# **Serverless Architecture**

# Automated Instance Management Using AWS Lambda and Boto3:

## **Set Up EC2 Instances:**

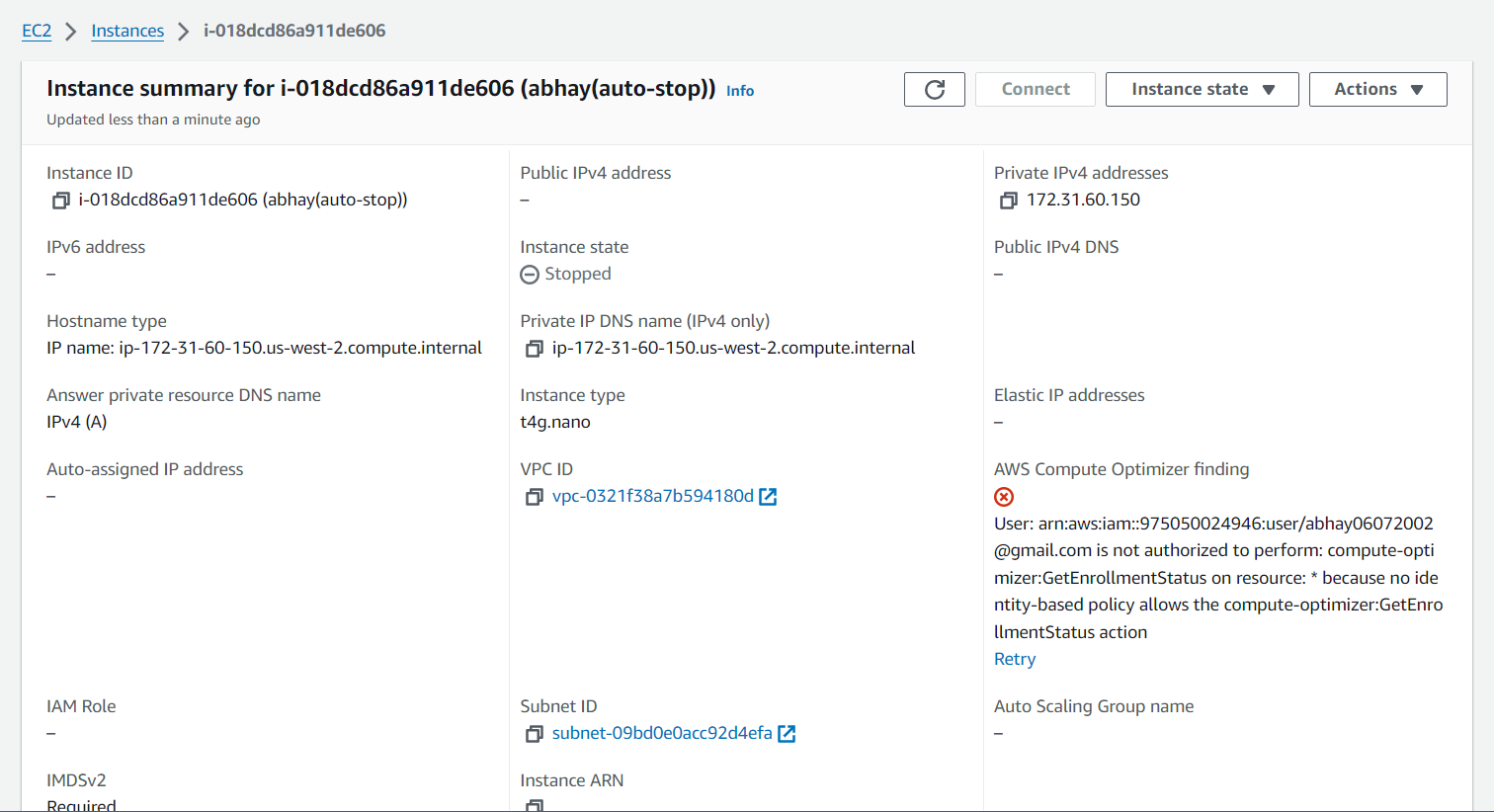
1. Create EC2 Instances:

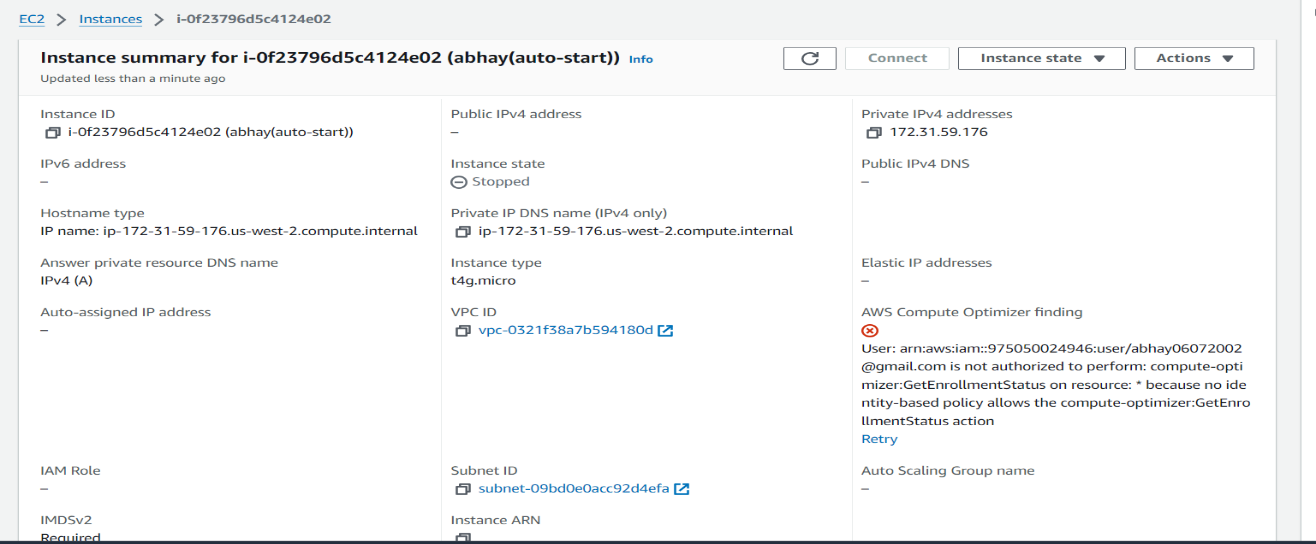
Create First Instance (Auto-Stop):

**Add Tags**: Under **Value**, type ‘Auto-Stop’.

Create second Instance (Auto-Start):

**Add Tags**: Under **Value**, type ‘Auto-Start’.





1. Set Up IAM Role for Lambda:

Create IAM Role: select AmazonEC2FullAccess policy.

