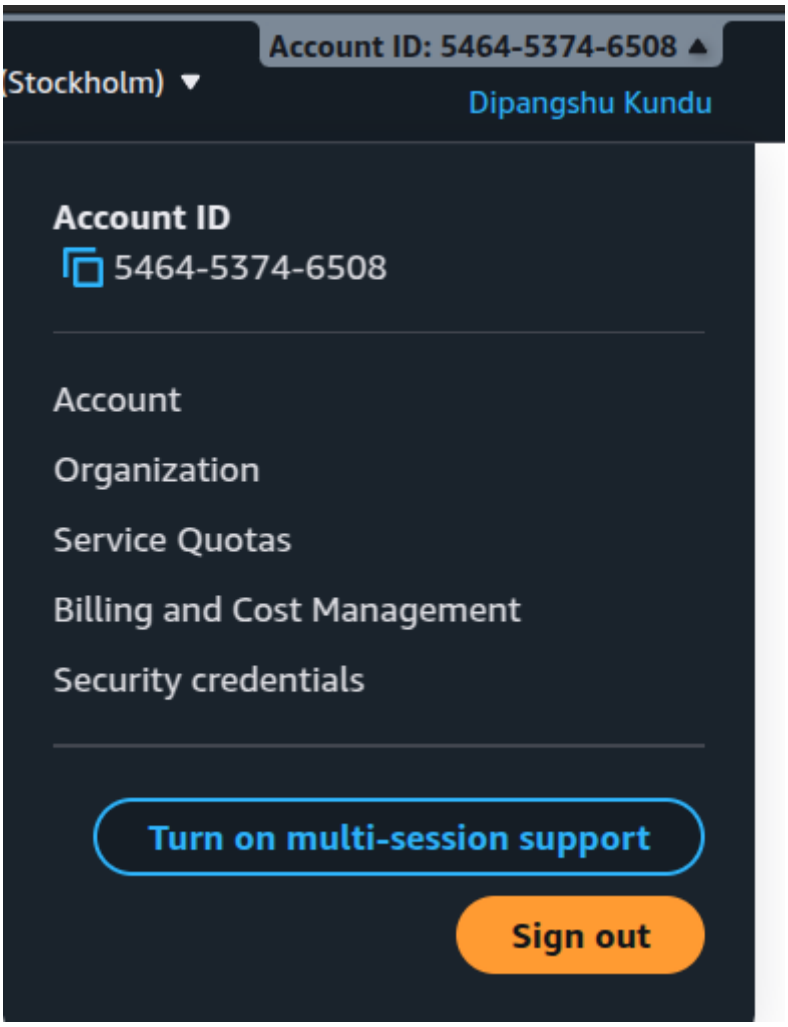
Name- Dipangshu Kundu
Reg No- 22BBS0148
AWS MID TERM
DATE:- 10/9/2025

Q) Create a serverless file content-type detector using S3 and AWS Lambda. Create an S3 bucket and configure S3 Event Notifications to trigger a Lambda function upon file upload. Write a Lambda function in Python or Node.js to retrieve the uploaded file's metadata and log the content type to CloudWatch. Test by uploading at least three different file types and verify that the logs display the detected content types. Document all configurations, code snippets, and test results with screenshots.

