

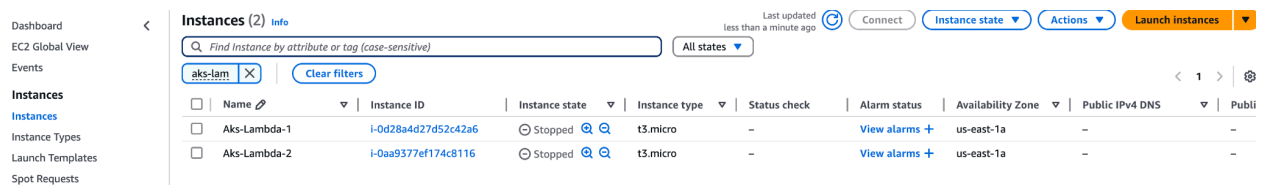
Lambda-Serverless-Architecture

Assignment 1: Automated Instance Management Using AWS Lambda and Boto3

Step1:

Created 2 EC2 instances t3.micro with tags

- Tag the first instance with a key `Action` and value `Auto-Stop`.
- Tag the second instance with a key `Action` and value `Auto-Start`.

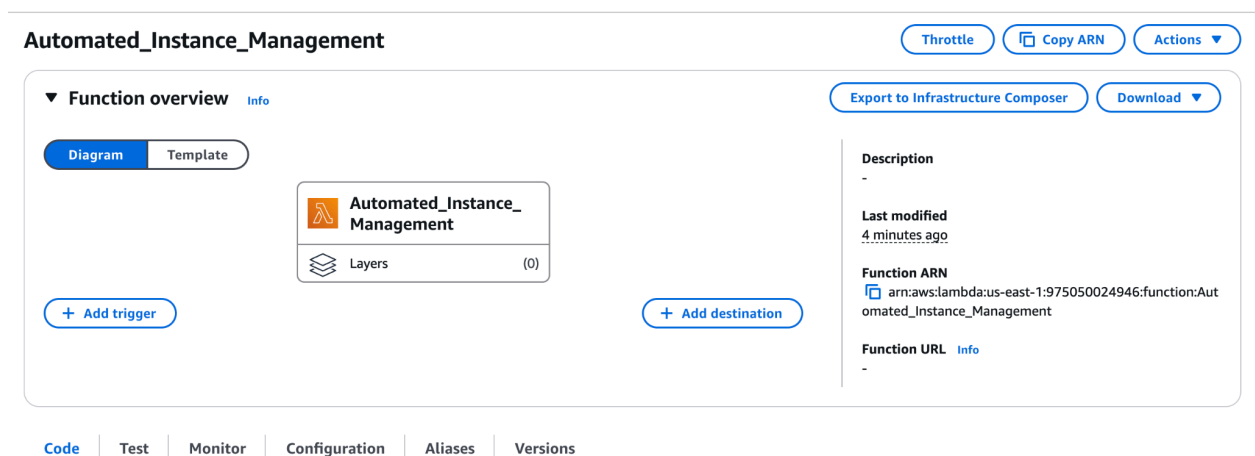


The screenshot shows the AWS Management Console 'Instances' page. Two EC2 instances are listed: 'Aks-Lambda-1' and 'Aks-Lambda-2', both with the 't3.micro' instance type and 'Stopped' state. The first instance has the tag 'Action' with the value 'Auto-Stop', and the second has 'Action' with the value 'Auto-Start'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Aks-Lambda-1	i-0d28a4d27d52c42a6	Stopped	t3.micro	-	View alarms +	us-east-1a	-	-
Aks-Lambda-2	i-0aa9377ef174c8116	Stopped	t3.micro	-	View alarms +	us-east-1a	-	-

Step2:

- Create a new function named "Automated_Instance_Management"
- Attach the `AmazonEC2FullAccess` policy to this function role.