Give the role a name ‘LambdaEC2Role’ and click Create Role.

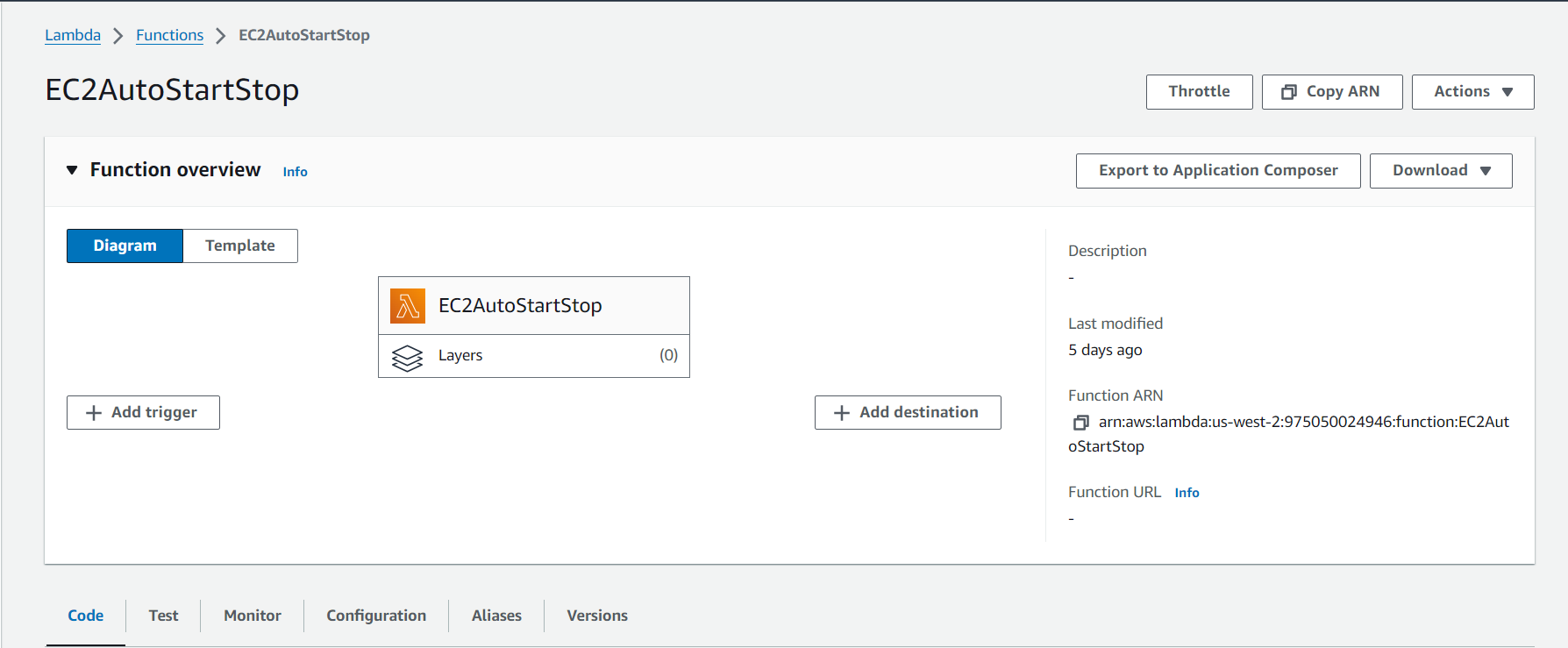
1. Create Lambda Function:

Create the Function

Function Name: EC2AutoStartStop.

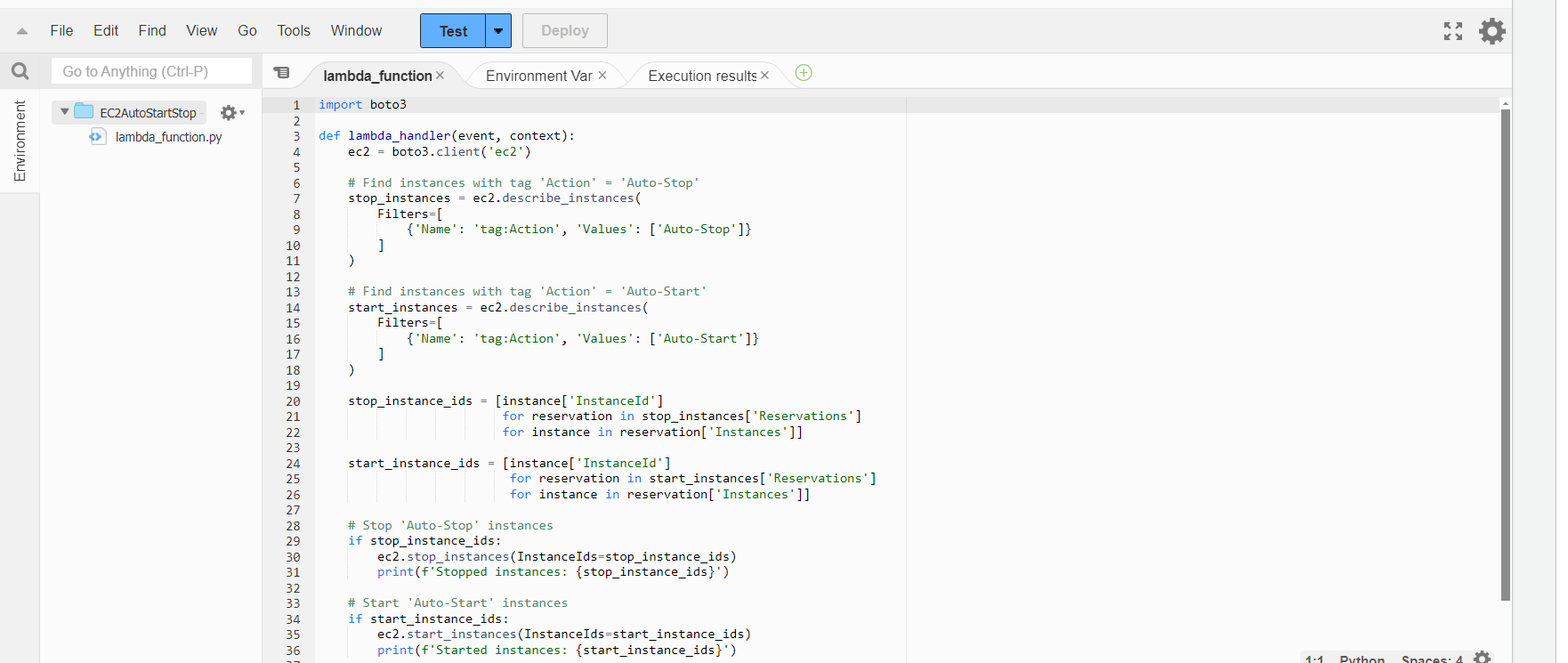
Runtime: Select Python 3.x.

