

Cross Region Replication

- Cross Region Replication is a feature that replicates the data from one bucket to another bucket which could be in a different region.
- It provides asynchronous copying of objects across buckets. Suppose X is a source bucket and Y is a destination bucket. If X wants to copy its objects to Y bucket, then the objects are not copied immediately.

Some points to be remembered for Cross Region Replication

- **Create two buckets:** Create two buckets within AWS Management Console, where one bucket is a source bucket, and other is a destination bucket.
- **Enable versioning:** Cross Region Replication can be implemented only when the versioning of both the buckets is enabled.
- **Amazon S3 encrypts the data in transit across AWS regions using SSL:** It also provides security when data traverse across the different regions.
- **Already uploaded objects will not be replicated:** If any kind of data already exists in the bucket, then that data will not be replicated when you perform the cross region replication.



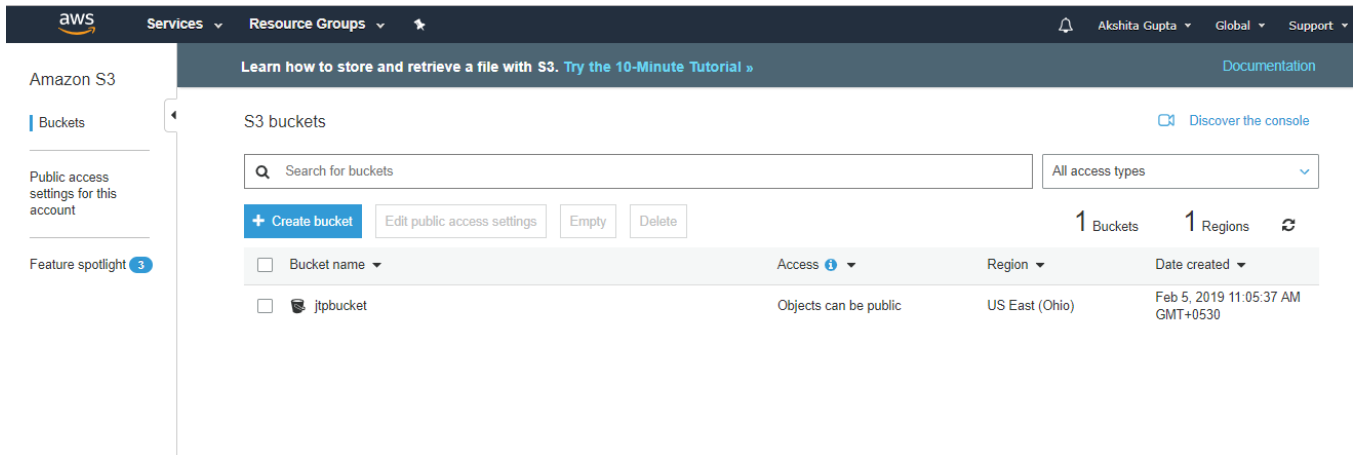
Use cases of Cross Region Replication

- **Compliance Requirements**
By default, Amazon S3 stores the data across different geographical regions or availability zone to have the availability of data. Sometimes there could be compliance requirements that you want to store the data in some specific region. Cross Region Replication allows you to replicate the data at some specific region to satisfy the requirements.
- **Minimize Latency**
Suppose your customers are in two geographical regions. To minimize latency, you need to maintain the copies of data in AWS region that are geographically closer to your users.
- **Maintain object copies under different ownership:** Regardless of who owns the source bucket, you can tell to Amazon S3 to change the ownership to AWS account user that owns the destination bucket. This is referred to as an owner override option.