The screenshot shows the 'Automated_Instance_Management' function overview in the AWS Lambda console. The function is in the 'Diagram' tab, showing a single function block with no layers or triggers. The 'Function overview' section includes the function name, last modified time (4 minutes ago), function ARN, and function URL.

Automated_Instance_Management

Function overview

Diagram | Template

Automated_Instance_Management

Layers (0)

+ Add trigger

+ Add destination

Export to Infrastructure Composer | Download

Description

Last modified 4 minutes ago

Function ARN arn:aws:lambda:us-east-1:975050024946:function:Automated_Instance_Management

Function URL Info

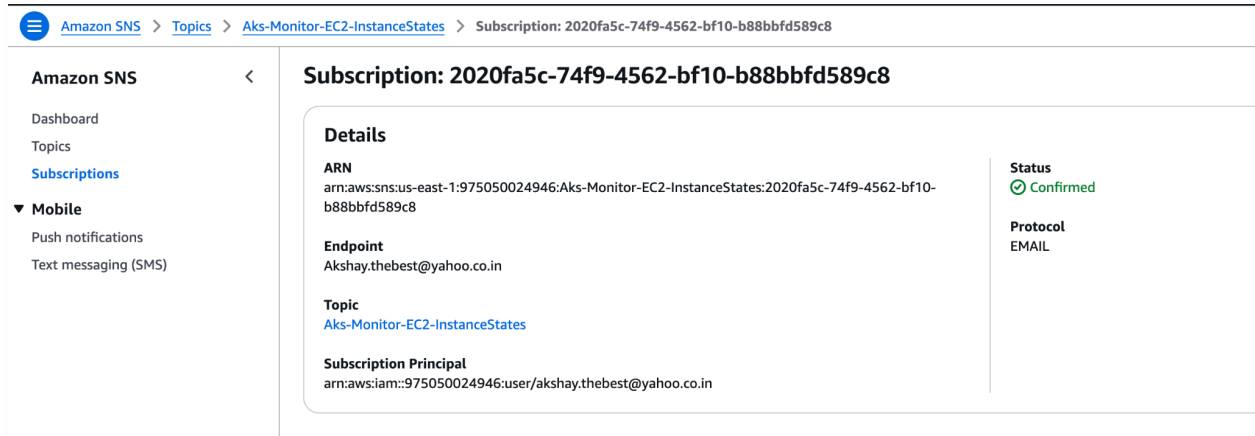
Code | Test | Monitor | Configuration | Aliases | Versions

- Code attached to git repo and deployed.
- Ran test to manually trigger it and confirm that the instances' states have changed according to their tags.

Assignment 14: Monitor EC2 Instance State Changes Using AWS Lambda, Boto3, and SNS

Step1.

SNS dashboard and create a new topic.

