# Assignment 3: Monitor Unencrypted S3 Buckets Using AWS Lambda and Boto3

## Objective

The objective of this assignment is to enhance AWS security by detecting Amazon S3 buckets that do not have server-side encryption enabled using AWS Lambda and Boto3.

## AWS Services Used

- Amazon S3

- AWS Lambda

- AWS IAM

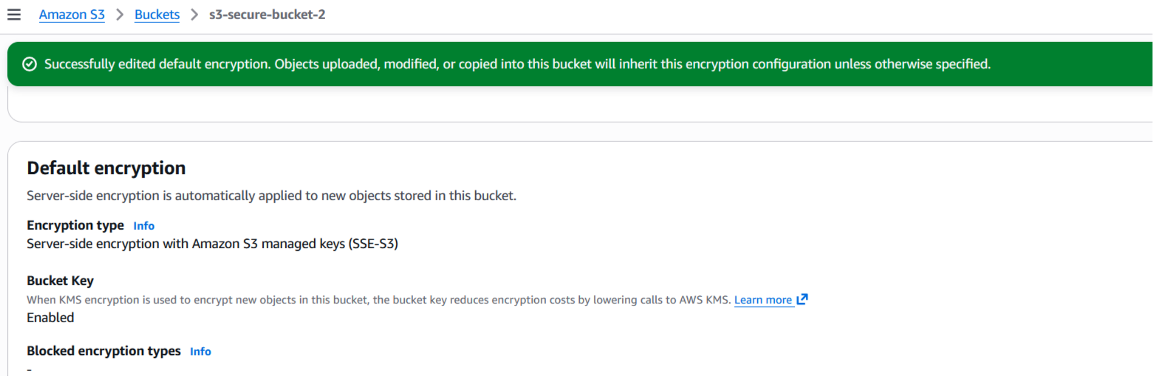
- Amazon CloudWatch Logs

- Boto3 (AWS SDK for Python)

## Step-by-Step Implementation

### Step 1: S3 Bucket Setup

Multiple S3 buckets were created. Server-side encryption was enabled on some buckets, while encryption was intentionally disabled on others to test the monitoring functionality.



Amazon S3 > Buckets > s3-secure-bucket-2

Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.

**Default encryption**

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

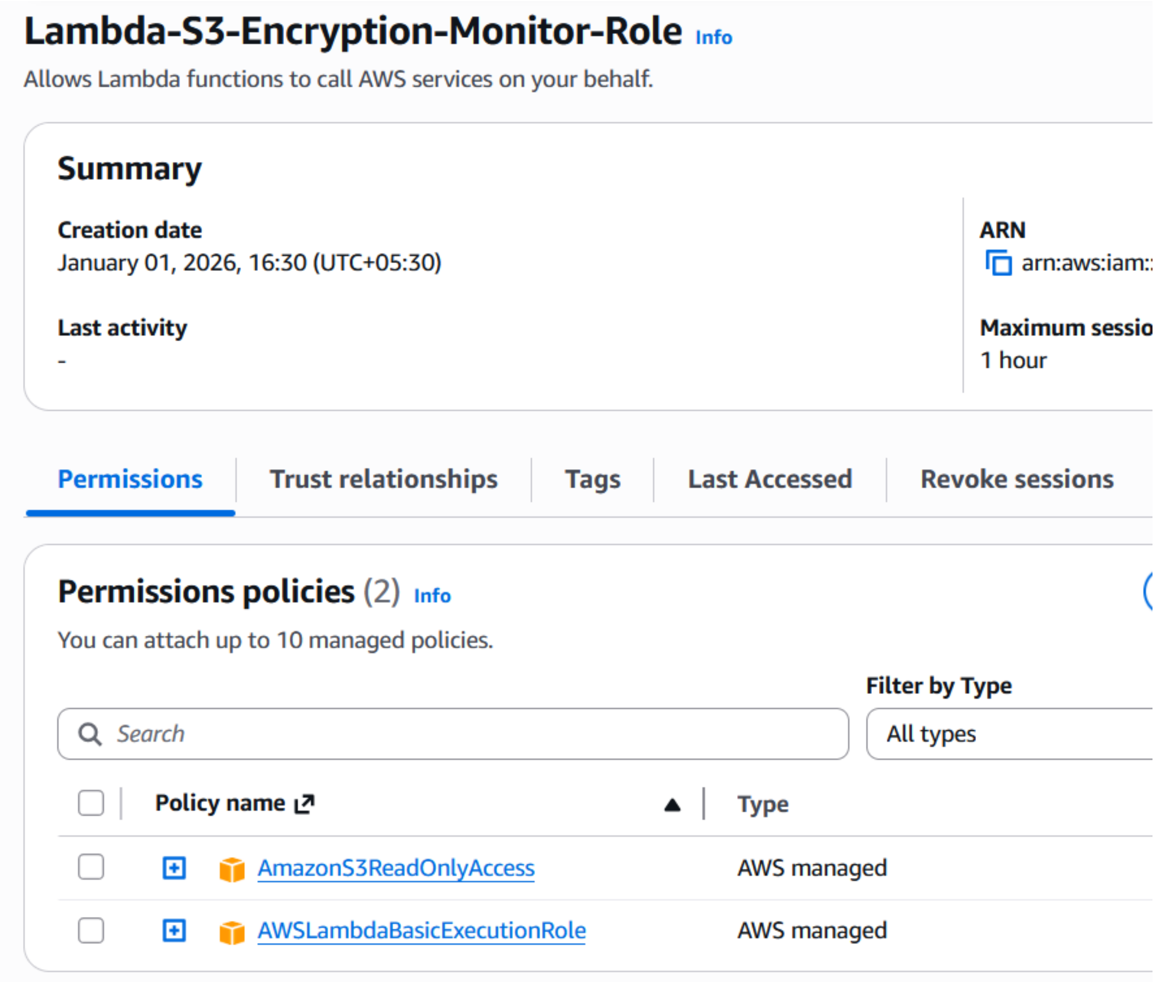
**Encryption type** Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)