Permissions: Choose the IAM role LambdaEC2Role.



1. Write the Lambda Code:

In the Lambda function code editor, paste the following Python code using Boto3:



Click **Deploy** to save the Lambda function

1. Test the Lambda Function

Manually Invoke the Function

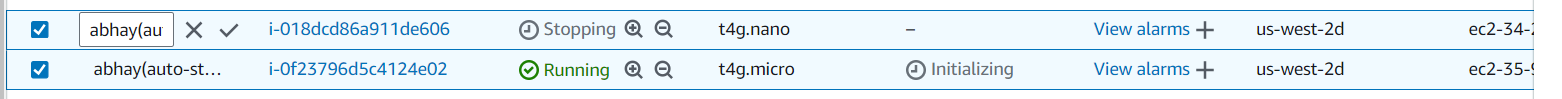
Create a new test event

Click **Test** again to run the function.

Check EC2 Instances

Go back to the EC2 Dashboard.

Verify that the instance tagged with Auto-Stop is **stopped**, and the one tagged with Auto-Start is **started**.



## **Implement a Log Cleaner for S3:**

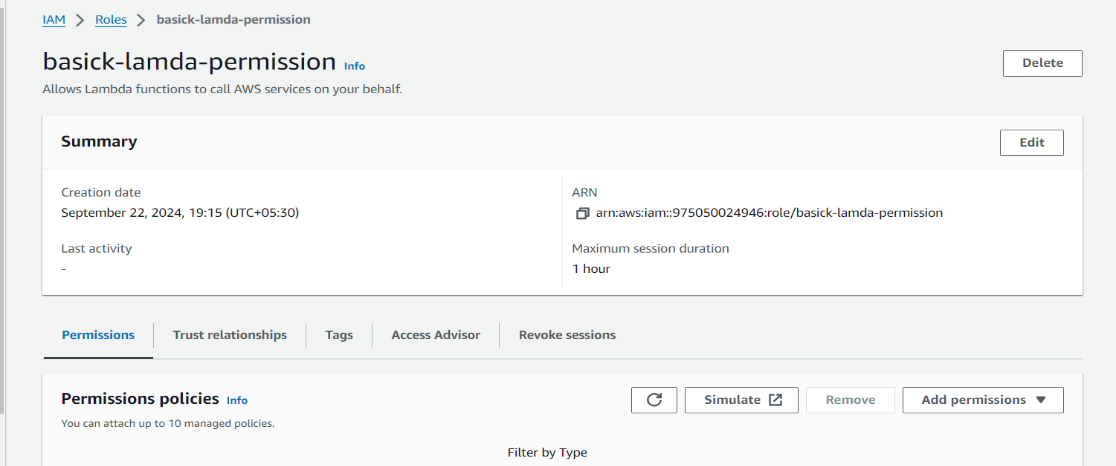
1. Set Up the IAM Role

Create a New Role:

Attach the AmazonS3FullAccess policy for now.

Attach the AWSLambdaBasicExecutionRole for logging in CloudWatch.

**Assign the Role** to your Lambda function under the “Permissions” tab.



1. Create a New Lambda Function

Go to AWS Lambda Console

Choose "Author from scratch":

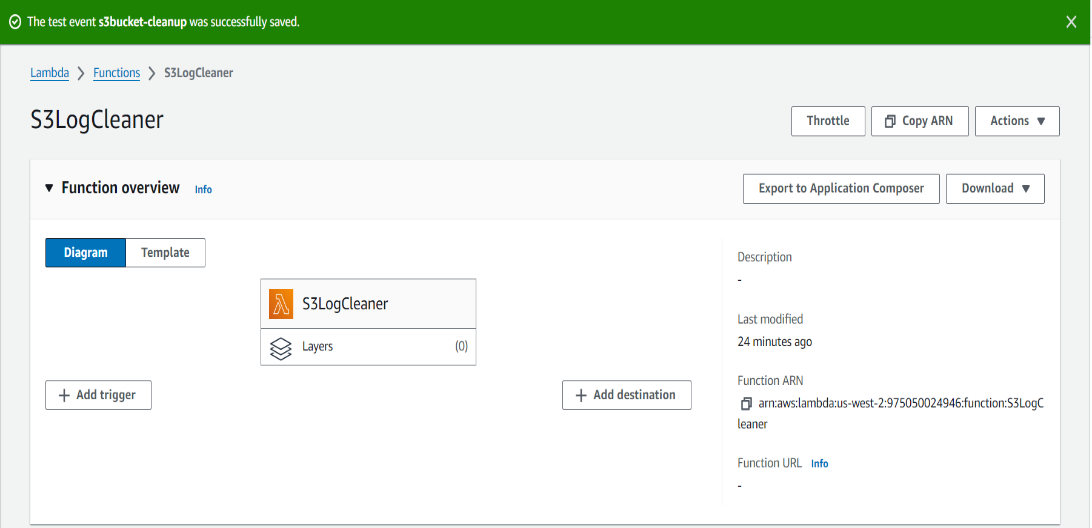
Name your function S3LogCleaner.

