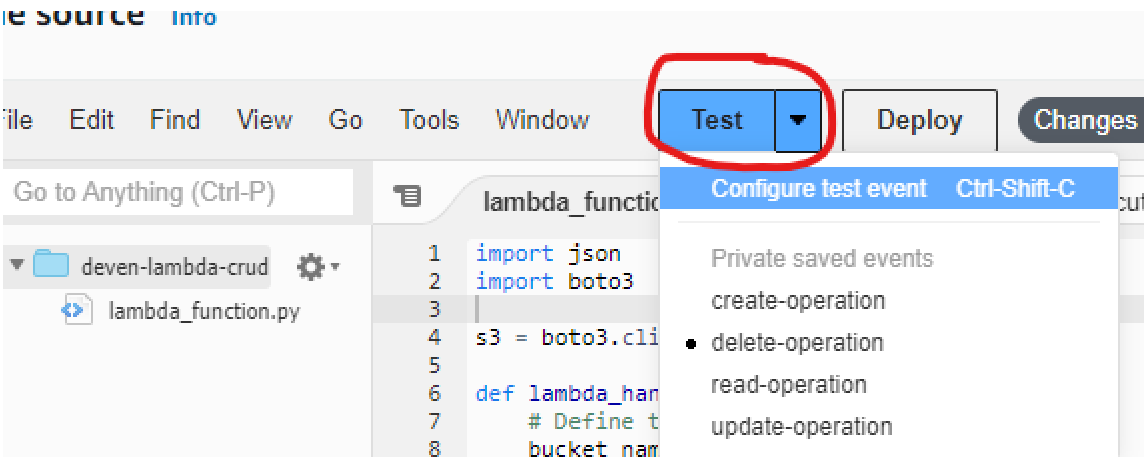
1. **Deploy the Lambda Functions:**
   - Copy the code into the Lambda function editor in the AWS Management Console deploy it.



```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    # Define the bucket name and object key
    bucket_name = 'deven-lambda-crud-operation'
    object_key = event.get('object_key')

    # Extract the action type from the event
    action = event.get('action')

    # Initialize response variables
    status_code = 500
    response_body = 'Error: Unknown action or missing parameters.'

    try:
        if action == 'upload':
            # Upload file content
            file_content = event.get('file_content', 'Hello, world!')
            s3.put_object(Bucket=bucket_name, Key=object_key, Body=file_content)
            status_code = 200
            response_body = 'File uploaded successfully!'

        elif action == 'read':
            # Read file content
```

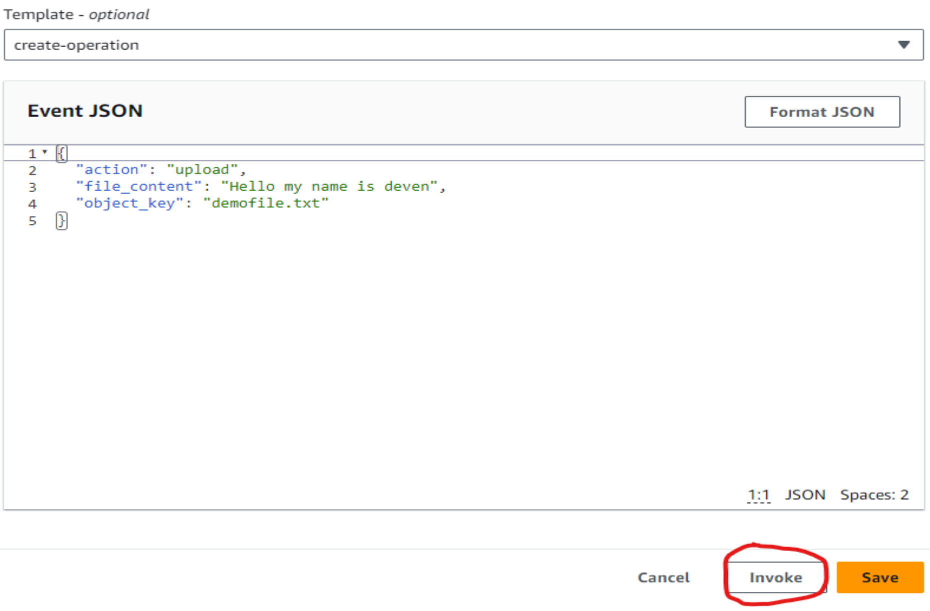
2. **Configure Test Events:**
   - In the Lambda console, go to the “Test” tab.