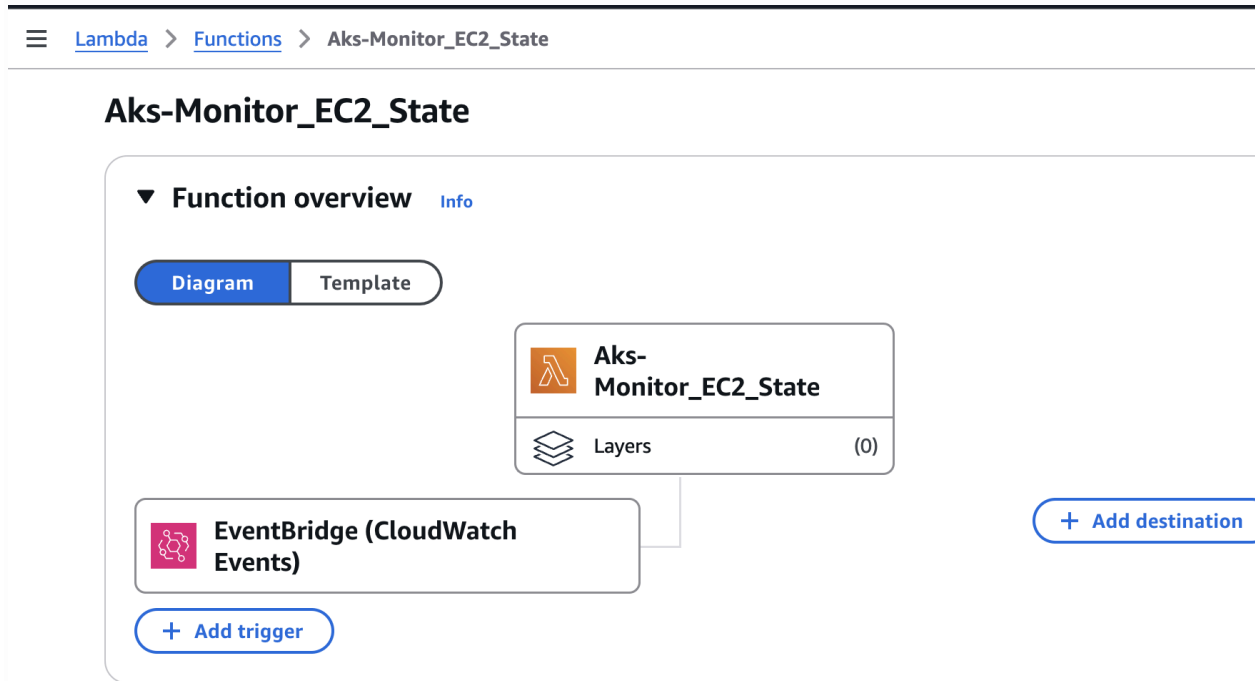


The screenshot shows the Amazon SNS console. The breadcrumb navigation is: Amazon SNS > Topics > Aks-Monitor-EC2-InstanceStates > Subscription: 2020fa5c-74f9-4562-bf10-b88bbfd589c8. The left sidebar shows the navigation menu with options: Dashboard, Topics, Subscriptions (selected), Mobile, Push notifications, and Text messaging (SMS). The main content area is titled "Subscription: 2020fa5c-74f9-4562-bf10-b88bbfd589c8" and contains the following details:

Details	
ARN arn:aws:sns:us-east-1:975050024946:Aks-Monitor-EC2-InstanceStates:2020fa5c-74f9-4562-bf10-b88bbfd589c8	Status ✔ Confirmed
Endpoint Akshay.thebest@yahoo.co.in	Protocol EMAIL
Topic Aks-Monitor-EC2-InstanceStates	
Subscription Principal arn:aws:iam::975050024946:user/akshay.thebest@yahoo.co.in	

Step2.

Create a lambda function and assign an IAM role with EC2 and SNS access.



The screenshot shows the AWS Lambda console for the function "Aks-Monitor_EC2_State". The breadcrumb navigation is: Lambda > Functions > Aks-Monitor_EC2_State. The function overview section shows the function icon, name, and layers (0). Below the function overview, there is a diagram showing the function connected to an EventBridge (CloudWatch Events) trigger. There are buttons for "Diagram", "Template", "+ Add trigger", and "+ Add destination".

- Code attached to git repo and deployed.

- Ran test to manually trigger it and confirm that the instances' states have changed and you have received the notification via email

Step3:

Created an Event Bridge rule to trigger your Lambda function whenever an EC2 instance state changes..

Amazon EventBridge > Rules > Aks-MonitorEC2_states

Amazon EventBridge

hboard [New](#)

reloper resources

rn

dbox

ck starts

ies

nt buses

es

bal endpoints

ives

lays

es

is

eduler

edules

edule groups

agration

near event source

Aks-MonitorEC2_states

Edit Disable

Rule details [Info](#)

Rule name
Aks-MonitorEC2_states

Description
MonitorEC2_states

Status
✔ Enabled

Rule ARN
[arn:aws:events:us-east-1:97505002494:6:rule/Aks-MonitorEC2_states](#)

Event bus name
[default](#)

Event bus ARN
[arn:aws:events:us-east-1:97505002494:6:event-bus/default](#)