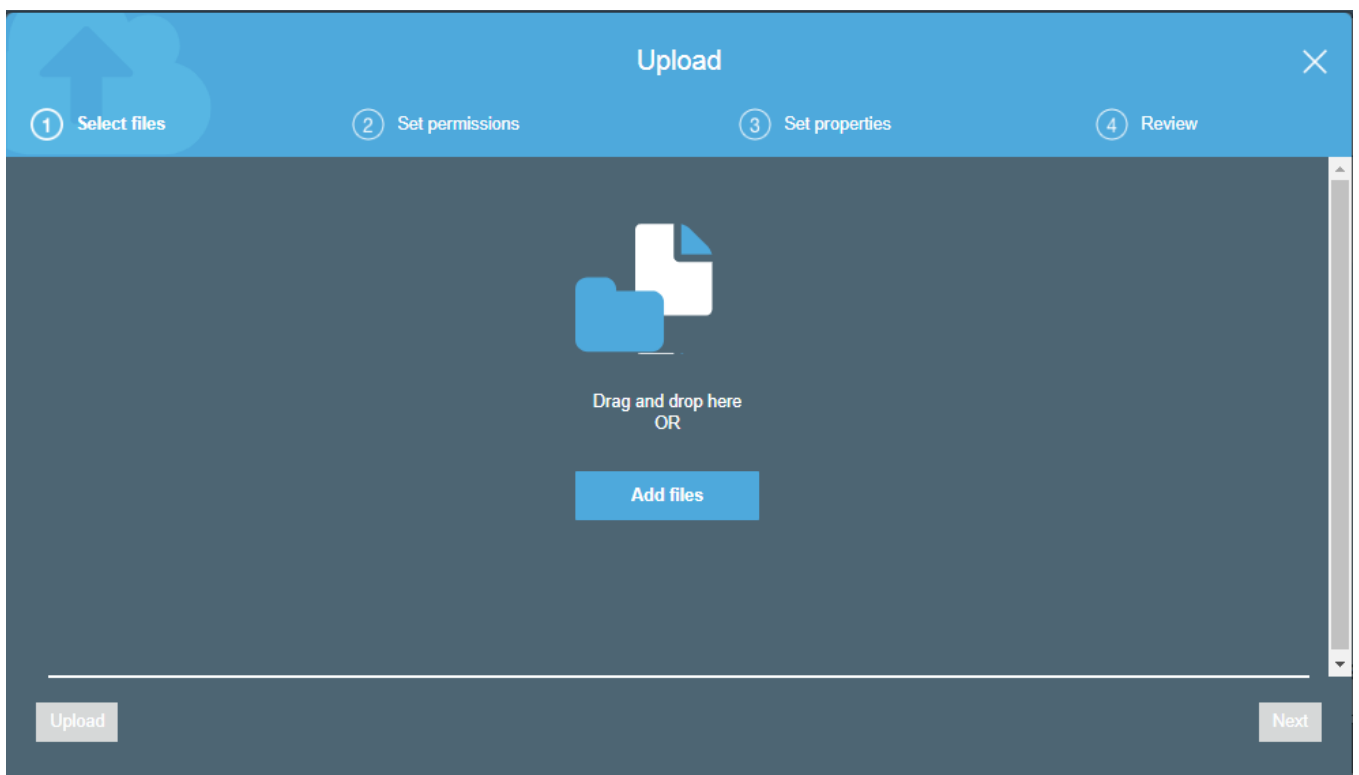
Let's understand the concept of Cross Region Replication through an example.

- Sign in to the AWS Management Console.