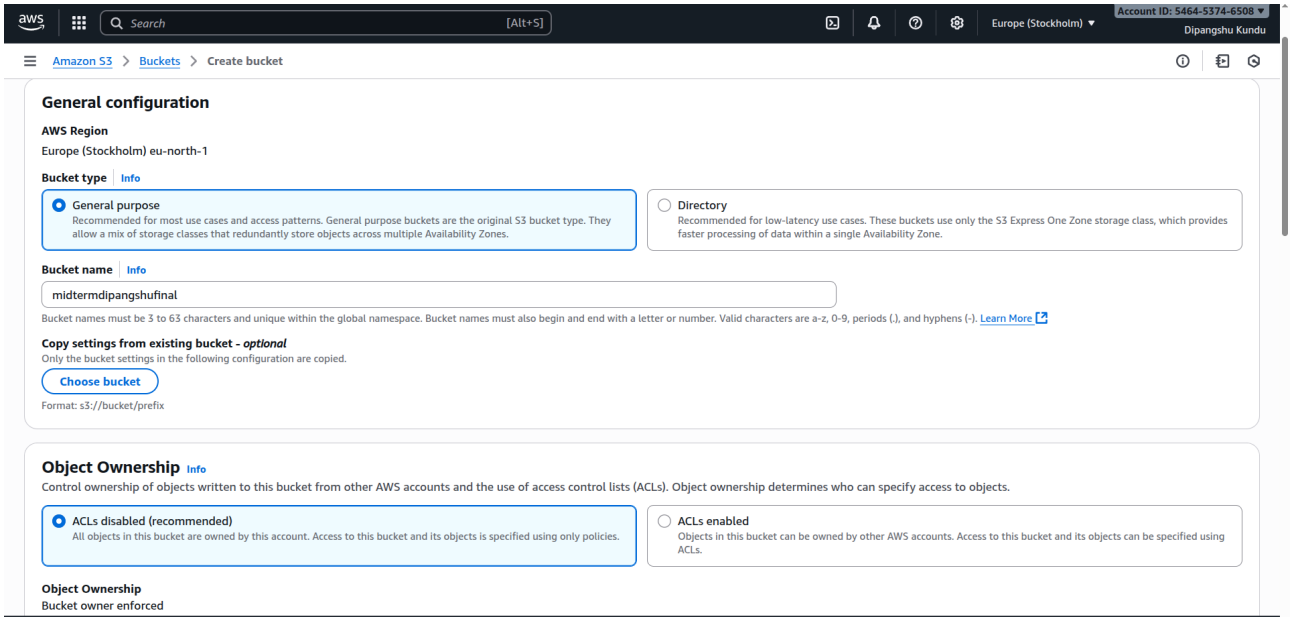
Solution:-

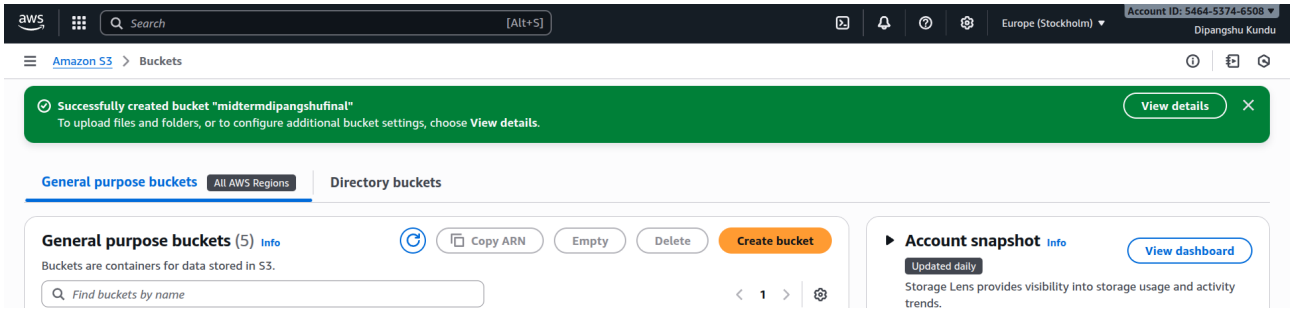


---

1. Create an S3 Bucket

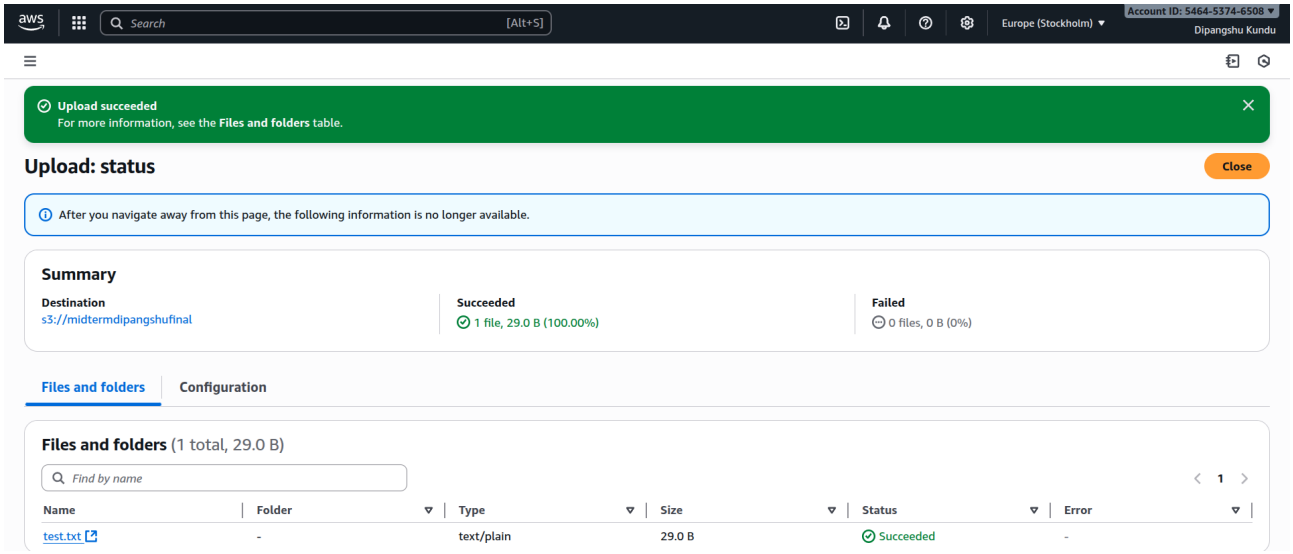
1. Go to **AWS Console → S3 → Create Bucket**.
2. Give it a unique name (e.g., my-lambda-demo-bucket).
3. Choose a region (keep it the same as Lambda).
4. Leave defaults, but **uncheck "Block all public access"** if you need public access.
5. **Create the bucket.**