Event pattern [Info](#)

```

1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"]
4 }

```

[Copy](#)

Edit Disable Delete Close

Rule details [Info](#)

Rule name
Aks-MonitorEC2_states

Description
MonitorEC2_states

Status
✔ Enabled

Rule ARN
[arn:aws:events:us-east-1:97505002494:6:rule/Aks-MonitorEC2_states](#)

Event bus name
[default](#)

Event bus ARN
[arn:aws:events:us-east-1:97505002494:6:event-bus/default](#)

Type
Standard

Event pattern **Targets** Monitoring Tags

Targets


Details	Target Name	Type	Arn	Input	Role
▼	Aks-Monitor_EC2_State 🔗	Lambda function	arn:aws:lambda:us-east-1:975050024946:function:Aks-Monitor_EC2_State	Matched event	-

Input to target: Matched event

Additional parameters: --


Dead-letter queue (DLQ): -

You will receive SNS over email specified with below content in it:

**Amazon Web Services**
amazonaws.com

Visit site →

● EC2 Instance State Change: i-0d28a4d27d52c42a6Akshay,the... /Deleted...

● **AWS Notifications** [amazonaws.com](#) >Thu, 19 Dec at 3:29 pm


From: no-reply@sns.amazonaws.com
To: akshay.thebest@yahoo.co.in

{"InstanceId": "i-0d28a4d27d52c42a6", "State": "stopping", "InstanceType": "t3.micro", "LaunchTime": "2024-12-19 09:55:07"}

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:3975050024946:Aks-Monitor-EC2-InstanceStates:2020fa5c-74f9-4562-bf10-b88bbfd589c8&Endpoint=Akshay.thebest@yahoo.co.in>

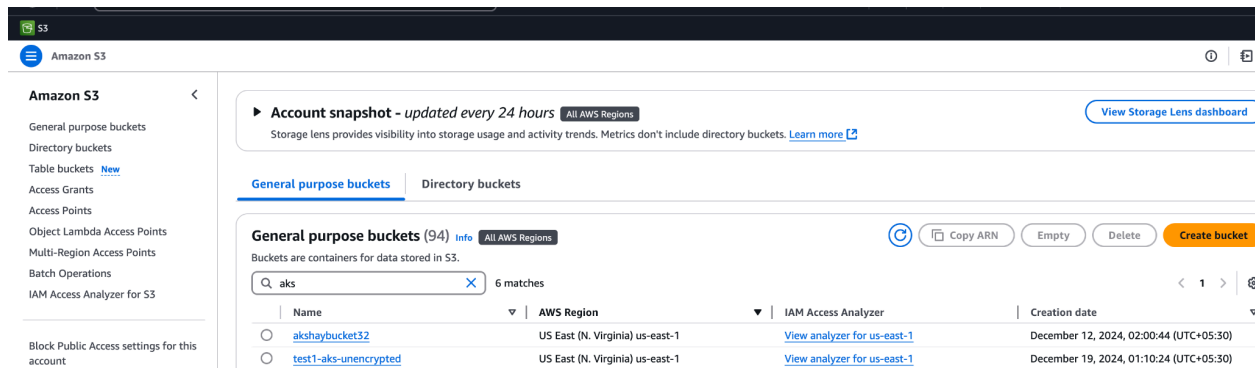
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>



