

Creating 2 S3 bucket - one for source-bucket and destination-bucket.

Creating IAM Role for lambda function to read the source-bucket and write the destination-bucket.

Then Creating Lambda Function

Python Code

```
import json
```

```
import boto3
```

```
def lambda_handler(event, context):
```

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

```
    # Get the bucket and file name from the event
```

```
    bucket = event['Records'][0]['s3']['bucket']['name']
```

```
    key = event['Records'][0]['s3']['object']['key']
```

```
    # Download the file from S3
```

```
    response = s3.get_object(Bucket=bucket, Key=key)
```

```
    file_content = response['Body'].read().decode('utf-8')
```

```
    # Count the number of words in the file
```

```
    word_count = len(file_content.split())
```

```
    # Prepare the result
```

```
    result = {
```

```
        'file_name': key,
```

```
        'word_count': word_count
```

```
    }
```

```
    # Convert result to JSON string
```

```
    result_json = json.dumps(result)
```

```
    # Upload the result to the destination bucket
```

```
destination_bucket = 'destination-bucket'
```

```
destination_key = f'{key.split('.')[0]}_word_count.json'
```

```
s3.put_object(Bucket=destination_bucket, Key=destination_key, Body=result_json)
```

```
return {
```

```
    'statusCode': 200,
```

```
    'body': json.dumps('Word count processed successfully!')
```

```
}
```

Ensuring Lambda Function has execution role like

s3:GetObject

s3:PutObject

Then Setting up the S3 to trigger the lambda function upon object creation

go to properties in s3 source and destination bucket and create event notification for lambda function.

Testing ==

Uploading the text file in the S3 source bucket and check the destination-S3-bucket for result JSON file containing WORD COUNT