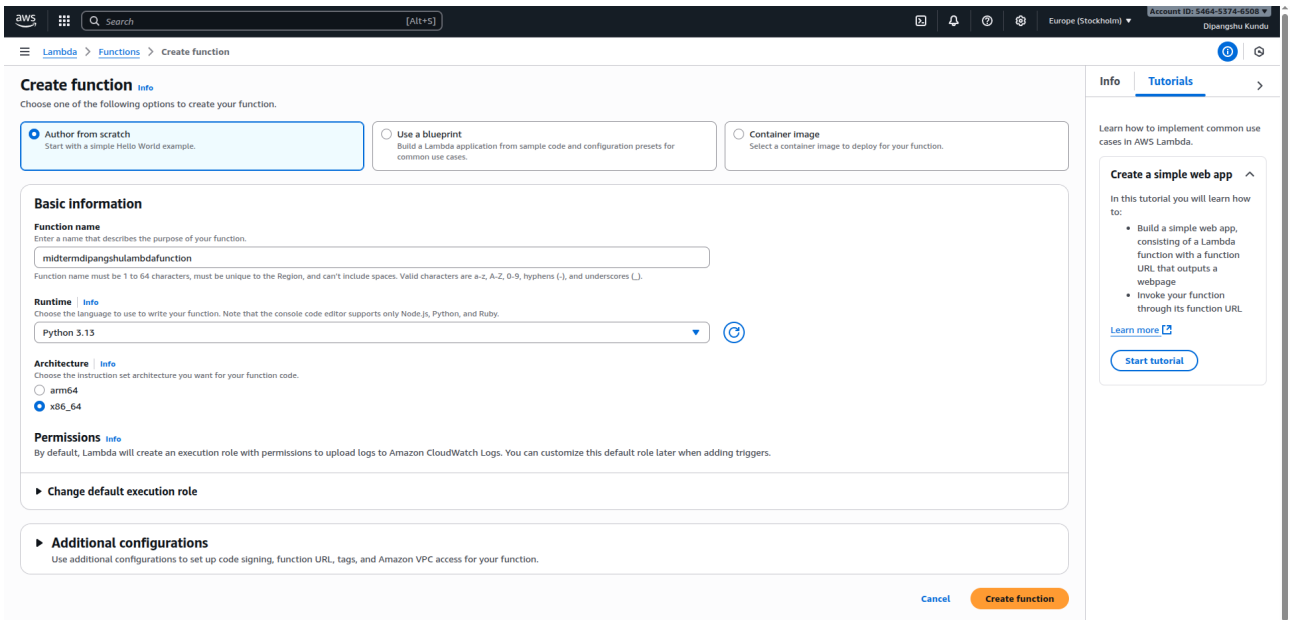
2. Upload a Test File to S3

- After bucket creation → **Upload** → choose a test file (e.g., test.txt).
- **We'll use this file to trigger Lambda.**