Runtime: Choose Python 3.10.

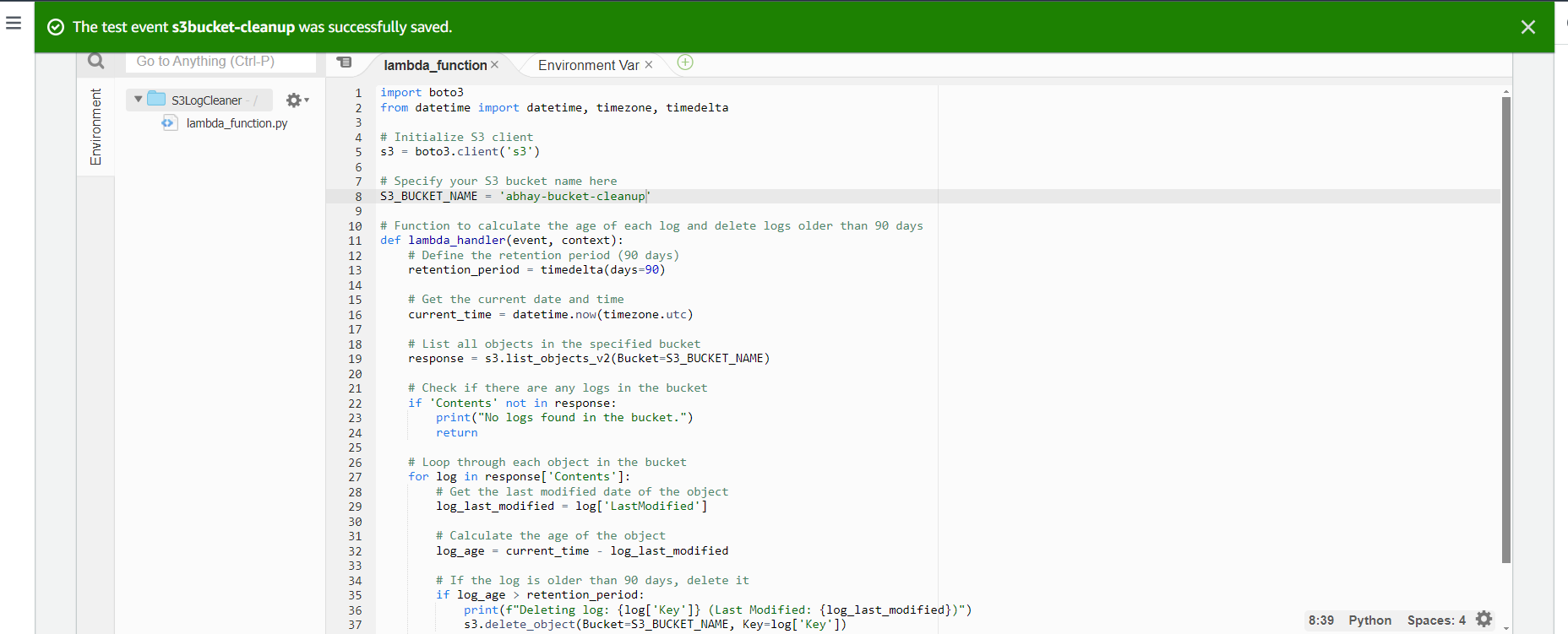
Set Permissions:

Create a new IAM role with basic Lambda execution permissions.

Attach the AmazonS3FullAccess policy for this exercise. In a real-world scenario, use a more restrictive policy.



1. Write the Boto3 Code for Log Cleaner



1. Schedule the Function with EventBridge:

To ensure the Lambda function runs weekly, we will schedule it using AWS EventBridge

Go to EventBridge:

Open the [Amazon EventBridge console](https://console.aws.amazon.com/events).

Click on Create rule.

Configure Rule Details:

Name: Name the rule WeeklyS3LogCleanup.

Event Source: Choose Event Source as EventBridge Scheduler.