**Bucket Key**
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

**Blocked encryption types** Info
-

### Step 2: IAM Role Creation

An IAM role was created for the Lambda function with the following permissions:
- AmazonS3ReadOnlyAccess
- AWSLambdaBasicExecutionRole
This role allows the Lambda function to list buckets and check encryption settings.



Lambda-S3-Encryption-Monitor-Role Info

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
January 01, 2026, 16:30 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::

Maximum sessio
1 hour

Permissions policies (2) Info

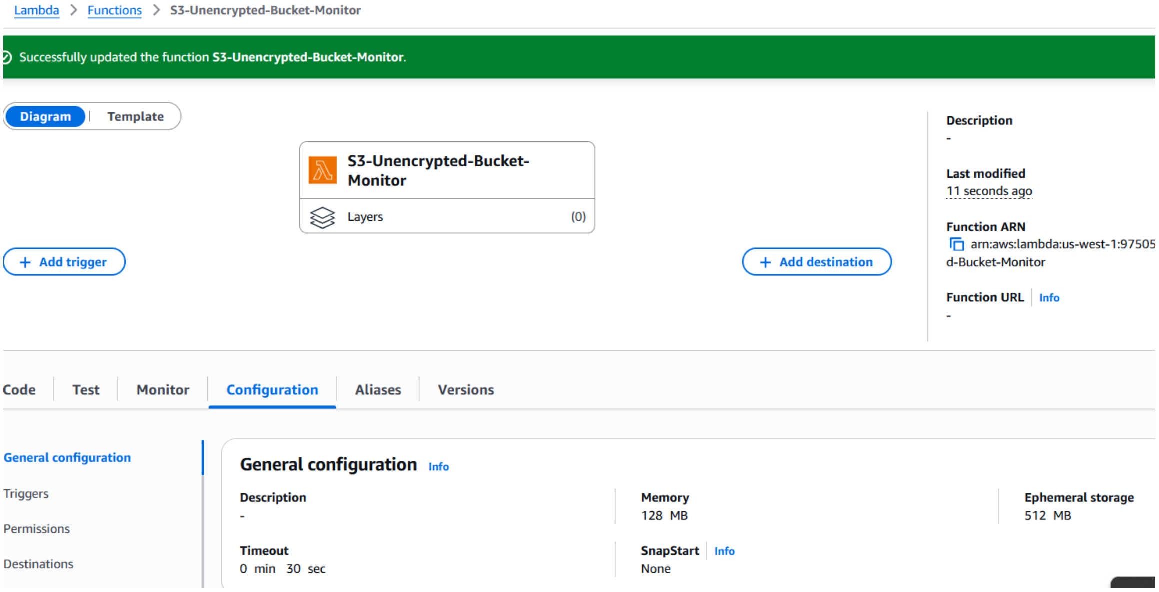
You can attach up to 10 managed policies.

