

Created 3 buckets

<input type="radio"/>	utshas3bucket1	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:48:03 (UTC+05:45)
<input type="radio"/>	utshas3bucket2	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:48:28 (UTC+05:45)
<input type="radio"/>	utshas3bucket3	US East (N. Virginia) us-east-1	Bucket and objects not public	February 20, 2024, 15:49:08 (UTC+05:45)

Created lambda function and implemented the operation for copying content and adding to new s3 bucket and changing the content to upper case and saving the content to next S3 bucket.

```
lambda_function x Environment Var x +
40
41 import json
42 import boto3
43
44 s3 = boto3.client('s3')
45 # s3_resource = boto3.resource('s3')
46
47 def lambda_handler(event, context):
48     # Get the bucket names from the event
49     source_bucket = event['Records'][0]['s3']['bucket']['name']
50     key = event['Records'][0]['s3']['object']['key']
51
52     # Download the file from the source bucket
53     response = s3.get_object(Bucket=source_bucket, Key=key)
54     content = response['Body'].read().decode('utf-8')
55
56     # Upload the file to the second bucket
57     destination_bucket = "utshas3bucket2"
58     s3.put_object(Body=content, Bucket=destination_bucket, Key=key)
59
60     # Convert content to uppercase
61     uppercase_content = content.upper()
62
63     # Upload the uppercase content to the third bucket
64     uppercase_bucket = "utshas3bucket3"
65     s3.put_object(Body=uppercase_content, Bucket=uppercase_bucket, Key=key)
66
67     # Download the file from the third bucket
68     # local_file_path = f"/tmp/{key}" # Temporary directory in Lambda
69     # s3_resource.meta.client.download_file(uppercase_bucket, key, local_file_path)
70     print('key', key)
71     local_file_path = '/tmp/utsha.txt'
72     s3.download_file(uppercase_bucket, key, local_file_path)
73     # s3_resource.meta.client.download_file(uppercase_bucket, key, r'/Users/utsha_mac/Downloads')
74
75     # Return a response
76     return {
77         'statusCode': 200,
78         'body': json.dumps('File replicated, content converted to uppercase, and downloaded successfully!')
79     }
80
```

Content of bucket 1

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use [Amazon Athena](#).

```
1 /* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
2 SELECT * FROM s3object s LIMIT 5
```

SQL Ln 1, Col 1 Errors: 0 Warnings: 0

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 2642 ms

Bytes returned: 15 B

1	utsha shrestha
2	

Content of bucket 2

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use [Amazon Athena](#).

```
1 /* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
2 SELECT * FROM s3object s LIMIT 5
```

SQL Ln 1, Col 1 Errors: 0 Warnings: 0

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 2733 ms

Bytes returned: 15 B

1	utsha shrestha
2	

Content of bucket 3

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use [Amazon Athena](#).

```
1 /* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
2 SELECT * FROM s3object s LIMIT 5
```

SQL Ln 1, Col 1 Errors: 0 Warnings: 0

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 2522 ms

Bytes returned: 15 B

1	UTSHA SHRESTHA
2	

