

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.

Steps Involved:

- 1) Go to S3> Bucket and Create a Bucket.

[Amazon S3](#) > [Buckets](#) > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

AWS Region

US East (N. Virginia) us-east-1 ▼

Bucket type [Info](#)

☒ **General purpose**

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory - New**

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

sonu-bucket-1

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

2) Enable Bucket Versioning.

objects

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☐ Disable

☒ Enable

3) Keep other things default as it. And click on Create Bucket.

► Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel Create bucket

4) Repeat these processes for creating 3 buckets.

General purpose buckets (3) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

< 1 > ⚙

	Name	AWS Region	Access	Creation date
<input type="radio"/>	sonu-bucket-1	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:48:52 (UTC+05:45)
<input type="radio"/>	sonu-bucket-2	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:49:42 (UTC+05:45)
<input type="radio"/>	sonu-bucket-3	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:50:01 (UTC+05:45)

5) Go to the Lambda to create a function. Give appropriate function name and the language you use to write your function.

[Lambda](#) > [Functions](#) > Create 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 you

Basic information

Function name [Info](#)
Enter a name that describes the purpose of your function.

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 3.12

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

- 6) In permissions part, "choose to use an existing role" as execution role and "LabRole" as an existing role.

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

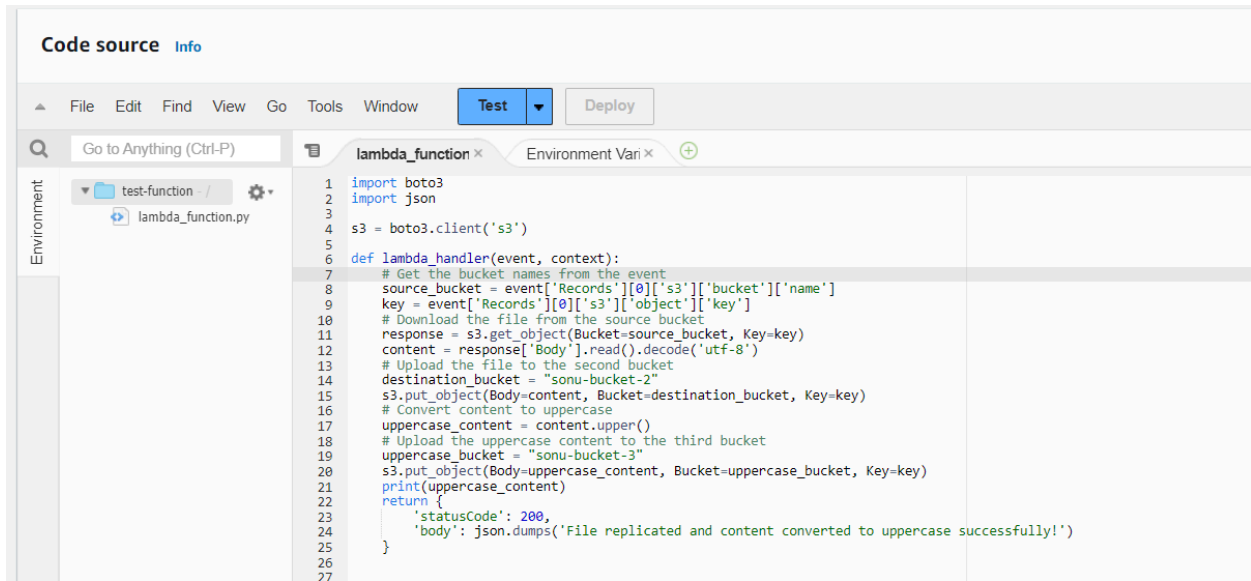
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

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

- 7) Select "Create function"