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

1. S3 Setup:

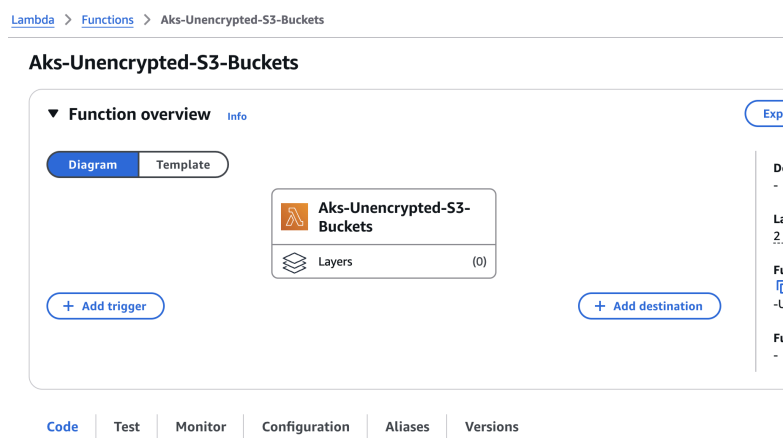
Tried creating S3 buckets but all have SSE enabled, unable to create a KMS key or choose any other option.



2. Lambda function:

-Created function named “Aks-Unencrypted-S3-Buckets”

-Using IAM role “Ralph-S3readonly” since permissions to create any role is not with this account.



- Code attached to git repo and deployed.
- Ran test to manually trigger it and confirm that all S3 buckets have SSE enabled or not.

Assignment 2: Automated S3 Bucket Cleanup Using AWS Lambda and Boto3

Step1:

Existing buckets in S3, named “akshaybucket32”

Step2:

Created IAM role “Aks-S3-bucket_cleaner-role-a7u6lpb1” with S3 full access.

Step3:

Created lambda function and applied above IAM role to it,

The screenshot displays two AWS console interfaces. The top interface is the Amazon S3 console for bucket 'akshaybucket32'. It shows a list of objects with the following details:

Name	Type	Last modified	Size	Storage class
lambda-challenge work.docx	docx	December 17, 2024, 01:31:37 (UTC+05:30)	14.0 KB	Standard
Unix-Shell-Scripting-Case-Study-Log-Analysis-Tool.pdf	pdf	December 17, 2024, 01:31:38 (UTC+05:30)	348.2 KB	Standard

The bottom interface is the AWS Lambda console showing the execution logs for the function 'Aks-S3-bucket_cleaner'. The logs indicate a successful execution with the following summary:

Code SHA-256	Execution time
ZrcP6OeUjIP5xwldqIHUvBuJGYDvbTrUHud2JV5t/Q=	33 seconds ago

Additional details from the logs include:

- Request ID: 0086902f-44ef-4325-a3a8-b1c0f45962c9
- Init duration: 282.48 ms
- Billed duration: 2720 ms
- Max memory used: 86 MB
- Function version: \$LATEST
- Duration: 2719.10 ms
- Resources configured: 128 MB

The log output shows the following actions:

```
Deleted object: Unix-Shell-Scripting-Case-Study-Log-Analysis-Tool.pdf
Deleted object: lambda-challenge work.docx
Deleted objects older than 30 days:
- Unix-Shell-Scripting-Case-Study-Log-Analysis-Tool.pdf
- lambda-challenge work.docx
No objects older than 30 days were found.
END RequestId: 0086902f-44ef-4325-a3a8-b1c0f45962c9
REPORT RequestId: 0086902f-44ef-4325-a3a8-b1c0f45962c9 Duration: 2719.10 ms Billed Duration: 2720 ms Memory Size: 128 MB Max Memory Used: 86 MB
Init Duration: 282.48 ms
```

Step4:

Triggered it to find the accuracy, wrote Boto3 Python script attached to git repo.

Amazon S3 > Buckets > akshaybucket32

Amazon S3
[General purpose buckets](#)
Legacy buckets
Le buckets [New](#)
Access Grants
Access Points
Select Lambda Access Points
Multi-Region Access Points
CloudFront Operations
Access Analyzer for S3

Check Public Access settings for this bucket

Storage Lens
Storage Dashboards

akshaybucket32 [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (0) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create](#)

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 permissions. [Learn more](#)

