

ASSIGNMENT 9

AIM: To understand Lambda Function and create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.

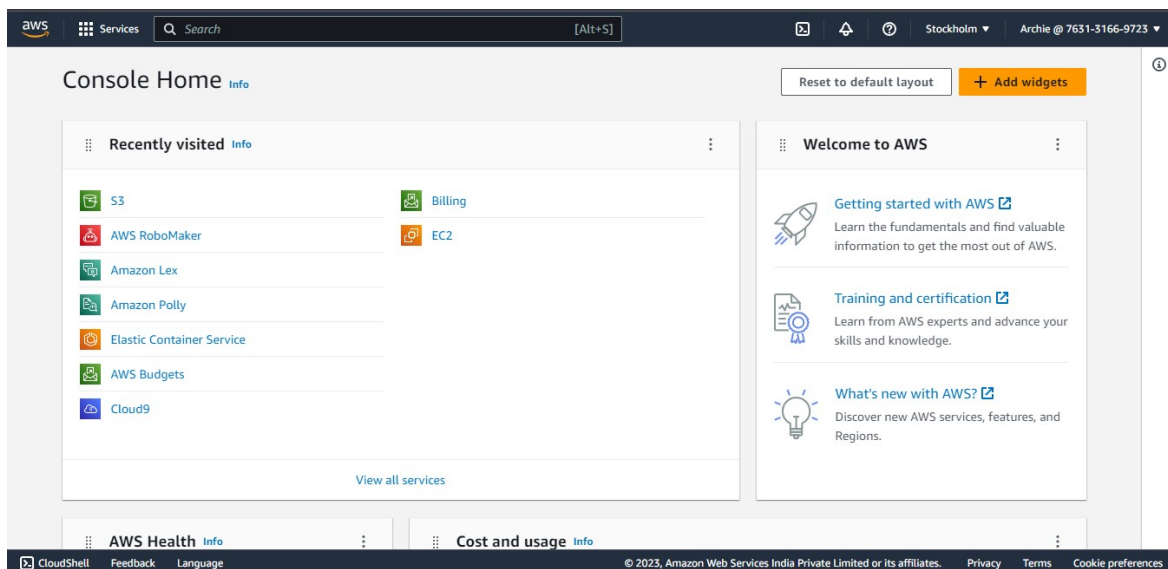
LO MAPPED: LO1, LO5

THEORY:

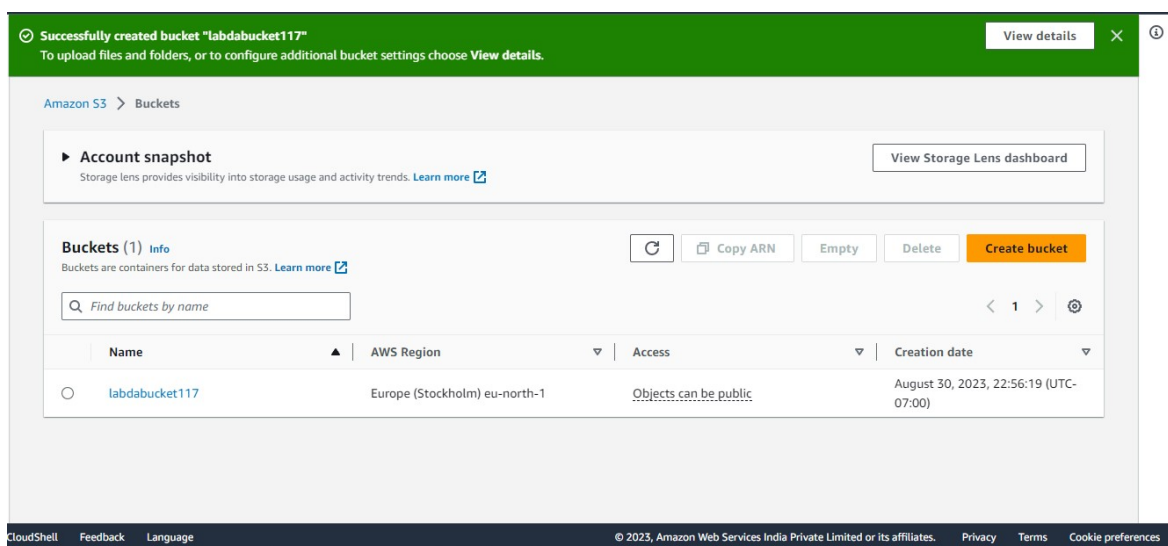
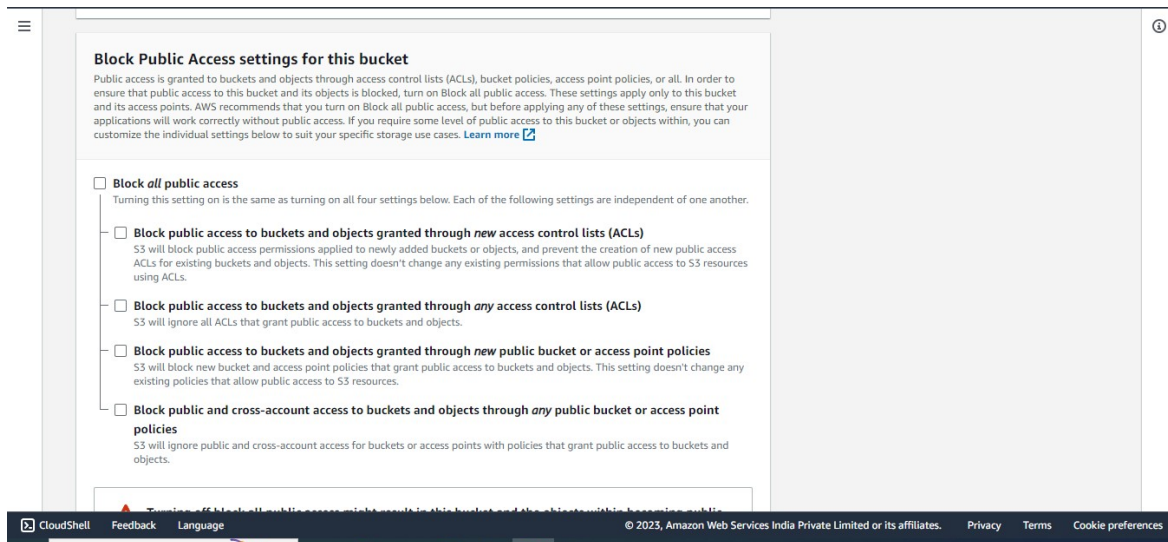
You can use Lambda to process event notifications from Amazon Simple Storage Service. Amazon S3 can send an event to a Lambda function when an object is created or deleted. You configure notification settings on a bucket, and grant Amazon S3 permission to invoke a function on the function's resource-based permissions policy.

STEPS TO FOLLOW:

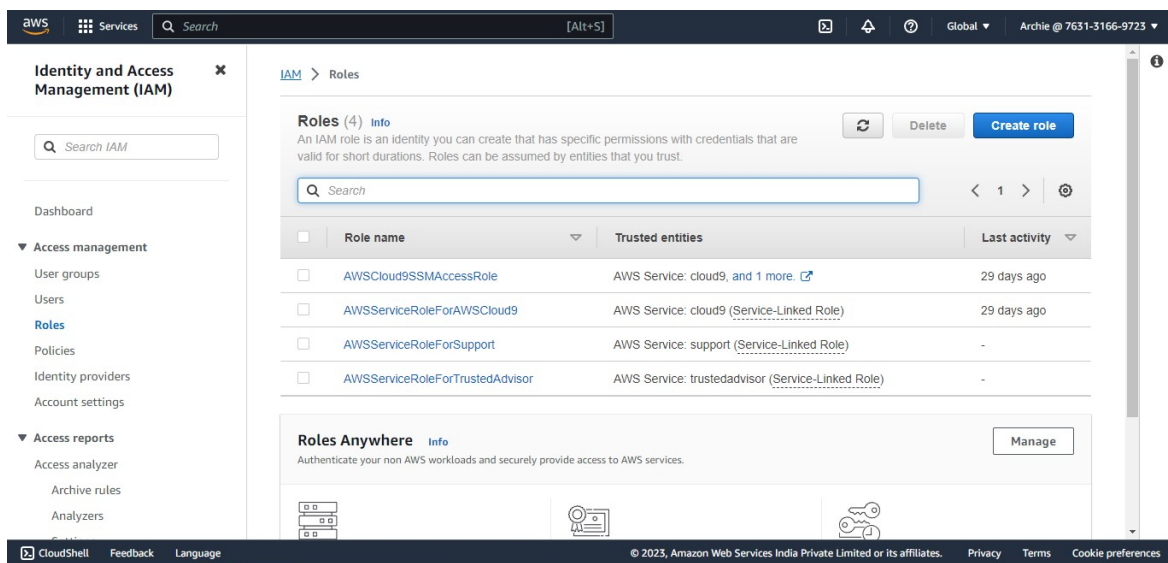
1. Log in as IAM User



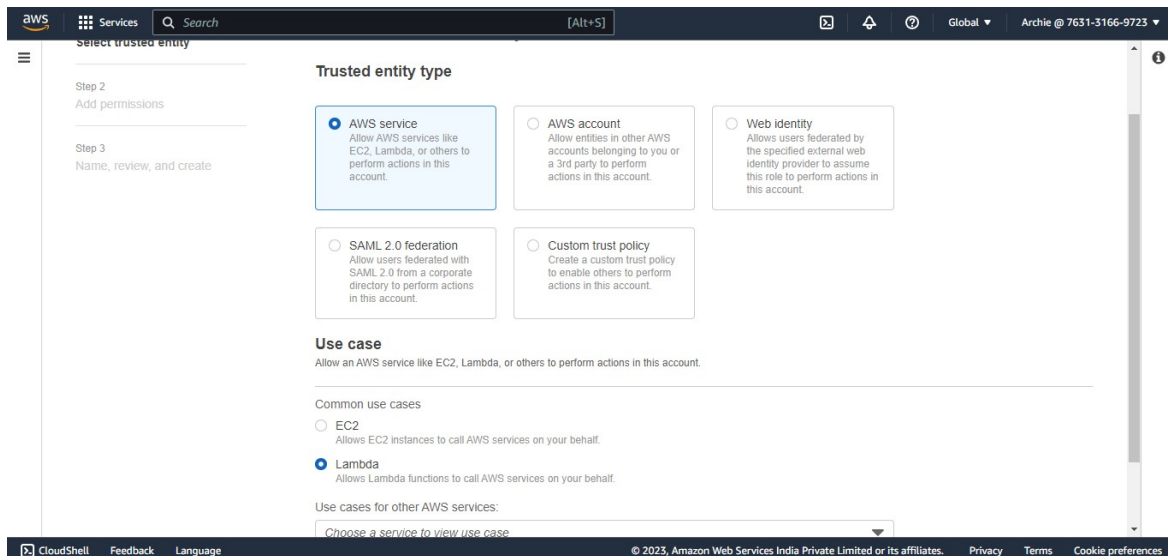
2. Create a S3 bucket and Enable the "Block all public access"