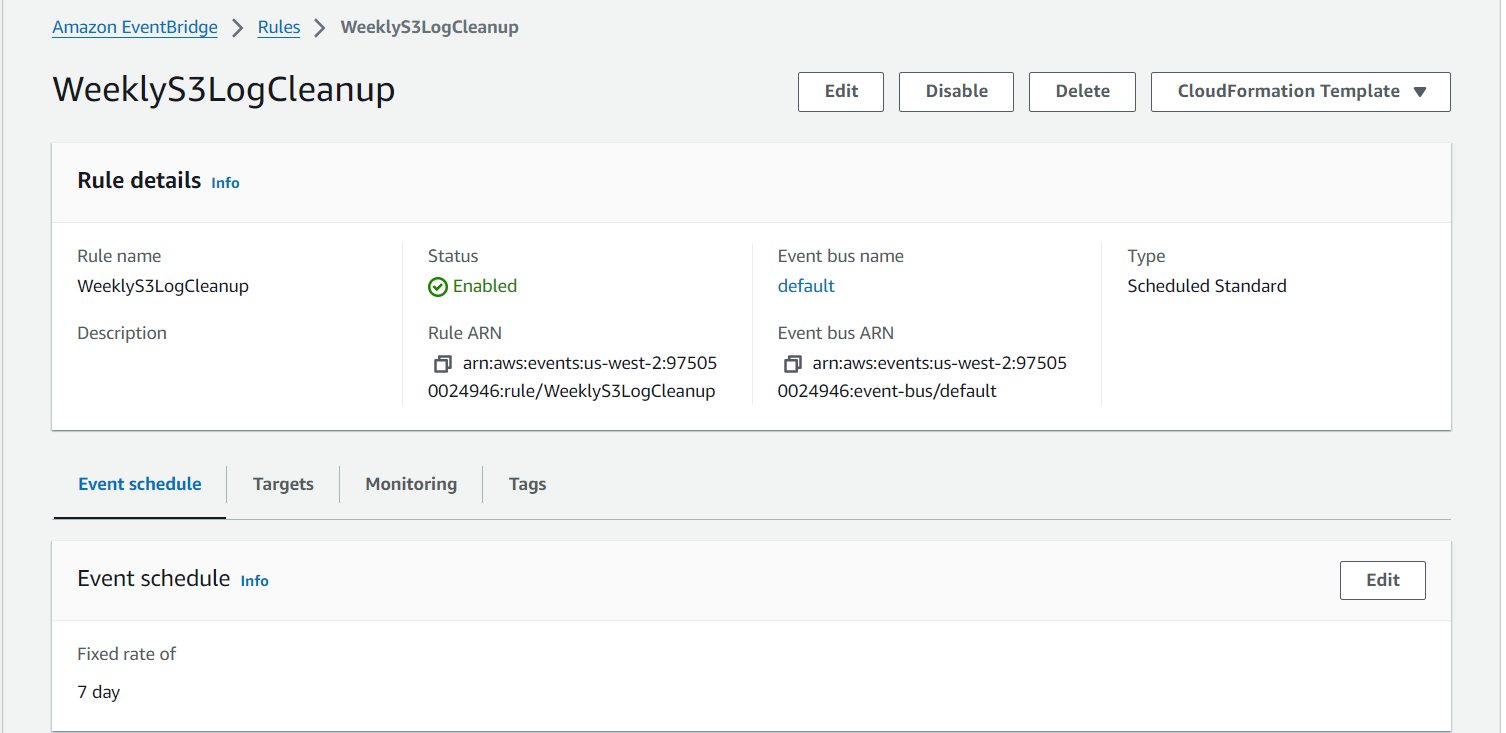
Schedule:

Choose the schedule expression as rate (7 days) to trigger the function weekly.

**Target**:

Add a **Target**.

Select Lambda Function as the target and choose the Lambda function you created.



1. Test the Function

After setting everything up, test the function manually to ensure it works before the weekly schedule kicks in.

1. Go to Lambda and click Test function.
2. Verify the Logs: Check the Lambda logs in **CloudWatch** to see which files were deleted.
3. Go to the S3 bucket and verify that files older than 90 days have been removed.

## **Automated S3 Bucket Cleanup Using AWS Lambda and Boto3:**

1. Set Up S3 Bucket

Log into AWS Console: Navigate to [AWS Management Console](https://aws.amazon.com/console/).

Go to the S3 Dashboard:

In the search bar, type S3 and click on S3.

Click Create Bucket.

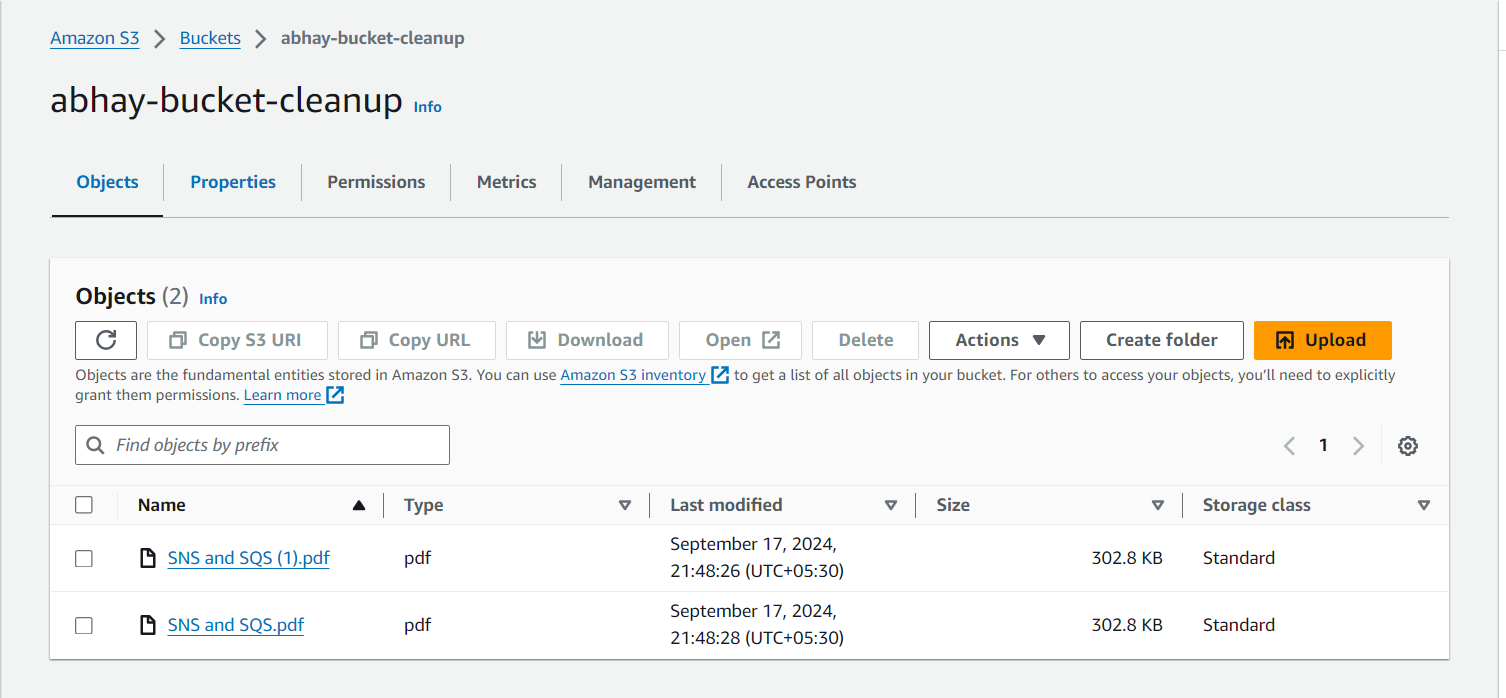
Create the Bucket:

Name of bucket my-bucket-cleanup and choose a region.