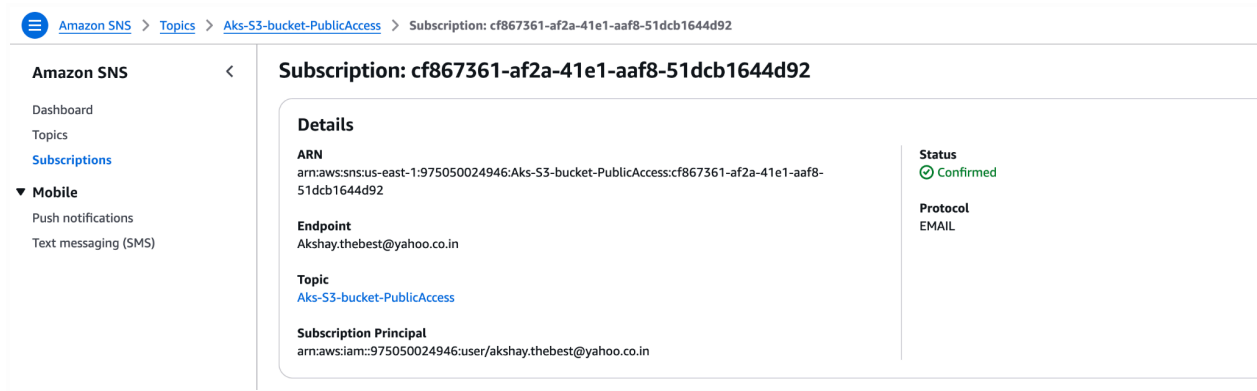
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
No objects					
You don't have any objects in this bucket.					

[Upload](#)

Assignment 13: Audit S3 Bucket Permissions and Notify for Public Buckets

Step1:

Created a new SNS topic “[Aks-S3-bucket-PublicAccess](#)”.



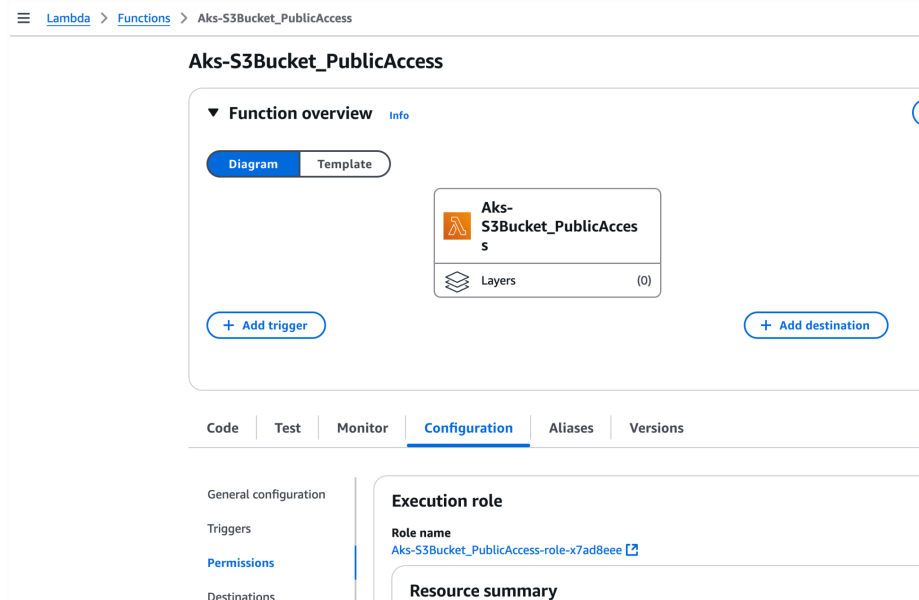
The screenshot shows the Amazon SNS console. The breadcrumb navigation is: Amazon SNS > Topics > Aks-S3-bucket-PublicAccess > Subscription: cf867361-af2a-41e1-aaf8-51dcb1644d92. The left sidebar shows the 'Subscriptions' link under the 'Topics' section. The main content area is titled 'Subscription: cf867361-af2a-41e1-aaf8-51dcb1644d92'. It contains a 'Details' section with the following information:

- ARN:** arn:aws:sns:us-east-1:975050024946:Aks-S3-bucket-PublicAccess:cf867361-af2a-41e1-aaf8-51dcb1644d92
- Endpoint:** Akshay.thebest@yahoo.co.in
- Topic:** [Aks-S3-bucket-PublicAccess](#)
- Subscription Principal:** arn:aws:iam::975050024946:user/akshay.thebest@yahoo.co.in

On the right side of the details, there is a 'Status' section showing a green checkmark and the word 'Confirmed', and a 'Protocol' section showing 'EMAIL'.