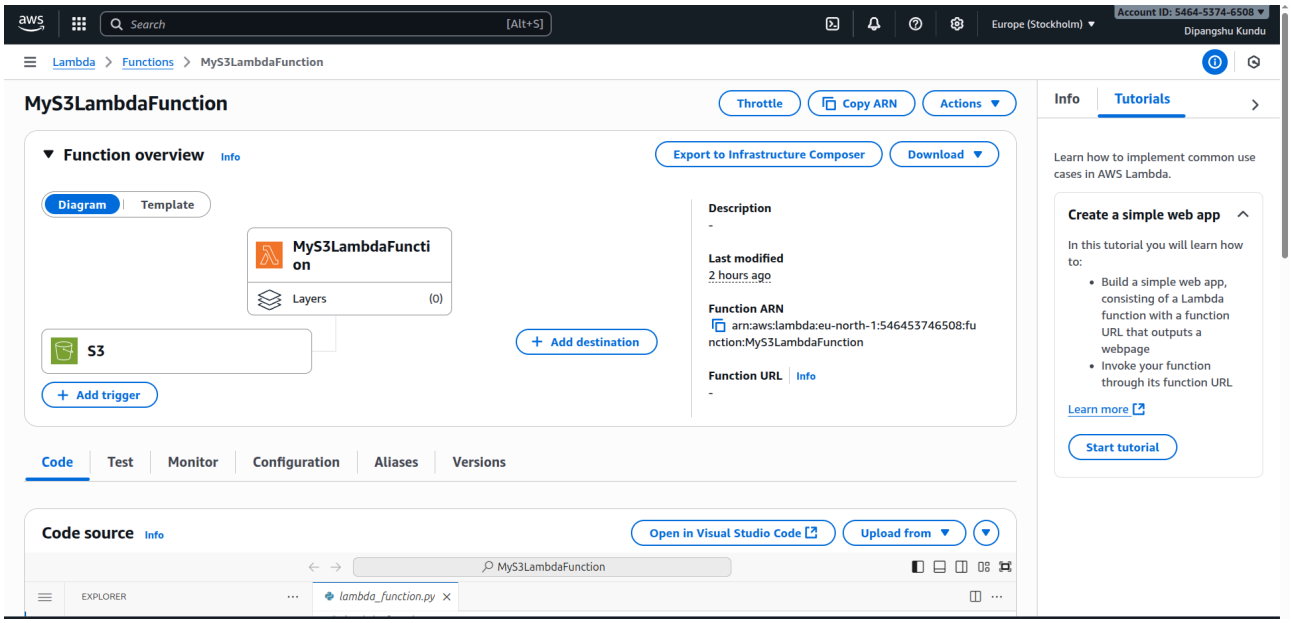
3. Create a Lambda Function

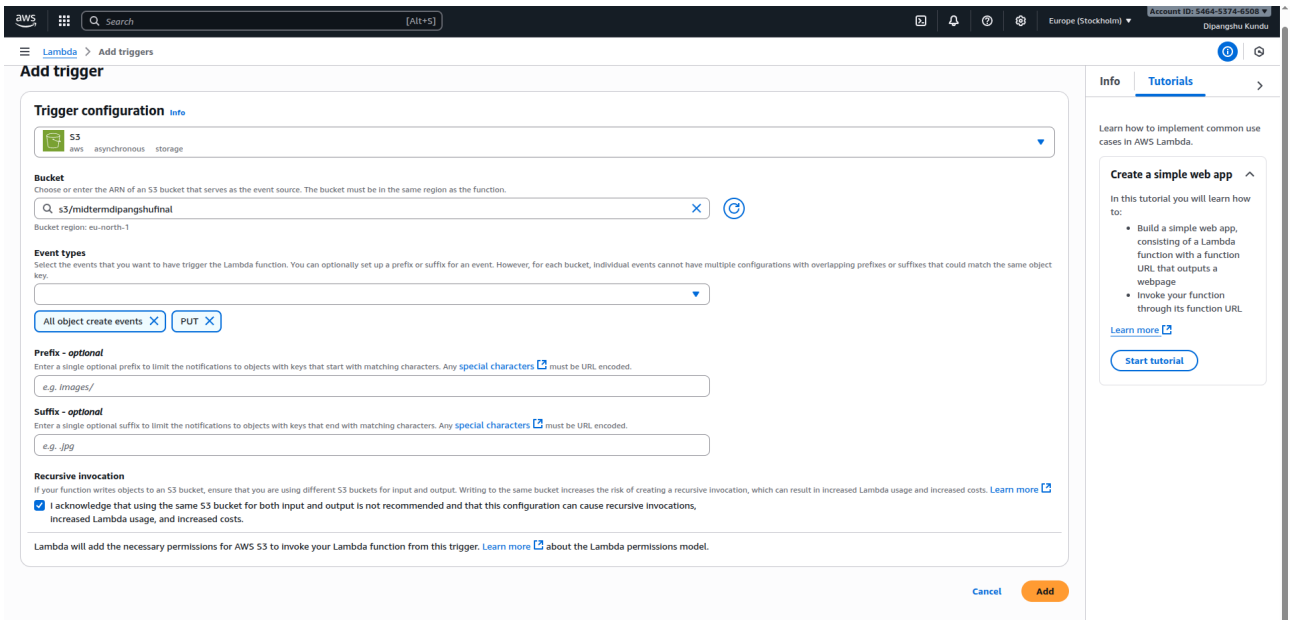
1. Go to **AWS Console → Lambda → Create Function**.
2. Choose **Author from scratch**.
3. Name: MyS3LambdaFunction.
4. Runtime: **Python 3.x** (or Node.js if you prefer).
5. Permissions: Create or use an existing role with **S3 and CloudWatch access**.
6. **Click Create Function.**