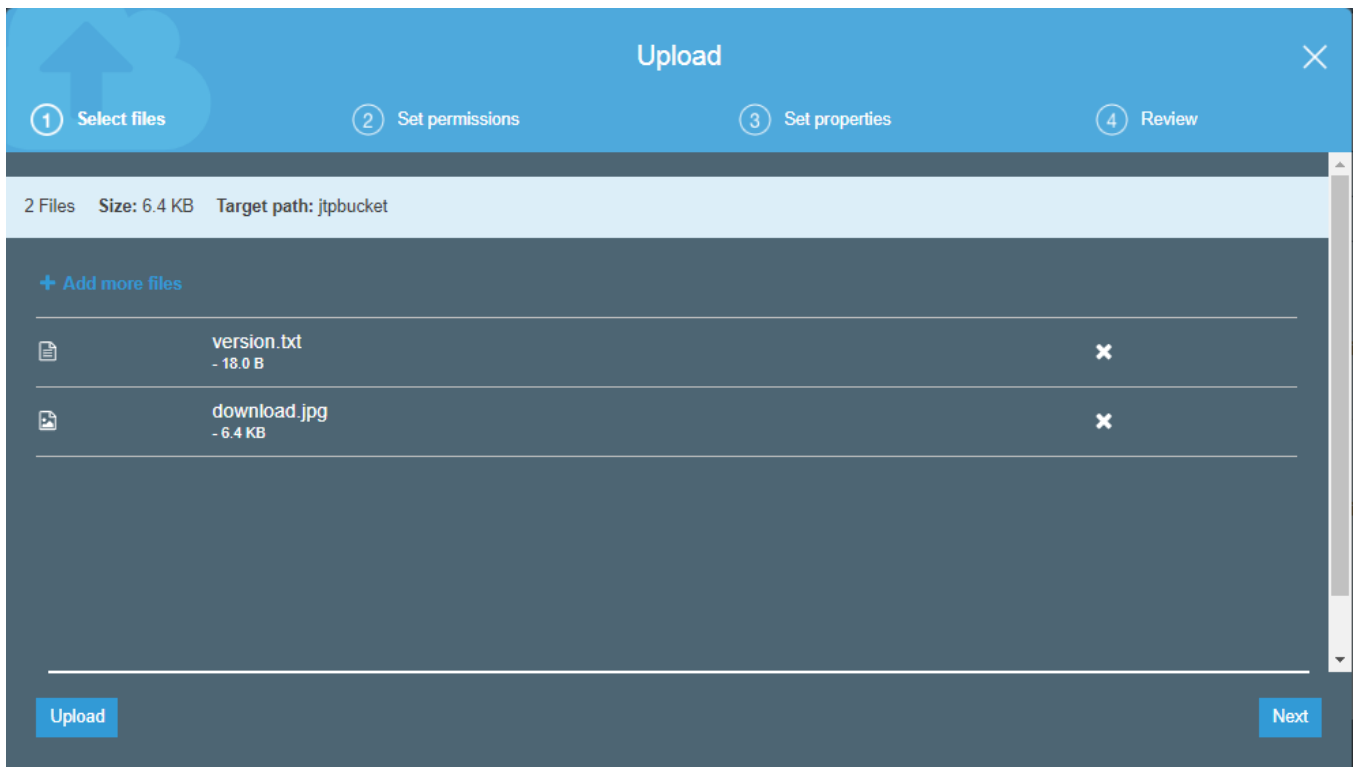
- Now, we upload the files in a **jtpbucket**. The jtpbucket is an s3 bucket created by us.



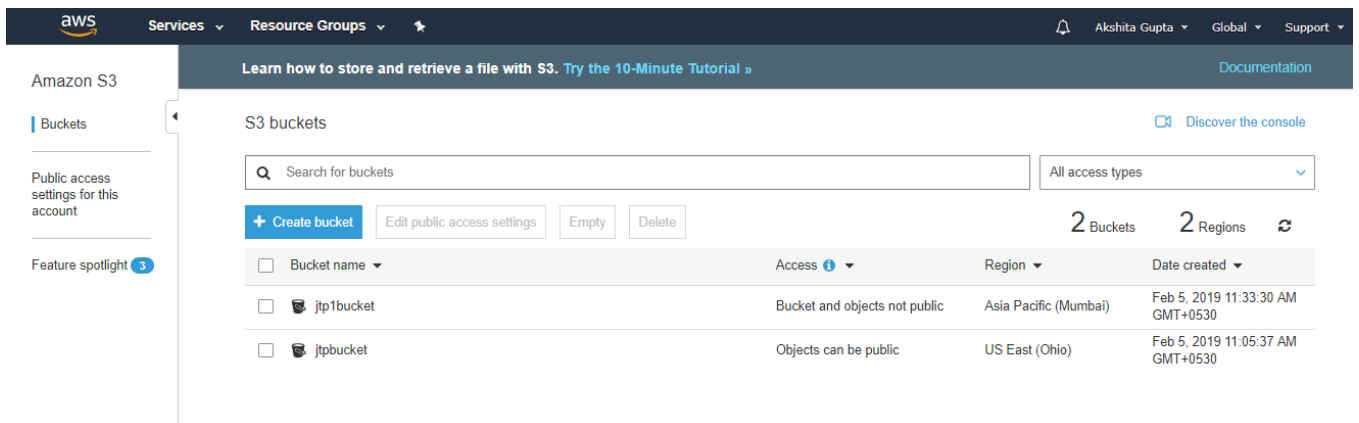
- Add the files in a bucket.



- Now, we add two files in a bucket, i.e., version.txt and download.jpg.