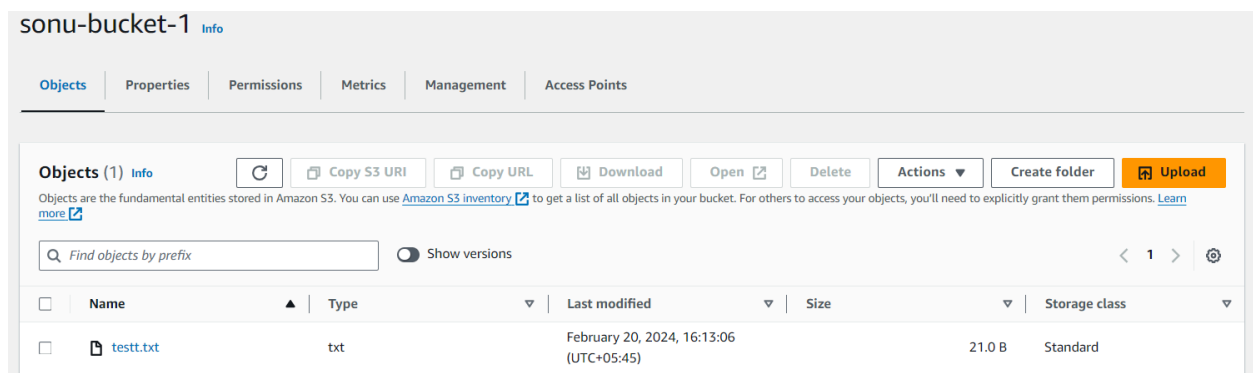
- 8) Go to the code source of lambda function and write the code that performs the related task.



```
1 import boto3
2 import json
3
4 s3 = boto3.client('s3')
5
6 def lambda_handler(event, context):
7     # Get the bucket names from the event
8     source_bucket = event['Records'][0]['s3']['bucket']['name']
9     key = event['Records'][0]['s3']['object']['key']
10    # Download the file from the source bucket
11    response = s3.get_object(Bucket=source_bucket, Key=key)
12    content = response['Body'].read().decode('utf-8')
13    # Upload the file to the second bucket
14    destination_bucket = "sonu-bucket-2"
15    s3.put_object(Body=content, Bucket=destination_bucket, Key=key)
16    # Convert content to uppercase
17    uppercase_content = content.upper()
18    # Upload the uppercase content to the third bucket
19    uppercase_bucket = "sonu-bucket-3"
20    s3.put_object(Body=uppercase_content, Bucket=uppercase_bucket, Key=key)
21    print(uppercase_content)
22    return {
23        'statusCode': 200,
24        'body': json.dumps('File replicated and content converted to uppercase successfully!')}
25
26
27
```

- 9) Now go to the first bucket you made and upload the .txt file from your computer.


The first bucket is the original bucket. In the second bucket, the text file is the replica of the first bucket whereas in the third bucket, the file should be transformed making all the characters to the upper case.



10) Then, add trigger configuration. Choose the bucket you want to trigger.

Add trigger

Trigger configuration [Info](#)

 **S3**
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.


× ↺

Bucket region: us-east-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 × PUT ×

11) Select I acknowledge part of Recursive invocation and click on Add button.

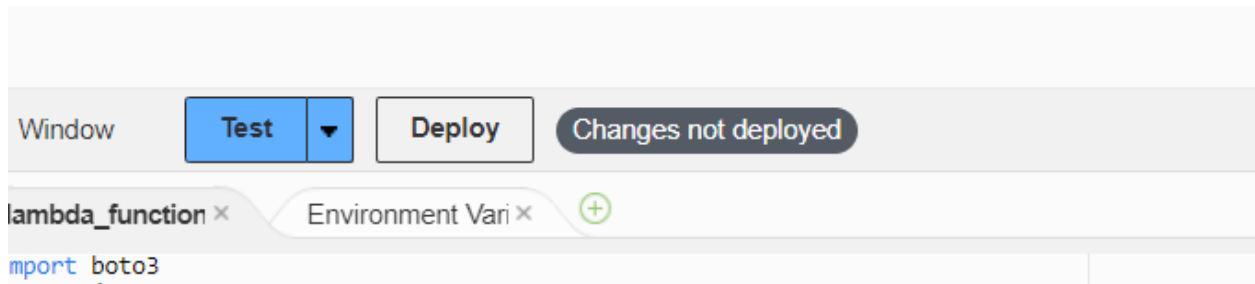
Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#) 