Filter by Type: All types

| □ | Policy name ▲ | Type |
|---|---|---|
| □ | AmazonS3ReadOnlyAccess | AWS managed |
| □ | AWSLambdaBasicExecutionRole | AWS managed |

### Step 3: Lambda Function Creation

A Lambda function was created using Python 3.x runtime and the IAM role created earlier was assigned to the function.



Lambda > Functions > S3-Unencrypted-Bucket-Monitor

Successfully updated the function **S3-Unencrypted-Bucket-Monitor**.

**Description**
-

**Last modified**
11 seconds ago

**Function ARN**
arn:aws:lambda:us-west-1:97505
d-Bucket-Monitor

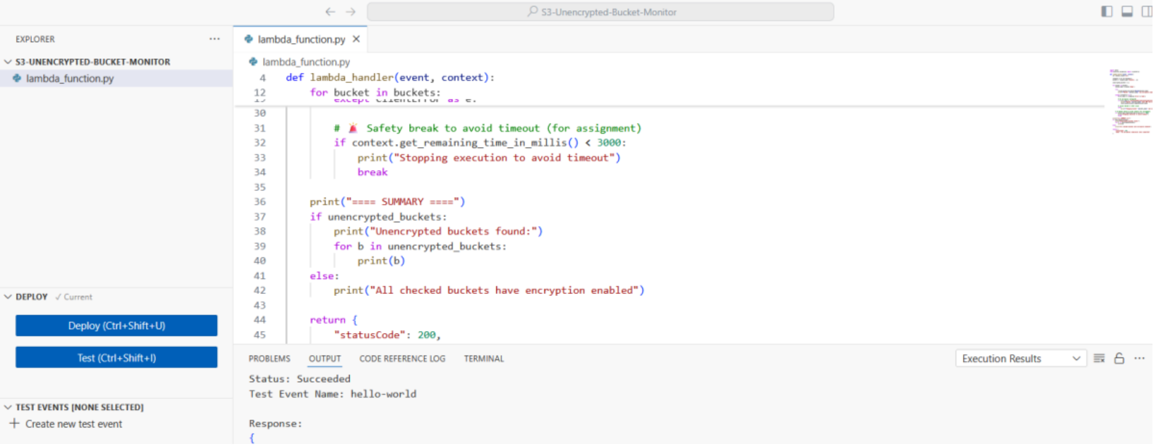
**Function URL** Info
-

**General configuration** Info

| | | |
|---|---|---|
| **Description**<br>- | **Memory**<br>128 MB | **Ephemeral storage**<br>512 MB |
| **Timeout**<br>0 min 30 sec | **SnapStart** Info<br>None | |

## Step 4: Lambda Code Logic

The Lambda function lists all S3 buckets and checks each bucket for server-side encryption. Buckets without encryption are logged in Amazon CloudWatch Logs.

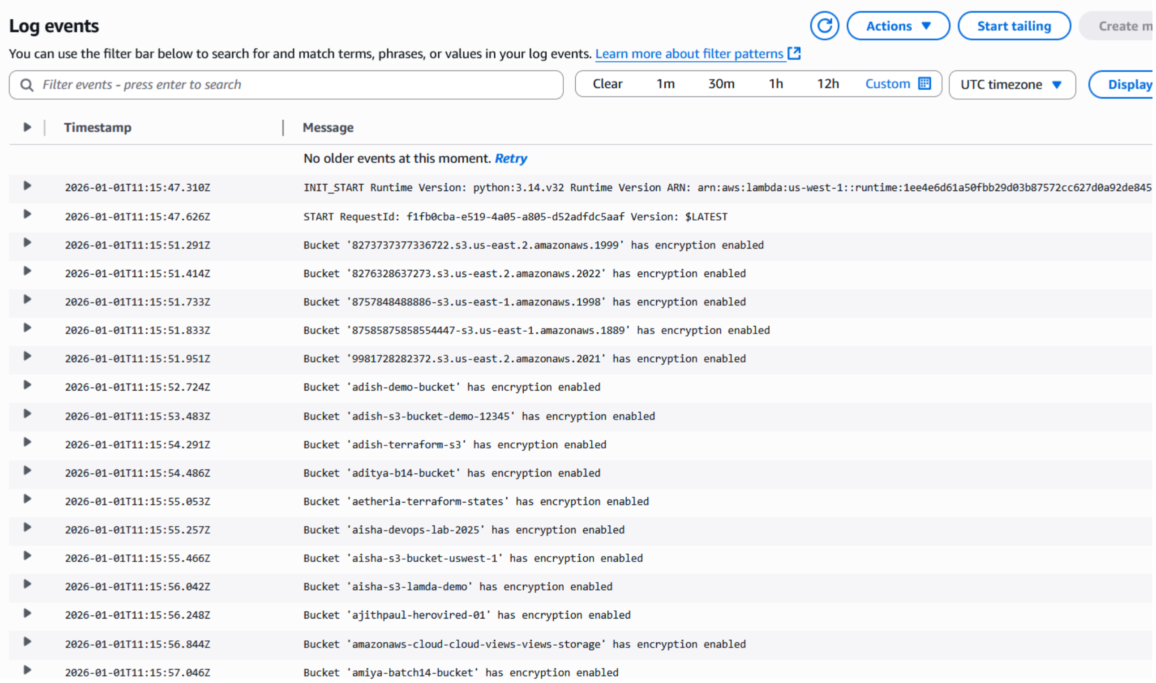


```python
def lambda_handler(event, context):
    for bucket in buckets:
        except ClientError as e:

            # 🚨 Safety break to avoid timeout (for assignment)
            if context.get_remaining_time_in_millis() < 3000:
                print("Stopping execution to avoid timeout")
                break

    print("==== SUMMARY ====")
    if unencrypted_buckets:
        print("Unencrypted buckets found:")
        for b in unencrypted_buckets:
            print(b)
    else:
        print("All checked buckets have encryption enabled")

    return {
        "statusCode": 200,
```

```
Status: Succeeded
Test Event Name: hello-world

Response:
{
```

