

3. Serverless Data Processing Pipeline

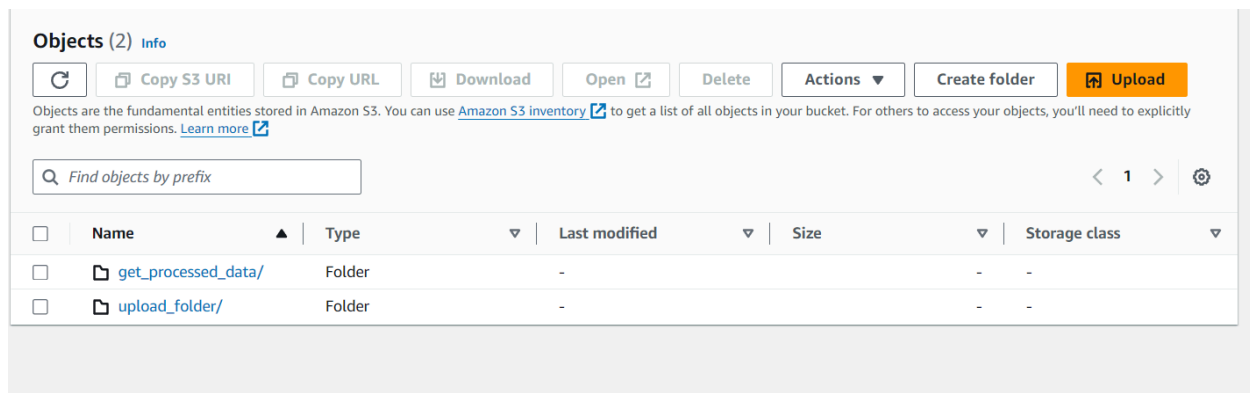
Objective: Build a serverless pipeline for processing data (e.g., log processing or ETL jobs).

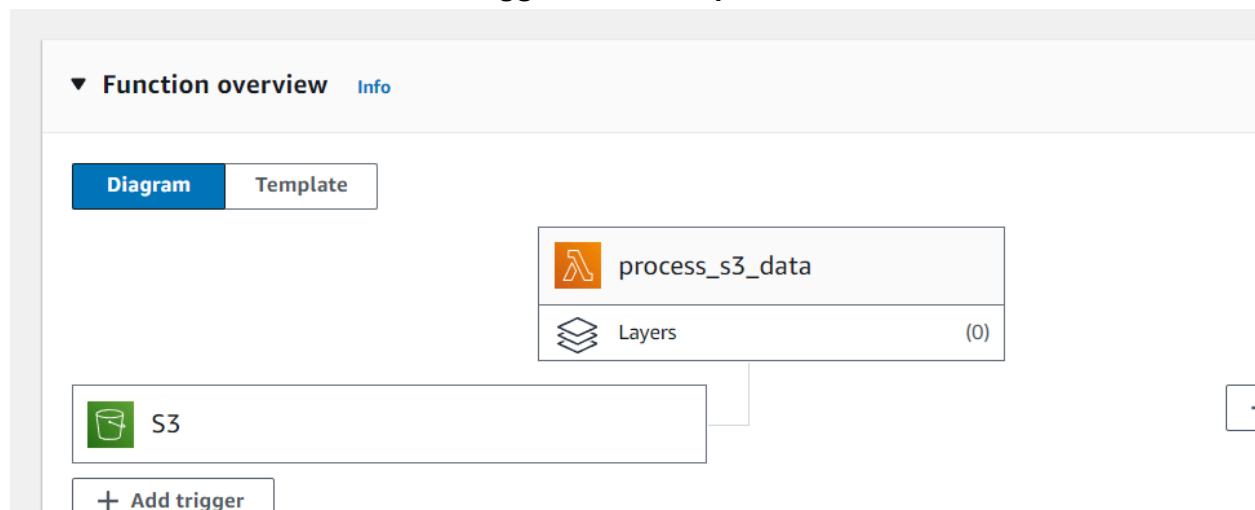
Approach:

- **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.

Create s3 bucket with two folder source and destination folder:



Create lambda function and add trigger to s3 file upload:**Lambda function 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

    # 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')

    # 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 upload code:

```
import boto3
```

```
# Initialize an S3 client
```

```
s3_client = boto3.client('s3')
```

```
# The string you want to upload
```

```
my_string = "This is a simple Tw sfso string."
```

```
# The S3 bucket name
```

```
bucket_name = 'myfirstbucket-1313'
```

```
# The key (file name) to use for the uploaded string, including the folder path
```

```
object_key = 'upload_folder/myfile13.txt'
```

```
# Upload the string
```

```
s3_client.put_object(Bucket=bucket_name, Key=object_key, Body=my_string)
```

Input Test:

```
# The string you want to upload
```

```
my_string = "This is a simple Tw sfso string."
```

Output text:

```
THIS IS A SIMPLE TW SFSO STRING.
```

Monitoring the lambda log with cloud watch:

CloudWatch > Log groups > /aws/lambda/process_s3_data > 2024/01/22/[\$LATEST]4fd1e10abc08407c8a72c075ddbd072b

Log events

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](#)

🔄

Actions ▾

Start tailing

Create metric filter

🔍 Filter events

Clear1m30m1h12hCustom📄Local timezone▼Display

▶	Timestamp	Message
No more records within selected time range Retry		
▶	2024-01-22T16:12:48.699+05:45	INIT_START Runtime Version: python:3.12.v16 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:c9875014cbcc77e3455765804516f064d18fe7
▶	2024-01-22T16:12:48.991+05:45	START RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85 Version: \$LATEST
▶	2024-01-22T16:12:51.098+05:45	[ERROR] NameError: name 'object_key' is not defined Traceback (most recent call last): File "/var/task/lambda_function.py", line 13, in
▶	2024-01-22T16:12:51.118+05:45	END RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85
▶	2024-01-22T16:12:51.118+05:45	REPORT RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85 Duration: 2127.45 ms Billed Duration: 2128 ms Memory Size: 128 MB Max Memory Used:
▶	2024-01-22T16:13:50.250+05:45	START RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85 Version: \$LATEST
▶	2024-01-22T16:13:50.317+05:45	[ERROR] NameError: name 'object_key' is not defined Traceback (most recent call last): File "/var/task/lambda_function.py", line 13, in
▶	2024-01-22T16:13:50.338+05:45	END RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85
▶	2024-01-22T16:13:50.338+05:45	REPORT RequestId: e6c27f79-6993-47db-ad53-60e803a8fe85 Duration: 87.32 ms Billed Duration: 88 ms Memory Size: 128 MB Max Memory Used: 80
No more records within selected time range Auto retry paused . Resume		

4