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

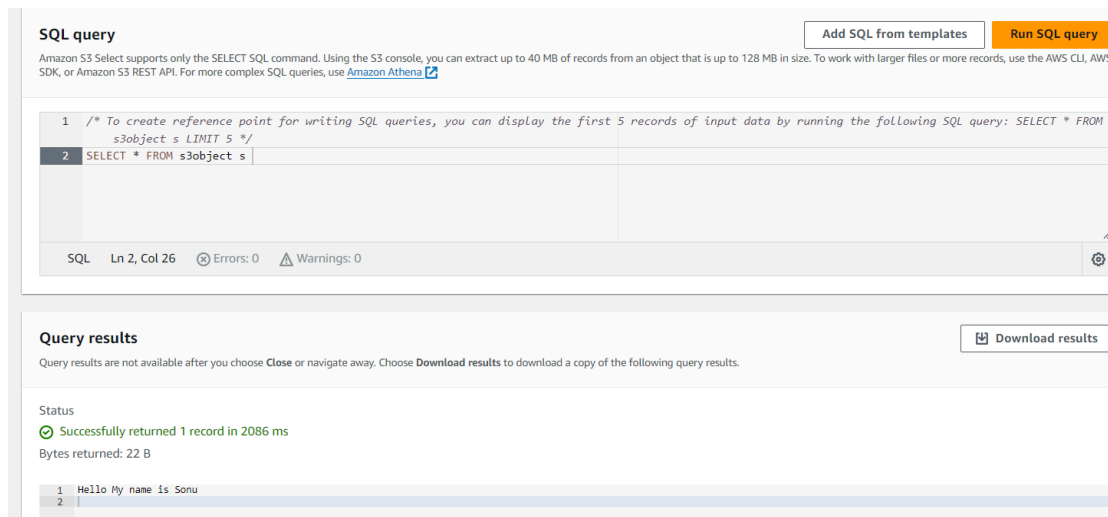
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#)  about the Lambda permissions model.

Cancel Add

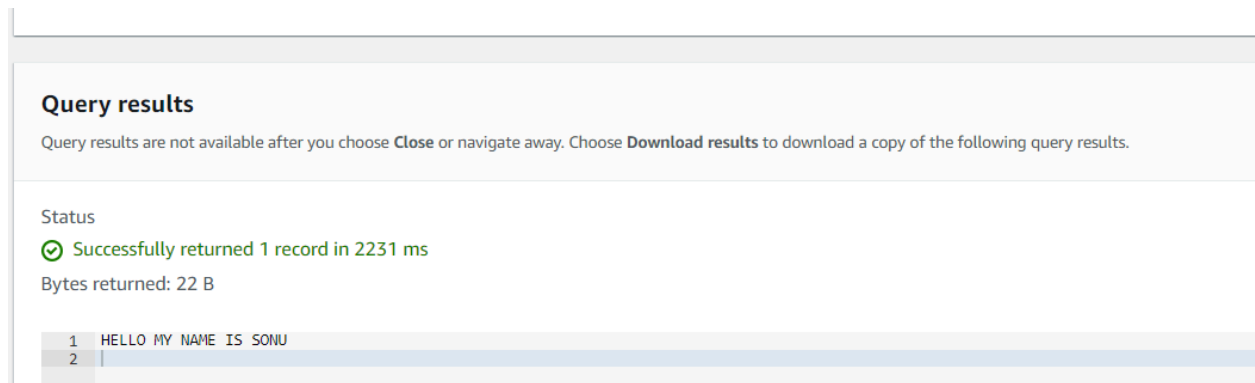
12) Now deploy the code



13) You can check text file in bucket by running SQL query too. This is bucket-2.



14) This is S3 bucket



15) You can go to the cloudwatch and check the log events.

[CloudWatch](#) > [Log groups](#) > [/aws/lambda/test-function](#) > 2024/02/20/[\$LATEST]486f9968ca24446bb410160082973533

Log events

Actions

Start tailing

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](#)

Filter events

Clear

1m

30m

1h

12h

Custom

Local timezone

▶	Timestamp	Message
		No older events at this moment. Retry
▶	2024-02-20T16:13:07.029+05:45	INIT_START Runtime Version: python:3.12.v18 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:776a3759221679a634181f858871d
▶	2024-02-20T16:13:07.498+05:45	START RequestId: 780f5d93-e3b9-45fd-9821-43b35a39c6d4 Version: \$LATEST
▶	2024-02-20T16:13:09.128+05:45	HELLO MY NAME IS SONU
▶	2024-02-20T16:13:09.129+05:45	END RequestId: 780f5d93-e3b9-45fd-9821-43b35a39c6d4
▶	2024-02-20T16:13:09.129+05:45	REPORT RequestId: 780f5d93-e3b9-45fd-9821-43b35a39c6d4 Duration: 1631.41 ms Billed Duration: 1632 ms Memory Size: 128 MB Max Mem
		No newer events at this moment. Auto retry paused. Resume