     Create a new test event and configure it if needed (for CRUD operations, this may not be required).



```
import json
import boto3

s3 = boto3.cli

def lambda_han
    # Define t
    bucket nam
```

3. **Invoke the Lambda Functions:**
   - Click the “Test” button to invoke your Lambda function and verify the results.



Template - *optional*

create-operation ▼

**Event JSON**

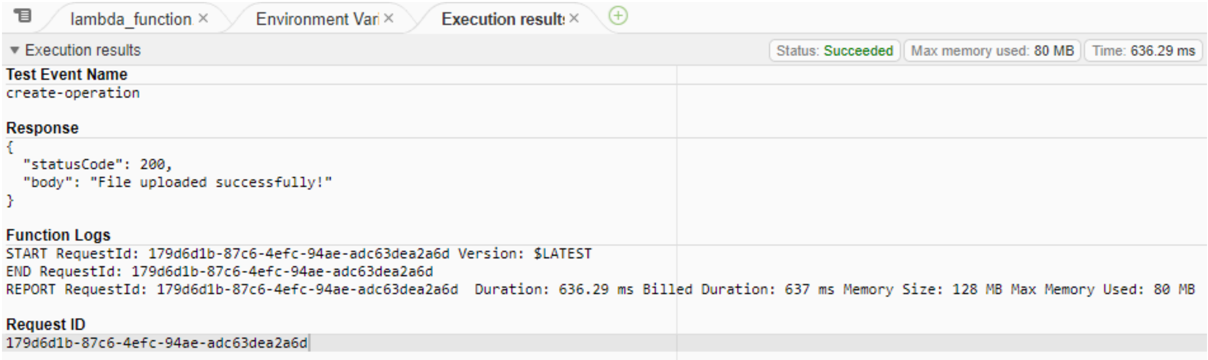
Format JSON

```
{
    "action": "upload",
    "file_content": "Hello my name is deven",
    "object_key": "demofile.txt"
}
```

1:1 JSON Spaces: 2

Cancel Invoke Save

## Create operation O/P :



▼ Execution results

Status: Succeeded | Max memory used: 80 MB | Time: 636.29 ms

**Test Event Name**

create-operation

**Response**

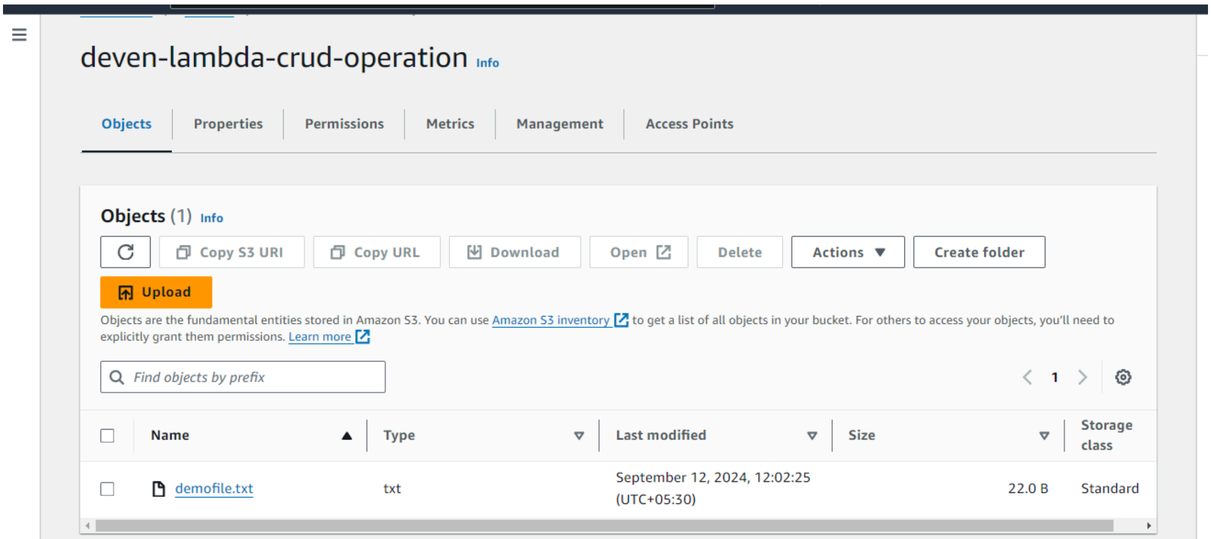
```
{
  "statusCode": 200,
  "body": "File uploaded successfully!"
}
```

