Project Tested:

Now Below script used to list all the objects

import boto3

# Create a session using your AWS credentials and region

session = boto3.Session(

region\_name='us-east-1' # Replace with your region

)

# Create an S3 client using the session

s3 = session.client('s3')

def list\_objects\_in\_bucket(bucket\_name):

try:

# List all objects in the specified S3 bucket

response = s3.list\_objects\_v2(Bucket=bucket\_name)

# Check if the bucket contains objects

if 'Contents' in response:

print(f"Objects in bucket '{bucket\_name}':")

for obj in response['Contents']:

print(f" - {obj['Key']}")

else:

print(f"No objects found in bucket '{bucket\_name}'.")

except Exception as e:

print(f"Error listing objects in bucket: {e}")

# Specify the bucket name

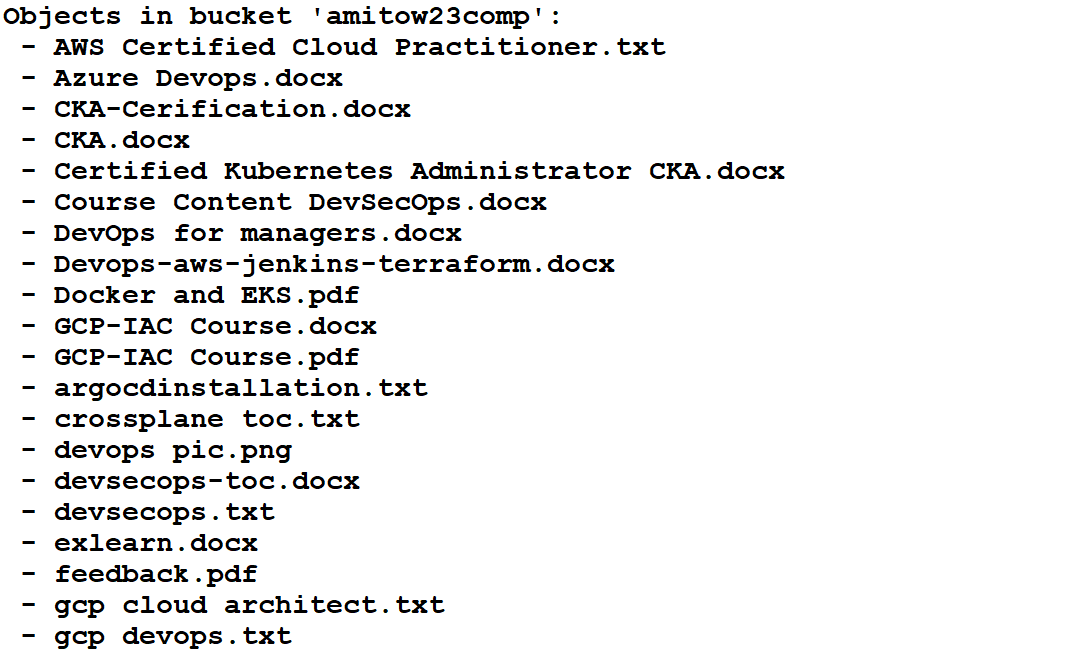
**bucket\_name = 'amitow23comp' # Replace with your bucket name**

# List objects in the specified bucket

list\_objects\_in\_bucket(bucket\_name)

# Apply the lifecycle rule

set\_lifecycle\_policy(bucket\_name)



Only want to do bulk restore for docx format

BUCKET\_NAME="amitow23comp"

# Get a list of all .docx files in the bucket (ensuring correct handling of spaces)

DOCX\_FILES=$(aws s3api list-objects --bucket "$BUCKET\_NAME" --query "Contents[?ends\_with(Key, '.docx')].Key" --output json | jq -r '.[]')

# Check if any .docx files exist

if [ -z "$DOCX\_FILES" ]; then

echo "No .docx files found in bucket: $BUCKET\_NAME"

exit 1

fi

# Loop through each .docx file and initiate restore

echo "Initiating restore for .docx files in bucket: $BUCKET\_NAME"

while IFS= read -r FILE; do

echo "Processing: $FILE"

aws s3api restore-object \

--bucket "$BUCKET\_NAME" \

--key "$FILE" \

--restore-request '{"Days":7, "GlacierJobParameters":{"Tier":"Standard"}}'

echo "Restore request sent for: $FILE"

done <<< "$DOCX\_FILES"

echo "All restore requests have been initiated."

