

**Project on AWS S3 Bucket**

**Quantiphi**

**A training report**

Submitted in partial fulfilment of the requirements for the award of degree of

**B.Tech. Computer Science Engineering**

**(Decision Science and Machine Learning)**

**Submitted to**

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**



**From 26/10/23 to 29/11/23**

**SUBMITTED BY**

**Name of student:** Mamidipally Rithvik Goud

**Registration Number:** 12200228

## **CONTENT**

- 1. Student Declaration**
- 2. Introduction to Company**
- 3. Objective of Project**
- 4. Creating AWS S3 Bucket**
- 5. Creating AWS Dynamo DB Table**
- 6. Creating AWS Lambda Function**
- 7. Lambda Code**
- 8. Creating IAM Role**
- 9. Cloud Logs**

## **DECLARATION**

We , Mamidipally Rithvik Goud(12200228), Piyush Kumar(12210053), Abhishek Roy(12221830) hereby declare that the work done by us on “AWS S3 cloud computing” from October,2023 to November, 2023, under the supervision of Quantiphi , Lovely professional University, Phagwara, Punjab, is a record of original work for the completion of project of CSQ-242 .

Name: - Mamidipally Rithvik Goud

# Introduction to Quantiphi Inc.

## Overview:

Quantiphi Inc. is a leading artificial intelligence and machine learning solutions provider with a global presence. Headquartered in Marlborough, Massachusetts, Quantiphi has established itself as a key player in delivering advanced analytics, automation, and data science solutions to enterprises across various industries.

## Founding and Growth:

Quantiphi was founded with a vision to harness the power of emerging technologies to solve complex business challenges. Since its inception, the company has experienced significant growth, expanding its offerings and client base. The commitment to innovation and a data-centric approach has positioned Quantiphi as a trusted partner for organizations seeking transformative solutions.

## Service Offerings:

Quantiphi specializes in providing end-to-end AI and ML solutions tailored to meet the unique needs of its clients. The company's service offerings include:

**Advanced Analytics:** Leveraging cutting-edge analytics to derive actionable insights from data.

**Automation:** Implementing intelligent automation solutions to enhance operational efficiency.

**Data Science:** Applying data science methodologies to drive informed decision-making.

**Computer Vision:** Developing solutions that enable machines to interpret and understand visual information.

**Natural Language Processing (NLP):** Creating systems that comprehend and respond to human language.

### **Clientele and Impact:**

Quantiphi has made a substantial impact on diverse industries, including healthcare, finance, retail, and technology. The company's client-centric approach and commitment to delivering tangible results have led to long-term partnerships with global enterprises.

### **Global Presence:**

With offices and delivery canterers strategically located around the world, including India and Canada, Quantiphi has established a global footprint. This presence enables the company to collaborate closely with clients and provide seamless implementation of AI and ML solutions.

### **Leadership:**

Quantiphi is led by a team of seasoned professionals who bring extensive expertise in artificial intelligence, machine learning, and business strategy. The leadership is dedicated to driving innovation and ensuring that Quantiphi remains at the forefront of technological advancements.

**Objective:** The goal of this project was to create a serverless application on AWS that could handle user CSV data, process it using Lambda functions, and store the results in DynamoDB. Additionally, a secondary Lambda function was implemented to delete corresponding records when a CSV file was deleted from the S3 bucket. CloudWatch was employed to monitor the performance of the Lambda functions.

## Step 1: AWS S3 and Dynamo DB Setup

### 1.1 Creating an S3 Bucket:

- Logged into the AWS Management Console and navigated to the S3 service.
- Created a bucket named "12200228rithvik" selected the region, and configured additional settings.

The screenshot shows the 'Create bucket' page in the AWS Management Console. The breadcrumb navigation is 'Amazon S3 > Buckets > Create bucket'. The page title is 'Create bucket' with an 'Info' link. Below the title, it says 'Buckets are containers for data stored in S3. [Learn more](#)'. The 'General configuration' section contains a 'Bucket name' field with the value '12200228rithvik', an 'AWS Region' dropdown set to 'Asia Pacific (Mumbai) ap-south-1', and a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. The 'Object Ownership' section has two radio buttons: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The bottom of the console shows 'CloudShell' and 'Feedback' links.

The screenshot shows the 'Buckets' list page in the Amazon S3 console. It includes an 'Account snapshot' section with a 'View Storage Lens dashboard' button. The 'Buckets (1)' section has a search bar and a table with one bucket. The table has columns for Name, AWS Region, Access, and Creation date. The bucket '12200228rithvik' is listed in the 'Asia Pacific (Mumbai) ap-south-1' region with 'Bucket and objects not public' access and a creation date of 'November 28, 2023, 21:29:35 (UTC+05:30)'. Action buttons like 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' are visible.

