

Serverless Data Processing Pipeline

Objective: Build a serverless pipeline for processing data (e.g., log processing or ETL jobs).

Approach:

- **Data Ingestion:** Use AWS services like S3 or Kinesis to ingest data.
- **Processing:** Create Lambda functions to process the ingested data.
- **Storage:** Store the processed data in an appropriate AWS service, like S3 or DynamoDB.
- **Monitoring:** Set up CloudWatch to monitor the pipeline's performance and to log any issues.

Goal: Learn to build a serverless data processing pipeline, understanding the flow of data through various AWS services.

1. First of all I created 3 different buckets namely Upesh bucket 1 2 and 3 respectively.

The screenshot displays the AWS S3 console interface for 'General purpose buckets'. At the top, there are tabs for 'General purpose buckets' and 'Directory buckets'. Below the tabs, there's a header section with 'General purpose buckets (5)' and an 'Info' link. To the right of the header are buttons for 'Refresh', 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. Below the header is a search bar with the placeholder text 'Find buckets by name'. The main content area is a table listing the buckets. The table has five columns: 'Name', 'AWS Region', 'Access', and 'Creation date'. There are five rows of data, each representing a bucket. The first row is 'namebucketq' in 'US East (N. Virginia) us-east-1' with 'Objects can be public' access. The second row is 'upeshbasiclab' in 'US East (N. Virginia) us-east-1' with 'Public' access. The third row is 'upeshbucket1' in 'US East (N. Virginia) us-east-1' with 'Bucket and objects not public' access. The fourth row is 'upeshbucket2' in 'US East (N. Virginia) us-east-1' with 'Bucket and objects not public' access. The fifth row is 'upeshbucket3' in 'US East (N. Virginia) us-east-1' with 'Bucket and objects not public' access. The creation dates are: February 23, 2024, 13:37:11 (UTC+05:45) for namebucketq; February 22, 2024, 22:28:41 (UTC+05:45) for upeshbasiclab; and February 20, 2024, 15:48:55 (UTC+05:45), February 20, 2024, 15:49:14 (UTC+05:45), and February 20, 2024, 15:49:34 (UTC+05:45) for the other three buckets respectively.

Name	AWS Region	Access	Creation date
namebucketq	US East (N. Virginia) us-east-1	Objects can be public	February 23, 2024, 13:37:11 (UTC+05:45)
upeshbasiclab	US East (N. Virginia) us-east-1	Public	February 22, 2024, 22:28:41 (UTC+05:45)
upeshbucket1	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:48:55 (UTC+05:45)
upeshbucket2	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:49:14 (UTC+05:45)
upeshbucket3	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:49:34 (UTC+05:45)

2. Then I uploaded files in the first bucket upeshbucket1

[Amazon S3](#) > [Buckets](#) > upeshbucket1

upeshbucket1 [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2) [Info](#)