Leave the default settings for Block Public Access and other configurations.

Click Create Bucket.

Upload Files to the Bucket



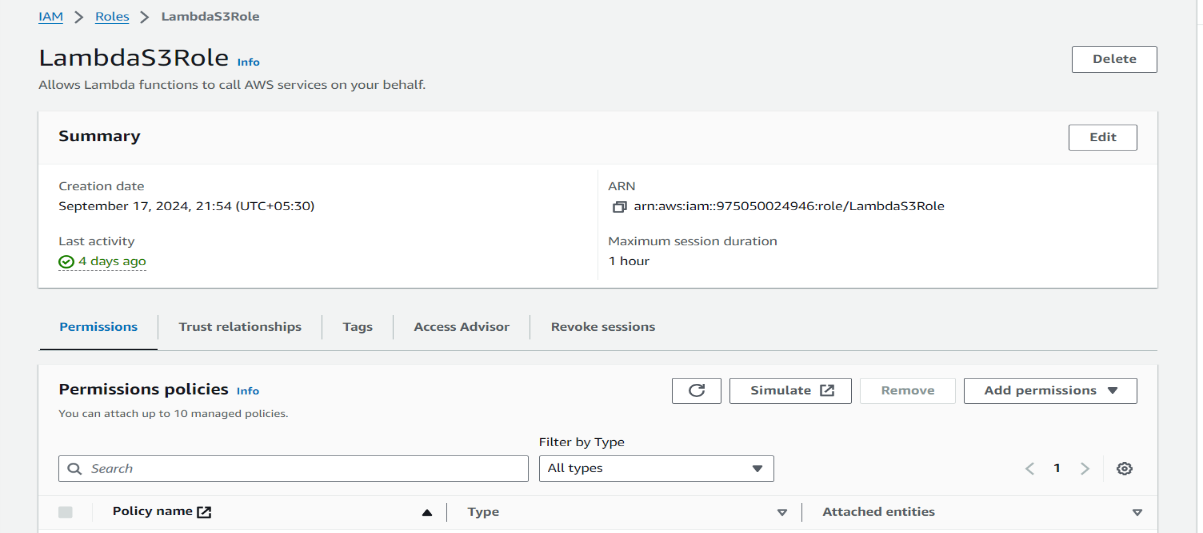
1. Set Up IAM Role for Lambda

In the AWS Console, open IAM Dashboard.

click Create Role select AWS Service and choose Lambda.

Click Next, then search for and select AmazonS3FullAccess policy.

Give the role a name LambdaS3Role and click Create Role.



1. Create Lambda Function

Create the Function

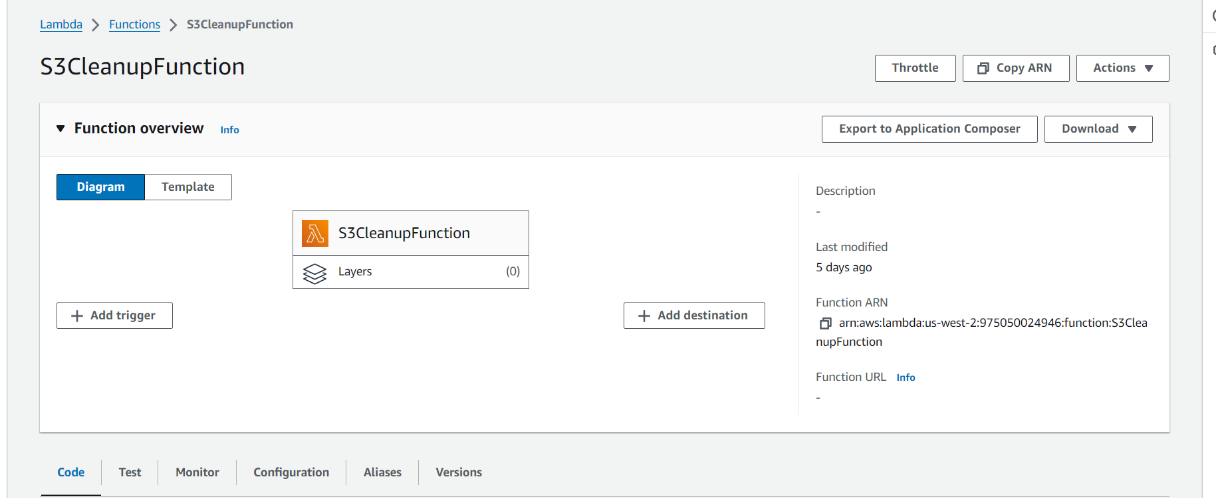
Go to the Lambda Dashboard

Function Name: S3CleanupFunction.

Runtime: Select Python 3.10.

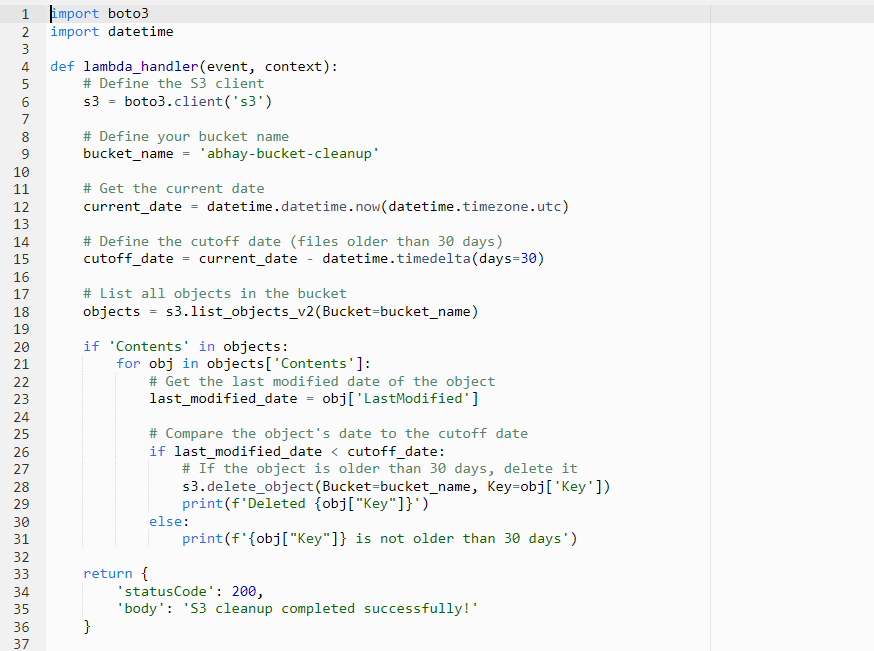
Permissions: Choose the IAM role LambdaS3Role.