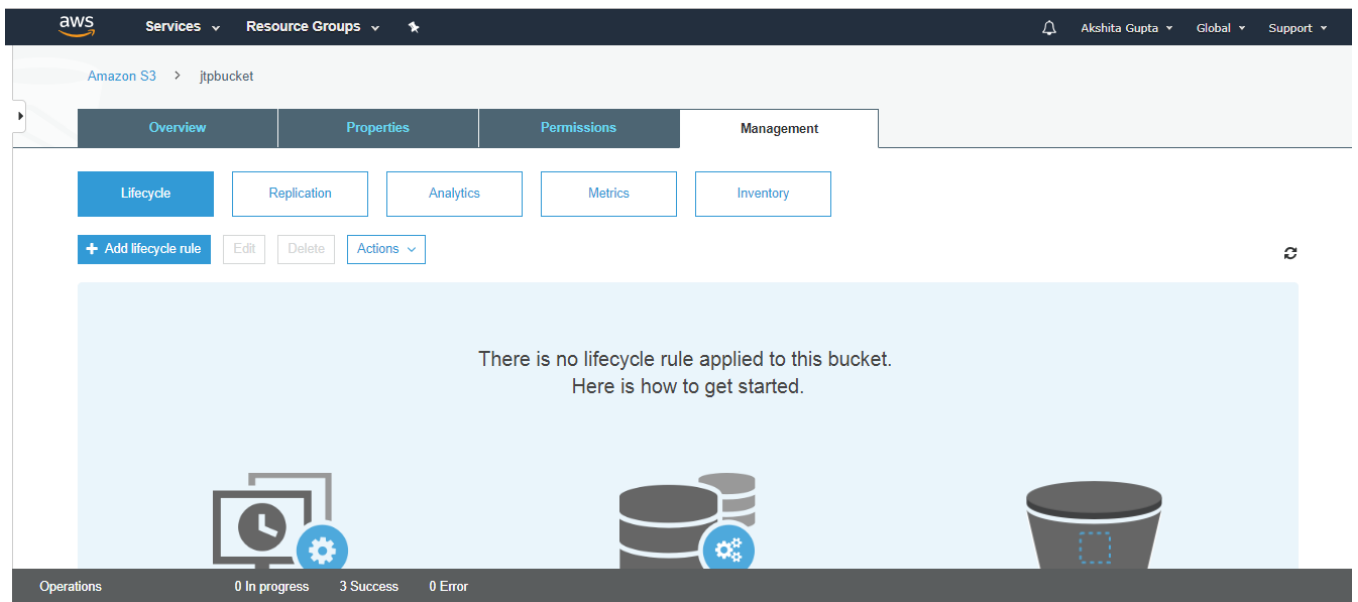


- Now, we create a new bucket whose name is **jtp1bucket** with a different region.