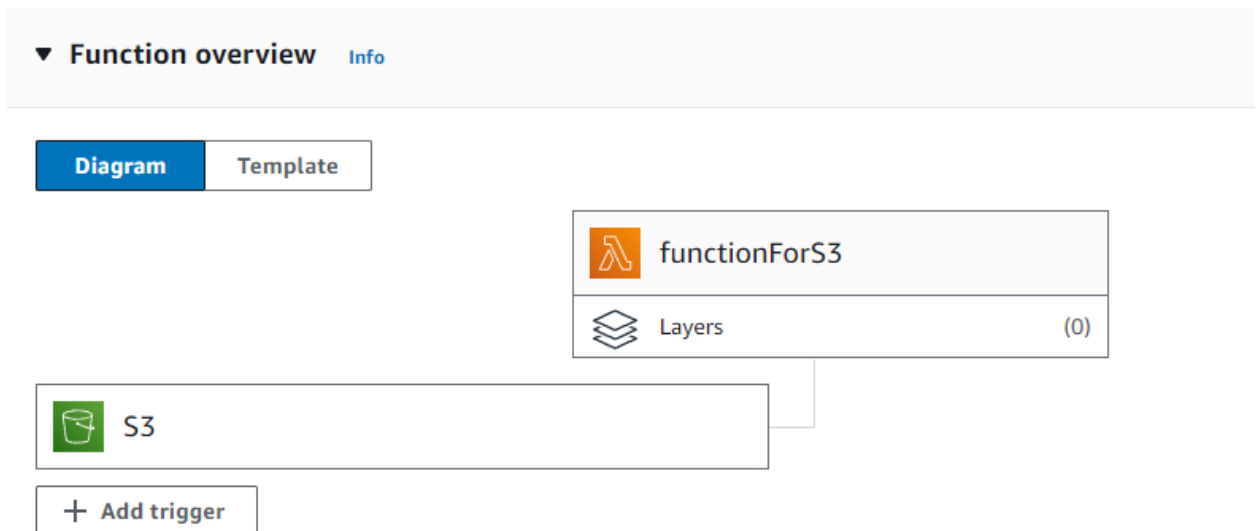
Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

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](#)

Show versions < 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Hello.txt	txt	February 20, 2024, 16:12:36 (UTC+05:45)	12.0 B	Standard
<input type="checkbox"/>	They.txt	txt	February 20, 2024, 16:12:38 (UTC+05:45)	9.0 B	Standard

3. Create a lambda function



4. I used the code to upload object in 1st bucket and the object would be replicated in the 2nd bucket. Where the 3rd bucket will have all the files in Uppercase.

Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl-P)

Environment

functionForS3 - /

lambda_function.py

```
1 import json
2
3 import boto3
4 import json
5
6 s3 = boto3.client('s3')
7
8 def lambda_handler(event, context):
9     # Get the bucket names from the event
10    source_bucket = event['Records'][0]['s3']['bucket']['name']
11    key = event['Records'][0]['s3']['object']['key']
12    # Download the file from the source bucket
13    response = s3.get_object(Bucket=source_bucket, Key=key)
14    content = response['Body'].read().decode('utf-8')
15    # Upload the file to the second bucket
16    destination_bucket = "upeshbucket2"
17    s3.put_object(Body=content, Bucket=destination_bucket, Key=key)
18    # Convert content to uppercase
19    uppercase_content = content.upper()
20    # Upload the uppercase content to the third bucket
21    uppercase_bucket = "upeshbucket"
22    s3.put_object(Body=uppercase_content, Bucket=uppercase_bucket, Key=key)
23    print(uppercase_content)
24    return {
25        'statusCode': 200,
26        'body': json.dumps('File replicated and content converted to uppercase successfully!')}
27
```

5. In the bucket 2 we can see that the bucket is now replicated from the first bucket

upeshbucket2 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (2) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

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](#)

☐ Show versions < 1 > [Settings](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Hello.txt	txt	February 20, 2024, 16:15:59 (UTC+05:45)	12.0 B	Standard
<input type="checkbox"/>	They.txt	txt	February 20, 2024, 16:15:39 (UTC+05:45)	9.0 B	Standard

6. In the upeshbucket3 now the files should be uploaded in the Uppercase characters. So

Hello.txt [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#)

[Properties](#) [Permissions](#) [Versions](#)

Object overview

Owner	awslabsc0w6975250t1703164493	S3 URI	s3://upeshbucket3/Hello.txt
AWS Region	US East (N. Virginia) us-east-1	Amazon Resource Name (ARN)	arn:aws:s3:::upeshbucket3/Hello.txt
Last modified	February 20, 2024, 16:07:13 (UTC+05:45)	Entity tag (Etag)	b59bc37d6441d96785bda7ab2ae98f75
Size	12.0 B	Object URL	https://upeshbucket3.s3.amazonaws.com/Hello.txt
Type	txt		
Key			

7. When we see the Hello.txt file we can see that all the characters are now in Uppercase characters.