**Function Logs**

```
START RequestId: 179d6d1b-87c6-4efc-94ae-adc63dea2a6d Version: $LATEST
END RequestId: 179d6d1b-87c6-4efc-94ae-adc63dea2a6d
REPORT RequestId: 179d6d1b-87c6-4efc-94ae-adc63dea2a6d  Duration: 636.29 ms Billed Duration: 637 ms Memory Size: 128 MB Max Memory Used: 80 MB
```

**Request ID**

179d6d1b-87c6-4efc-94ae-adc63dea2a6d



deven-lambda-crud-operation Info

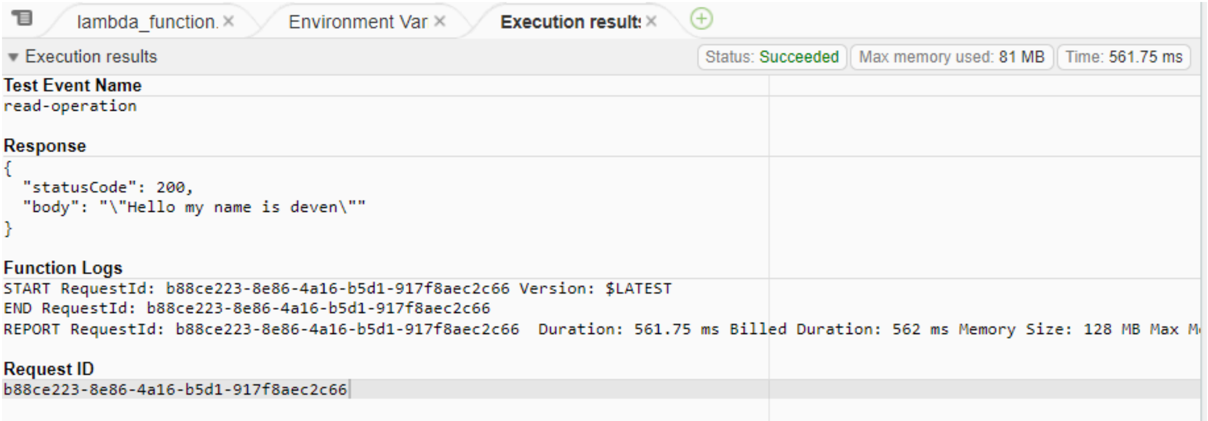
**Objects** (1) Info

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

*Find objects by prefix*

| □ | Name ▲ | Type | Last modified | Size | Storage class |
|---|---|---|---|---|---|
| □ | demofile.txt | txt | September 12, 2024, 12:02:25 (UTC+05:30) | 22.0 B | Standard |

## Read operation O/P :



**Test Event Name**

```
read-operation
```

**Response**

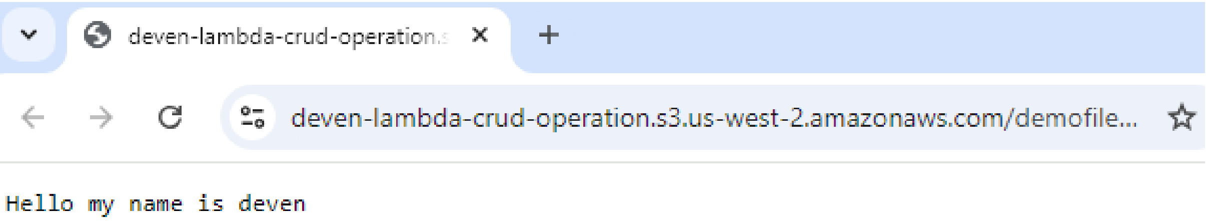
```
{
  "statusCode": 200,
  "body": "\"Hello my name is deven\""
}
```