Name	AWS Region	Access	Creation date
12200228rithvik	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	November 28, 2023, 21:29:35 (UTC+05:30)

## 1.2 Creating a DynamoDB Table:

- Accessed the DynamoDB console.
- Created a table named "RithvikTable" defined a primary key, and configured additional settings.

[DynamoDB](#) > [Tables](#) > [Create table](#)

## Create table

### Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.

RithvikTable

Between 3 and 255 characters, containing only letters, numbers, underscores (`_`), hyphens (`-`), and periods (`.`).

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

id

String

1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name

String

1 to 255 characters and case sensitive.

The RithvikTable table was created successfully.

[DynamoDB](#) > [Tables](#)

**Tables (1) [Info](#)**

Find tables by table name

Any tag key

Any tag value

< 1 >

[RithvikTable](#)

Active

id (S)

-

0

Off

Provisioned (5)

Provisioned (5)

0

	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mo...	T...
	<a href="#">RithvikTable</a>	Active	id (S)	-	0	Off	Provisioned (5)	Provisioned (5)	0

## 2.1 Creating a Lambda Function:

- ## Create function

[Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **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

### Basic information

**Function name**

Enter a name that describes the purpose of your function.

RithvikLambda

Use only letters, numbers, hyphens, or underscores with no spaces.

**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.

- Added an S3 trigger to the Lambda function.
- Selected the S3 bucket created in Step 1.
- Defined the event type as "ObjectCreated."

RithvikLambda

Throttle

Copy ARN

Action

The trigger 12200228rithvik was successfully added to function RithvikLambda. The function is now receiving events from the trigger.

▼ Function overview

Info

Export to Application Composer

Download function

Diagram

Template

RithvikLambda

Layers (0)

S3

+ Add trigger

+ Add destination

Description

-

Last modified

6 minutes ago

Function ARN

arn:aws:lambda:ap-south-1:457028026855:function:RithvikLambda

Function URL

Info

-



# 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.



s3/12200228rithvik



Bucket region: ap-south-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.

.csv

### 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

## Edit basic settings

### Basic settings [Info](#)

#### Description - optional

#### Memory [Info](#)

Your function is allocated CPU proportional to the memory configured.

128

MB

Set memory to between 128 MB and 10240 MB

#### Ephemeral storage [Info](#)

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512

MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

#### SnapStart [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

None

Supported runtimes: Java 11, Java 17, Java 21.

#### Timeout

1

min

3

sec

#### Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☒ Use an existing role

Timeout

min  sec

Execution role


Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).


☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.





[View the dev role](#) on the IAM console.

Cancel **Save**

## 2.3 Writing Lambda Function Code:

- Wrote Python code to process CSV data and store the top 10 rows in DynamoDB.
- Utilized the boto3 library for AWS SDK to interact with S3 and DynamoDB.

### Code:

```
import boto3
```

```
import csv
```

```
from io import StringIO
```

```
def lambda_handler(event, context):
```

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

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

```
    bucket_name = '12200228rithvik'
```

```
    table_name = 'RithvikTable'
```

```
    response = s3.list_objects_v2(Bucket=bucket_name)
```

```
    objects = response.get('Contents', [])
```

```
    csv_objects = [obj for obj in objects if obj['Key'].lower().endswith('.csv')]
```

```
    top_10_csv_objects = sorted(csv_objects, key=lambda x: x['Size'], reverse=True)[:10]
```

```
    for csv_obj in top_10_csv_objects:
```

```
        csv_content = s3.get_object(Bucket=bucket_name,
Key=csv_obj['Key'])['Body'].read().decode('utf-8')
```

```
        csv_reader = csv.DictReader(StringIO(csv_content))
```

```
        row_count = 0
```

```
for row in csv_reader:

    if row_count >= 10:

        break

    dynamodb.put_item(

        TableName=table_name,

        Item={

            'object_key': {'S': csv_obj['Key']},

            'id': {'S': row['id']},

            'name': {'S': row['name']},

            'age': {'S': row['age']},

            'email': {'S': row['email']}

        }

    )

    row_count += 1

return {

    'statusCode': 200,

    'body': 'Top 10 CSV objects processed and stored in DynamoDB.'

}
```

Go to Anything (Ctrl-P)