Click Create Function.



1. Write the Lambda Code

Click **Deploy** to save the Lambda function.



1. Test the Lambda Function

**Manually Invoke the Function**

In the Lambda function dashboard, click **Test**.

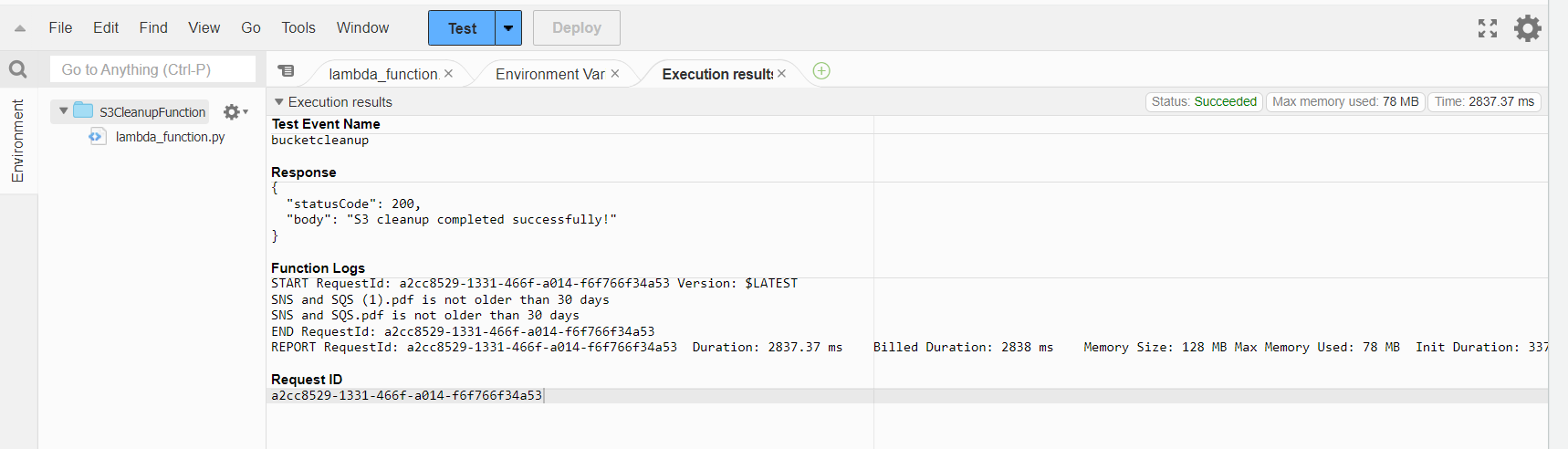
Create a new test event.

Click **Test** again to run the function.

**Check S3 Bucket**

Go back to the **S3 Dashboard**.

Verify that files older than 30 days have been deleted, and newer files are still present.



## **Automatic EBS Snapshot and Cleanup Using AWS Lambda and Boto3:**

1. Set Up EBS Volume

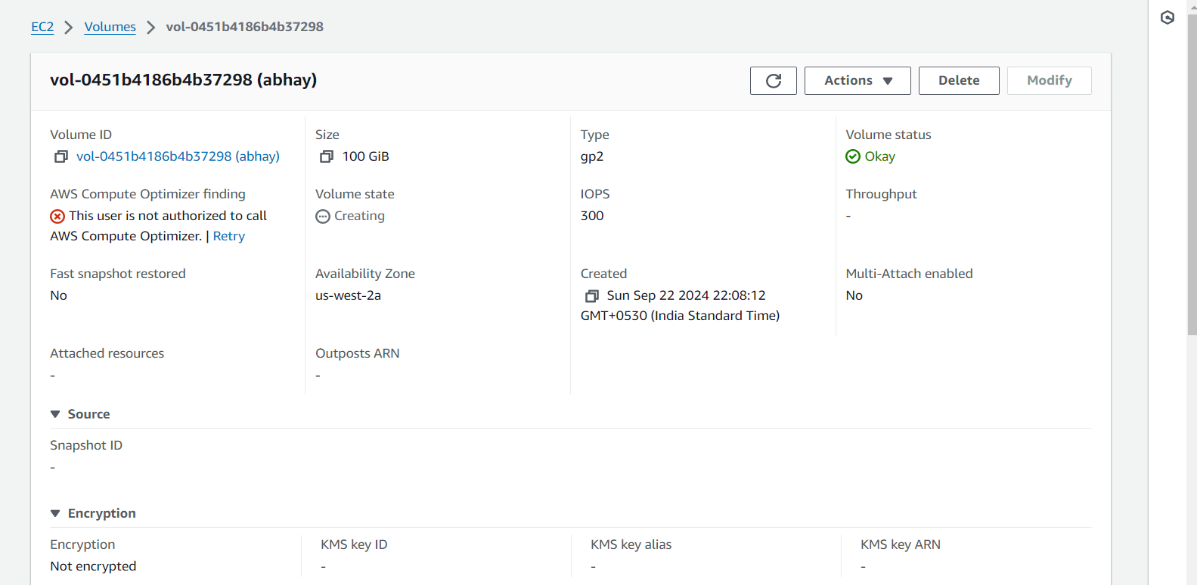
Go to EC2 Dashboard:

In the search bar, type EC2 and click on EC2 Dashboard.