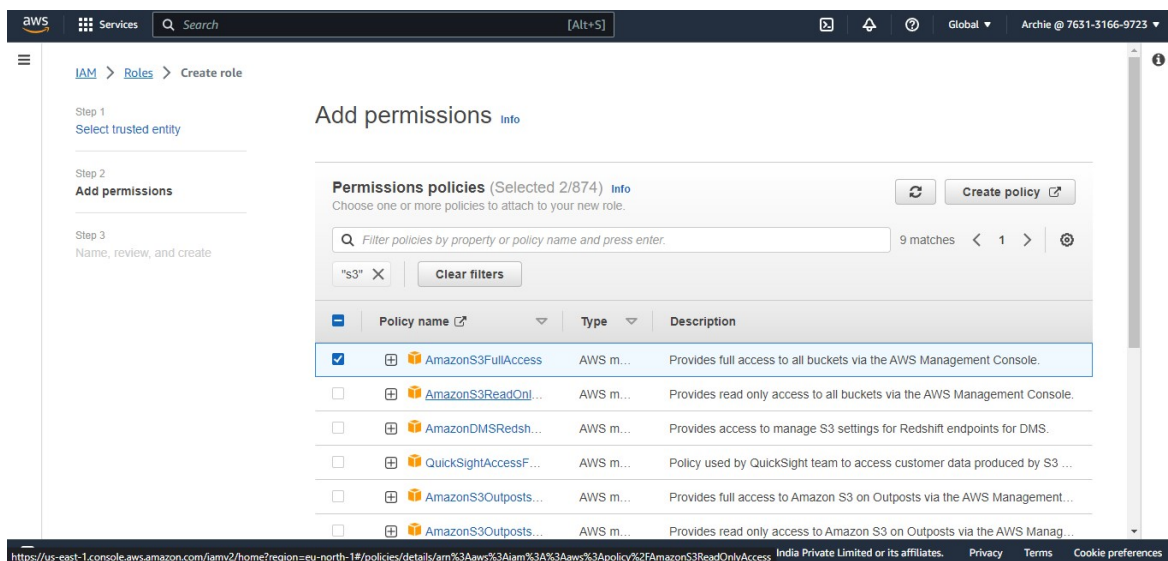
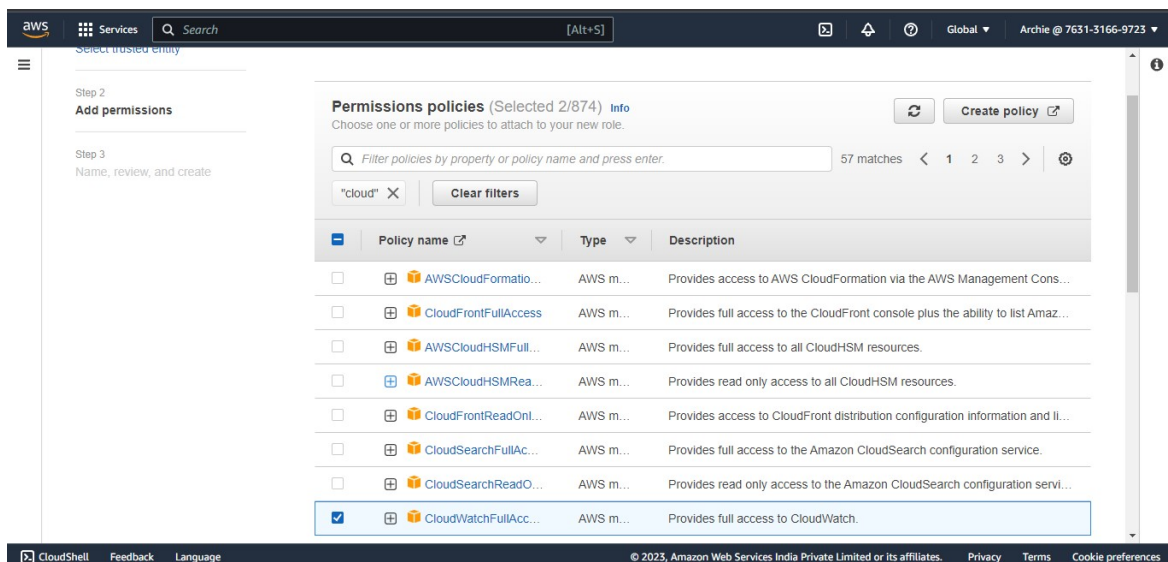
3. Search IAM on the console and go to "Roles". Click on Create Roles



4. Select the options of "AWS service" and "lambda"



5. Enable the "CloudWatchFullAccess" and "AmazonS3FullAccess"



6. Click on Create Role and you will be redirected to this dashboard. Give the Role a Name and Click on Done.

The screenshot shows the AWS IAM console 'Create role' page. The breadcrumb navigation is 'IAM > Roles > Create role'. The left sidebar shows three steps: 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. The main content area is titled 'Name, review, and create' and contains 'Role details'. Under 'Role name', there is a text input field with 'LambdaUser' entered. Below it, a description field contains 'Allows Lambda functions to call AWS services on your behalf.' At the bottom, there is a section for 'Step 1: Select trusted entities' with an 'Edit' button. The footer includes 'CloudShell', 'Feedback', 'Language', and copyright information for 2023.

7. Search for Lambda in the console and click on Create Function.

The screenshot shows the AWS Lambda console 'Get started' page. The header includes the AWS logo, 'Services', a search bar, and user information. The main content area has a dark blue background with the text 'AWS Lambda lets you run code without thinking about servers.' and a 'Create a function' button. Below this, there is a 'How it works' section with a 'Run' button and a 'Next: Lambda responds to events' button. The footer includes the URL 'https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/create/function?firstrun=true' and copyright information for 2023.

8. Change the settings of the Lambda Function.

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.
- ☐ **Browse serverless app repository**
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.
LambdaFunctionUser

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 3.11

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

9. Add the python code and click on deploy to save the changes.

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

Code source [info](#)

Upload from

File Edit Find View Go Tools Window **Test** Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