RithvikLambda - /  
lambda\_function.py

lambda\_function x Environment Var x +

```
1 import boto3
2 import csv
3 from io import StringIO
4
5 def lambda_handler(event, context):
6     # Initialize S3 and DynamoDB clients
7     s3 = boto3.client('s3')
8     dynamodb = boto3.client('dynamodb')
9
10    # S3 bucket and DynamoDB table names
11    bucket_name = '12200228rithvik'
12    table_name = 'RithvikTable'
13
14    # Get the list of objects in the S3 bucket
15    response = s3.list_objects_v2(Bucket=bucket_name)
16    objects = response.get('Contents', [])
17
18    # Filter only CSV files
19    csv_objects = [obj for obj in objects if obj['Key'].lower().endswith('.csv')]
20
21    # Sort CSV objects by size and get the top 10
22    top_10_csv_objects = sorted(csv_objects, key=lambda x: x['Size'], reverse=True)[:10]
23
24    # Store top 10 CSV objects in DynamoDB
25    for csv_obj in top_10_csv_objects:
26        # Read CSV content from S3
27        csv_content = s3.get_object(Bucket=bucket_name, Key=csv_obj['Key'])['Body'].read().decode('utf-8')
28
29        # Parse CSV content
30        csv_reader = csv.DictReader(StringIO(csv_content))
31
32        # Counter for limiting to 10 rows
33        row_count = 0
34
35        for row in csv_reader:
36            # Break after processing 10 rows
37            if row_count >= 10:
```

lambda\_function x Environment Var x +

```
22 top_10_csv_objects = sorted(csv_objects, key=lambda x: x['Size'], reverse=True)[:10]
23
24 # Store top 10 CSV objects in DynamoDB
25 for csv_obj in top_10_csv_objects:
26     # Read CSV content from S3
27     csv_content = s3.get_object(Bucket=bucket_name, Key=csv_obj['Key'])['Body'].read().decode('utf-8')
28
29     # Parse CSV content
30     csv_reader = csv.DictReader(StringIO(csv_content))
31
32     # Counter for limiting to 10 rows
33     row_count = 0
34
35     for row in csv_reader:
36         # Break after processing 10 rows
37         if row_count >= 10:
38             break
39
40         dynamodb.put_item(
41             TableName=table_name,
42             Item={
43                 'object_key': {'S': csv_obj['Key']],
44                 'id': {'S': row['id']],
45                 'name': {'S': row['name']],
46                 'age': {'S': row['age']],
47                 'email': {'S': row['email']}}
48         )
49
50         row_count += 1
51
52     return {
53         'statusCode': 200,
54         'body': 'Top 10 CSV objects processed and stored in DynamoDB.'
55     }
```

## 2.4 Creating IAM Role:

- Created an IAM role named DataProcessingRole to grant necessary permissions to the Lambda function.
- Attached policies: AmazonDynamoDBFullAccess, AmazonS3FullAccess, AWSLambda\_FullAccess, AWSLambdaBasicExecutionRole, AWSLambdaDynamoDBExecutionRole.

dev Info Delete

Summary Edit

Creation date  
November 17, 2023, 21:50 (UTC+05:30)

Last activity  
✓ 6 hours ago

ARN  
 arn:aws:iam::457028026855:role/service-role/dev

Maximum session duration  
1 hour

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (6) Info

Simulate  Remove Add permissions ▼

You can attach up to 10 managed policies.

Filter by Type  
All types ▼

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	<a href="#">AmazonDynamoDBFullAccess</a>	AWS managed	<u>2</u>
<input type="checkbox"/>	<a href="#">AmazonS3FullAccess</a>	AWS managed	<u>2</u>
<input type="checkbox"/>	<a href="#">AWSLambda_FullAccess</a>	AWS managed	<u>1</u>
<input type="checkbox"/>	<a href="#">AWSLambdaBasicExecutionRole</a>	AWS managed	<u>2</u>
<input type="checkbox"/>	<a href="#">AWSLambdaDynamoDBExecutionRole</a>	AWS managed	<u>1</u>

