Using Lambda

1. Upload the text file in 1st bucket. The file should be replicated in the 2nd bucket. The list_object or get_object should be used to get the list of files in the bucket.
2. There should be the transformation of the file making all the characters to the upper case. The output should be printed in the console and be saved in the 3rd bucket. There should be total 3 buckets.

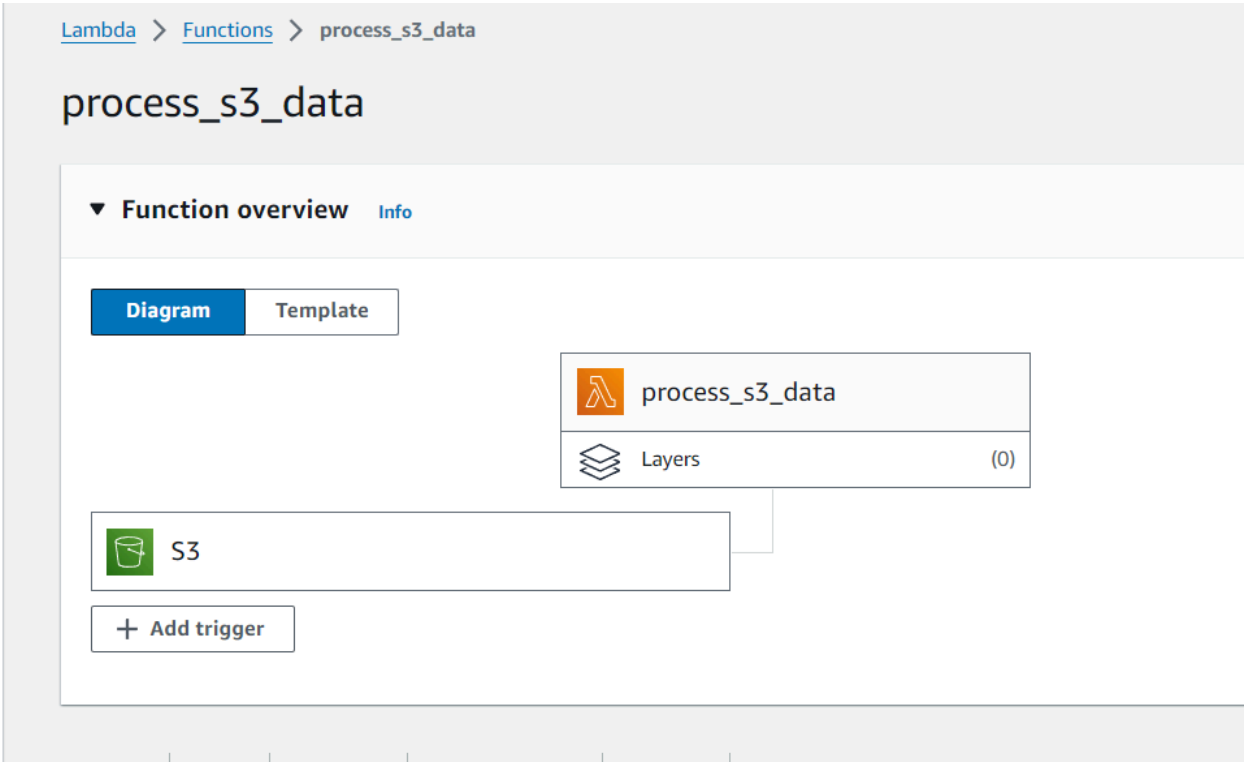
   This task should be done using s3 and lambda

Script to upload the folder:

```
import boto3

bucket_name = 'myfirstbucket-1313'
upload_folder = 'upload_folder/'
key = f'{upload_folder}file1.txt'
content = 'my name is silence'

def upload_text_to_s3(bucket, key, content):
    s3_client = boto3.client('s3')

    try:
        s3_client.put_object(Bucket=bucket, Key=key, Body=content)
        print(f"Successfully uploaded content to {bucket}/{key}")
    except Exception as e:
        print(f"Failed to upload content to {bucket}/{key}")
        print(e)

def list_files_in_folder(bucket, folder):
    s3_client = boto3.client('s3')

    try:
        response = s3_client.list_objects_v2(Bucket=bucket, Prefix=folder)
        if 'Contents' in response:
            print(f"Files in {folder}:")
            for item in response['Contents']:
                print(item['Key'])
        else:
            print(f"No files found in {folder}.")
    except Exception as e:
        print(f"Failed to list files in {folder}")
        print(e)
```

```
# Upload the file first
upload_text_to_s3(bucket_name, key, content)

# Then list all the files in the upload_folder
list_files_in_folder(bucket_name, upload_folder)
```

Lambda structure:

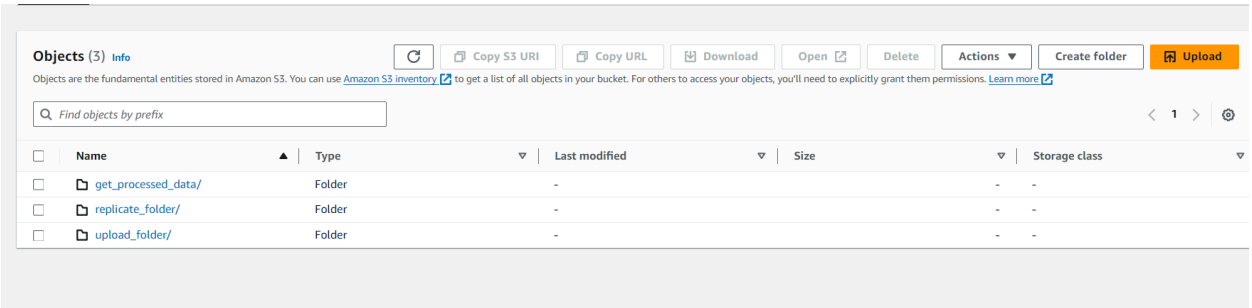


Lambda code:

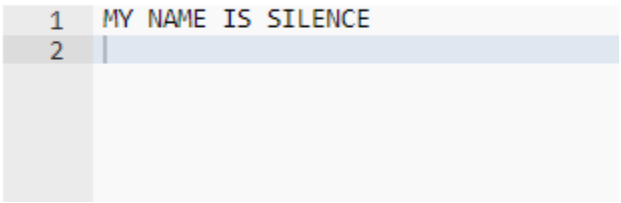
```
import boto3
import urllib.parse
import os

def lambda_handler(event, context):
    s3_client = boto3.client('s3')

    # Get bucket name and object key from the S3 event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
encoding='utf-8')

    # Extract the filename from the object key
```

```
filename = os.path.basename(object_key)

# Define destination key (output folder and file name)
destination_key = 'get_processed_data/' + filename
replicate_key = 'replicate_folder/' + filename
# Get the file from S3
file_obj = s3_client.get_object(Bucket=bucket_name, Key=object_key)
file_content = file_obj['Body'].read().decode('utf-8')

#replicate the content to replicate folder
s3_client.put_object(Bucket=bucket_name, Key=replicate_key, Body=file_content)

# Convert content to uppercase
upper_content = file_content.upper()

# Upload the modified content back to S3
s3_client.put_object(Bucket=bucket_name, Key=destination_key, Body=upper_content)

return {
    'statusCode': 200,
    'body': 'File processed and uploaded successfully'
}
```

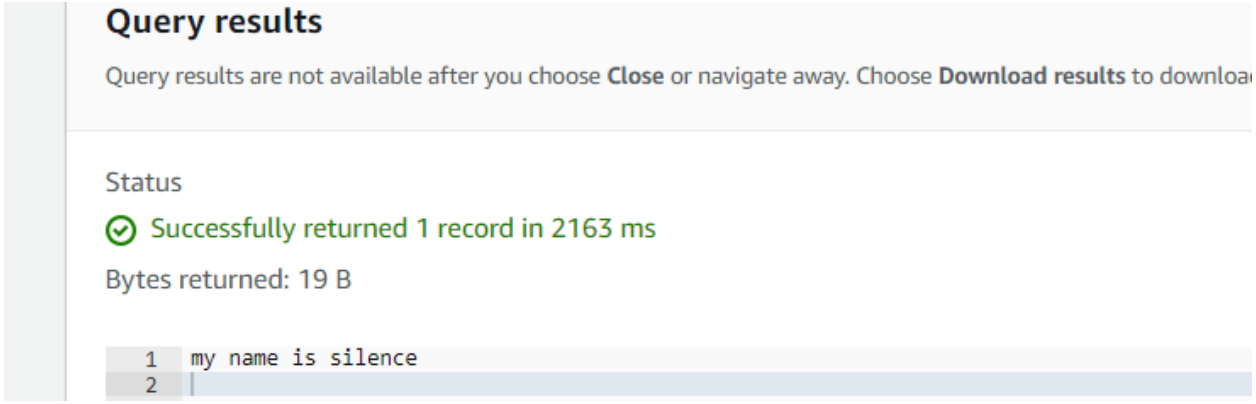
S3 folder structure:



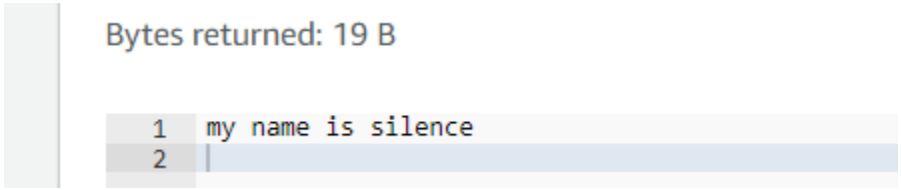
Body of file1.txt inside get_processed_data.



Body of file1.txt inside upload_folder folder:



Body of file1.txt inside replicate_folder folder:



Running the uploadtxt.py script in EC2 instance console:

```
[ec2-user@ip-172-31-19-179 ~]$ python3 uploadtxt.py
Successfully uploaded content to myfirstbucket-1313/upload_folder/file1.txt
Files in upload_folder/:
upload_folder/
upload_folder/file1.txt
upload_folder/myfile.txt
upload_folder/myfile113.txt
upload_folder/myfile13.txt
upload_folder/myfile4.txt
```