4. Add S3 Trigger to Lambda

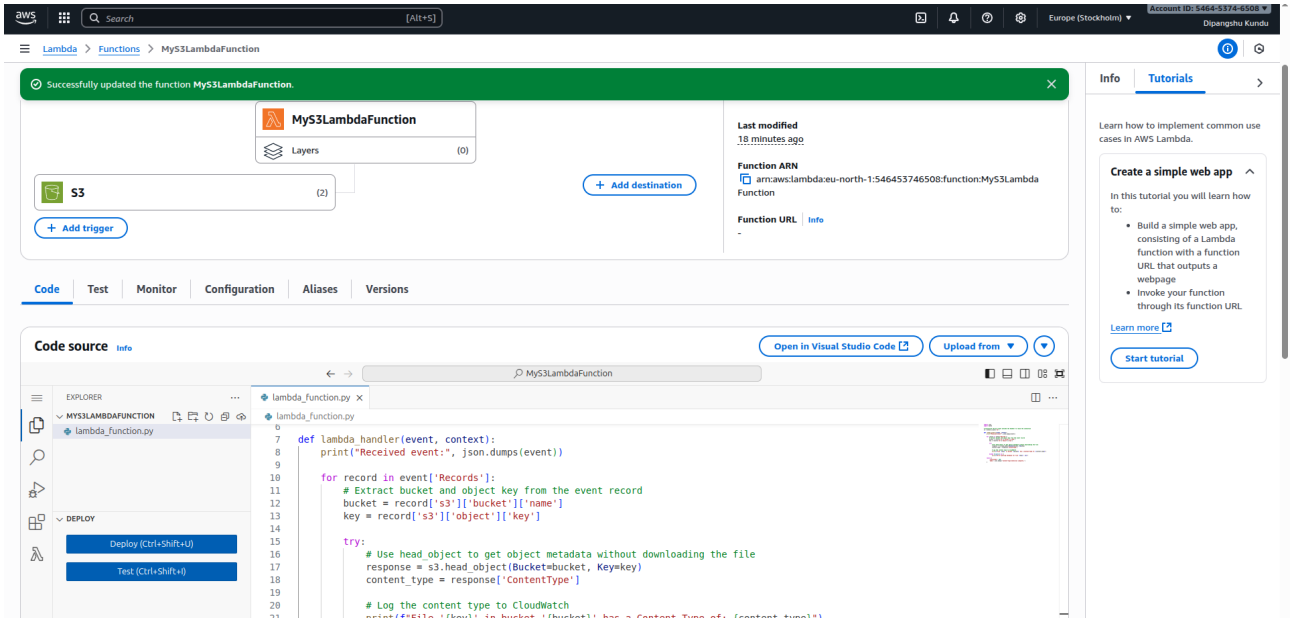
1. In your Lambda function → **Configuration** → **Triggers** → **Add Trigger**.
2. Select **S3**.
3. Choose your bucket (my-lambda-demo-bucket).
4. Event type: **PUT (All object create events)**.
5. Check "Enable trigger".
6. **Save.**