## Step 3: Uploading CSV file, DynamoDB Table Output, Cloud log

### Uploading CSV File:

1	id	name	age	email		
2	1	Aarav Shar	25	aarav.sharma@email.in		
3	2	Aisha Pate	30	aisha.patel@email.in		
4	3	Vikram Sin	22	vikram.singh@email.in		
5	4	Ananya Ve	28	ananya.verma@email.in		
6	5	Rahul Kapo	35	rahul.kapoor@email.in		
7	6	Neha Gupt	27	neha.gupta@email.in		
8	7	Arjun Shar	32	arjun.sharma@email.in		
9	8	Pooja Yada	29	pooja.yadav@email.in		
10	9	Rohan Kha	26	rohan.khanna@email.in		
11	10	Sneha Sing	31	sneha.singh@email.in		
12	11	Aditya Kun	23	aditya.kumar@email.in		
13	12	Shreya Pat	33	shreya.patel@email.in		
14	13	Vivek Vern	24	vivek.verma@email.in		
15	14	Naina Kapo	28	naina.kapoor@email.in		
16	15	Rajat Yada	30	rajat.yadav@email.in		
17	16	Meera Sha	27	meera.sharma@email.in		
18	17	Amit Kuma	34	amit.kumar@email.in		
19	18	Anushka Si	25	anushka.singh@email.in		
20	19	Rohit Vern	29	rohit.verma@email.in		
21	20	Preeti Kha	26	preeti.khanna@email.in		

# Upload Info

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, 834.0 B)

Remove

Add files

Add folder

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	aws_pro_1.csv	-	text/csv	834.0 B

Upload succeeded  
View details below.

## Upload: status

Close

The information below will no longer be available after you navigate away from this page.

### Summary

Destination  
[s3://12200228rithvik](#)

Succeeded  
 1 file, 834.0 B (100.00%)

Failed  
 0 files, 0 B (0%)

Files and folders Configuration

### Files and folders

Find by name

< 1 >

Name	Folder	Type	Size	Status	Error
<a href="#">aws_pro_1.csv</a>	-	text/csv	834.0 B	Succeeded	-

## DynamoDB Table Output:

RithvikTable

Items returned (10)

↺

1

↻

⚙

✖

<input type="checkbox"/>	id (String) ▲	age ▼	email ▼	name ▼	object_key ▼
<input type="checkbox"/>	<a href="#">1</a>	25	aarav.shar...	Aarav Sharma	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">10</a>	31	sneha.singh...	Sneha Singh	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">2</a>	30	aisha.patel...	Aisha Patel	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">3</a>	22	vikram.sing...	Vikram Singh	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">4</a>	28	ananya.ver...	Ananya Ver...	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">5</a>	35	rahul.kapoo...	Rahul Kapoor	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">6</a>	27	neha.gupta...	Neha Gupta	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">7</a>	32	arjun.sharm...	Arjun Sharma	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">8</a>	29	pooja.yadav...	Pooja Yadav	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">9</a>	26	rohan.khan...	Rohan Kha...	aws_pro_1.csv

## Cloud log:

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](#)

↺

Actions ▼

Start tailing

Create metric filter

🔍 Filter events

Clear

1m

30m

1h

12h

Custom

Local timezone ▼

Display ▼

⚙

▶	Timestamp	Message
No older events at this moment. <a href="#">Retry</a>		
▶	2023-11-28T21:50:06.789+05:30	INIT_START Runtime Version: python:3.11.v18 Runtime Version ARN: arn:aws:lambda:ap-south-1::runtime:6ebff6b58cf714d30879a40cc3...
▶	2023-11-28T21:50:07.042+05:30	START RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7 Version: \$LATEST
▶	2023-11-28T21:50:09.996+05:30	END RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7
▶	2023-11-28T21:50:09.996+05:30	REPORT RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7 Duration: 2954.11 ms Billed Duration: 2955 ms Memory Size: 128 MB Max M...
No newer events at this moment. <a href="#">Auto retry paused</a> . <a href="#">Resume</a>		