## Step 5: Testing and Verification

The Lambda function was manually invoked. CloudWatch logs were reviewed to identify S3 buckets without server-side encryption enabled.



Log events

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

```
   Timestamp                   Message
                               No older events at this moment. Retry
▶  2026-01-01T11:15:47.310Z    INIT_START Runtime Version: python:3.14.v32 Runtime Version ARN: arn:aws:lambda:us-west-1::runtime:1ee4e6d61a50fbb29d03b87572cc627d0a92de845
▶  2026-01-01T11:15:47.626Z    START RequestId: f1fb0cba-e519-4a05-a805-d52adfdc5aaf Version: $LATEST
▶  2026-01-01T11:15:51.291Z    Bucket '8273737377336722.s3.us-east.2.amazonaws.1999' has encryption enabled
▶  2026-01-01T11:15:51.414Z    Bucket '8276328637273.s3.us-east.2.amazonaws.2022' has encryption enabled
▶  2026-01-01T11:15:51.733Z    Bucket '8757848488886-s3.us-east-1.amazonaws.1998' has encryption enabled
▶  2026-01-01T11:15:51.833Z    Bucket '87585875858554447-s3.us-east-1.amazonaws.1889' has encryption enabled
▶  2026-01-01T11:15:51.951Z    Bucket '9981728282372.s3.us-east.2.amazonaws.2021' has encryption enabled
▶  2026-01-01T11:15:52.724Z    Bucket 'adish-demo-bucket' has encryption enabled
▶  2026-01-01T11:15:53.483Z    Bucket 'adish-s3-bucket-demo-12345' has encryption enabled
▶  2026-01-01T11:15:54.291Z    Bucket 'adish-terraform-s3' has encryption enabled
▶  2026-01-01T11:15:54.486Z    Bucket 'aditya-b14-bucket' has encryption enabled
▶  2026-01-01T11:15:55.053Z    Bucket 'aetheria-terraform-states' has encryption enabled
▶  2026-01-01T11:15:55.257Z    Bucket 'aisha-devops-lab-2025' has encryption enabled
▶  2026-01-01T11:15:55.466Z    Bucket 'aisha-s3-bucket-uswest-1' has encryption enabled
▶  2026-01-01T11:15:56.042Z    Bucket 'aisha-s3-lamda-demo' has encryption enabled
▶  2026-01-01T11:15:56.248Z    Bucket 'ajithpaul-herovired-01' has encryption enabled
▶  2026-01-01T11:15:56.844Z    Bucket 'amazonaws-cloud-cloud-views-views-storage' has encryption enabled
▶  2026-01-01T11:15:57.046Z    Bucket 'amiya-batch14-bucket' has encryption enabled
```

## Result

The Lambda function successfully identified and logged all S3 buckets that did not have server-side encryption enabled, helping improve security visibility.

## GitHub Repository

https://github.com/Rushiargade/aws-lambda-s3-encryption-monitor.git