On the left side, click Volumes under Elastic Block Store.

Identify Volume:

Select an existing volume you want to back up, or click Create Volume.



1. Set Up IAM Role for Lambda

Create IAM Role

In the AWS Console, search for IAM and open IAM Dashboard.

On the left sidebar, click Roles, then click Create Role.

Under Trusted Entity Type, select AWS Service and choose Lambda.

Click Next, then search for and select AmazonEC2FullAccess policy.

Give the role a name LambdaEBSRole, and click Create Role.

1. Create Lambda Function

Create the Function

Go to the Lambda Dashboard

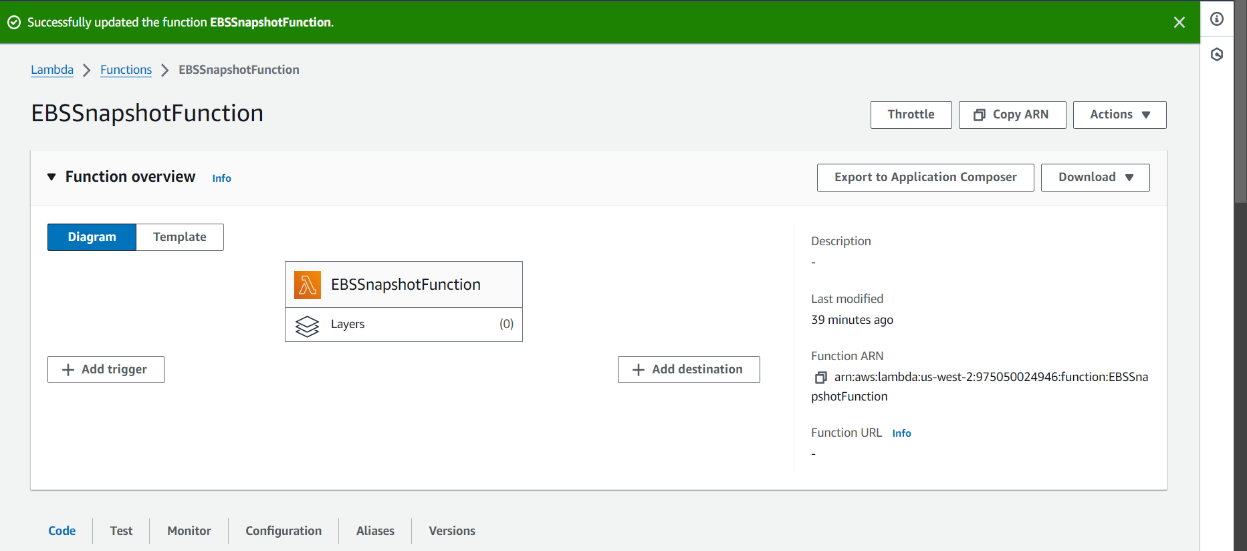
Fill in the details:

Function Name: EBSSnapshotFunction.

Runtime: Select Python 3.10.

Permissions: Choose the IAM role you created earlier LambdaEBSRole.

Click Create Function.



1. Write the Lambda Code

Click Deploy to save the Lambda function.



1. Test the Lambda Function

Manually Invoke the Function

In the Lambda function dashboard, click Test.

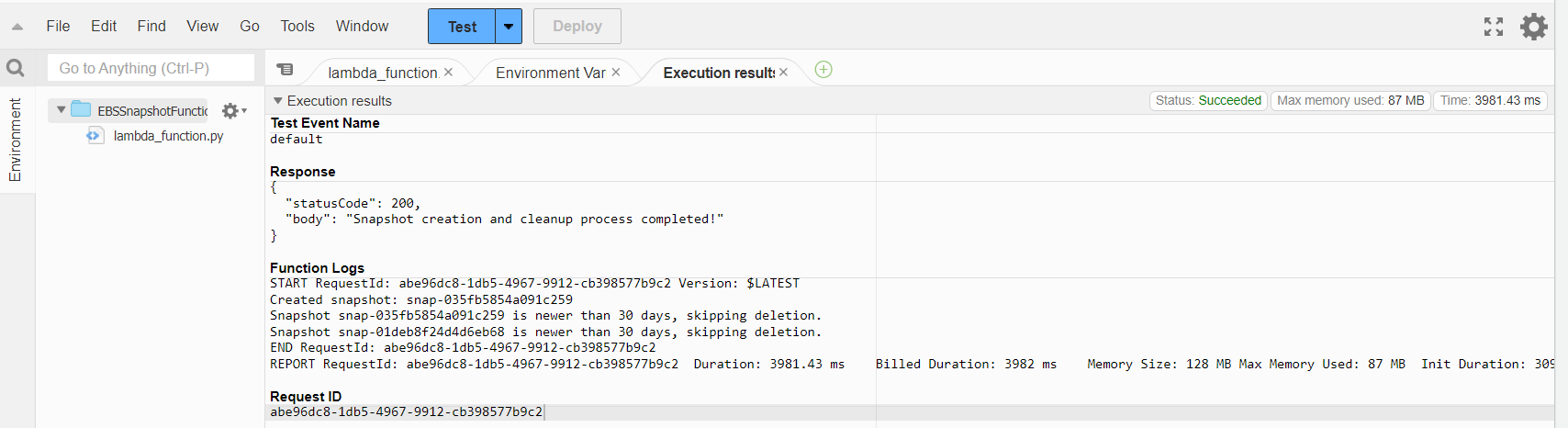
Create a new test event.

Click Test again to run the function.

Check Snapshots

Go to the EC2 Dashboard and select Snapshots under Elastic Block Store.

Confirm that a new snapshot was created, and older snapshots were deleted.



1. Set Up Event Source (Bonus)

Go to Amazon CloudWatch in the AWS Console.

On the left sidebar, click Rules, then Create Rule.

Under Event Source, choose Event Source Type as Schedule:

Define your schedule.

Under Targets, choose Lambda Function, and select your EBSBackupCleanup Lambda function.

Click Create to save the CloudWatch rule.