## Step 4: CloudWatch Configuration

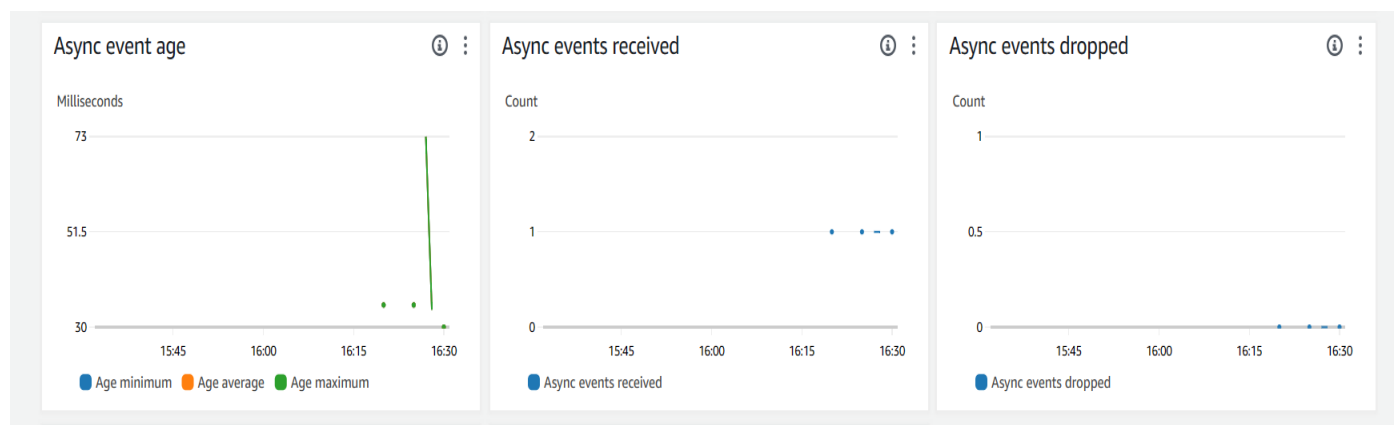
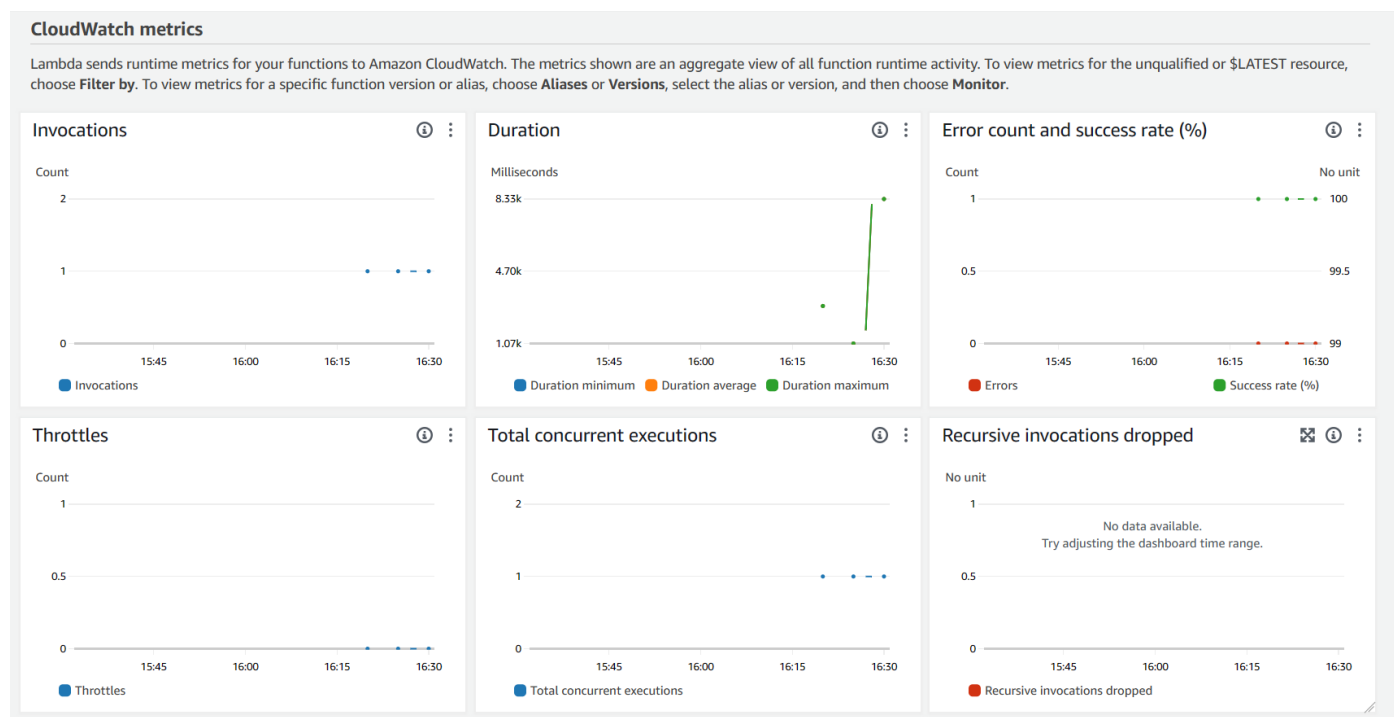
### 4.1 Enabling CloudWatch Logs:

- In the Lambda function settings, configured CloudWatch Logs to capture logs generated by the function.