**Function Logs**

```
START RequestId: b88ce223-8e86-4a16-b5d1-917f8aec2c66 Version: $LATEST
END RequestId: b88ce223-8e86-4a16-b5d1-917f8aec2c66
REPORT RequestId: b88ce223-8e86-4a16-b5d1-917f8aec2c66  Duration: 561.75 ms Billed Duration: 562 ms Memory Size: 128 MB Max M
```

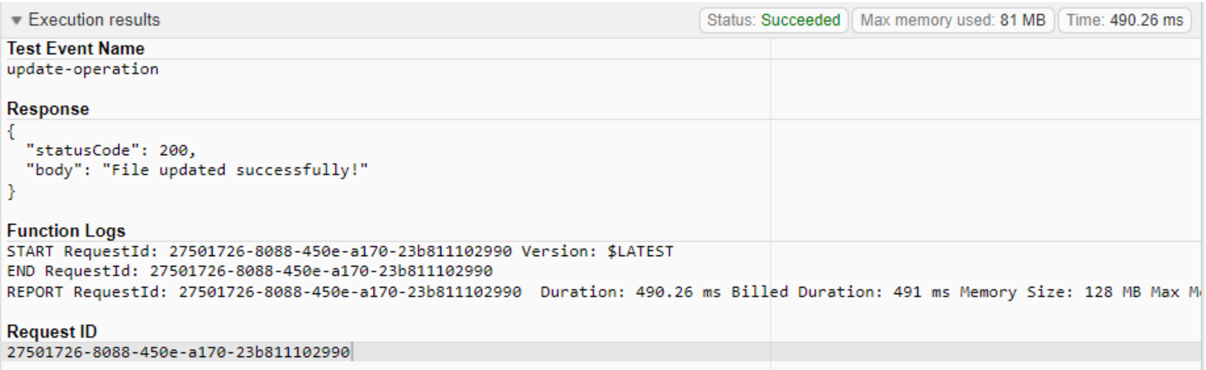
**Request ID**

```
b88ce223-8e86-4a16-b5d1-917f8aec2c66
```


deven-lambda-crud-operation.s
deven-lambda-crud-operation.s3.us-west-2.amazonaws.com/demofile...

```
Hello my name is deven
```

## Update operation O/P :



**Test Event Name**

```
update-operation
```

**Response**

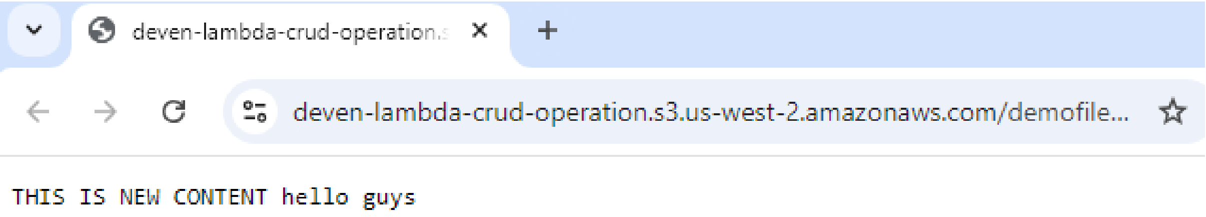
```
{
  "statusCode": 200,
  "body": "File updated successfully!"
}
```

**Function Logs**

```
START RequestId: 27501726-8088-450e-a170-23b811102990 Version: $LATEST
END RequestId: 27501726-8088-450e-a170-23b811102990
REPORT RequestId: 27501726-8088-450e-a170-23b811102990  Duration: 490.26 ms Billed Duration: 491 ms Memory Size: 128 MB Max M
```