Step2:

Created a function with an IAM role for S3fullaccess, cloudwatch and SNSfullaccess.



The screenshot shows the AWS Lambda console for the function 'Aks-S3Bucket_PublicAccess'. The breadcrumb navigation is: Lambda > Functions > Aks-S3Bucket_PublicAccess. The 'Function overview' section shows the function name and a 'Layers' section with '(0)' layers. There are buttons for '+ Add trigger' and '+ Add destination'. Below the overview, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Configuration' tab is selected, showing the 'Execution role' section with the role name 'Aks-S3Bucket_PublicAccess-role-x7ad8eee' and a link to view it. There is also a 'Resource summary' section.

Step3:

Schedule your Lambda function to run daily via Eventbridge rule.

Amazon EventBridge

<

hboard

New

Developer resources

rn

dbox

ck starts

ies

nt buses

as

bal endpoints

ives

lays

es

is

Aks_S3_PublicAccess_buckets

Edit

Disable

Delete

CloudFormation Template

Rule details

Info

Rule name

Aks_S3_PublicAccess_buckets

Description

Status

Enabled

Rule ARN

arn:aws:events:us-east-1:975050024946:rule/Aks_S3_PublicAccess_buckets

Event bus name

default

Event bus ARN

arn:aws:events:us-east-1:975050024946:event-bus/default

Type

Scheduled Standard

Event schedule

Targets

Monitoring

Tags

Event schedule

Info

Fixed rate of

24 hour

Aks_S3_PublicAccess_buckets

Edit

Disable

Delete

Rule details

Info

Rule name

Aks_S3_PublicAccess_buckets

Description

Status

Enabled

Rule ARN

arn:aws:events:us-east-1:975050024946:rule/Aks_S3_PublicAccess_buckets

Event bus name

default

Event bus ARN

arn:aws:events:us-east-1:975050024946:event-bus/default

Type

Scheduled Standard

Event schedule

Targets

Monitoring

Tags

Targets

Details	Target Name	Type	Arn	Input	Role
▼	Aks-S3Bucket_PublicAccess	Lambda function	arn:aws:lambda:us-east-1:975050024946:function:Aks-S3Bucket_PublicAccess	Matched event	-
Input to target:		Matched event			
Additional parameters:		--			
Dead-letter queue (DLQ):		-			

Step4:

Triggered it to find the accuracy, wrote Boto3 Python script attached to git repo.

✔ Executing function: succeeded ([logs](#) ²)

▼ Details

The area below shows the last 4 KB of the execution log.

null

Summary

Code SHA-256 wely3o+N+XWP+IMbWejGTjUVbxXu7hiGB1goeM2a79M=	Execution time 9 minutes ago
Request ID c241a6c4-b48d-4c6e-ba14-38d717b05342	Function version \$LATEST
Init duration 279.16 ms	Duration 77502.84 ms
Billed duration 77503 ms	Resources configured 128 MB
Max memory used 90 MB	

Log output

The section below shows the logging calls in your code. [Click here](#) ² to view the corresponding CloudWatch log group.

```
START RequestId: c241a6c4-b48d-4c6e-ba14-38d717b05342 Version: $LATEST
Public buckets with read or write permissions:
- dilna-devops9
- mahesh--devops09--bucket1
- prashant-batch9
- salman-batch9
- santosh-herovired-batch9
- vishalbbatch9
- yashhelm
END RequestId: c241a6c4-b48d-4c6e-ba14-38d717b05342
```