### 4.2 Monitoring Lambda Performance:

- Accessed the CloudWatch console.
- Navigated to "Logs" and selected the log group associated with the Lambda function.
- Analysed logs to monitor the performance of the Lambda function during executions

## Cloud Watch Metrics:





Cloud log for 5 Csv Files:

▶ Favorites and recents

Dashboards

▶ Alarms

1

7

0

▶ Logs

Log groups 

New

Log Anomalies 

New

Live Tail

Logs Insights

▶ Metrics 

New

All metrics

Explorer

Streams

▶ X-Ray traces

▶ Events

▶ Application monitoring

▶ Insights

Settings

Getting Started

What's new

▶

Timestamp

Message

No more records within selected time range 

Retry

▶

2023-11-28T21:50:06.789+05:30

INIT\_START Runtime Version: python:3.11.v18 Runtime Version ARN: arn:aws:lambda:ap-south-1::runtime:6ebff6b58cf714d30879a40...

▶

2023-11-28T21:50:07.042+05:30

START RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7 Version: \$LATEST

▶

2023-11-28T21:50:09.996+05:30

END RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7

▶

2023-11-28T21:50:09.996+05:30

REPORT RequestId: e48c6e64-0479-45b0-8b14-f63ab9c686e7 Duration: 2954.11 ms Billed Duration: 2955 ms Memory Size: 128 MB Ma...

▶

2023-11-28T21:55:07.326+05:30

START RequestId: 5995c521-750a-4ab2-ae09-cb09a5949a97 Version: \$LATEST

▶

2023-11-28T21:55:08.396+05:30

END RequestId: 5995c521-750a-4ab2-ae09-cb09a5949a97

▶

2023-11-28T21:55:08.396+05:30

REPORT RequestId: 5995c521-750a-4ab2-ae09-cb09a5949a97 Duration: 1070.07 ms Billed Duration: 1071 ms Memory Size: 128 MB Ma...

▶

2023-11-28T21:57:56.887+05:30

START RequestId: 6450204d-5eb2-4380-bc56-fcb93cf7de86 Version: \$LATEST

▶

2023-11-28T21:57:58.596+05:30

END RequestId: 6450204d-5eb2-4380-bc56-fcb93cf7de86

▶

2023-11-28T21:57:58.596+05:30

REPORT RequestId: 6450204d-5eb2-4380-bc56-fcb93cf7de86 Duration: 1709.52 ms Billed Duration: 1710 ms Memory Size: 128 MB Ma...

▶

2023-11-28T21:58:09.642+05:30

START RequestId: 84f4be0b1-ed20-4989-bca8-e7cf5591c74c Version: \$LATEST

▶

2023-11-28T21:58:17.716+05:30

END RequestId: 84f4be0b1-ed20-4989-bca8-e7cf5591c74c

▶

2023-11-28T21:58:17.716+05:30

REPORT RequestId: 84f4be0b1-ed20-4989-bca8-e7cf5591c74c Duration: 8074.40 ms Billed Duration: 8075 ms Memory Size: 128 MB Ma...

▶

2023-11-28T22:00:14.087+05:30

START RequestId: 4294035b-6c5f-49a0-83fb-684315e5ab25 Version: \$LATEST

▶

2023-11-28T22:00:22.416+05:30

END RequestId: 4294035b-6c5f-49a0-83fb-684315e5ab25

▶

2023-11-28T22:00:22.416+05:30

REPORT RequestId: 4294035b-6c5f-49a0-83fb-684315e5ab25 Duration: 8328.79 ms Billed Duration: 8329 ms Memory Size: 128 MB Ma...

## Step5: Uploading 5 CSV Files,Display dynamodb table

### Uploading 5 CSV Files:

12200228rithvik

Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (5)

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](#)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	aws_pro_1.csv	csv	November 28, 2023, 21:50:06 (UTC+05:30)	834.0 B	Standard
<input type="checkbox"/>	aws_pro_2.csv	csv	November 28, 2023, 21:55:06 (UTC+05:30)	861.0 B	Standard
<input type="checkbox"/>	aws_pro_3.csv	csv	November 28, 2023, 21:57:56 (UTC+05:30)	845.0 B	Standard
<input type="checkbox"/>	aws_pro_4.csv	csv	November 28, 2023, 21:58:09 (UTC+05:30)	843.0 B	Standard
<input type="checkbox"/>	aws_pro_5.csv	csv	November 28, 2023, 22:00:14 (UTC+05:30)	848.0 B	Standard