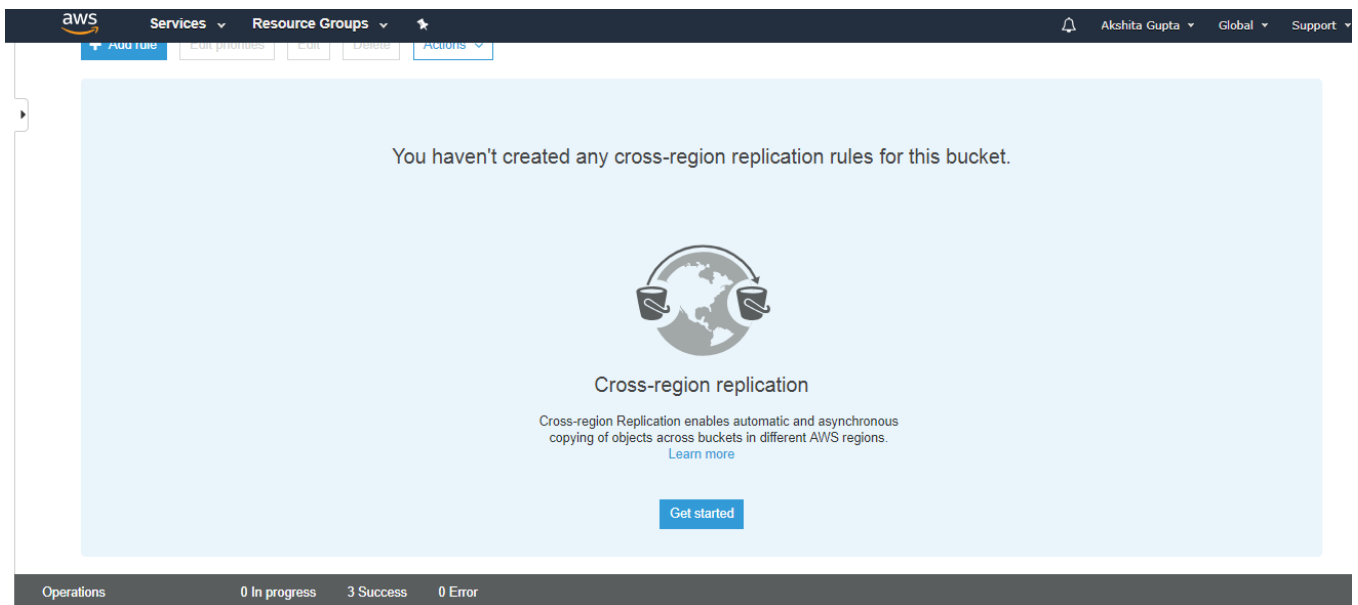


Now, we have two buckets, i.e., jtpbucket and jtp1bucket in s3.

- Click on the **jtpbucket** and then move to the Management of the **jtpbucket**.



- Click on the **Replication**. On clicking, the screen appears as shown below:



- Click on the **Get started** button.
- Enable the versioning of both the buckets.
- You can either replicate the entire bucket or tags to the destination bucket. Suppose you want to replicate the entire bucket and then click on the Next.

Replication rule

1

Set source

2

Set destination


3


Configure options

4


Review


Set source

☒ Entire bucket  jtpbucket

☐ Prefix or tags 

Replication criteria

☐ Replicate objects encrypted with AWS KMS 

 **Your CRR rule will be created using the new schema**

Cross-region replication (CRR) now has a new schema that supports replication based on prefixes, one or more object tags or a combination of the two. As part of the new schema, you can set overlapping rules with priorities. The new schema does not support delete marker replication, which would prevent any delete actions from replicating. [Learn more](#) about cross-region replication.

Cancel

Next

- o Enter your destination bucket, i.e., jtp1bucket.

Replication rule

Set source Set destination **Configure options** Review

Destination bucket

jtp1bucket

Options

☐ Change the storage class for the replicated object(s)

☐ Change object ownership to destination bucket owner ⓘ

Previous Next

- Create a new IAM role, and the role name is S3CRR and then click on the Next.

Replication rule

✓ Set source

✓ Set destination

3 Configure options

4 Review

IAM role

Create new role

Rule name

S3CRR

Status

☒ Enabled

☐ Disabled

Previous

Next

- After saving the settings, the screen appears as shown below:

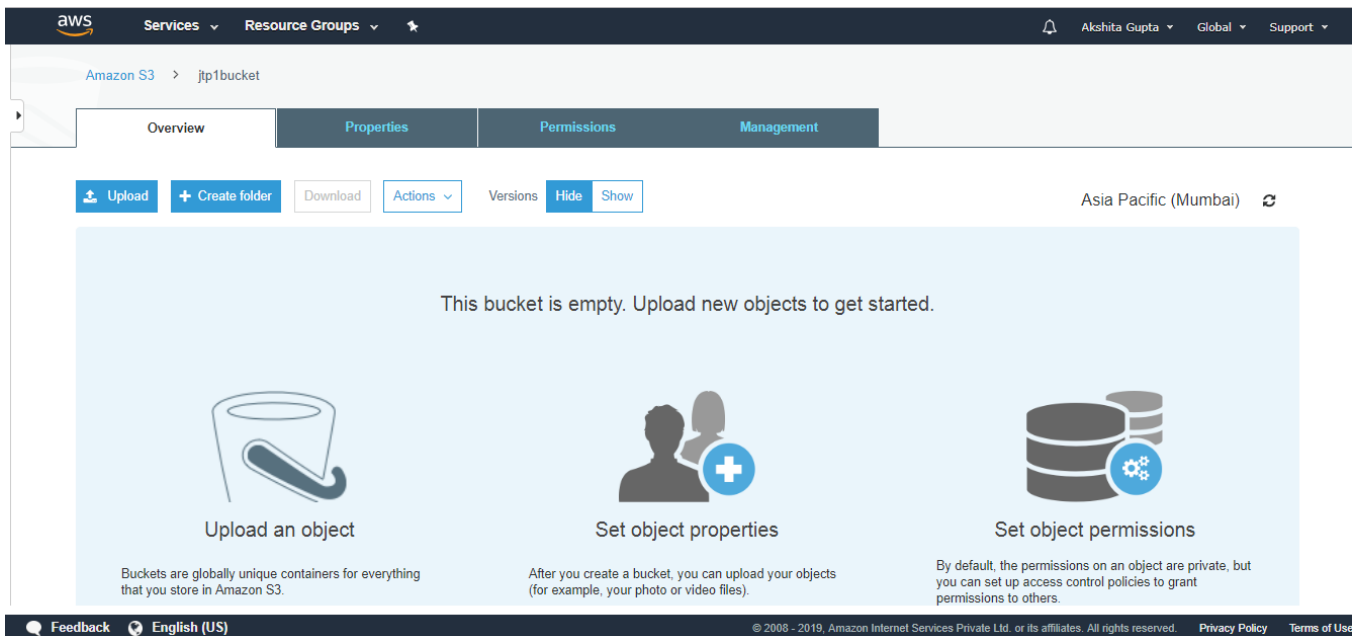
Cross-region replication updated successfully.

| Source | Destination | Permissions | Edit global settings |
|--------------------------|---------------------------------|--|--------------------------------------|
| Bucket jtpbucket | Bucket jtp1bucket | IAM role s3crr_role_for_jtpbucket_to_jtp1bucket | |
| Region US East (Ohio) | Region Asia Pacific (Mumbai) | Bucket policy Copy | |

[+ Add rule](#) [Edit priorities](#) [Edit](#) [Delete](#) [Actions](#)

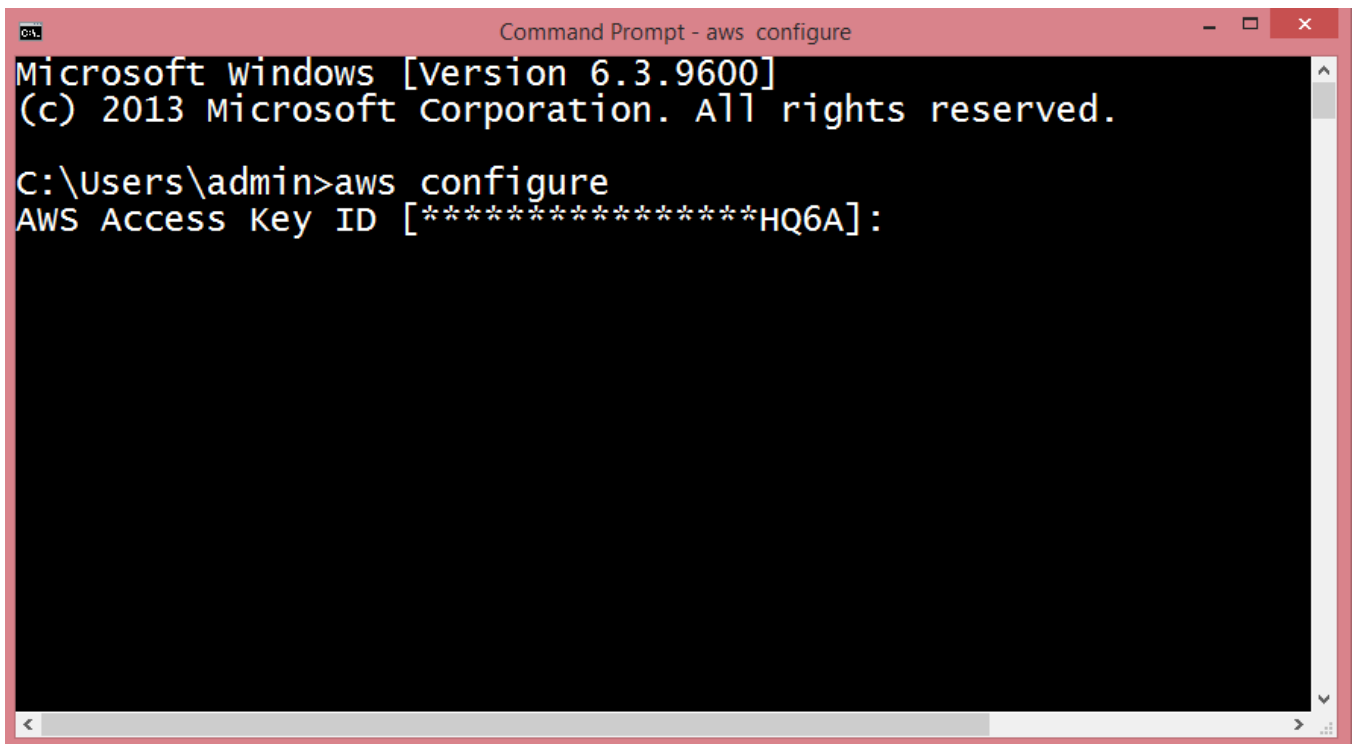
The above screen shows that the Cross region replication has been updated successfully. We can also see the source bucket and destination with their associated permissions.