---

5. Write lambda Code



-

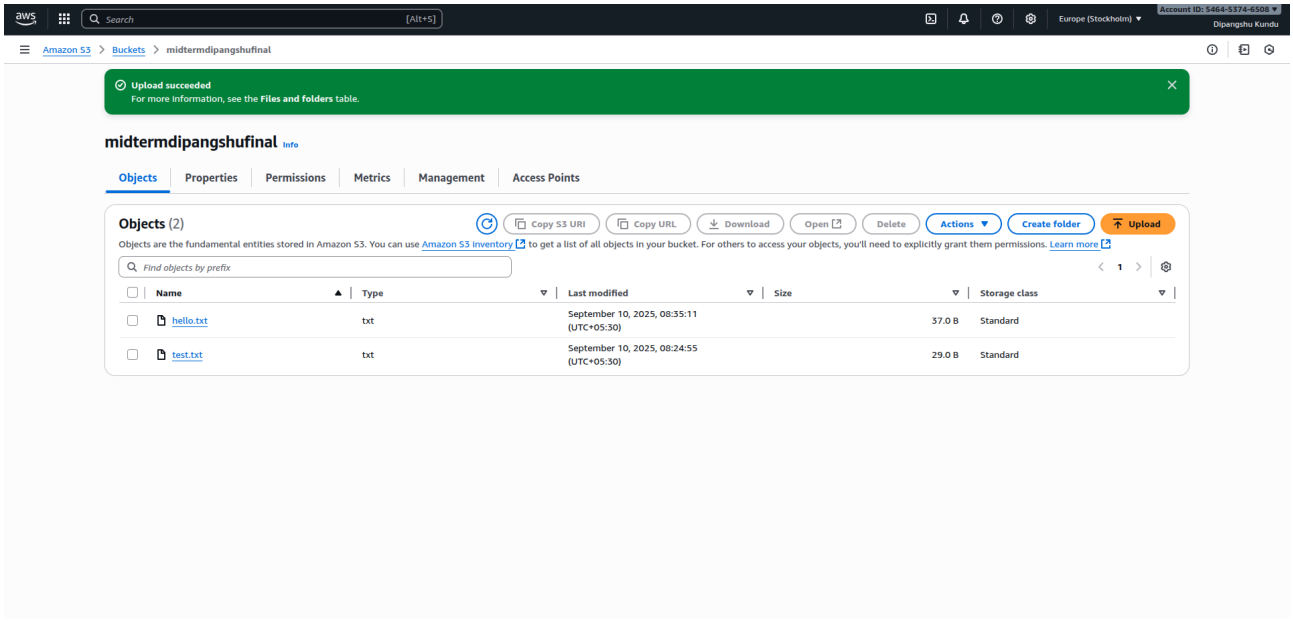
```
import json
import boto3

# Initialize the S3 client outside the handler to reuse the connection
s3 = boto3.client('s3')

def lambda_handler(event, context):
    print("Received event:", json.dumps(event))
```

```
    for record in event['Records']:
        # Extract bucket and object key from the event record
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        try:
            # Use head_object to get object metadata without downloading the file
            response = s3.head_object(Bucket=bucket, Key=key)
            content_type = response['ContentType']

            # Log the content type to CloudWatch
            print(f"File '{key}' in bucket '{bucket}' has a Content-Type of: {content_type}")

        except Exception as e:
            print(f"Error getting metadata for file '{key}': {e}")

    return {
        'statusCode': 200,
        'body': json.dumps('Content-type detection complete.')
    }
```