- LambdaFunctionUser
 - lambda_function.py

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     print("Image successfully uploaded in s3 bucket")
  
```

10. Scroll above and click on ADD TRIGGER. Select the following options and click on Done.

Add trigger

Trigger configuration info

S3
aws asynchronous storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/labdabucket117
Bucket region: eu-north-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](#)

11. Go to S3 bucket and click on Add files. Select a image and click on Upload.

Amazon S3 **labdabucket117** info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (0)
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](#)

Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

Upload

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 266.2 KB) Remove Add files Add folder

All files and folders in this table will be uploaded.

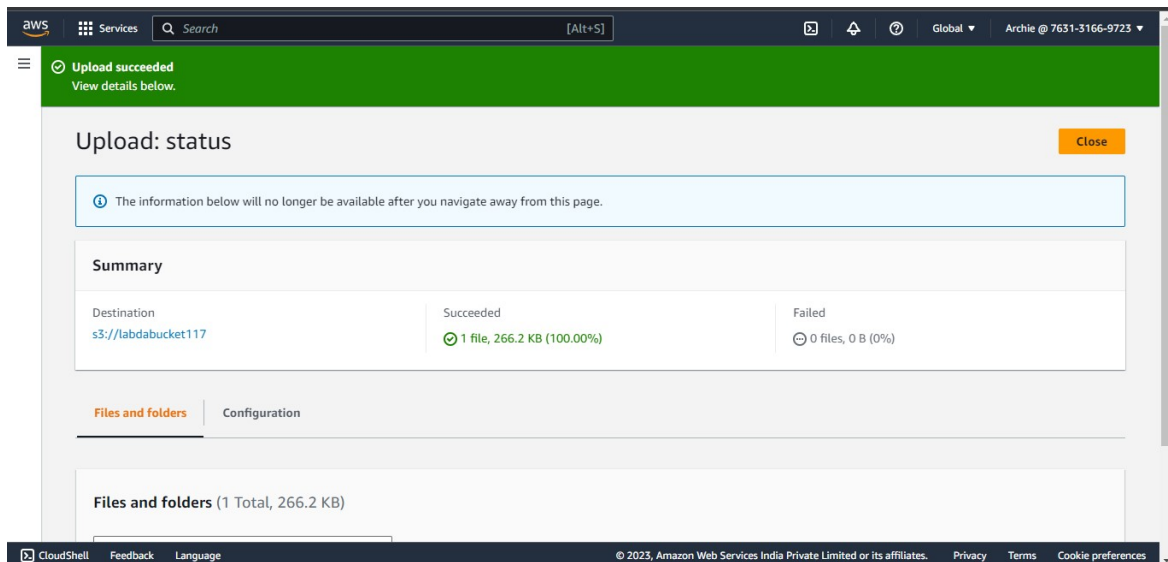
Find by name

Name	Folder	Type	Size
Screenshot (3).png	-	image/png	266.2 KB

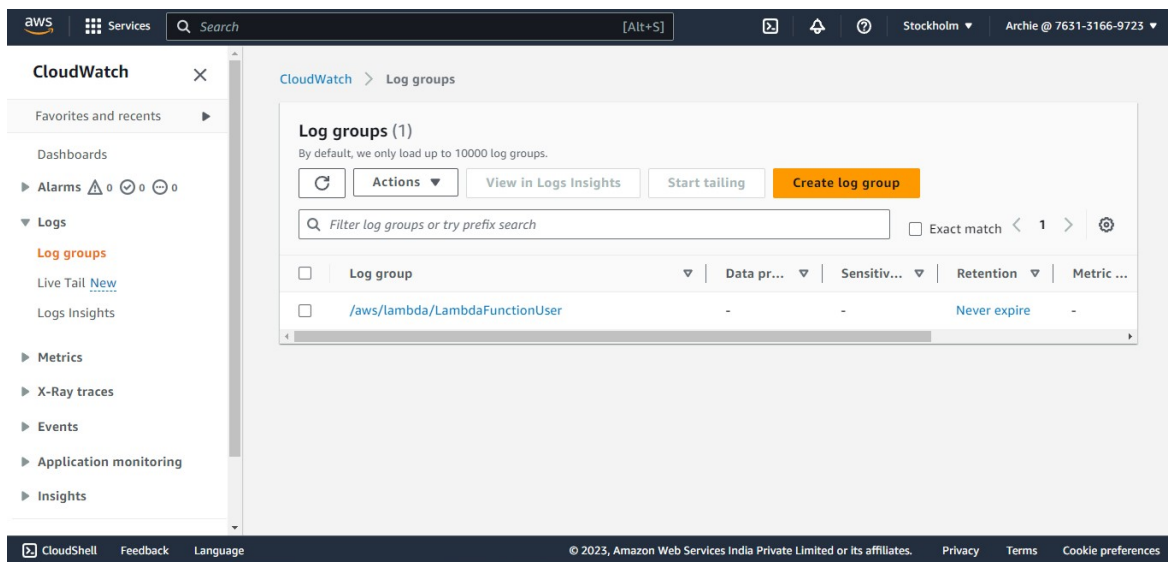
Destination

Destination
s3://labdabucket117

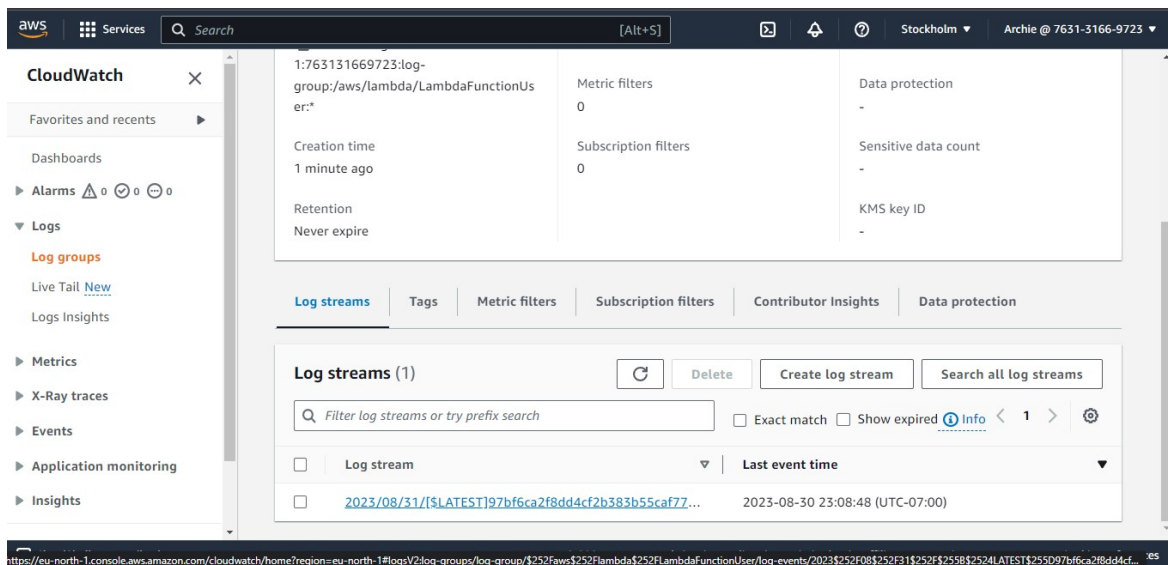
Destination details
Bucket settings that impact new objects stored in the specified destination.



12. Search CloudWatch and go to Log groups. Select the existing Log Group.



13. Click on the link provided and then you will see the message displayed.



The screenshot displays the AWS CloudWatch console interface. The left-hand navigation pane includes sections for 'Favorites and recents', 'Dashboards', 'Alarms', 'Logs' (with sub-links for 'Log groups', 'Live Tail', and 'Logs Insights'), 'Metrics', 'X-Ray traces', 