- Now, we will see whether the files have been replicated from jtpbucket to the jtp1bucket. Click on the **jtp1bucket**.



The above screen shows that the bucket is empty. Therefore, we can say that the objects do not replicate from one bucket to another bucket automatically, we can replicate only by using AWS CLI (Command Line Interface). To use the AWS CLI, you need to install the CLI tool.

- After installation, open the cmd and type **aws configure**.



- Now, we need to generate the Access Key ID which is a user name and secret access key which is a password. To achieve this, we first need to create an IAM Group.

The screenshot shows the AWS IAM console interface. On the left, there is a navigation menu with options like Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, and Credential report. The main area displays a 'Create New Group' button and a 'Group Actions' dropdown. Below this is a search bar and a table header with columns: Group Name, Users, Inline Policy, and Creation Time. The table currently shows 'Showing 0 results' and 'No records found.'

- Set the Group Name, i.e., javatpoint.

This screenshot shows the 'Set Group Name' step of the 'Create New Group Wizard'. The wizard has three steps: Step 1: Group Name, Step 2: Attach Policy, and Step 3: Review. The 'Group Name' field is set to 'jvatpoint' (note the typo in the image). Below the field, there is a note: 'Example: Developers or ProjectAlpha' and 'Maximum 128 characters'. The instruction says: 'Specify a group name. Group names can be edited any time.'

- Check the **AdministratorAccess** policy to access the AWS console through AWS CLI.
- Now, create an IAM User.
- Add the user name with programmatic access.

The screenshot shows the 'Add User' wizard in the AWS IAM console. It has five steps, with the first step 'Set user details' being the active one. In this step, the 'User name*' is set to 'Akshita'. Below this is a link to 'Add another user'. The next step is 'Select AWS access type', which shows two options: 'Programmatic access' (selected) and 'AWS Management Console access'. The 'Programmatic access' option is described as: 'Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.'

- Add the user to a group, i.e., javatpoint.

aws Services ▾ Resource Groups ▾

1 2 3 4 5

▼ Set permissions

Add user to group
 Copy permissions from existing user
 Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

Create group Refresh

Search Showing 1 result

| Group ▾ | Attached policies |
|--|---------------------|
| <input checked="" type="checkbox"/> javatpoint | AdministratorAccess |

- Finally, the user is created.

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://582304292942.signin.aws.amazon.com/console>

Download .csv

| | User | Access key ID | Secret access key |
|---|---------|----------------------|----------------------------|
| ▶ | Akshita | AKIAIKO2P7RT7WSDYAMA | ***** Show |

From the above screen, we observe that access key and secret access key have been generated.