### Display dynamodb table:

<input type="checkbox"/>	id (String)	age	email	name	object_key
<input type="checkbox"/>	<a href="#">1</a>	25	aarav.shar...	Aarav Sharma	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">10</a>	31	sneha.singh...	Sneha Singh	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">2</a>	30	aisha.patel...	Aisha Patel	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">21</a>	28	varun.shar...	Varun Sharma	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">22</a>	31	divya.kapo...	Divya Kapoor	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">23</a>	29	raj.sharma...	Raj Sharma	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">24</a>	26	sonia.verma...	Sonia Verma	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">25</a>	33	kunal.singh...	Kunal Singh	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">26</a>	27	ayesha.pate...	Ayesha Patel	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">27</a>	30	krishna.ku...	Krishna Ku...	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">28</a>	32	anita.yadav...	Anita Yadav	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">29</a>	25	aditi.khann...	Aditi Khanna	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">3</a>	22	vikram.sing...	Vikram Singh	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">30</a>	34	rahul.verma...	Rahul Verma	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">4</a>	28	ananya.ver...	Ananya Ver...	aws_pro_1.csv

<input type="checkbox"/>	id (String) ▲	age ▼	email ▼	name ▼	object_key ▼
<input type="checkbox"/>	<a href="#">50</a>	28	deepak.yad...	Deepak Yadav	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">6</a>	27	neha.gupta...	Neha Gupta	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">61</a>	31	priya.khann...	Priya Khanna	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">62</a>	27	kunal.verm...	Kunal Verma	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">63</a>	29	suman.pate...	Suman Patel	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">64</a>	34	rajat.kumar...	Rajat Kumar	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">65</a>	28	sneha.singh...	Sneha Singh	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">66</a>	26	ajay.khanna...	Ajay Khanna	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">67</a>	30	anita.yadav...	Anita Yadav	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">68</a>	32	vikrant.shar...	Vikrant Sha...	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">69</a>	25	deepika.ka...	Deepika Ka...	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">7</a>	32	arjun.sharm...	Arjun Sharma	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">70</a>	33	arjun.singh...	Arjun Singh	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">8</a>	29	pooja.yadav...	Pooja Yadav	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">81</a>	28	kavita.patel...	Kavita Patel	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">82</a>	26	kishan.khan...	Kishan Kha...	aws_pro_5.csv

<input type="checkbox"/>	id (String) ▲	age ▼	email ▼	name ▼	object_key ▼
<input type="checkbox"/>	<a href="#">30</a>	34	rahul.verma...	Rahul Verma	aws_pro_2.csv
<input type="checkbox"/>	<a href="#">4</a>	28	ananya.ver...	Ananya Ver...	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">41</a>	28	nisha.yadav...	Nisha Yadav	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">42</a>	26	karan.shar...	Karan Sharma	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">43</a>	30	swati.patel...	Swati Patel	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">44</a>	33	sumit.kuma...	Sumit Kumar	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">45</a>	27	kirti.verma...	Kirti Verma	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">46</a>	29	rajeev.singh...	Rajeev Singh	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">47</a>	31	sapna.khan...	Sapna Khan...	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">48</a>	25	vivek.patel...	Vivek Patel	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">49</a>	34	anjali.shar...	Anjali Sharma	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">5</a>	35	rahul.kapoo...	Rahul Kapoor	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">50</a>	28	deepak.yad...	Deepak Yadav	aws_pro_3.csv
<input type="checkbox"/>	<a href="#">6</a>	27	neha.gupta...	Neha Gupta	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">61</a>	31	priya.khann...	Priya Khanna	aws_pro_4.csv

Items returned (50)

Actions

Create item

<

1

...

>

<input type="checkbox"/>	id (String)	age	email	name	object_key
<input type="checkbox"/>	<a href="#">69</a>	25	deepika.ka...	Deepika Ka...	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">7</a>	32	arjun.sharm...	Arjun Sharma	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">70</a>	33	arjun.singh...	Arjun Singh	aws_pro_4.csv
<input type="checkbox"/>	<a href="#">8</a>	29	pooja.yadav...	Pooja Yadav	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">81</a>	28	kavita.patel...	Kavita Patel	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">82</a>	26	kishan.khan...	Kishan Kha...	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">83</a>	31	anju.sharm...	Anju Sharma	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">84</a>	29	vikas.yadav...	Vikas Yadav	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">85</a>	32	rakhi.kapoo...	Rakhi Kapoor	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">86</a>	25	sachin.sing...	Sachin Singh	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">87</a>	33	asha.patel...	Asha Patel	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">88</a>	27	rajesh.verm...	Rajesh Verma	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">89</a>	30	shikha.khan...	Shikha Kha...	aws_pro_5.csv
<input type="checkbox"/>	<a href="#">9</a>	26	rohan.khan...	Rohan Kha...	aws_pro_1.csv
<input type="checkbox"/>	<a href="#">90</a>	34	amit.kapoo...	Amit Kapoor	aws_pro_5.csv

## Bonus Step: Delete Record Lambda Function

**Objective:** The goal is to automatically delete corresponding records in DynamoDB when a CSV file is deleted from the S3 bucket.