'Events', 'Application monitoring', and 'Insights'. The main content area is titled 'Log events' and provides instructions on using the filter bar. It features buttons for 'Actions', 'Start tailing', and 'Create metric filter'. A search bar labeled 'Filter events' is present, along with filters for 'Clear', time intervals ('1m', '30m', '1h', '12h', 'Custom'), and a 'Display' dropdown. The log events are presented in a table with two columns: 'Timestamp' and 'Message'. The messages include 'INIT_START Runtime Version: python:3.11.v10', 'START RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Version: \$LATEST', 'Image successfully uploaded in s3 bucket', 'END RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9', and 'REPORT RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Duration: 1.51 ms Billed D...'. The interface also shows messages about no older events and auto retry status. The footer contains links for 'CloudShell', 'Feedback', 'Language', and copyright information for Amazon Web Services India Private Limited.

Timestamp	Message
2023-08-30T23:08:48.141-07:00	INIT_START Runtime Version: python:3.11.v10 Runtime Version ARN: arn:aws:lambda:e...
2023-08-30T23:08:48.249-07:00	START RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Version: \$LATEST
2023-08-30T23:08:48.249-07:00	Image successfully uploaded in s3 bucket
2023-08-30T23:08:48.250-07:00	END RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9
2023-08-30T23:08:48.250-07:00	REPORT RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Duration: 1.51 ms Billed D...

CONCLUSION: In this assignment, we learnt how to create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.