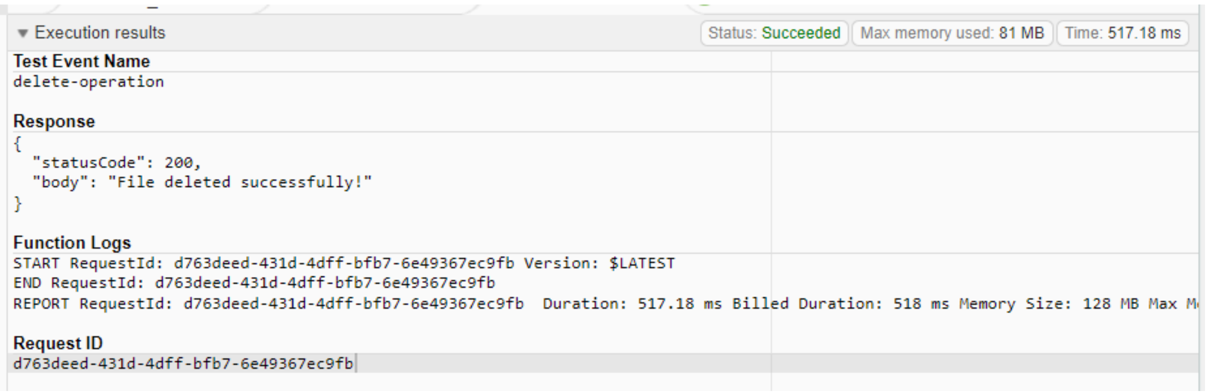
**Request ID**

```
27501726-8088-450e-a170-23b811102990
```


deven-lambda-crud-operation.s
deven-lambda-crud-operation.s3.us-west-2.amazonaws.com/demofile...

```
THIS IS NEW CONTENT hello guys
```

## Delete operation O/P :



**Test Event Name**

```
delete-operation
```

**Response**

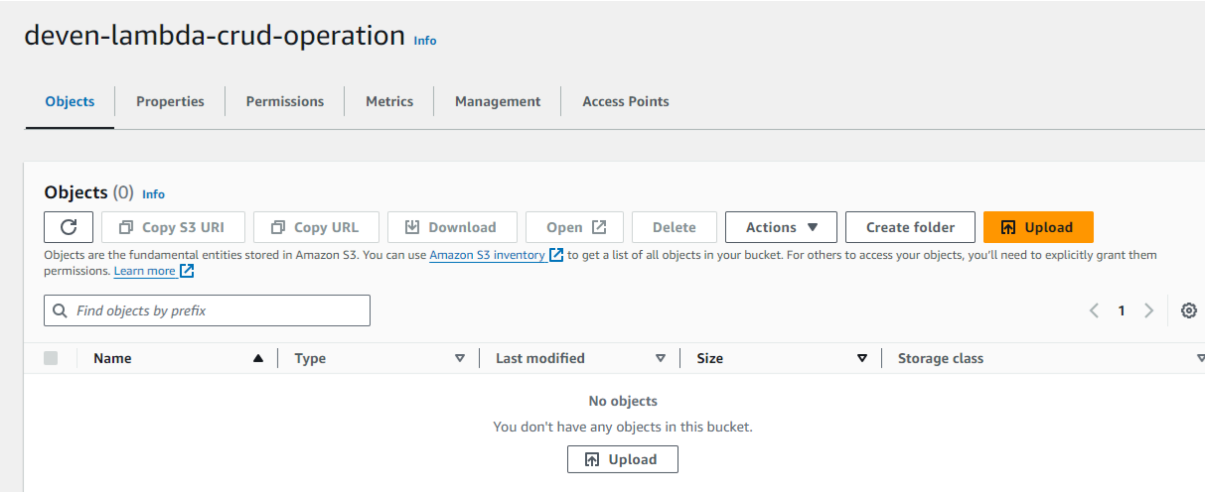
```
{
  "statusCode": 200,
  "body": "File deleted successfully!"
}
```

**Function Logs**

```
START RequestId: d763deed-431d-4dff-bfb7-6e49367ec9fb Version: $LATEST
END RequestId: d763deed-431d-4dff-bfb7-6e49367ec9fb
REPORT RequestId: d763deed-431d-4dff-bfb7-6e49367ec9fb  Duration: 517.18 ms Billed Duration: 518 ms Memory Size: 128 MB Max M
```

**Request ID**

```
d763deed-431d-4dff-bfb7-6e49367ec9fb
```



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

No objects

You don't have any objects in this bucket.