### 4.1 Creating a Second Lambda Function:

- Named the function "RithvikDeleteLambda" selected Python as the runtime, and created a role with DynamoDB permissions.

Lambda > Functions > RithvikDeleteLambda

RithvikDeleteLambda

ThrottleCopy ARNActions

Function overviewInfo

Export to Application ComposerDownload function

DiagramTemplate

RithvikDeleteLambda

Layers(0)

+ Add trigger

+ Add destination

Description

-

Last modified

1 minute ago

Function ARN

arn:aws:lambda:ap-south-1:457028026855:function:RithvikDeleteLambda

Function URLInfo

-

## Code:

```
lambda_function × (+)
1 import boto3
2
3 dynamodb = boto3.resource('dynamodb')
4 table_name = 'RithvikGoud'
5 primary_key='id'
6 ddb_table = dynamodb.Table(table_name)
7
8 def lambda_handler(event, context):
9     # Scan DynamoDB table to get all items
10    for record in event['Records']:
11        # Assuming the event is an S3 event triggered by the deletion of a file
12        s3_bucket = record['s3']['bucket']['name']
13        s3_key = record['s3']['object']['key']
14        response = ddb_table.scan()
15        print(response)
16
17    # Delete each item
18    for item in response['Items']:
19        ddb_table.delete_item(
20            Key={
21                primary_key: item[primary_key]
22                # Assuming your DynamoDB table has a primary key named 'primaryKey'
23                # Adjust this based on your actual table schema
24            }
25        )
26
27    return {
28        'statusCode': 200,
29        'body': 'All items deleted successfully'}
```

**import boto3**

**dynamodb = boto3.resource('dynamodb')**

**table\_name = 'RithvikTable'**

**primary\_key='id'**

**ddb\_table = dynamodb.Table(table\_name)**

**def lambda\_handler(event, context):**

**# Scan DynamoDB table to get all items**

**for record in event['Records']:**

**# Assuming the event is an S3 event triggered by the deletion of a file**

**s3\_bucket = record['s3']['bucket']['name']**

**s3\_key = record['s3']['object']['key']**

**response = ddb\_table.scan()**

**print(response)**

**# Delete each item**

```
for item in response['Items']:
```

## `ddb_table.delete_item()`

**Key={**

```
primary_key: item[primary_key]
```

## # Assuming your DynamoDB table has a primary key named 'primaryKey'

### # Adjust this based on your actual table schema

}

)

```
return {
```

```
'statusCode': 200,
```

```
'body': 'All items deleted successfully'}
```

## Adding Trigger:

+ Add trigger

S3

+ Add destination

Code
Test
Monitor
**Configuration**
Aliases
Versions

General configuration

**Triggers**

Permissions

Destinations

Function URL

Environment variables

Tags


VPC

Monitoring and operations tools

Triggers (1) Info

Find triggers

Trigger


**S3:** [12200228rithvik](#)  
arn:aws:s3:::12200228rithvik

▼ Details

☐

Bucket arn: **arn:aws:s3:::12200228rithvik**

Event types: **s3:ObjectRemoved:\***

Notification name: **74a503c0-ca81-4780-b764-2841bf38b6d2**

Service principal: **s3.amazonaws.com**

Source account: **457028026855**

Statement ID: **lambda-9b3e5626-5a8c-42dd-9120-3f04ca81dd25**

# Delete Objects from S3:

Successfully deleted objects  
View details below.

Delete objects: status

Close

The information below will no longer be available after you navigate away from this page.

Summary

Source

s3://12200228rithvik

Successfully deleted

1 object, 834.0 B

Failed to delete

0 objects

Failed to delete

Configuration

Failed to delete (0)

Find objects by name

< 1 >

Name

Folder

Type

Last modified

Size

Error

No objects failed to delete.

# DynamoDB Table View:

DynamoDB > Explore items > RithvikTable

Tables (1)

Any tag key

Any tag value

Find tables by table name

< 1 >

RithvikTable

RithvikTable

Autopreview

View table details

Scan or query items

Expand to query or scan items.

Completed. Read capacity units consumed: 0.5

Items returned (0)

Actions

Create item

< 1 >

The query did not return any results.