- Copy the access key and secret access key to the cmd.

```
Command Prompt

C:\Users\admin>aws configure
AWS Access Key ID [*****FEIQ]: AKIAJLCEBPT723YDFEI
AWS Secret Access Key [*****0Ztq]: tZ0wpUB6JhidBGW
m70Ztq
Default region name [us-east-1]: us-east-1
Default output format [json]:

C:\Users\admin>
```

- To view the S3 buckets, run the command **aws s3 ls**.

```
Command Prompt

C:\Users\admin>aws configure
AWS Access Key ID [*****FEIQ]: AKIAJLCEBPT723YDFEI
AWS Secret Access Key [*****0Ztq]: tZ0wpUB6JhidBGW
m70Ztq
Default region name [us-east-1]: us-east-1
Default output format [json]:

C:\Users\admin>aws s3 ls
2019-02-05 11:33:30 jtp1bucket
2019-02-05 11:05:37 jtpbucket

C:\Users\admin>
```

- To copy the objects of **jtpbucket** to **jtp1bucket**, run the command **aws s3 cp?recursive s3://jtpbucket s3://jtp1bucket**.

```

C:\Users\admin>aws configure
AWS Access Key ID [*****FEIQ]: AKIAJLCEBPT723YDFEI
AWS Secret Access Key [*****0Ztq]: tZ0wpUB6JhidBGW
m70Ztq
Default region name [us-east-1]: us-east-1
Default output format [json]:

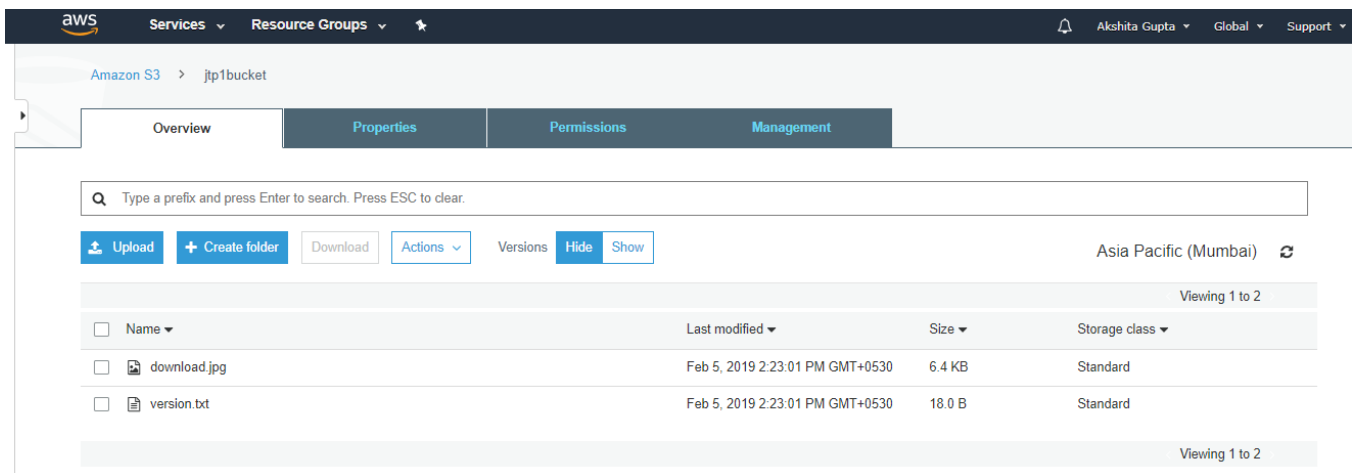
C:\Users\admin>aws s3 ls
2019-02-05 11:33:30 jtp1bucket
2019-02-05 11:05:37 jtpbucket

C:\Users\admin>aws s3 cp --recursive s3://jtpbucket s3://jtp1
copy: s3://jtpbucket/version.txt to s3://jtp1bucket/version.t
copy: s3://jtpbucket/download.jpg to s3://jtp1bucket/download
C:\Users\admin>

```

The above screen shows that the objects of **jtpbucket** have been copied to the **jtp1bucket**.

- Click on the "**jtp1bucket**".



From the above screen, we observed that all the files in the original bucket have been replicated to another bucket, i.e., **jtp1bucket**.

Note: If any further changes made in the original bucket will always be copied to its replicated bucket.

Important points to be remembered:

- Versioning must be enabled on both the source and destination buckets.
- The regions of both the buckets must be unique.

- All the files in an original bucket are not replicated automatically, and they can be replicated through AWS CLI. All the subsequent files are replicated automatically.
- Files in a file cannot be replicated to multiple buckets.
- Delete markers are not replicated.
- Delete versions or Delete markers are not replicated.

[< Prev](#)[Next >](#)

For Videos Join Our Youtube Channel: [Join Now](#)

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share



Learn Latest Tutorials