---

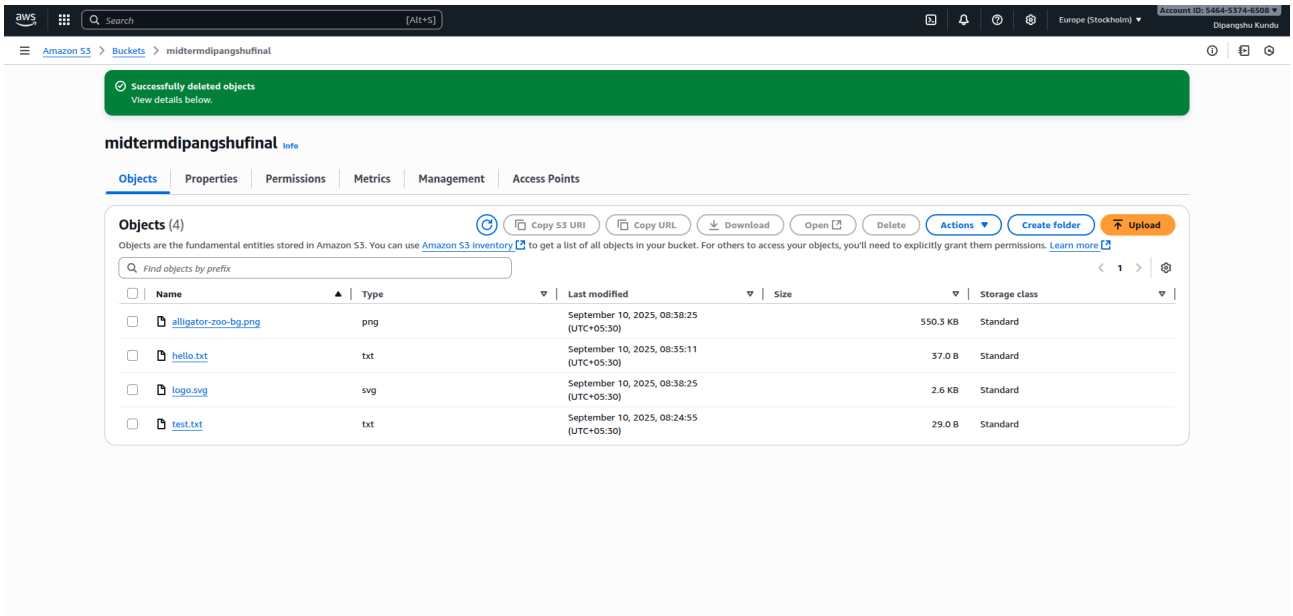
6. Test the Integration

1. We Go to our **S3 bucket**.
2. Upload a new file (e.g., hello.txt).

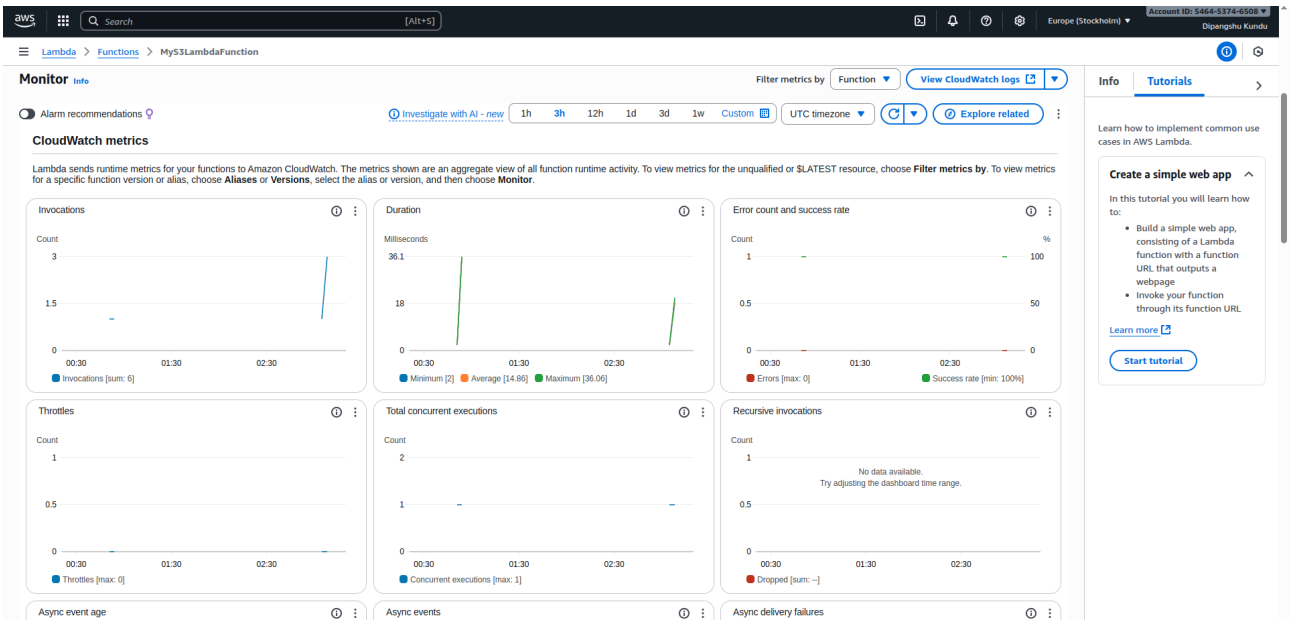


---

7. UPLOADING THREE DIFFERENT TYPES OF FILES



8. → Go back to Lambda → Monitor → Logs (CloudWatch).
→ We should see logs showing the file name, size, and bucket



→ VERIFYING THAT THE LOGS ARE OF CORRECT DETECTED TYPES

hello.txt

-

alligator-zoo-bg.png

-

logo.svg